THE COMPLEXITY OF
MULTI-LAYERED PERCEPTRONS

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de Rector Magnificus, prof. dr. J.H. van Lint, voor een commissie aangewezen door het College van Dekanen in het openbaar te verdedigen op vrijdag 15 april 1994 om 16.00 uur

door
Patrick Johannes Zwietering
geboren te Sittard
Dit proefschrift is goedgekeurd
door de promotoren
prof.dr. E.H.L. Aarts
en
prof.dr. J. Wessels

CIP-DATA KONINKLIJKE BIBLIOTHEEK, DEN HAAG
Zwietering, Patrick Johannes
The complexity of multi-layered perceptrons / Patrick
Johannes Zwietering. - Eindhoven : Technische
Universiteit Eindhoven
ISBN 90-386-9293-6
Subject headings: neural networks ; complexity.

Printed and bound by Wibro Dissertationsdrukkerij, Helmond

Copyright © 1994 by P.J. Zwietering, Alphen aan den Rijn, The Netherlands
to Eveline
my personal iff condition
Contents

1 Introduction .......................................................... 1
  1.1 Multi-layered perceptrons ........................................ 1
  1.2 Complexity ....................................................... 3
  1.3 Problem formulation ............................................. 5
  1.4 Solution approach and obtained results ....................... 5
  1.5 Outline .......................................................... 7

2 Multi-Layered Perceptrons ........................................ 9
  2.1 Introduction ..................................................... 9
  2.2 One-layered perceptrons ........................................ 10
    2.2.1 The construction of one-layered perceptrons ............... 11
    2.2.2 Classification with 1LPs ................................ 12
    2.2.3 An approach for 1LP Lပ% based on the convex-hull ........ 14
    2.2.4 An approach for FLP, based on the Perceptron Convergence Procedure ............................................. 17
    2.2.5 An approach for HLP, based on linear programming ....... 24
    2.2.6 Unsolvable finite classification problems ................ 25
  2.3 Multi-layered perceptrons .................................... 27
    2.3.1 The construction of multi-layered perceptrons ........... 27
    2.3.2 Classification with MLPs .................................. 30
    2.3.3 Finite loading of MLPs ................................... 32
    2.3.4 The complexity of loading MLPs .......................... 35
    2.3.5 Approximation with MLPs ................................ 37
Chapter 1

Introduction

This thesis is concerned with complexity of multi-layered perceptrons, required for solving a given problem. In particular we study the complexity of multi-layered perceptrons that solve combinatorial optimization problems. The remainder of this chapter is organized as follows. In Section 1.1, we give an introduction of the multi-layered perceptron. In Section 1.2, we define the corresponding notion of complexity. The problem formulation is discussed in Section 1.3, while the solution approach and obtained results are summarized in Section 1.4. The chapter ends with an outline of the thesis.

1.1 Multi-layered perceptrons

In general, a neural network consists of a network of elementary nodes that are connected through weighted connections. The nodes represent computational units, which are capable of performing a simple computation that consists of a summation of the weighted inputs of the node, followed by the addition of a constant called the threshold or bias, and the application of a non-linear response function. The result of the computation of a unit constitutes the output of the corresponding node. Subsequently, the output of a node is used as an input for the nodes to which it is connected through an outgoing arc.

Over the years there has grown a large variety of neural network models, which can be roughly divided into recurrent neural networks and feedforward neural networks. In recurrent neural networks such as the Hopfield network [Hopfield, 1982; Hopfield, 1984; Hopfield & Tank, 1986] and the Boltzmann machine [Aarts & Korst, 1989; Fahlman, Hinton & Sejnowski, 1983; Hinton & Sejnowski, 1985; Hinton & Sejnowski, 1985; Hinton, Sejnowski & Ackley, 1984], information that flows out of a node can return to that node, whereas in feedforward neural networks such as the multi-layered perceptron [Rumelhart, Hinton & Williams, 1986] and the Kohonen feature map [Kohonen, 1988], information only flows in one direction; see also the books by Aarts & Korst [1989], Hecht-Nielsen [1990], and Kosko [1992], for a more detailed review of the different neural network models.

In this thesis we consider the subclass of feedforward neural networks known as
the multi-layered perceptrons or multi-layered neural networks; see also Rumelhart, Hinton & Williams [1986]. In a multi-layered perceptron the nodes are arranged in layers, and the connections are not allowed to cross a layer, i.e., there are connections between the inputs of the network and the nodes in the first layer and between subsequent layers only. This implies that the inputs of a node in the first layer correspond to the inputs of the network, while the inputs of the nodes in a higher layer are the outputs of the nodes in the preceding layer. The outputs of the nodes in the highest layer form the outputs of the network. The nodes that are not output nodes are called the hidden nodes, and the corresponding layers the hidden layers; see Figure 1.1 for an example of a three-layered perceptron.

Figure 1.1: An example of a three-layered perceptron with three inputs, four nodes in the first hidden layer, three nodes in the second hidden layer, and two output nodes. The weights on the connections are discarded, as well as the thresholds. The symbols $x_1$, $x_2$, and $x_3$ represent the three input values of the network, and $f_1(x_1, x_2, x_3)$ and $f_2(x_1, x_2, x_3)$ the corresponding output values.

The inputs and the weights of a multi-layered perceptron are in general real valued. The outputs of the nodes can also be real valued, depending on the choice
1.2 Complexity

of the response functions that are used. For most problems discussed in this thesis we consider multi-layered perceptrons that use hard-limiting response functions. The output of the hard-limiting response function is 1, if its input is positive or zero, and 0 otherwise. Applications of multi-layered perceptrons usually consider some kind of sigmoidal response function, which is often some kind of continuous approximation of the hard-limiting response function, such as the well-known logistic function. Therefore, we also present some results that relate the capabilities of multi-layered perceptrons with hard-limiting response functions, to the capabilities of multi-layered perceptrons that use some kinds of sigmoidal response functions.

Multi-layered perceptrons can be viewed as an extension of the single-layered perceptrons designed by Rosenblatt; see [Rosenblatt, 1958; Rosenblatt, 1962]. Rosenblatt showed that perceptrons can be used for adaptive pattern classification, by proving his famous perceptron convergence theorem. This theorem states that the perceptron convergence procedure finds the weights of a one-layered perceptron that solves a given classification problem if such a solution exists. Among others, Minsky & Papert [1969] demonstrated the limitations of one-layered perceptrons by showing that they can only classify sets that are linearly separable. Minsky & Papert suggested the use of multi-layered perceptrons to overcome these difficulties. After the convincing argument of Minsky & Papert and the lack of a convergence procedure for multi-layered perceptrons, interest in the perceptrons dropped to a modest level.

Recently, multi-layered perceptrons regained interest due to the discovery of suitable learning algorithms such as the back-propagation algorithm, that can be used to find the weights that correspond to a given input-output behavior [Rumelhart, Hinton & Williams, 1986; Werbos, 1990]. However, despite many successful practical applications of the back-propagation algorithm there is still a large number of unresolved questions about the use of multi-layered perceptrons. The optimal choice of the number of layers and the number of nodes in each layer, are examples of such open problems.

1.2 Complexity

Complexity is a general notion that expresses a measure for the amount of computational resources used by some implementation of an algorithm. The considered resource may be time, memory, space, number of transistors, the quality of the used transistors, etc. Given such a complexity measure, the complexity of a certain problem with respect to that measure is defined as the size of the smallest, with respect to that measure, implementation of any algorithm that solves the problem. Usually, while determining the complexity of a problem with respect to a certain measure, the available amount of some of the other resources is restricted. For instance, the sequential time complexity of a problem may be defined as the number of basic operations required by any algorithm that solves the problem, given that these algorithms are allowed to run on a single processor only.

In this section we introduce the notion of complexity for multi-layered perceptrons, based on an approach that is often used for describing the complexity of Boolean circuits; see also Chandra, Stockmeyer & Vishkin [1984], Johnson [1990] and the
The complexity of a multi-layered perceptron expresses the amount of physical resources that is required to construct this multi-layered perceptron. It may consider for instance the number of nodes, the number of layers, the number of connections, the maximum number of incoming connections for every node, the maximum number of outgoing connections for every node, the size of the weights, and the accuracy required of the computation in the node. Depending on the particular situation, some resources may be more important than others. Usually, the number of nodes is the most important complexity measure, also because it provides an upper bound on most of the other quantities.

The complexity of a multi-layered perceptron is not interesting without specifying its input-output behavior. Let the function \( f : \mathcal{D} \rightarrow \mathcal{R} \), for some domain \( \mathcal{D} \subseteq \mathbb{R}^N \), range \( \mathcal{R} \subseteq \mathbb{R}^K \), and \( N, K \in \mathbb{N} \), represent the desired input-output behavior of a multi-layered perceptron, where \( \mathbb{R} \) and \( \mathbb{N} \) denote the set of real numbers and positive integers, respectively. Then the MLP-complexity of \( f \) with respect to a given complexity measure is the size of the smallest multi-layered perceptron that computes \( f \). For instance, the MLP-complexity of \( f \) with respect to the number of nodes is the smallest number of nodes of any multi-layered perceptron that computes \( f \). Similarly, the MLP-complexity of \( f \) with respect to the size of the weights is the minimum, taken over all multi-layered perceptrons that compute \( f \), of the maximum absolute weight of the multi-layered perceptron. Note that since every multi-layered perceptron that computes a given function has the same number of inputs \( N \) and outputs \( K \), it makes no sense to consider these as complexity measures.

We consider the multi-layered perceptrons as a computational model that can be used for solving problems of the following kind. A problem is defined as an infinite set of functions \( \{ f_N \}_{N \in \mathbb{N}} \), with \( f_N : \mathcal{P} \mathcal{D}^N \rightarrow \mathcal{P} \mathcal{R}^K(N) \), for some \( \mathcal{P} \mathcal{D}, \mathcal{P} \mathcal{R} \subseteq \mathbb{R} \) and \( K(N) \in \mathbb{N} \). \( \mathcal{P} \mathcal{D} \) and \( \mathcal{P} \mathcal{R} \) stand for problem domain and problem range, respectively. We say that \( \{ f_N \}_{N \in \mathbb{N}} \) is solved by a family \( \{ M_N \}_{N \in \mathbb{N}} \) of multi-layered perceptrons, if \( M_N \) computes \( f_N \), for all \( N \in \mathbb{N} \). Using the above definition of the MLP-complexity of a function, we can define the MLP-complexity of a problem. The MLP-complexity of a problem \( \{ f_N \}_{N \in \mathbb{N}} \) with respect to a given complexity measure, is a function of \( N \) that for each \( N \in \mathbb{N} \) is equal to the MLP-complexity of \( f_N \) with respect to that complexity measure. One often considers the MLP-complexity of a problem with respect to one of the complexity measures, given a set of restrictions on some of the other complexity measures. For instance, one may consider the layer complexity of a given problem where all considered multi-layered perceptrons have a number of nodes that is bounded by a polynomial in the number of inputs.

When considering a family of multi-layered perceptrons \( \{ M_N \}_{N \in \mathbb{N}} \) for solving a given problem, we in general do not require that a description of \( M_N \) can be generated given \( N \). Therefore, this computational model can be viewed a nonuniform approach for solving problems, similar to the nonuniform approaches based on Boolean circuits; see also Johnson [1990] and Wegener [1987]. One may also consider uniform variants, in which case the complexity of the generation of \( M_N \), given \( N \), must also be examined.
1.3 Problem formulation

Using the definition of MLP-complexity defined in the previous section, we can formulate the main questions posed in this thesis. What is the MLP-complexity with respect to the number of nodes and what is the MLP-complexity with respect to the number of layers of a given problem, provided that the response functions used by all considered multi-layered perceptrons are restricted to a certain set of functions? Since the MLP-complexity of a problem is based on the MLP-complexity of functions, the above question can be reformulated as follows. What is the minimal number of nodes and what is the minimal number of layers required by any multi-layered perceptron that computes a given function, provided that the response functions used by all considered multi-layered perceptrons are restricted to a certain set of functions?

Note that in the above questions the node complexity and layer complexity are dealt with separately. We do not consider the questions of finding the minimum number of layers given a certain number of nodes, or vice versa. If one considers Boolean functions only, the above questions are the same as the ones posed in [Wegener, 1987] for circuits.

In addition to the above questions, we also consider the question of finding methodical approaches for constructing a multi-layered perceptron that computes a given function, as well as for determining the complexity of such an approach.

1.4 Solution approach and obtained results

We consider results on the MLP-complexity of various problems, with respect to the required number of nodes and layers. The first results are obtained by a neural network approach, based on the early neural network literature and some of the more recent papers. In particular, we study the results of Minsky & Papert [1969], Parberry [1991], and Baum [1988], who consider multi-layered perceptrons that use hard-limiting response functions for solving finite classification problems. This corresponds to the problem of finding a multi-layered perceptron that computes a function $f : D \to R$, for some finite $D \subseteq R^n$ and $R = B$, where $B = \{0, 1\}$. We discuss the well-known linear-separability condition, which gives a necessary and sufficient condition for the existence of a one-layered perceptron that computes the given function. Furthermore, we show that, in case a one-layered perceptron exists that solves a given finite classification problem, it can be found in polynomial time. The analysis used for these results show that in some cases the weight complexity of a one-layered perceptron can become exponential in the size of the input. However, in many cases there does not exist a one-layered perceptron that solves a given finite classification problem, which makes it necessary to consider multi-layered perceptrons.

It is well-known that any finite classification problem can be solved with a two-layered perceptron; see for instance [Haum, 1988]. Hence, the layer complexity of a finite classification problem is determined by the linear-separability condition. Unfortunately, the node complexity of a finite classification problem is still unknown,
except for some rough upper bounds. Furthermore, the problem of finding a multi-layered perceptron with a certain number of layers and nodes, that solves a given finite classification problem, is shown to be \( \mathcal{NP} \)-hard. Finally, we discuss some of the results found in literature on the capabilities of multi-layered perceptrons for solving infinite classification problems. To this end, we discuss the approximation results of Cybenco [1989], Hornik, Stinchcombe & White [1989], Kurkova [1992], and Barron [1991], who consider the problem of approximating a given function \( f : \mathbb{R}^N \to [0,1] \) with two- and three-layered perceptrons that use certain nonlinear response functions. The discussed results can be used to obtain bounds for the number of nodes required to approximately solve a given infinite classification problem with two- and three-layered perceptrons, respectively.

A second set of results is obtained using an approach based on the introduction of a number of complexity classes, which are similar to the classes introduced for Boolean circuits. In fact we use the approach presented in [Parberry, 1991], which introduces a general framework for both Boolean circuits and multi-layered perceptrons. When considered for computing Boolean functions, the multi-layered perceptrons that use hard-limiting response functions are shown to be equally powerful as the circuits that use Boolean threshold functions. Results from [Maass, Schnitger & Sontag, 1991] indicate that this also holds for multi-layered perceptrons that use response functions similar to the sigmoid function.

The major part of this thesis concerns results obtained for functions \( f : \mathbb{R}^N \to \mathbb{B}^N \), based on a geometrical analysis of the capabilities of multi-layered perceptrons that use hard-limiting response functions. The corresponding problems are infinite classification problems. The analysis is based on the case that \( K = 1 \), corresponding to binary classification problems, and later extended to all values of \( K \). We formulate necessary and sufficient conditions for the existence of an \( m \)-layered perceptron that computes a given function \( f : \mathbb{R}^N \to \mathbb{B} \), for all values of \( m \in \mathbb{N} \). The necessary and sufficient conditions are equal, and fairly trivial, for all \( m \neq 2 \). The non-trivial case \( m = 2 \) is analyzed in great detail. We present results that indicate that the gap between the presented necessary and sufficient conditions is small. Using some polyhedral theory we derive a lower bound for the number of nodes in the first hidden layer, that is required for solving a given classification problem. In some cases this lower bound can be shown to be tight. Finally, we discuss a methodical approach for finding the construction of a two-layered perceptron that solves a given classification problem in some cases.

By reformulating a combinatorial optimization problem as an infinite classification problem, we can use the results obtained for the MLP-complexity of classification problems to derive results on the MLP-complexity of combinatorial optimization problems. We derive sufficient conditions for a combinatorial optimization problem to be solvable by a three-layered perceptron and necessary conditions for a combinatorial optimization problem to be solvable by a two-layered perceptron. For a large class of combinatorial optimization problems we derive an expression for the minimal number of first-layer nodes required by any multi-layered perceptron that solves the problem, which is based on the minimal exact neighborhood of the considered problem. The minimal exact neighborhood is a notion that stems from local
1.5 Outline

search, an approach for handling combinatorial optimization problems through the exploration of neighborhoods; see Aarts, Korst & Zwietering [1992] for an introduction into this field. The number of nodes in the first-hidden layer is obviously a lower bound for the total number of hidden nodes, which in some cases can be used to show that no multi-layered perceptron exists that solves a given problem with a polynomial number of nodes. Furthermore, we present a construction of a three-layered perceptron with this minimum number of nodes in the first hidden layer that solves the considered combinatorial optimization problem.

The general results on the MLP-complexity of combinatorial optimization problems are applied to five well-known combinatorial optimization problems; SORTING, MINIMUM COST SPANNING TREE, SHORTEST NETWORK PATH, SHORTEST NETWORK ROUTE, and DISCRETE DYNAMIC LOTSIZING. We prove that the first two problems have polynomial node complexity, i.e., there exist multi-layered perceptrons with a polynomial number of nodes that use hard-limiting response functions, that solve SORTING and MINIMUM COST SPANNING TREE. For the other three problems we show that the minimum number of first-layer nodes is exponential, which implies that no multi-layered perceptron exists with a polynomial number of nodes that solves these problems. Finally, we show that the layer complexity of SORTING and DISCRETE DYNAMIC LOTSIZING is three, i.e., no two-layered perceptron exists that solves these problems.

1.5 Outline

The thesis is roughly divided into two parts. The first part consists of the Chapters 2, 3, and 4, that can be considered as introductory chapters. The results discussed in these chapters are mainly based on results found in literature. The second part consists of the Chapters 5, 6, and 7. These chapters present new results on the MLP-complexity of combinatorial classification problems and combinatorial optimization problems. The contents of each of the chapters is briefly described below.

In Chapter 2, we introduce the multi-layered perceptrons in a historical context. We discuss some results on the complexity of multi-layered perceptrons that solve finite classification problems, both exact and approximative. Most of these results stem from the neural network literature.

In Chapter 3, we introduce our formulation of a real valued combinatorial optimization problems, and introduce the five combinatorial optimization problems discussed in Chapter 4 and Chapter 7.

Chapter 4 introduces a general framework for both Boolean circuits and multi-layered perceptrons. We discuss complexity results for Boolean circuits that are relevant for the study of the complexity of multi-layered perceptrons, and examine how these results apply to the multi-layered perceptrons. Furthermore, we introduce some complexity classes for multi-layered perceptrons similarly to those defined for Boolean circuits.

Chapter 5 examines the complexity of multi-layered perceptrons that solve infinite classification problems. Necessary and sufficient conditions are derived for a function
\( f : \mathbb{R} \to \mathbb{B} \) to be representable by an \( m \)-layered perceptron, or equivalently, to classify the subset \( f^{-1}(1) \). Furthermore, we derive a lower bound on the number of first-layer nodes that is necessary for a multi-layered perceptron to be able to classify a given subset, whatever the number of layers that is used. Part of the results presented in this chapter have been published in an earlier paper; see Zwietering, Aarts & Wessels [1992a].

In Chapter 6 we use the results obtained in Chapter 5 to obtain general results about the complexity of multi-layered perceptrons that solve combinatorial optimization problems. Firstly, the classifiability conditions of subsets are used to obtain conditions for a combinatorial optimization problem to be solvable with a multi-layered perceptron, and to determine the minimal number of layers. Secondly, an expression is presented for the minimal number of first-layer nodes that is required by any multi-layered perceptron that solves a given combinatorial optimization problem.

In Chapter 7 we apply the results obtained in Chapter 6 to the five problems introduced in Chapter 3: **sorting**, **minimum cost spanning tree**, **shortest network path**, **shortest network route**, and **discrete dynamic lot sizing**. Preliminary results for **sorting** and **discrete dynamic lot sizing** have been presented in [Zwietering, Aarts & Wessels, 1993; Zwietering, Aarts & Wessels, 1992b] and [Zwietering, Van Kraaij, Aarts & Wessels, 1991], respectively.

Chapter 8 concludes this thesis with a discussion of the obtained results.
Chapter 2

Multi-Layered Perceptrons

2.1 Introduction

In this chapter we introduce the multi-layered perceptrons and consider their capabilities for solving classification problems. The objective of this chapter is to provide some feeling for this type of neural network, and for its capabilities and complexity, by browsing through some relevant parts of the history of multi-layered perceptrons.

The introduction of the multi-layered perceptrons is rather informal, and consists of a constructional description using a stratified approach. For a formal introduction of the multi-layered perceptrons as a computational model, we refer to Chapter 4. A multi-layered perceptron consists of a number of interconnected nodes that are arranged in layers, and can be viewed as a function with a number of inputs and outputs. Different types of problems can be considered, to be solved by a multi-layered perceptron, depending on the allowed input and output values. If the allowed output values are Boolean, i.e., 0 or 1, then the considered problems are usually known as classification problems, whereas in the case of real-valued outputs one usually speaks of representation problems. This chapter considers some results on the capabilities of multi-layered perceptrons for solving classification problems. The major part of the discussed results concerns finite classification problems and is found in literature. Finite classification problems are problems in which a vector has to be classified with respect to a number of finite subsets; see also Chapter 5, where we discuss the infinite classification problems.

The problem of finding a multi-layered perceptron that solves a given classification problem is denoted as a loading problem; see also Judd [1991]. We start with the loading of one-layered perceptrons that have one output, corresponding to the original perceptron, which is based on the work of Rosenblatt [1958] and Minsky & Papert [1969]. We discuss three approaches for solving finite one-layered perceptron loading problems. The first approach is a geometrical approach based on the computation of the convex hull of the subsets that have to be classified, the second approach is based on the Perceptron Convergence Procedure [Rosenblatt, 1962], following the lines of [Minsky & Papert, 1969], and the third approach is based on linear programming. We show that the first two approaches yield exponential run-
cryptron, and the neuron; see also [Muroga, 1971]. The formulation of the 1LP given above is due to Minsky & Papert [1969], who considered the 1LP as a special case of their perceptrons, namely as a perceptron of order 1.

The functioning of the 1LP can be summarized as follows. Consider a 1LP with one output, \( N \) inputs, \( N \) connection weights \( a_1, \ldots, a_N \), and threshold \( b \). Then the output \( f \) as function of the input \( x = (x_1, \ldots, x_N) \) is given by

\[
f(x) = \theta(a \cdot x + b),
\]

where \( a \cdot x \) denotes the vector inner product and \( \theta \) is the step-function defined by

\[
\theta(x) = \begin{cases} 
1, & \text{if } x \geq 0, \\
0, & \text{if } x < 0.
\end{cases}
\]

For later use we define the weight of this 1LP to be \( \| (a_1, \ldots, a_N, b) \|_\infty \), which denotes \( \max\{|a_1|, \ldots, |a_N|, |b|\} \).

An extension to the above definition a 1LP is to consider other functions to be applied to the weighted sum, instead of the step function \( \theta \). Let \( \gamma : \mathbb{R} \to \mathbb{R} \) be an arbitrary function, then the output of a 1LP with response function \( \gamma \) is defined by

\[
f(x) = \gamma(a \cdot x + b).
\]

1LPs that use the response function \( \gamma \) are denoted by \( \gamma \)-1LPs. Similarly, 1LPs that use the standard response function \( \theta \) are denoted by \( \theta \)-1LPs. The step-function \( \theta \) is also known as the hard-limiting response function. Other response functions are typically some kind of continuous approximation of the step function, i.e., one often uses functions \( \gamma : \mathbb{R} \to [0, 1] \) that are continuous and non-decreasing. A well-studied example of such a function is the sigmoid function \( \sigma : \mathbb{R} \to [0, 1] \) defined by

\[
\sigma(x) = \frac{1 + \exp(-x)}{1 - \exp(-x)}.
\]

We return to the various response functions when we discuss the multi-layered perceptrons in Section 2.3. In the remainder of this section we only consider \( \theta \)-1LPs, which are denoted by 1LPs for reasons of brevity.

### 2.2.2 Classification with 1LPs

Since a 1LP with \( N \) inputs maps the input space \( \mathbb{R}^N \) onto the two values 0 and 1, it can be viewed as a classifying device. A 1LP with \( N \) inputs classifies the elements of \( \mathbb{R}^N \) in two sets \( f^-(0) \) and \( f^+(1) \), denoting the subset of \( \mathbb{R}^N \) that is mapped to 0 and 1, respectively. Hence, a 1LP could be used to solve classification problems. Below we define a binary classification problem (BCP), where the objective is to classify between two subsets. In Chapter 5 we consider an arbitrary number of subsets, when we discuss Combinatorial Classification Problems (CCPs).

**Definition 2.2.1.** (Binary Classification Problem (BCP))

**INPUT:** Two disjoint sets \( \Omega_0, \Omega_1 \subseteq \mathbb{R}^N \) and a vector \( x \in \mathbb{R}^N \), for some \( N \in \mathbb{N} \).

**OUTPUT:** 1, if \( x \in \Omega_1 \), and 0, if \( x \in \Omega_0 \).

Note that the variable \( N \) in Definition 2.2.1 implicitly runs over all integers; see also Chapter 4, where we give a general definition of a problem. In case that \( x \notin \Omega_1 \cup \Omega_0 \), the above problem is infeasible. We assume that "infeasible" is the desired output in case that a problem is unsolvable.
2.2 One-layered perceptrons

In Chapter 5 we introduce \((\mathbb{R}^N, \mathcal{B}, \{\Omega_0, \Omega_1\})\) as a shorthand notation for a BCP, where \(\mathbb{R}^N\) denotes the set of possible inputs, \(\mathcal{B}\) denotes the set of possible outputs, and \(\{\Omega_0, \Omega_1\}\) denote the two sets between which has to be classified. The objective is to solve a BCP with a 1LP. Solving a BCP is defined in general as follows.

**Definition 2.2.2.** Let \(N \in \mathbb{N}\) and \(\Omega_0, \Omega_1 \subseteq \mathbb{R}^N\) be two disjoint sets. A function \(f: \mathbb{R}^N \rightarrow \mathcal{B}\) is said to solve the BCP given by \((\mathbb{R}^N, \mathcal{B}, \{\Omega_0, \Omega_1\})\), if

\[
 f(x) = \begin{cases} 
 1, & \text{if } x \in \Omega_1 \\
 0, & \text{if } x \in \Omega_0.
\end{cases}
\]

The value of \(f(x)\) for \(x \notin \Omega_1 \cup \Omega_0\) is undefined.

Loosely speaking we say that \(f\) labels \(\Omega_1\) with a 1 and \(\Omega_0\) with a 0. If we search for a 1LP that solves a certain BCP, we are in fact solving the following problem.

**Definition 2.2.3.** (1LP Loading Problem (1LPLP); Search Variant)

**INPUT:** Two disjoint sets \(\Omega_0, \Omega_1 \subseteq \mathbb{R}^N\), for some \(N \in \mathbb{N}\).

**OUTPUT:** \(a \in \mathbb{R}^N\) and \(b \in \mathbb{R}\), such that \(a \cdot x + b \geq 0\), for all \(x \in \Omega_1\), and \(a \cdot x + b < 0\), for all \(x \in \Omega_0\).

This problem is called a loading problem because one wants to load the 1LP with a set of weights and a threshold, such that it solves the considered BCP; see also [Judd, 1991]. In a loading problem the classification of all the vectors in \(\Omega_1 \cup \Omega_0\) must be correct, this in contrast to a learning problem, where one allows a certain amount of mistakes or misclassifications. We do not discuss results on learning problems, since they are considered beyond the scope of this thesis; see for definitions and results of various learning problems [Valiant, 1984; Baum, 1991; Gallant, 1990; Judd, 1991].

The above loading problem is the search variant of 1LPLP, because we search for an explicit presentation of the weights \((a, b)\), with \(a \in \mathbb{R}^N\) and \(b \in \mathbb{R}\), of a 1LP that labels \(\Omega_1\) with a 1 and \(\Omega_0\) with a 0. Alternatively, one may consider the decision variant of 1LPLP, where one wants to answer the question whether such a 1LP exists.

**Definition 2.2.4.** (1LP Loading Problem (1LPLP); Decision Variant)

**INPUT:** Two disjoint sets \(\Omega_0, \Omega_1 \subseteq \mathbb{R}^N\), for some \(N \in \mathbb{N}\).

**OUTPUT:** Does there exist \(a \in \mathbb{R}^N\) and \(b \in \mathbb{R}\), such that \(a \cdot x + b \geq 0\), for all \(x \in \Omega_1\), and \(a \cdot x + b < 0\), for all \(x \in \Omega_0\)?

The output of this problem is "yes" or "no".

In the following subsections we consider some approaches for solving both the decision variant and the search variant of 1LPLP, which are denoted by 1LPLP\(_{\text{d}}\) and 1LPLP\(_{\text{s}}\), respectively. These approaches give insight in the complexity of both variants. Usually, the complexity of the search variant and the complexity of the decision variant of a problem are polynomially related, i.e., if one of them is solvable in polynomial time then so is the other; see for more about the different variants and their complexity [Garey & Johnson, 1979; Johnson, 1990; Papadimitriou & Steiglitz, 1982]. This also holds for the finite version of 1LPLP, where one considers instances for which \(\Omega_0\) and \(\Omega_1\) are finite. In fact we prove that both the search variant and
the decision variant of the finite version of \( \text{ILPLP} \) are solvable in polynomial time using an approach based on linear programming.

Before we consider the approach based on linear programming, we consider two other approaches that are in general not polynomial, even for finite instances. The first approach solves an instance of \( \text{ILPLP}_d \) by determining the convex-hull of \( \Omega_0 \) and \( \Omega_1 \). The second approach solves an instance of \( \text{ILPLP}_d \) under the condition that it is a "yes"-instance of the corresponding decision variant, and is based on the well-known Perceptron Convergence Procedure. The reason for considering these two approaches is that they give a historical insight into the complexity of the considered problems. Furthermore, although we prove the correctness of the two approaches for the finite versions of the above loading problems, they may be adaptable to one of the more general formulations. A third reason is that they provide an introduction to the notation and the results presented in the remainder of this thesis, in particular to some of the results of Chapter 4.

In the forthcoming discussion we consider some problems related to \( \text{ILPLP} \). For these problems we do not explicitly notify whether they are search problems or decision problems, since this follows from the formulation of the problem. Usually, we define the decision variant of the considered problem only, and assume that the search variant can be defined similarly.

### 2.2.3 An approach for \( \text{ILPLP}_d \) based on the convex-hull

Consider an instance \( \Omega_0, \Omega_1 \subseteq \mathbb{R}^N \) of \( \text{ILPLP}_d \). It is obvious that this is a "yes"-instance, if and only if the two sets \( \Omega_1 \) and \( \Omega_0 \) can be separated by a hyperplane \( \hat{a} \cdot x + \hat{b} = 0 \). If a separating hyperplane for \( \Omega_1 \) and \( \Omega_0 \) exists we say that \( \Omega_1 \) and \( \Omega_0 \) are linearly separable. Figure 2.2a shows an example of two linearly separable subsets. Intuitively it seems obvious that the two considered sets \( \Omega_1 \) and \( \Omega_0 \) can be linearly separated, if their convex-hulls do not intersect. We refer to this condition as the "convex-hull condition". The convex-hull of a given subset is the smallest convex set that contains that subset, and is defined as follows.

![Diagram](image)

**Figure 2.2:** Linearly separable subsets. In (a) two infinite subsets, in (b) two finite subsets.
2.2 One-layered perceptrons

Definition 2.2.5. Let \( V \subseteq \mathbb{R}^N \). The convex-hull of \( V \) is denoted by \( \text{conv.hull}(V) \) and is given by

\[
\text{conv.hull}(V) = \{x \in \mathbb{R}^N \mid x = \sum_{i=1}^r \lambda_i v_i, v_i \in V, \lambda_i \geq 0, \sum_{i=1}^r \lambda_i = 1, r \geq 1\}.
\]

One can easily construct examples that prove that the "convex-hull condition" does not hold for arbitrary subsets in \( \mathbb{R}^N \). Furthermore, it may in general be complicated to determine the convex-hull of an arbitrary subset of \( \mathbb{R}^N \). Below, we prove that the "convex-hull condition" is correct for finite subsets. Figure 2.2b shows an example of a finite binary classification problem. The corresponding loading problem can be defined as follows.

Definition 2.2.6. (Finite 1LP Loading Problem (F1LPLP))

INPUT: Two finite disjoint sets \( \Omega_0, \Omega_1 \subseteq \mathbb{R}^N \), for some \( N \in \mathbb{N} \).

OUTPUT: Does there exist \( a \in \mathbb{R}^N \) and \( b \in \mathbb{R} \), such that \( a \cdot x + b \geq 0 \), for all \( x \in \Omega_1 \), and \( a \cdot x + b \leq 0 \), for all \( x \in \Omega_0 \)?

We prove below that an instance \( \Omega_0, \Omega_1 \subseteq \mathbb{R}^N \) of F1LPLP is a "yes"-instance, if and only if \( \text{conv.hull}(\Omega_1) \cap \text{conv.hull}(\Omega_0) = \emptyset \). To this end we require the well-known Farkas' lemma, which is usually stated as follows; see Papadimitriou & Steiglitz [1982], Nemhauser & Wolsey [1988], or Schrijver [1986].

Proposition 2.2.1 (Farkas' Lemma). Let \( A \) be an \( M \times N \) matrix, and \( b \in \mathbb{R}^M \). Then, either \( \{x \in \mathbb{R}^N \mid Ax = b\} \neq \emptyset \), or, exclusively, \( \{v \in \mathbb{R}^M \mid vA \geq 0, vb < 0\} \neq \emptyset \).

There are many variants of Farkas' Lemma. One of the variants is presented in Proposition 2.2.2 below, because it is convenient for the result proved below. It can be straightforwardly proved using Proposition 2.2.1, or directly using the standard approach based on the strong duality property of the primal and dual formulation of a linear program; see for such a proof and for a number of other variants [Nemhauser & Wolsey, 1988].

Proposition 2.2.2 (Variant of Farkas' Lemma). Let \( A \) be an \( M \times N \) matrix, and \( b \in \mathbb{R}^M \). Then, either \( \{x \in \mathbb{R}^N \mid Ax \leq b\} \neq \emptyset \), or, exclusively, \( \{v \in \mathbb{R}^M \mid vA = 0, vb < 0\} \neq \emptyset \).

Using the variant of Farkas' Lemma given above we can show the following result for F1LPLP.

Theorem 2.2.1. An instance \( \Omega_0, \Omega_1 \subseteq \mathbb{R}^N \) of F1LPLP is a "yes"-instance, if and only if

\[
\text{conv.hull}(\Omega_1) \cap \text{conv.hull}(\Omega_0) = \emptyset.
\]

Proof. Using that \( \Omega_1 \) and \( \Omega_0 \) are finite, one easily verifies that there exist \( a \in \mathbb{R}^N \) and \( b \in \mathbb{R} \) such that

\[
\begin{align*}
  a \cdot x + b &\geq 0, & \text{if } x &\in \Omega_1, \\
  a \cdot x + b &< 0, & \text{if } x &\in \Omega_0,
\end{align*}
\]

if and only if there exist \( a \in \mathbb{R}^N \) and \( b \in \mathbb{R} \) such that

\[
\begin{align*}
  a \cdot x + b &\geq +1, & \text{if } x &\in \Omega_1, \\
  a \cdot x + b &\leq -1, & \text{if } x &\in \Omega_0.
\end{align*}
\]
Let \( \Omega_1 = \{z_1, \ldots, z_i\} \) and \( \Omega_0 = \{z_{i+1}, \ldots, z_{i+l}\} \), for some \( z_1, \ldots, z_{i+l} \in \mathbb{R}^N \) and \( i, l \in \mathbb{N} \). Define the \((l_i + l_0) \times (N + 1)\) matrix \( A \) by

\[
A = \begin{bmatrix}
-z_1 & \cdots & -z_i & z_{i+1} & \cdots & z_{i+l_0}
-1 & \cdots & -1 & 1 & \cdots & 1
\end{bmatrix}^T,
\]

then \( a \in \mathbb{R}^N \) and \( b \in \mathbb{R} \) satisfy (2.2), if and only if

\[
A \begin{pmatrix}
a
b
\end{pmatrix} \leq -1,
\]

where \( 1 \in \mathbb{R}^{l_i + l_0} \) denotes a vector of all ones. Using Proposition 2.2.2, we find that there exist \( a \in \mathbb{R}^N \) and \( b \in \mathbb{R} \) that satisfy (2.3), if and only if there do not exist \( \lambda_i \geq 0, i \in \{1, \ldots, l_i + l_0\} \), that satisfy

\[
\sum_{i=1}^{l_i} \lambda_i x_i = \sum_{i=l_i+1}^{l_i+l_0} \lambda_i x_i,
\]

\[
\sum_{i=1}^{l_i} \lambda_i = \sum_{i=l_i+1}^{l_i+l_0} \lambda_i,
\]

\[
\sum_{i=1}^{l_i+l_0} \lambda_i > 0.
\]

From this and Definition 2.2.5 the result easily follows.

Using the results of Section 5.6, it can be shown that the "convex-hull condition" is correct for compact, i.e., bounded and closed, subsets that have piece-wise linear bounds. Furthermore, these results show that for arbitrary subsets \( \Omega_0, \Omega_1 \) with piece-wise linear bounds, \( \Omega_1 \cap \Omega_0 \) is linearly separable, if and only if \( \text{conv.hull}(\Omega_1) \cap \text{conv.hull}(\Omega_0) = \emptyset \), where \( \text{conv.hull}(\Omega_i) \) denotes the closure of the convex-hull of \( \Omega_i \); see Section 5.6.

In Section 5.6 we also describe some procedures that can be used to verify whether \( \text{conv.hull}(\Omega_1) \cap \text{conv.hull}(\Omega_0) = \emptyset \). However, this is not an efficient approach for solving \text{FPLP}, since it leads to an algorithm that has an exponential running time. Below we discuss an efficient approach based on linear programming, that leads to a polynomial-time algorithm for solving \text{FPLP}, after we have introduced some related problems and discussed the running time of the \text{Perceptron Convergence Procedure}.

The first related problem is the following.

**Definition 2.2.7.** (Finite Perceptron Loading Problem (FPLP))

**INPUT:** A finite set \( V \subseteq \mathbb{R}^N \), for some \( N \in \mathbb{N} \).

**OUTPUT:** Does there exist an \( a \in \mathbb{R}^N \), such that \( a \cdot x > 0 \), for all \( x \in V \)?

One can easily show that \text{FPLP} and \text{FILPLP} are essentially the same problem.

**Proposition 2.2.3.** \text{FILPLP} and \text{FPLP} are equivalent in the following sense.

(i) An instance \( \Omega_0, \Omega_1 \subseteq \mathbb{R}^N \) of \text{FILPLP} is a "yes"-instance, if and only if the instance \( V \subseteq \mathbb{R}^{N+1} \) of \text{FPLP} defined by

\[
V = \{ (x,1) \mid x \in \Omega_1 \} \cup \{ (-x,-1) \mid x \in \Omega_0 \},
\]

is a "yes"-instance.
An instance \( V \subseteq \mathbb{R}^N \) of FPLP is a "yes"-instance, if and only if the instance \( \Omega_0, \Omega_1 \subseteq \mathbb{R}^N \) of F1LPLP defined by \( \Omega_0 = \emptyset \) and \( \Omega_1 = \{0\} \) is a "yes"-instance.

Using Theorem 2.2.1 and Proposition 2.2.3 we directly obtain the following result.

**Corollary 2.2.1.** An instance \( V \subseteq \mathbb{R}^N \) of FPLP is a "yes"-instance, if and only if \( 0 \not\in \text{conv.bull}(V) \).

Next, we consider the problem of finding a 1LP with integer valued weights that solves a given finite classification problem. It is well-known that if a vector can be linearly separated from a finite subset, it can be linearly separated from this subset with a hyperplane with integer coefficients; see also Proposition 2.2.5 and [Papadimitriou & Steiglitz, 1982; Schrijver, 1986]. Thus the existence of a real vector \( a \) that solves FPLP guarantees the existence of an integer vector \( a' \) that solves FPLP. Proofs of this result in a perceptron-like context usually consider explicit approximations of reals by rationals; see for instance Minsky & Papert [1969], Parberry [1991], and [Parberry, 1990]. We consider the following problem.

**Definition 2.2.8. (Finite Integer Perceptron Loading Problem (F1LPLP))**

**INPUT:** A finite set \( V \subseteq \mathbb{R}^N \), for some \( N \in \mathbb{N} \).

**OUTPUT:** Does there exist an \( a \in \mathbb{Z}^N \), such that \( a \cdot x \geq 1 \), for all \( x \in V \)?

The above remarks show that FPLP and F1LPLP are equal, in the sense that if an instance is a "yes"-instances of FPLP, it is also a "yes"-instances of F1LPLP, and vice versa.

**Proposition 2.2.4.** \( \text{FPLP}_d = \text{F1LPLP}_d \).

In the following section we consider an approach for solving FPLP, and F1LPLP, that is based on the Perceptron Convergence Procedure.

### 2.2.4 An approach for FPLP, based on the Perceptron Convergence Procedure

In this section we consider an approach for solving the introduced loading problems that is based on the well-known Perceptron Convergence Procedure (PCP). Since it enables a more convenient formulation, we restrict the vectors in the sets \( \Omega_0, \Omega_1 \), and \( V \) to 0-1 vectors and \( \{-1,0,1\} \) vectors, respectively, the latter because of (2.4); see the problem formulations of H1LPLP, HPLP, and H1LPLP below. We note, however, that the results can be generalized to arbitrary finite sets, and thus to F1LPLP and FPLP; see Minsky & Papert [1969].

**Definition 2.2.9. (Hypercube 1LP Loading Problem (H1LPLP))**

**INPUT:** Two disjoint sets \( \Omega_0, \Omega_1 \subseteq \mathbb{B}^N \), for some \( N \in \mathbb{N} \).

**OUTPUT:** Does there exist \( a \in \mathbb{R}^N \) and \( b \in \mathbb{R} \), such that \( a \cdot x + b \geq 0 \), for all \( x \in \Omega_1 \), and \( a \cdot x + b < 0 \), for all \( x \in \Omega_0 \)?

**Definition 2.2.10. (Hypercube Perceptron Loading Problem (HPLP))**

**INPUT:** A set \( V \subseteq \{-1,0,1\}^N \), for some \( N \in \mathbb{N} \).
**OUTPUT:** Does there exist an \( a \in \mathbb{R}^N \), such that \( a \cdot x > 0 \), for all \( x \in V \)?

**Definition 2.2.11. (Hypercube Int. Perceptron Loading Problem (HILP))**

**INPUT:** A set \( V \subseteq \{-1,0,1\}^N \), for some \( N \in \mathbb{N} \).

**OUTPUT:** Does there exist an \( a \in \mathbb{Z}^N \), such that \( a \cdot x \geq 1 \), for all \( x \in V \)?

---

```
procedure PCP(input: \( V \subseteq \{-1,0,1\}^N \); output: \( a \in \mathbb{Z}^N \));
begin
  \( a := 0 \);
  repeat
    \( \text{GENERATE}(x \in V) \);
    if \( a \cdot x \leq 0 \) then \( a := a + x \)
  until \( [a \cdot x \geq 1 \text{ for all } x \in V] \)
end;
```

---

**Figure 2.3:** The perceptron convergence procedure.

The perceptron convergence procedure (PCP) is presented in Figure 2.3. This remarkably simple procedure is due to Rosenblatt [1958], although similar procedures for solving a set of linear inequalities had appeared long before Rosenblatt stated his results; see [Minsky & Papert, 1969; Nilsson, 1965] for a historical overview of the literature on the PCP. Minsky & Papert also noted that the PCP can be viewed as a "hill-climbing" algorithm, and indeed the similarity with a local search procedure is evident; see Aarts, Korst & Zwietering [1992] for the usual formulation of a local search procedure, and also Section 6.3.1. The following theorem due to Minsky & Papert [1969] proves that the PCP solves an instance of HILP, if this instance is a "yes"-instance of HILP. Furthermore, it provides an upper bound on the size of the solution found and on the running time of the algorithm. Note that we use \( \|a\|_2 = \sqrt{a \cdot a} \) to denote the Euclidean norm.

**Theorem 2.2.2 (Perceptron Convergence Theorem).**

Let \( N \in \mathbb{N} \), \( V \subseteq \{-1,0,1\}^N \), and \( \hat{a} \in \mathbb{R}^N \), such that \( \hat{a} \cdot x \geq 1 \), for all \( x \in V \). Then the following results hold.

(i) The PCP halts on input \( V \), and thus finds an \( a \in \mathbb{Z}^N \) with \( a \cdot x \geq 1 \), for all \( x \in V \).

(ii) The output vector \( a \) satisfies \( \|a\|_2 \leq N\|\hat{a}\|_2 \).

(iii) The running time of the algorithm is \( O(|V|N^2\|\hat{a}\|_2^2 \log(N\|\hat{a}\|_2)) \).

**Proof.** Let \( a_k \) denote the value of the variable \( a \) after \( k \) vectors \( x \in V \) have been added. Then \( a_0 = 0 \) and obviously \( a_k \in \mathbb{Z}^N \). Furthermore, there exists an \( x \in V \) such that

\[
\hat{a} \cdot a_{k+1} = \hat{a} \cdot (a_k + x) = \hat{a} \cdot a_k + \hat{a} \cdot x \geq \hat{a} \cdot a_k + 1,
\]

which implies that

\[
k \leq \hat{a} \cdot a_k. \tag{2.5}
\]
2.2 One-layered perceptrons

Similarly, we have for some $x \in V$

$$\|a_{k+1}\|_2^2 = (a_k + x) \cdot (a_k + x) = \|a_k\|_2^2 + 2a_k \cdot x + \|x\|_2^2 \leq \|a_k\|_2^2 + N,$$

since $a_k \cdot x \leq 0$ and $\|x\|_2^2 \leq N$, for all $x \in V$ added to $a$. Hence, we find

$$\|a_k\|_2^2 \leq Nk.$$  \hspace{1cm} (2.6)

Combining (2.5) and (2.6) with Cauchy's inequality, we obtain

$$k \leq a \cdot a_k \leq \|a\|_2 \|a_k\|_2 \leq \|a\|_2 \sqrt{Nk}.$$  \hspace{1cm} (2.7)

This yields

$$k \leq N\|\hat{a}\|_2^2,$$  \hspace{1cm} (2.7)

which proves that the number of times an $x$ is added is finite, and, hence, the PCP must terminate.

Result (ii) follows directly from (2.6) and (2.7).

To prove (iii) we observe that after $k$ additions of an $x$ in $V \subseteq \{-1,0,1\}^N$ we have $\|a_k\|_\infty \leq k$. Hence, at this stage the computation of

$$\text{if } a \cdot x \leq 0 \text{ then } a := a + x$$

can be done in time $O(N \log k)$. This implies that the total time to reach the $k$-th addition of an $x$ is bounded by $O(|V|/Nk \log k)$, for all $k \geq 1$, and thus, result (iii) follows from (2.7).

Next, we consider the question whether the PCP is a polynomial-time algorithm. In general, an algorithm is a polynomial-time algorithm if for each input the running time of the algorithm is bounded by a polynomial in the size of that input. The size $S$ of the input of the PCP is equal to the number of bits that is required to represent a subset $V \subseteq \{-1,0,1\}^N$, which is bounded by $\Omega(2^{|V|})$. Hence, using Theorem 2.2.2, it follows that the running time of the PCP is bounded by $O(S^2 \|\hat{a}\|_2 \log(\|\hat{a}\|_2^2))$. If, for all $V \subseteq \{-1,0,1\}^N$ for which an $\hat{a} \in \mathbb{R}^N$ exists such that $\hat{a} \cdot x \geq 1$, for all $x \in V$, there exists such an $\hat{a} \in \mathbb{R}^N$ with length $\|\hat{a}\|_2$ bounded by a polynomial in $N + |V|$, then the above would prove that the PCP is a polynomial-time algorithm. We prove below that this is not the case, by showing that there exists a subset $V \subseteq \{-1,0,1\}^N$ of polynomial size, for which every $\hat{a} \in \mathbb{R}^N$ with $\hat{a} \cdot x \geq 1$, for all $x \in V$, has a length $\|\hat{a}\|_2$ that is exponential in $N$. Furthermore, we use this example to prove that the PCP is not a polynomial-time algorithm. To prove that the PCP is not a polynomial-time algorithm, we need a lower bound for the number of additions of the PCP. In Corollary 2.2.2, Corollary 2.2.3, and Corollary 2.2.4 we present three lower bounds that can be obtained from the proof of Theorem 2.2.2. These bounds differ with respect to the conditions imposed; the stronger the conditions, the stronger the bound. We define $\hat{a} \leq a$, for two vectors $a, \hat{a} \in \mathbb{R}^N$, if $\hat{a}_i \leq a_i$, for all $i = 1, \ldots, N$.

**Corollary 2.2.2.** Let $\hat{a}$ be as in Theorem 2.2.2, and assume that $\hat{a} \in \mathbb{Z}^N$, and $\|\hat{a}\|_\infty \leq \|a\|_\infty$, for all $a \in \mathbb{Z}^N$ that satisfy $a \cdot x \geq 1$, for all $x \in V$. Then the number of additions $k$ of the PCP is bounded by

$$\|\hat{a}\|_\infty \leq k \leq N\|\hat{a}\|_2^2.$$  \hspace{1cm} (2.8)

**Corollary 2.2.3.** Let $\hat{a}$ be as in Theorem 2.2.2, and assume that $\hat{a} \in \mathbb{Z}^N$, and $\|\hat{a}\|_2 \leq \|a\|_2$, for all $a \in \mathbb{Z}^N$ that satisfy $a \cdot x \geq 1$, for all $x \in V$. Then the number
of additions $k$ of the PCP is bounded by\[ \frac{1}{N} \| \tilde{a} \|^2 \leq k \leq N \| \tilde{a} \|^2. \]

**Corollary 2.2.4.** Let $\tilde{a}$ be as in Theorem 2.2.2, and assume that $\tilde{a} \in \mathbb{N}^n$, $\tilde{a} \cdot x = 1$, for all $x \in V$, and $\tilde{a} \leq a$, for all $a \in \mathbb{Z}^n$ that satisfy $a \cdot x \geq 1$, for all $x \in V$. Then the number of additions $k$ of the PCP is bounded by\[ \| \tilde{a} \|^2 \leq k \leq N \| \tilde{a} \|^2. \]

We give two examples of pairs $(V, \tilde{a})$ that satisfy the conditions of Corollary 2.2.4. The first example is based on the classical construction [Minsky & Papert, 1969] of a two-layered device for the parity function, that consists of AND components in the first layer and a perceptron in the second layer. It is sometimes referred to as an example that shows that the PCP is not a polynomial-time algorithm, because the weight of the perceptron gets exponential; see for instance [Judd, 1991]. We argue below that this is not a correct conclusion since the input size of the considered example is also exponential. The second example presented in Theorem 2.2.4 below proves that the PCP indeed requires a time which is not polynomial in the input size.

**Theorem 2.2.3 (Minsky & Papert [1969])**. Let $N = 2^n$, for some $n \in \mathbb{N}$. Define the function $p : \mathbb{B}^n \rightarrow \{-1, +1\}$ by\[ p(y) = \begin{cases} +1 & \text{if } \sum_{i=1}^n y_i \equiv 0 \pmod{2} \\ -1 & \text{if } \sum_{i=1}^n y_i \equiv 1 \pmod{2}, \end{cases} \]
for all $y \in \mathbb{B}^n$. Define the function $u : \mathbb{R}^n \rightarrow \mathbb{R}^N$ by\[ u(y) = \left( \frac{1}{1}, \frac{y_1}{y_1}, \ldots, \frac{y_n}{y_n}, \frac{y_1y_2}{y_2}, \ldots, \frac{y_1y_2 \cdots y_{n-1}y_n}{y_n}, \ldots, \frac{y_1 \cdots y_n}{y_n} \right), \]
for all $y \in \mathbb{R}^n$, and an arbitrary fixed order of the overbraced terms. Let, finally, $k \in \mathbb{N}$ denote the number of additions that is executed by the PCP, when it is applied to the subset $V \subseteq \{-1, 0, 1\}^N$ defined by\[ V = \{ p(y)u(-y) \mid y \in \mathbb{B}^n \}. \]
Then $5^n \leq k \leq 10^n$.

**Proof.** Let $\tilde{a} = u(2, \ldots, 2)$, then $\tilde{a} \in \mathbb{N}^N$. First, we show that $\tilde{a} \cdot x = 1$, for all $x \in V$. Let $y \in \mathbb{B}^n$ and assume that $\sum_{i=1}^n y_i = k$, for some $k \in \{0, \ldots, n\}$, then one verifies that\[ \tilde{a} \cdot [p(y)u(-y)] = p(y) \sum_{i=0}^k \binom{k}{i} (-2)^i = p(y)(-1)^k = 1. \]
Next, we prove that $\tilde{a} \leq a$, for all $a \in \mathbb{Z}^N$ that satisfy $a \cdot x \geq 1$, for all $x \in V$. Let $a \in \mathbb{Z}^N$ with $a \cdot x \geq 1$, for all $x \in V$. Let $l \in \{1, \ldots, 2^n\}$, then we must show that $a_l \geq \tilde{a}_l = 2^k$, where $k \in \{0, \ldots, n\}$ is such that\[ \sum_{i=0}^{k-1} \binom{n}{i} < l \leq \sum_{i=0}^k \binom{n}{i}. \]
One easily verifies that whatever the order of the overbraced terms in the definition of \( u(y) \), there exists precisely one \( y \in B^* \) such that \( u_i(y) = 1 \), and \( u_i(y) = 0 \), for all \( i > 1 \). Furthermore, this \( y \) satisfies \( \sum_{i=1}^{k} y_i = k \). Let \( Z = \{ z \in B^* \mid z \leq y \} \), then obviously \( |Z| = 2^k \). Finally, if \( u_j(z) = 1 \), for some \( z \in Z \) and \( k' \in \{ 0, \ldots, k \} \) is such that
\[
\sum_{i=0}^{k'} \binom{n}{i} < j \leq \sum_{i=0}^{k} \binom{n}{i},
\]
then one verifies that
\[
\left( \sum_{z \leq y} p(z) u(-z) \right)_j = \sum_{i=0}^{k'} (-1)^i \binom{k-k'}{i}
\]
\[
= \begin{cases} 
0, & \text{if } 0 \leq k' < k \\
1, & \text{if } k' = k.
\end{cases}
\]
Hence, using \( a \cdot x \geq 1 \), for all \( x \in V \), we find
\[
\alpha_i = a \cdot \left[ \sum_{z \leq y} p(z) u(-z) \right]
\]
\[
= \sum_{z \leq y} a \cdot [p(z) u(-z)]
\]
\[
\geq \sum_{z \leq y} 1
\]
\[
= 2^k.
\]
The result now follows from Corollary 2.2.4, and using that
\[
\|\tilde{a}\|_2^2 = \sum_{i=0}^{n} \binom{n}{i} 2^i = 5^n,
\]
and \( N = 2^n \). □

Although Theorem 2.2.3 shows that the running time of the PCP must be exponential in \( n \), the size of the input is also exponential in \( n \) since \( N|V| = 4^n \). In order to prove that the PCP is not polynomial, we must find an example with \( |V| \) polynomial in \( N \) and \( \|\tilde{a}\|_2^2 \) exponential in \( N \). It is unlikely that the above example can be “thinned”, i.e., reducing the number of vectors in \( V \) such that it keeps the same minimal integer separating vector \( \delta \), because the proof of Theorem 2.2.3 "uses" all the vectors in \( V \). Minsky & Papert provide some results from which examples with the desired properties can be constructed, but these examples are quite complicated. A straightforward example was found in theorem by Parberry [1991], but it is not known whether it has been used in this context before.

**Theorem 2.2.4.** Let \( N = 2n \), for some \( n \in \mathbb{N} \), and define the vectors \( u_i, v_i \in \{-1, 0, 1\}^N \), \( i = 1, \ldots, n \), by
\[
u_i = e_i + \sum_{j=i+1}^{N-i} e_j - e_N,
\]
\[
\nu_i = -\sum_{j=1}^{i-1} e_j - \sum_{j=i+1}^{N-i} e_j + e_N,
\]
respectively, where \( e_j \in B^N \) denotes the \( j \)-th unit vector, i.e., with all zeros except a one on its \( j \)-th position. Let \( k \in \mathbb{N} \) denote the number of additions that is executed by the PCP, when it is applied to the subset \( V \subseteq \{-1, 0, 1\}^N \) defined by \( V = \{ u_i, v_i \}_{i=1}^n \).
Then \( \|\hat{a}\|_2^2 \leq k \leq N\|\hat{a}\|_2^2 \), where \( \hat{a} \in \mathbb{R}^N \) is defined by

\[
\hat{a} = \sum_{j=1}^{n} a^j e_j + \sum_{j=n+1}^{N-1} 2^{N-j} e_j + (2^n - 1)e_N
\]

\[
= (2^1, 2^2, \ldots, 2^{n-1}, 2^n, 2^{n-1}, \ldots, 2^2, 2^1, 2^n - 1).
\]

Proof. If \( a = (a_1, \ldots, a_N) \in \mathbb{R}^N \), then by definition

\[
a \cdot u_i = a_i + \sum_{j=n+1}^{N-1} a_j - a_N,
\]

\[
a \cdot v_i = -\sum_{j=1}^{i-1} a_j - \sum_{j=n+1}^{N-1} a_j + a_N.
\]

Hence, for all \( i = 1, \ldots, n \)

\[
\hat{a} \cdot u_i = 2^i + \sum_{j=n+1}^{N-1} 2^{N-j} - 2^n + 1 = 1
\]

\[
\hat{a} \cdot v_i = -\sum_{j=1}^{i-1} 2^j - \sum_{j=n+1}^{N-1} 2^{N-j} + 2^n - 1 = 1,
\]

which proves that \( \hat{a} \cdot x = 1 \) for all \( x \in V \).

Next, we show that \( \hat{a} \leq a \) for all \( a \in \mathbb{R}^N \) with \( a \cdot x \geq 1, x \in V \). Let \( a \in \mathbb{R}^N \) and assume that \( a \cdot u_i \geq 1 \) and \( a \cdot v_i \geq 1 \), for all \( i = 1, \ldots, n \). Then firstly, \( a \cdot u_i + a \cdot v_i \geq 2 \), for all \( i = 1, \ldots, n \). Hence, using (2.8) and (2.9) it follows that \( a_i \geq 2 + \sum_{j=1}^{i-1} a_j \), or, equivalently, \( a_i - 2^i \geq \sum_{j=1}^{i-1} (a_j - 2^j) \), for all \( i = 1, \ldots, n \). Hence, by induction, \( a_i \geq 2^i = \hat{a}_i \), for all \( i = 1, \ldots, n \).

Secondly, \( a \cdot u_i + a \cdot v_{i+1} \geq 2 \), for all \( i = 1, \ldots, n - 1 \). Hence, using (2.8) and (2.9) it follows that \( a_{N-i} \geq 2 + \sum_{j=1}^{i-1} a_j \), or, equivalently, \( a_{N-i} - 2^i \geq \sum_{j=1}^{i-1} (a_j - 2^j) \), for all \( i = 1, \ldots, n - 1 \). Using \( a_j \geq 2^j \), it follows that \( a_{N-i} \geq 2^i = \hat{a}_{N-i} \), for all \( i = 1, \ldots, n - 1 \).

Finally, from \( a \cdot v_n \geq 1 \) we deduce that \( a_N \geq 1 + \sum_{j=1}^{n-1} a_j \geq 1 + \sum_{j=1}^{n-1} 2^j = 2^n - 1 = \hat{a}_N \). Now the result follows directly from Corollary 2.2.4.

We conjecture that the lower bound of Theorem 2.2.4 is tight, i.e., we conjecture that there exists a GENERATION scheme of the PCP such that the number of additions for the example discussed in Theorem 2.2.4 is exactly \( \|\hat{a}\|_2^2 \). This conjecture is based on the observation that there exist numbers \( p_i, q_i \in \mathbb{N}, i = 1, \ldots, n \), such that \( \sum_{j=1}^{n} p_i u_j + \sum_{i=1}^{n} q_i v_i = \hat{a} \), which can be verified straightforwardly.

Theorem 2.2.4 gives an instance of \( \text{HFLP}_P \) of size \( O(N^2) \), for which the running time of the PCP is \( \Omega(\|\hat{a}\|_2^2) = \Omega(2^n) \). This proves the following corollary.

Corollary 2.2.5. The Perceptron Convergence Procedure is not a polynomial-time algorithm.

Since the numbers that occur in the instance defined in Theorem 2.2.4 are bounded, it follows that the PCP is not even a pseudo-polynomial-time algorithm; see for a formal discussion of pseudo-polynomial-time algorithms Garey & Johnson [1979].
2.2 One-layered perceptrons

We used the example presented in Theorem 2.2.4 to show that the PCP is not a polynomial-time algorithm. Originally, the example was used by Parberry [1991] to show that there exist $N$-input ILPs with weight $\|a\|_\infty$ as large as $2^{N+1}/2^N$, that cannot be replaced by a 1LP with a smaller weight that has the same functional behavior on the set $B^N$. Muroga [1971] has shown examples of ILPs with minimal weight bounded by $\Omega(2^N)$, but it is not known whether these examples can be used to obtain a stronger result than the one given in Theorem 2.2.4. Below we will show that both HPLP, and HPLP, can be solved efficiently. Hence, there seems no reason to consider the PCP any longer. However, we close our discussion on the PCP by showing that it is a quasi-polynomial-time algorithm—and very likely a polynomial-time algorithm—for instances of size $|V| = 2^{O(N)}$. To this end we need the following result from [Papadimitriou & Steiglitz, 1982], Chapter 13, Lemma 13.1.

**Proposition 2.2.5.** Let $\alpha, N \in \mathbb{N}$, $V \subseteq \{0, \pm1, \pm2, \ldots, \pm\alpha\}^N$ and $I = (\alpha N)^{N+1}$. Then the following statements are equivalent.

(i) There exists a vector $a \in \mathbb{R}^N$, such that $a \cdot x > 0$, for all $x \in V$.

(ii) There exists a vector $a \in \{0, \pm1, \pm2, \ldots, \pm\alpha\}^N$, such that $a \cdot x \geq 1$, for all $x \in V$.

If we take $\alpha = 1$, we find that the maximum minimum size of any solution of an instance of HPLP, is at most $N^{N+1}$. This, combined with the result of the Perceptron Convergence Theorem 2.2.2, implies that if there exists an $a \in \mathbb{R}^N$ such that $a \cdot x > 0$, for all $x \in V$, then the PCP has a running time which can be bounded by $O(|V|N^{2N+1} \log N)$. If $|V| = 2^{O(N)}$, then $|V|N^{2N+1} \log N = 2^{O(\log S \log \log S)}$, where $S = \Omega(N|V|)$ is the size of the input. Hence, in this case the PCP is a quasi-polynomial-time algorithm.

Using a more precise analysis, Muroga, Toda & Takasu [1960] have shown that the maximum minimum weight of a 1LP is at most $(N + 1)^{N/2}/2^N$; see also [Muroga, 1971; Parberry, 1991]. Weaker versions of this result have recently been rediscovered by various authors; see Hong [1987], Natarajan [1991], and Raghavan [1988], which are also cited in [Judd, 1991; Parberry, 1991; Shawe-Taylor, Anthony & Kern., 1992]. However, it has been conjectured that this upper bound is still too pessimistic and that a tighter upper bound is $2^{O(N)}$; see Hampson & Volper [1986], Tesar & Jansems [1988], and Tesauro [1987]. If this conjecture is true the PCP is in fact a polynomial-time algorithm in case that $|V| = 2^{O(N)}$, and the PCP may then even be competitive with the approach discussed in the following section.

In addition to a worst-case exponential running time, the PCP requires the existence of a solution. If the PCP is applied to a "no"-instance of HPLP, the PCP does not terminate. Although Minsky & Papert [1969] have shown that the vector $a$ cannot attain an infinite number of values, the above indicates that the PCP is not a good choice for testing the separability of two arbitrary finite sets. Especially since there exist efficient procedures for solving HPLP, and HPLP, these procedures are based on a linear programming approach, that is discussed in the following section.
2.2.5 An approach for HLP, based on linear programming

In this section we discuss some efficient approaches for solving HLP, and HLP+, that are based on some of the more recent results obtained in linear programming; see [Papadimitriou & Steiglitz, 1982; Schrijver, 1986] for an explanation of the basic concepts of linear programming. Consider the following problem.

Definition 2.2.12. (LINEAR PROGRAMMING FORMULATION 1)

**INPUT:** A finite set \( V \subseteq \{-1, 0, 1\}^N \), for some \( N \in \mathbb{N} \).

**OUTPUT:** \( \arg \max_{a} \ 0 \cdot a \)

\[ s.t. \quad x \cdot a \geq 1, \quad \text{for all} \ x \in V \]

\[ a \in \mathbb{R}^N \]

The above problem belongs to the class of linear programming problems, which are known to be solvable in polynomial time; see also [Papadimitriou & Steiglitz, 1982; Schrijver, 1986]. Furthermore, we may assume that any algorithm that solves this problem gives the output “infeasible”, if there does not exist an \( a \in \mathbb{R}^N \) with \( x \cdot a \geq 1 \), for all \( x \in V \), and gives the output \((0, a)\), if an \( a \in \mathbb{R}^N \) is found such that \( x \cdot a \geq 1 \), for all \( x \in V \). In other words, any algorithm that solves the above problem solves HLP+. Consequently, HLP+ can be solved in polynomial time.

The above LINEAR PROGRAMMING FORMULATION 1 is the most trivial linear programming formulation equivalent with HLP+. There are, however, various alternatives. One is given below, which corresponds to the problem of finding a weight vector \( a \) with minimal total weight \( \|a\| \).

Definition 2.2.13. (LINEAR PROGRAMMING FORMULATION 2)

**INPUT:** A finite set \( V \subseteq \{-1, 0, 1\}^N \), for some \( N \in \mathbb{N} \).

**OUTPUT:** \( \arg \min_{u, v} \ 1 \cdot (u + v) \)

\[ s.t. \quad x \cdot (u - v) \geq 1, \quad \text{for all} \ x \in V \]

\[ u, v \in \mathbb{R}_+^N \]

Finally, we consider HLP+. A standard approach is to add the constraints “\( a \) integer” and “\( u, v \) integer” to the two problem formulations given above, respectively. Although this leads to two perfectly correct problem formulations that are equivalent to HLP+, these problems belong to the class of Integer Linear Programming Problems, which are in general \( \mathcal{NP} \)-complete; see for instance Papadimitriou & Steiglitz [1982] or Schrijver [1986]. Furthermore, the constraint matrix of these problems, which consists of the elements of \( V \) taken as columns, is not totally unimodular (TUM); see [Papadimitriou & Steiglitz, 1982; Schrijver, 1986] for a definition of TUM and its properties. This implies that there is not a straightforward polynomial-time solution of the integer linear programming formulations. Still, it is not hard to see that HLP+ can be solved in polynomial time.

In order to solve an instance of HLP+, we first solve the corresponding instance of HLP, which can be done in polynomial time. This yields a solution \( a \) which has rational components, because any solution of a linear programming problem with
integer coefficients is rational. Multiply $a$ with the absolute value of the product of all the denominators of the components of $a$, then we obtain an integer vector $a'$ that solves the instance of HIPLP$_d$. The size of the obtained vector $a'$ can be easily estimated. Using the techniques of [Muroga, Toda & Takasu, 1960], it can be shown that both the nominator and the denominator of the rational components of $a$ are $O(N^N)$; see also Muroga [1971] and Parberry [1991]. Hence, the components of $a'$ are bounded by $2^{O(N^2 \log N)}$, which proves that this procedure is polynomial.

Remarks.

1. The linear programming can obviously be extended to solve instances of FLP, in polynomial time. However, the polynomial-time procedure described above for solving instances of HIPLP$_d$ does not apply to instances of HIPLP$_d$, since the bounds on the size of $a$ are valid for subsets of $\{-1, 0, 1\}^N$ only. We do not know whether FLP can be solved in polynomial time, although we know that HIPLP$_d = FLP_d$ can.

2. In the discussion about the different variants of the ILP loading problems, we have assumed that the sets $\Omega_0, \Omega_1$, and $V$ are explicitly specified by a listing of their elements. However, if the sets are specified in a more compact manner, for instance by giving a Boolean function in disjunctive normal form that defines $\Omega_1$ and $\Omega_0$, HIPLP is $\mathcal{NP}$-hard; see Pined & Simeone [1985].

3. Kearns, Li, Pitt & Valiant [1987] have discussed HBPLP, where the problem is to find a separating 0-1 vector, and showed that the decision variant of this problem is $\mathcal{NP}$-complete.

2.2.6 Unsolvable finite classification problems

So far we have only considered classification problems that can be solved by ILPs. However, many classification problems, including finite classification problems, cannot be solved by ILPs. In fact most finite classification problems cannot be solved by ILPs. This statement is explained below. Consider some finite set $V \subseteq \mathbb{R}^N$. A partition of $V$ in two disjoint subsets $\Omega_0$ and $\Omega_1$ is called a dichotomy of $V$; see Baum [1988]. If $V$ contains $s$ vectors, the total number of dichotomies is $2^s$. Now the question rises how many of these dichotomies are linearly separable, i.e., for how many choices of a partition $\Omega_0, \Omega_1 \subseteq V$ is the corresponding instance of ILP a “yes”-instance? To answer this question we first consider the special case in which the vectors in $V$ are in general position, which is defined as follows. A set of vectors in $\mathbb{R}^N$ is in general position, if there is no hyperplane that contains more than $N$ of the given vectors. Furthermore, we say that a dichotomy is homogeneously linearly separable, if it is separable by a hyperplane passing through the origin. A well-known result due to Winder gives the exact number of homogeneously linearly separable dichotomies of a set of vectors in general position; see Cover [1965] and Muroga [1971].

Proposition 2.2.6 (Winder). The number of homogeneously linearly separable dichotomies of a set $V \subseteq \mathbb{R}^N$ of $s$ vectors in general position is

$$d(s, N) = 2 \sum_{i=0}^{N+1} \binom{s-1}{i},$$

(2.10)
From this we can obtain an answer for the above question. Firstly, the number of linearly separable dichotomies of a set $V \subseteq \mathbb{R}^N$ of $s$ vectors is equal to the number of homogeneously linearly separable dichotomies of a set $V' \subseteq \mathbb{R}^{N+1}$ of $s$ vectors; cf. Proposition 2.2.3. Secondly, the number of homogeneously linearly separable dichotomies of a set $V' \subseteq \mathbb{R}^{N+1}$ of $s$ vectors is less than or equal to the number of homogeneously linearly separable dichotomies of a set $V'' \subseteq \mathbb{R}^N$ of $s$ vectors in general position. This yields the following corollary.

**Corollary 2.2.6 (Baum [1988]).** The number of linearly separable dichotomies of a set $V \subseteq \mathbb{R}^N$ of $s$ vectors is less than or equal to $d(s, N + 1)$, which for $s \geq 3N$ is less than $4s^N / N!$.

This result implies that for a given set $V \subseteq \mathbb{R}^N$ of $s$ vectors, out of the $2^s$ different instances of 1LP with $\Omega_0 \cup \Omega_1 = V$, only polynomially many of them are "yes"-instances. If we consider the special case when $V = \mathbb{B}^N$, then the result becomes even more pronounced; while the number of Boolean functions $f : \mathbb{B}^N \rightarrow \mathbb{B}$ of $N$ variables is $2^{2^N}$, the number of distinct 1LPs of $N$ variables is approximately $2^{\mathcal{N}}$, see Muroga [1971], who uses a similar approach as above to obtain this upper bound, and gives a matching lower bound. Obviously, the gap becomes enormous for large $N$. For $N = 2$ one can easily show that 14 of the 16 Boolean functions can be represented as a 1LP. One of the two functions of two variables that cannot be represented as a 1LP is the EOR-function. In Figure 2.4 a corresponding classification problem is given. One can easily see that the subsets $\Omega_1$ and $\Omega_0$ shown in Figure 2.4 cannot be linearly separated.

![Figure 2.4: The Exclusive-Or problem presented as an infinite classification problem.](image)

In order to prove that the EOR-problem cannot be solved with a 1LP, we consider the 1LP with $\Omega_0 = \{(1, 0), (0, 1)\}$ and $\Omega_1 = \{(0, 0), (1, 1)\}$. The corresponding 1LP instance is given by $V = \{(1, 0, 1), (0, 1, 1), (0, 0, -1), (-1, -1, -1)\}$; see Proposition 2.2.3. That is instance is a "no"-instance follows from Corollary 2.2.1, since we have that

$$0 = \frac{1}{4}(1, 0, 1) + \frac{1}{4}(0, 1, 1) + \frac{1}{4}(0, 0, -1) + \frac{1}{4}(-1, -1, -1).$$

Minsky & Papert [1969] were among the first to show the limitations of the 1LPs by noting that it could not solve the EOR-problem, nor the parity problem and the connectivity problem. This was used by Minsky & Papert to plead for perceptrons with more layers, although they could not extend their analytic results to such devices.
2.3 Multi-layered perceptrons

After Minsky & Papert [1969] had successfully demonstrated the weaknesses of one-layered perceptrons, the interest in this kind of devices dropped to a modest level. At the time they published their results on one-layered perceptrons, multi-layered perceptrons were already being studied; see the work of Gamba and co-workers [Borsellino & Gamba, 1961; Gamba, Gambarini, Palmieri & Sanna, 1961], and also [Nilsson, 1965]. It was soon to be recognized that the capabilities of multi-layered perceptrons were superior to those of one-layered perceptrons; see also [Minsky & Papert, 1969]. Furthermore, Amari [1967] showed that learning algorithms like the Perceptron Convergence Procedure, discussed in Section 2.2, could be generalized to more complex devices like the multi-layered perceptrons; see also [Amari, 1990]. However, for some reasons multi-layered perceptrons and in general neural networks fell into oblivion.

Interest in neural networks revived through the work of Hopfield, which was partly based on other work of Amari, and by the work of Rumelhart, Hinton and the other members of the PDP-research group; see [Hopfield, 1982; Rumelhart & McClelland, 1986], and also the books by Aarts & Korst (1989), Hecht-Nielsen [1990] and Kosko [1992]. Particularly, the (re-)discovery of the back-propagation algorithm, a learning algorithm for multi-layered perceptrons, and its publication by Rumelhart, Hinton & Williams [1986] caused an avalanche of papers on the theory and application of multi-layered perceptrons; see for a recent review Xu, Klaas & Yuille [1992]. For some references to earlier appearances of algorithms related to the back-propagation algorithm we refer to [Baum, 1988; Rumelhart, Hinton & Williams, 1986; Werbos, 1990]. In this section we introduce the multi-layered perceptron and some relevant complexity results that have recently appeared in the neural network literature.

2.3.1 The construction of multi-layered perceptrons

In Section 2.2 we studied one-layered perceptrons with one output only. In general, a 1LP may have an arbitrary number of output nodes by simply considering a number of 1LPs that use the same set of inputs. If the number of outputs of a certain 1LP corresponds to the number of inputs of another 1LP, they can be combined to construct a two-layered perceptron (2LP). To be more specific we consider two 1LPs, the first with N inputs and L output nodes, the second with L inputs and K output nodes. If we place the second 1LP "on top" of the first, and connect the outputs of the first to the inputs of the second, we obtain a 2LP with N inputs and K output nodes; see Figure 2.5 for an example of a 2LP with $N = 3$ inputs and $K = 4$ outputs. The output nodes of the 1LP that forms the first layer become the hidden nodes of the 2LP. The layer of hidden nodes is called the hidden layer.

The above construction of 2LPs can easily be extended to construct three-layered perceptrons (3LPs) and in general $m$-layered perceptrons ($m$LPs), for some $m \in \mathbb{N}$, by putting $m$ one-layered perceptrons with matching sizes "on top" of each other. In a $m$LP there are $m - 1$ hidden layers, the first one being connected to the inputs, the $(m - 1)$th one being connected to the outputs. A multi-layered perceptron (MLP) is an $m$-layered perceptron with an arbitrary number of layers, i.e., $m$ may be any
Figure 2.5: An example of a two-layered perceptron with two inputs ($N = 2$), three hidden nodes ($L = 3$), and one output node ($K = 1$).

The nodes used in the above construction of MLPs may use arbitrary response functions; see also Section 2.2. Let $\Sigma \subseteq \mathbb{R} \rightarrow \mathbb{R}$ be a collection of response functions. Then we speak of $\Sigma$-MLPs, if the nodes all use a response function in $\Sigma$. The following definition identifies all possible functions computed by $\Sigma$-MLPs.

**Definition 2.3.1.** Let $m, N, K \in \mathbb{N}$ and let $A_N$ denote the set of all affine functions from $\mathbb{R}^N$ to $\mathbb{R}$ defined by

$$A_N = \{f : \mathbb{R}^N \rightarrow \mathbb{R} \mid f(x) = a \cdot x + b, x \in \mathbb{R}^N, a \in \mathbb{R}^N, b \in \mathbb{R}\}.$$  

Then the set $\Sigma$-$\mathcal{R}_{m,N,K} \subseteq \mathbb{R}^N \rightarrow \mathbb{R}^K$ of all functions that can be computed by a $\Sigma$-MLP with $m$ layers, $N$ inputs, and $K$ outputs, is defined iteratively by

$$\Sigma$-$\mathcal{R}_{1,N,K} = \{(\gamma_1 \circ f_1, \ldots, \gamma_K \circ f_K) \mid \gamma_i \in \Sigma, f_i \in A_N, i = 1, \ldots, K\},$$

and, for $m > 1$,

$$\Sigma$-$\mathcal{R}_{m,N,K} = \{g \circ h \mid g \in \Sigma$-$\mathcal{R}_{1,L,K}, h \in \Sigma$-$\mathcal{R}_{m-1,N,L}, L \in \mathbb{N}\}.$$  

The general form of $\Sigma$-$\mathcal{R}_{m,N,K}$ given in the above definition is due to the strict stratification we have imposed on the considered MLPs, which implies that there are only connections between subsequent layers. This restriction corresponds with the definition of feednext networks of [Muroga, 1971]. We do not consider the more general unrestricted feedforward networks in which connections between nodes may cross a layer. The reasons for posing the strict stratification restrictions are the
2.3 Multi-layered perceptrons

following.

- It simplifies the analysis of the capabilities of multi-layered perceptrons.
- Most papers on multi-layered perceptrons use this restriction.
- Usually, nodes in the same layer of a multi-layered perceptron use the same type of response function. By imposing the above restriction the inputs of a node always are all of the same “type”, i.e., they all originate from the same type of response function, which implies that there are no “dimension” clashes.

Although the major part of this chapter concerns results for MLPs with the above restriction, we sometimes give the corresponding result for MLPs without the restriction, in order to compare their capabilities. Furthermore, we consider $\Sigma$-MLPs for which $\Sigma$ contains a set of arbitrary non-linear functions. It makes no sense to use linear or affine response functions in the hidden layers, as one can easily see that such MLPs could be collapsed onto one layer. This can be formulated as follows.

**Proposition 2.3.1.** Let $\Sigma = A_1$. Then $\Sigma$-R$^{m,N,1}_N = A_N$.

However, sometimes linear response functions are used in the output layer; see Section 2.3.5. In the remainder of this chapter we consider $\Sigma$-MLPs with one output only, but the results are easily extended to $\Sigma$-MLPs with more than one output. Finally, we note that we often use MLPs to denote $\theta$-MLPs, i.e., multi-layered perceptrons that use the hard-limiting response function denoted by $\theta$.

![Diagram of two-layered perceptrons](image)

Figure 2.6: Two-layered perceptrons for EOR and parity. In (a) and (b) the output equals $f(x) = \text{EOR}(x_1, x_2) = \text{PAR}^2(x_1, x_2)$, in (c) the output equals $f(x) = \text{PAR}^3(x_1, x_2, x_3)$; see the text for a definition of PAR. All unlabeled connections have weight 1.

It is only useful to consider MLPs if they have greater capabilities than 1LPs, in the sense that they can deal with a larger set of problems. A first and easy way of proving that MLPs indeed have greater capabilities than 1LPs is by showing that the EOR-problem can be solved with a 2LP. In Figure 2.6a an example of a 2LP with 2 hidden nodes and 1 output node is given, that computes the EOR-function defined by $f(x) = \text{EOR}(x_1, x_2)$, for all $x_1, x_2 \in B$. The 2LP in Figure 2.6a is
minimal with respect to the number of nodes, as each 2LP with only 1 hidden node and without connections that cross the first layer can be reduced to a 1LP that represents the same function. Here we see the first consequence of our restriction to feednext MLPs. In case that we consider unrestricted feedforward networks, we can construct a solution with only one hidden node; see Figure 2.6b. The 2LP of Figure 2.6a can easily be extended to find a 2LP with $N$ hidden nodes and one output node that computes the parity function of $N$ inputs, which equals 1, if the sum of inputs is odd, and 0 otherwise; see Figure 2.6c for the case $N = 3$. For arbitrary $N \in \mathbb{N}$ we have

\[
\text{PAR}^N(x_1, \ldots, x_N) = \theta(-1 + \sum_{i=1}^{N} (-1)^{i-1} \theta(-i + \sum_{j=1}^{N} x_j)).
\]

(2.11)

A less well-known 2LP implementation of the parity function is, for $N = 2k$,

\[
\text{PAR}^N(x_1, \ldots, x_N) = \theta(-N + 1 + \sum_{i=1}^{N} 2\theta(-1 + \sum_{j=0}^{k-1} 2(x_{i+j} - x_{i+j+k}))),
\]

(2.12)

where $x_l = x_{l-N}$ if $l > N$. This result can be extended to odd values using the identity $\text{PAR}^N(x_1, \ldots, x_N) = \text{PAR}^{N+1}(x_1, \ldots, x_N, 0)$. The verification of (2.11) and (2.12) is left to the reader. The unrestricted feedforward network presented in Figure 2.6b can either be extended to find a feedforward network of two layers and $\lceil N/2 \rceil + 1$ nodes for the parity function, or a feedforward network of $\lceil \log_2 N \rceil + 1$ layers and the same number of nodes, where $\lfloor x \rfloor$ denotes the largest integer smaller than or equal to $x$; see also Muroga [1971], and Chapter 4 for more about the parity function.

### 2.3.2 Classification with MLPs

MLPs may be considered for solving classification problems in the same way as 1LPs. Again we restrict ourselves to binary classification problems (BCPs); see Definition 2.2.1 and Chapter 3. Let $(\mathbb{R}^N, \mathbb{B}, \{\Omega_0, \Omega_1\})$ represents a BCP, for some $N$, then similar as for 1LPs we name the problem of finding an MLP that solves this BCP, for all $x \in \mathbb{R}^N$, a **loading problem**. Let $m$ be a fixed integer, then we define the following decision problem.

**Definition 2.3.2.** *(mLP Loading Problem (mLPLP); Decision Variant)*

**INPUT:** Two disjoint sets $\Omega_0, \Omega_1 \subseteq \mathbb{R}^N$, for some $N \in \mathbb{N}$.

**OUTPUT:** Does there exist an mLP that labels $\Omega_1$ with 1, and $\Omega_0$ with 0?

The corresponding search problem is defined as follows.

**Definition 2.3.3.** *(mLP Loading Problem (mLPLP); Search Variant)*

**INPUT:** Two disjoint sets $\Omega_0, \Omega_1 \subseteq \mathbb{R}^N$, for some $N \in \mathbb{N}$.

**OUTPUT:** The weights of an mLP that labels $\Omega_1$ with 1, and $\Omega_0$ with 0.

The output of mLPLP, is a list containing the size of each of the $m - 1$ hidden layers and all the connection weights. Strictly speaking mLFLP, is not correctly formulated.
since the dimension of the output is unbounded; see also Chapter 4. This is mended below when we consider the complexity of some of the finite MLP loading problems.

The capabilities of MLPs for solving classification problems can be examined by describing their decision regions. The decision regions of an MLP with \( m \) layers, \( N \) inputs, 1 output, and represented by \( f \in \Theta-R_{m,N} \), are \( f^- \) and \( f^+ \). Lippmann [1987] was one of the first to describe the decision regions of MLPs graphically. He argued that the decision regions of 1LP, 2LP, and 3LPs are affine halfspaces, intersections of affine halfspaces, and arbitrary subsets with piece-wise linear bounds, respectively, or the complements of such subsets. Furthermore, he argued that there is never a need for more than three layers, because a further increase of the number of layers does not further increase the complexity of the decision region. Later, it was shown that 2LPs can also classify subsets that are not convex or concave; see for examples of such subsets [Huang & Lippmann, 1987; Wieland & Leighton, 1987; Makhoul, Schwartz & El-Jaroudi, 1989; Li, 1991]. Some other examples are given in Figure 2.7, see also Chapter 5, where we present some methodical approaches for verifying that these subsets are indeed classifiable with a 2LP.

![Figure 2.7: Three examples of subsets in \( \mathbb{R}^2 \) that can be classified with a 2LP.](image)

Note that these examples show that linear separability is not a necessary condition for two subsets \( \Omega_1 \) and \( \Omega_0 \) to be classifiable with a 2LP. Finally, we mention the result of Gibson & Cowan [1990], who give a "no"-instance of 2PLP, i.e., they present a partition of \( \mathbb{R}^2 \) that cannot correspond to the decision regions of a 2LP; see also Section 5.5.1. Despite the mentioned results it turned out that the characterization of the decision regions of 2LPs is not a trivial problem. It is a problem in which combinatorial and geometrical properties are subtly mixed. To show some of this subtlety we consider the subset given in Figure 2.8 and prove that it is the decision region of a 2LP.

**Proposition 2.3.2.** Let \( \Omega_1 \) be the shaded region given in Figure 2.8 and \( \Omega_0 = \mathbb{R}^2 - \Omega_1 \). Then this is a "yes"-instance of 2PLP.

We give an informal proof of Proposition 2.3.2. It is sufficient if we show that a 2LP represented by \( f \in \Theta-R_{2,2,1} \) exists such that \( f^- (1) \) corresponds to the shaded region.
given in Figure 2.8. Consider the 2LP given by
\[ f(x_1, x_2) = \theta(-37 + 5\theta(3x_1 + x_2 + 0) + 5\theta(-3x_1 - x_2 + 51) + 5\theta(-2x_1 + x_2 + 18) + 4\theta(-2x_1 + 8x_2 - 67) + 4\theta(-3x_1 - 3x_2 + 30) + 2\theta(x_1 + x_2 - 19) + 2\theta(x_1 - 3x_2 + 14) + \theta(x_1 + 3x_2 - 20) + \theta(3x_1 - x_2 - 25) + \theta(x_1 - 3x_2 - 3) + \theta(-x_1 - x_2 + 7) + \theta(-2x_1 + x_2 + 5)), \]
then one can easily verify that \( f^* (1) \) corresponds to the shaded region given in Figure 2.8, for instance using a computer drawing program.

The subtlety of the subset given by Figure 2.8 lies in the choice of the slopes of the lines marked with an arrow. These slopes are such that the lines intersect at a point just outside the convex hull of the given subset. If the slopes of these lines are altered such that the point of intersection falls within the convex hull, it can still be classified with a 2LP but more hidden nodes are required. A detailed study of the decision regions of 2LPs is presented in Chapter 5, that considers arbitrary infinite classification problems. In the following subsection we discuss the capabilities of MLPs for solving finite classification problems.

2.3.3 Finite loading of MLPs

In Section 2.2.6 we showed that 1LPs can only solve a small fraction of the total number of finite classification problems. In this section we investigate whether the capabilities of MLPs are superior compared to those of 1LPs, with respect to the
number of finite classification problems that can be solved. We present some results from literature that show that they are indeed capable of solving a much larger fraction of the total number of finite classification problems. In fact, we show that all finite classification problems can be solved by some 2LP, that has a sufficient large number of hidden nodes. Furthermore, we present bounds on the number of hidden nodes that any MLP must have, in order to be able to solve all finite classification problems.

We start with the definition of the decision variant of two finite MLP loading problems. The first problem considers an arbitrary fixed $m \in \mathbb{N}$, denoting the number of layers of the considered MLP.

**Definition 2.3.4. (Finite MLP loading problem (FMLLP))**

**INPUT:** Two finite disjoint sets $\Omega_0, \Omega_1 \subseteq \mathbb{R}^N$, and $m - 1$ numbers $l_1, \ldots, l_{m-1}$, for some $N \in \mathbb{N}$.

**OUTPUT:** Does there exist an MLP with $m$ layers, $N$ inputs, 1 output, and $l_i$ nodes in the $i$-th hidden layer, that labels $\Omega_1$ with 1 and $\Omega_0$ with 0?

In the second problem the number of layers is one of the inputs of the problem.

**Definition 2.3.5. (Finite MLP loading problem (FMLP))**

**INPUT:** Two finite disjoint sets $\Omega_0, \Omega_1 \subseteq \mathbb{R}^N$, and $M - 1$ numbers $l_1, \ldots, l_{M-1}$, for some $N, M \in \mathbb{N}$.

**OUTPUT:** Does there exist an MLP with $M$ layers, $N$ inputs, 1 output, and $l_i$ nodes in the $i$-th hidden layer, that labels $\Omega_1$ with 1 and $\Omega_0$ with 0?

First, we consider $\text{FMLLP}$. Below we present we present a result based on the work of Baum [1988], that gives an upper and a lower bound for the number of hidden nodes required by a 2LP that can solve all finite classification problems. It is based on the theory of dichotomies; see for more on dichotomies Section 2.2.6, and the papers by Baum [1988], Baum [1990], Cover [1965], Huang & Huang [1991], and Makhoul, Schwartz & El-Jaroudi [1989]. Using a counting argument Baum derives a lower bound $c s / ((N + 1) \log s)$, for some $c > 0$, for the number of hidden nodes that a 2LP must have to be able to form any dichotomy of $s$ points in $\mathbb{R}^N$. However, Baum gives no explicit proof of this result. In the following theorem we prove this lower bound with $c = \frac{1}{2}$. However, it is not hard to improve on this value using a more refined analysis.

**Theorem 2.3.1 (Baum [1988]).** Let $L(s, N) \in \mathbb{N}$ be the smallest number such that all instances $\Omega_0, \Omega_1 \subseteq \mathbb{R}^N$ of $\text{FMLLP}$, with $|\Omega_1 \cup \Omega_0| = s \geq 2$ and $l_i \geq L(s, N)$ are “yes”-instances. Then we have

$$s \frac{3(N + 1) \log s}{3(N + 1) \log s} \leq L(s, N) \leq s - 1. \quad (2.13)$$

**Proof.** It is a well-known result that $s - 1$ hidden nodes are sufficient to classify two disjoint finite subsets with a total of $s$ vectors; see Nilsson [1965] or Huang & Huang [1991].

Using Corollary 2.2.6, it follows that the number $K(s, N)$ of different 1LPs defined on $s$ points in $\mathbb{R}^N$ satisfies $K(s, N) \leq 4s^N / N! \leq s^N$, if $s \geq 3N \geq 3$. On the other
hand, if \( s \leq 3N \), then (2.10) implies that
\[
K(s, N) \leq d(s, N + 1) = 2 \sum_{i=0}^{N} \left( \frac{s - 1}{i} \right) \\
\leq 2 \sum_{i=0}^{N} \left( \frac{3N - 1}{i} \right) \\
\leq 2 \sum_{i=0}^{3N-1} \left( \frac{3N - 1}{i} \right) \\
= 2^{3N} \\
\leq s^{3N},
\]

since \( s \geq 2 \).

Consequently, the number of different 2LPs with \( l \) hidden nodes and defined on \( s \) points, is at most
\[
[K(s, N)]^{l} K(s, l) \leq s^{3(N+1)l} = 2^{(N+1)l} \log s.
\]

This proves that if \( 3(N+1)l \log s < s \), then the number of different 2LPs with \( l \) hidden nodes and defined on \( s \) points, is less than \( 2^{s} \), which implies that some dichotomies of the \( s \) points cannot be formed.

Baum has shown that much tighter bounds can be derived if the points in the set \( V = \Omega \cup \Omega_{0} \) are in general position; see Baum [1988] and also Section 2.2.6. In this case it can be shown that
\[
\left\lfloor \frac{(s - 1)}{N} \right\rfloor \leq L(s, N) \leq \left\lfloor \frac{s}{N} \right\rfloor + 1,
\]

where \( \lfloor x \rfloor \) denotes the smallest integer greater than or equal to \( x \).

Next, we discuss some similar results for MLPs with more than two layers. Of course one hopes that the required number of hidden nodes is less than for the 2LP. Using a similar approach as used in the proof of Theorem 2.3.1, one can show that this hope is in vain.

**Proposition 2.3.3** (Baum [1988]). There exists a \( c > 0 \) for which the following holds. Let the total number of connections \( N_{1} + l_{1} + l_{2} + \cdots + l_{M-1} + l_{M-1} \) be less than \( cs/\log s \), for some \( N, M \in \mathbb{N} \) and \( s \geq 2 \). Then there exist "no"-instances \( \Omega_{0}, \Omega_{1} \subseteq \mathbb{R}^{N} \) of \( vMLP_{LP} \) with \( \left| \Omega_{1} \cup \Omega_{0} \right| = s \).

**Proposition 2.3.4** (Baum [1988]). There exists a \( c > 0 \) for which the following holds. Let the total number of nodes \( l_{1} + l_{2} + \cdots + l_{M-1} + 1 \) be less than \( c\sqrt{s}/\log s \), for some \( N, M \in \mathbb{N} \) and \( s \geq 2 \). Then there exist "no"-instances \( \Omega_{0}, \Omega_{1} \subseteq \mathbb{R}^{N} \) of \( vMLP_{LP} \) with \( \left| \Omega_{1} \cup \Omega_{0} \right| = s \).

Muroga has shown that a similar result holds for unrestricted feedforward networks. In fact, he proves the same lower bound as given in Proposition 2.3.4, with \( c = 2 \); see [Muroga, 1971]. It is often argued that one can use MLPs with a smaller total number of nodes if one uses more layers. The above results imply that adding layers is not the ultimate answer for having too many hidden nodes. On the other hand, for a specific instance defined by \( \Omega_{0}, \Omega_{1} \subseteq \mathbb{R}^{N} \), the number of hidden nodes required to classify \( \Omega_{1} \) and \( \Omega_{0} \) may be much smaller than the bounds given above. The next
question is then how one can find such an MLP. In the following section we consider the complexity of loading problems, and show that already for MLPs with two layers and two nodes, the loading problem becomes \(\mathcal{NP}\)-complete.

### 2.3.4 The complexity of loading MLPs

In this section we discuss the complexity of the loading problems for MLPs defined in the previous subsection. To this end we first consider the following related loading problems.

**Definition 2.3.6. (Hypercube MLP Loading Problem (hMLP))**

**INPUT:** Two disjoint sets \(\Omega_0, \Omega_1 \subseteq \mathbb{B}^N\) and \(m-1\) \(L\)-bit unary numbers \(l_1, \ldots, l_{m-1}\), for some \(N, L \in \mathbb{N}\).

**OUTPUT:** Does there exist an MLP with \(m\) layers, \(N\) inputs, 1 output, and \(l_i\) nodes in the \(i\)-th hidden layer, that labels \(\Omega_i\) with 1 and \(\Omega_0\) with 0?

**Definition 2.3.7. (Hypercube MLP Loading Problem (hMLP))**

**INPUT:** Two disjoint sets \(\Omega_0, \Omega_1 \subseteq \mathbb{B}^N\) and \(m-1\) \(L\)-bit unary numbers \(l_1, \ldots, l_{m-1}\), for some \(N, M, L \in \mathbb{N}\).

**OUTPUT:** Does there exist an MLP with \(M\) layers, \(N\) inputs, 1 output, and \(l_i\) nodes in the \(i\)-th hidden layer, that labels \(\Omega_i\) with 1 and \(\Omega_0\) with 0?

In Theorem 2.3.2 below we show that hMLP and hMLP are in \(\mathcal{NP}\); see for a definition of \(\mathcal{NP}\) [Garey & Johnson, 1979]. The main idea of the proof of Theorem 2.3.2 is based on similar approaches presented by Judd [1991] and Blum & Rivest [1992]. The unary encodings of \(l_i\) are essential in the proof, because with the standard binary encoding the size of any certificate of a "yes"-instance, which corresponds to a description of an MLP, is exponential in the size of the input.

**Theorem 2.3.2.** hMLP \(\in \mathcal{NP}\) and hMLP \(\in \mathcal{NP}\).

**Proof.** We prove the result for hMLP, the result for hMLP follows by replacing \(M\), a variable, with \(m\), a fixed constant.

Consider an instance of hMLP given by two disjoint sets \(\Omega_0, \Omega_1 \subseteq \mathbb{B}^N\) and \(M-1\) \(L\)-bit unary numbers \(l_1, \ldots, l_{M-1}\), for some \(N, M, L \in \mathbb{N}\). Assume that \(|\Omega_0 \cup \Omega_1| = s \geq 1, M \geq 2\), and let \(S\) denote the size of this instance. Then

\[ S = \Omega_l sN + (M-1)L. \]  

(2.14)

Assume that this instance is a "yes"-instance. Then there exist an MLP with \(M\) layers, \(N\) inputs, 1 output, and \(l_i\) nodes in the \(i\)-th hidden layer, that labels \(\Omega_i\) with 1 and \(\Omega_0\) with 0. Let further \(K = N + L\). Then, \(N \leq K\) and \(L \leq K\), for all \(i = 1, \ldots, M-1\). Hence, using the results of Section 2.2, it follows that there exists an MLP with \(M\) layers, \(N\) inputs, 1 output, and \(l_i\) nodes in the \(i\)-th hidden layer, with the additional constraint that all the weights are integer and bounded by \(K^{K+1}\), that labels \(\Omega_i\) with 1 and \(\Omega_0\) with 0; see Proposition 2.2.5 and also Proposition 2.2.4.

The summation of \(K\) integers of size at most \(K^{K+1}\) can be done in a time bounded by \(O(K^2)\). Consequently, the verification that this MLP labels \(\Omega_i\) with 1 and \(\Omega_0\) with 0
with 0, can be done in a time bounded by \( O(sMLK^3) = O(sML(N + L)^3) \), which by (2.14) is bounded by a polynomial in \( S \), the size of the instance. \( \square \)

Below we prove that \( HMLPLP \) is \( \mathcal{NP} \)-complete, after we have discussed some related results that have appeared in literature. The first one is the result by Judd, who considers a variant of \( HMLPLP \) for unrestricted feedforward networks; see [Judd, 1988; Judd, 1991]. Furthermore, in his version not only the number of nodes is specified, but also the connection pattern. Note that this makes the unary encoding superfluous. Judd proves that his version of \( HMLPLP \) is \( \mathcal{NP} \)-complete by a reduction from \( 3SAT \); see for a definition of \( 3SAT \) Garey & Johnson [1979]. Given an instance of \( 3SAT \) with \( v \) variables and \( c \) clauses, he constructs a feedforward network with two layers, \( 2v + c \) hidden nodes, and \( 4v + c \) outputs, that solves a certain instance of the corresponding loading problem, if and only if there exists a truth assignment for the \( 3SAT \) instance.

A stronger result is given by Blum & Rivest [1992], who consider a variant of \( H2LPLP \) which we call \( H2L2HLP \). This problem is defined as follows.

**Definition 2.3.8.** (Hypercube 2LP 2-Hidden Loading Problem (H2L2HLP))

**INPUT:** Two disjoint sets \( \Omega_0, \Omega_1 \subseteq \mathbb{R}^N \), for some \( N \in \mathbb{N} \).

**OUTPUT:** Does there exist a 2LP with two hidden nodes that labels \( \Omega_1 \) with 1, and \( \Omega_0 \) with 0?

**Proposition 2.3.5 (Blum & Rivest [1992]).** \( H2L2HLP \) is \( \mathcal{NP} \)-complete.

Proving that \( H2L2HLP \) is in \( \mathcal{NP} \) follows similarly as for \( H2LPLP \); see Theorem 2.3.2. Blum & Rivest prove the \( \mathcal{NP} \)-hardness of \( H2L2HLP \) by a reduction from \( SET-SPLITTING \); see Blum & Rivest [1992] and also Garey & Johnson [1979], for a definition of \( SET-SPLITTING \). Since \( H2L2HLP \) is a special case of both \( H2LPLP \) and \( HMLPLP \), we obtain the following results.

**Corollary 2.3.1.** \( H2LPLP \) is \( \mathcal{NP} \)-complete.

**Corollary 2.3.2.** \( HMLPLP \) is \( \mathcal{NP} \)-complete.

Since \( H2LPLP \) and \( HMLPLP \) are special cases of \( F2LPLP \) and \( FMLPLP \), respectively, this yields the following result.

**Corollary 2.3.3.** \( F2LPLP \) is \( \mathcal{NP} \)-hard.

**Corollary 2.3.4.** \( FMLPLP \) is \( \mathcal{NP} \)-hard.

Blum & Rivest have also shown that the following restricted problems are \( \mathcal{NP} \)-complete; see [Blum & Rivest, 1992].

(i) \( H2L2HLP \) in which some or all of the weights of the connections between the inputs and the first hidden node are required to equal the corresponding weights of the second hidden node, with possibly different thresholds, and in which some or all of the weights may be restricted to the set \( \{-1, +1\} \).

(ii) \( H2L2HLP \) in which the output node is forced to compute the AND function of the outputs of the hidden nodes.
2.3 Multi-layered perceptrons

Lin & Vitter [1991] have proved the \( \mathcal{NP} \)-completeness of the loading problem corresponding to an unrestricted feedforward network with two nodes, of which one is a hidden node and the other is the output node.

This completes our discussion of the complexity of loading. Several approaches have been proposed to solve MLP loading problems. Some early approaches reformulated FMLP as a mixed integer linear programming problem, for which some solution approaches are known that can solve small instances; see Muroga [1971] and Hughes [1967]. Since the results were discouraging, people lost their interest in MLPs. MLPs were brought back to life by the publication of the back-propagation learning algorithm by Rumelhart, Hinton & Williams [1986]. As several authors have pointed out, this algorithm can be viewed as an algorithm for approximately solving FMLP, that is in many ways similar to simulated annealing, a well-known procedure for approximately solving combinatorial optimization problems; see Bottou [1991], White [1989], and, for an introduction to simulated annealing, Aarts & Korst [1989]. In addition to its capabilities for approximately solving FMLP, the back-propagation algorithm has a very elegant and efficient implementation, such that it can be executed by "the MLP itself"; see Rumelhart, Hinton & Williams [1986]. It has had a great impact on the research of MLPs and neural networks in general, both theoretically and practically; see for discussions of the application of the back-propagation algorithm [Xu, Klasa & Yuille, 1992; Tesauro, 1987; Tesauro & Janssens, 1988; Weigend, Huberman & Rumelhart, 1990].

The back-propagation algorithm uses an arbitrary differentiable response function, so it is natural to investigate the capabilities of MLPs that use an arbitrary response function, for approximating a given function. Some of the results are discussed in the following section, because they can also give insight in the complexity of classification problems.

2.3.5 Approximation with MLPs

In this section we discuss some of the recent results about the approximation capabilities of MLPs. More specifically, we discuss a few results that show that 2LPs are universal approximators, and consider the bounds on the number of hidden nodes that is required to approximate a given function with a 3LP or a 2LP, as obtained recently by Kurkova [1992] and Barron [1991], respectively.

The first result we discuss is that 2LPs are universal approximators, i.e., given a sufficiently smooth real-valued function defined on a compact subset of \( \mathbb{R}^N \), it can be approximated with arbitrary precision by a 2LP, if the number of available hidden nodes can be made arbitrarily large. Here, approximating with arbitrary precision means that the difference between given function and the function computed by the 2LP can be made arbitrarily small. The first results of this kind have been obtained using a well-known result of Kolmogorov (1957) or one of its more recent refinements by Sprecher (1965) and Lorentz (1966, 1976), who state that every real-valued continuous function defined on the \( N \)-dimensional hypercube can be represented exactly by a \( \Sigma \)-3LP, with \( N \) nodes in the first hidden layer, \( 2N+1 \) nodes in the second hidden layer, and one output, provided that \( \Sigma \) is a suitable chosen set
of response functions, some of which depend on the function at hand; see [Hecht-Nielsen, 1989], and also [Cybenko, 1989; Cybenko, 1990; Funahashi, 1989; Hornik, Stinchcombe & White, 1989; Hornik, 1991; Kurkova, 1992]. If the requirement of exact representation is dropped, \( \Sigma \) can be made independent of the function at hand and the required number of layers can be lowered to 2, at the cost of a need for an arbitrary large number of hidden layers.

The intuitive idea behind the universal approximation of 2LPs is straightforward. It is well known that any sufficiently smooth function defined on the \( N \)-dimensional hypercube can be approximated with arbitrary precision by a superposition of a sufficiently large number of bell-shaped functions of the right width and steepness, placed at the right position; consider for instance a Fourier series approximation of the given function, which can be viewed as an superposition of bell-shaped functions. Furthermore, if \( \gamma : \mathbb{R} \to [0,1] \) is a given sigmoidal function, i.e., a measurable, non-decreasing, and non-constant response function, it is easily verified that

\[
\phi_{\alpha,\delta}(x) = \gamma(\delta x + \alpha) - \gamma(\delta x - \alpha)
\]

for some \( \delta > 0 \) and \( \alpha \in \mathbb{R} \), is a bell-shaped, positive and integrable function, of which the width and steepness can be easily tuned; see also [Funahashi, 1989]. Thus the given function can be approximated by a linear sum of \( \phi_{\alpha,\delta} \)'s, which directly implies that the given function can be approximated by a linear sum of \( \gamma(\delta x + \alpha) \)'s. Several results of this sort have been obtained, we give the formulation of Cybenko [1989].

**Proposition 2.3.6 (Cybenko [1989]).** Let \( \gamma : \mathbb{R} \to \mathbb{R} \) be any continuous sigmoidal function. Then finite sums of the form

\[
\hat{g}(x) = \sum_{i=1}^{N} \alpha_i \gamma(a_i^T x + b_i)
\]

are dense in \( C([0,1]^N) \). In other words, given any \( f \in C([0,1]^N) \) and \( \varepsilon > 0 \), there is a sum, \( \hat{g}(x) \), of the above form, for which

\[
\| f - \hat{g} \|_\infty = \sup_{x \in [0,1]^N} |f(x) - \hat{g}(x)| < \varepsilon.
\]

**Proposition 2.3.7 (Cybenko [1989]).** Let \( \gamma : \mathbb{R} \to \mathbb{R} \) be any bounded sigmoidal function. Then finite sums of the form

\[
\hat{g}(x) = \sum_{i=1}^{N} \alpha_i \gamma(a_i^T x + b_i)
\]

are dense in \( L^1([0,1]^N) \). In other words, given any \( f \in L^1([0,1]^N) \) and \( \varepsilon > 0 \), there is a sum, \( \hat{g}(x) \), of the above form, for which

\[
\| f - \hat{g} \|_{L^1} = \int_{[0,1]^N} |f(x) - \hat{g}(x)| dx < \varepsilon.
\]

The above results show that \( \{ \gamma, \beta \} \).2LPs are universal approximators, where \( \gamma \) is the considered response function used in the hidden layer and \( \beta \) is the identity function used in the output layer.

Similar results have been obtained by various authors; see for instance Funahashi [1989], Hornik, Stinchcombe & White [1989], Kurkova [1992], and also the recent review of Hornik [1991]. The class of considered non-linear response functions \( \gamma \) used in the hidden layer, is usually restricted in one way or the other. In Table 2.1
we have listed a number of papers, the corresponding first author, the name the authors use for the considered functions, and the restrictions posed by them.

<table>
<thead>
<tr>
<th>Author + Citation</th>
<th>Name</th>
<th>bnds</th>
<th>mon</th>
<th>non-d</th>
<th>cont</th>
<th>diff</th>
<th>( \gamma^\prime(\xi) &gt; 0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Funahashi [1989]</td>
<td>Sigmoid</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Shawe-Taylor [1992]</td>
<td>Activation</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Cybenko [1989]</td>
<td>Sigmoidal</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>x</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Hornik [1989]</td>
<td>Squashing</td>
<td>x</td>
<td>-</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Judd [1991]</td>
<td>Quasi-linear</td>
<td>x</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Cybenko [1989]</td>
<td>Sigmoidal</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Kurkova [1992]</td>
<td>Sigmoidal</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Rumelhart [1986]</td>
<td>Squashing</td>
<td>-</td>
<td>-</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td>Maass [1991]</td>
<td>Sigmoid</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Toda [1991]</td>
<td>Sigmoid</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

Table 2.1: Different names for different non-linear functions. An "x" ("-" implies that the restriction is required (not required). The abbreviation "bnds" stands for the restriction that \( \gamma \) is bounded to \([0,1]\) with \( \lim_{x \to 0} \gamma(x) = 1 \) and \( \lim_{x \to \infty} \gamma(x) = 0 \). Furthermore, "mon", "non-d", "cont", and "diff" stand for "monotonic", "non-decreasing", "continuous", and "differentiable", respectively. Finally, \( \gamma^\prime(\xi) > 0 \), stands for the existence of an \( \xi \in \mathbb{R} \) with this property.

The above results of the approximation of functions with 2LPs, can be used to prove that \((\gamma, \theta)\)-2LPs can approximately solve any classification problem with arbitrary precision; see for instance Cybenko [1989]. However, as with the above results, no bounds are given on the required number of hidden nodes. The first results that gave an explicit upper bound on the number of hidden nodes that is required to approximate a function with a given precision, were obtained for 3LPs. This is because for 3LPs it is much easier to derive such upper bounds. Below we present a result of Kurkova [1992] that is based on the work of Kolmogorov that we discussed earlier. First, we need the following definition.

**Definition 2.3.9 (Kurkova [1992]).** The function \( \omega_f : (0, \infty) \to \mathbb{R} \) is called the modulus of continuity of the function \( f : [0,1]^N \to \mathbb{R} \), and is defined by

\[
\omega_f(\delta) = \sup \{|f(x) - f(y)| : x, y \in [0,1]^N, ||x-y||_\infty < \delta\}.
\]

Kurkova proves a bound on the number of hidden nodes that is sufficient for the approximation of continuous functions with 3LPs. One may use arbitrary response functions that are measurable and satisfy the "bnds" restriction; see Table 2.1.

**Proposition 2.3.8 (Kurkova [1992]).** Let \( \gamma : \mathbb{R} \to [0,1] \) be a sigmoidal function, let \( N \in \mathbb{N} \), with \( N \geq 2 \), let \( f : [0,1]^N \to \mathbb{R} \) be a continuous function, and let \( \varepsilon > 0 \) be a positive real number. Then for every \( K \in \mathbb{N} \) and \( \nu > 0 \) such that

\[
(i) \quad K \geq 2N + 1,
(ii) \quad \frac{N}{K-N} + \nu < \frac{\varepsilon}{\|f\|_\infty}, \text{ and}
(iii) \quad \frac{\omega_f}{K} < \nu \frac{K-N}{2K-3N},
\]

...
there exists a \( (\gamma, \beta) \)-SLP, with \( NK(K+1) \) \( \gamma \)-nodes in the first hidden layer, \( K^2(K+1)^N \) \( \gamma \)-nodes in the second hidden layer, and one \( \beta \)-output, represented by \( g : [0,1]^N \rightarrow \mathbb{R} \), such that \( \|f - g\|_{\infty} < \epsilon \).

The above result is remarkable if compared with the previous approximation results, because it gives an explicit bound on the number of hidden nodes that is required. However, this is entirely due to the choice of 3LPs as follows from the following result, which can be proven straightforwardly.

**Proposition 2.3.9.** Let \( N \in \mathbb{N} \) and let \( f : [0,1]^N \rightarrow \mathbb{R} \) be a continuous function. Then the \( (\theta, \beta) \)-SLP defined for all \( K \in \mathbb{N} \) by

\[
g(x_1, \ldots, x_N) = \sum_{i_1, \ldots, i_N = 0}^{K-1} f(\frac{i_1}{K}, \ldots, \frac{i_N}{K}) \theta(\sum_{p=1}^{N}(\theta(x_p - \frac{i_p}{K}) - \theta(x_p - \frac{i_p + 1}{K})))
\]

has \( N(K+1) \) \( \theta \)-nodes in the first hidden layer, \( K^N \) \( \theta \)-nodes in the second hidden layer, one \( \beta \)-output, and satisfies

\[
\|f - g\|_{\infty} \leq \omega f\left(\frac{1}{K}\right).
\]

The idea behind the above result is to partition the \( N \)-dimensional hypercube into \( K^N \) cubes of size \( \frac{1}{K} \times \frac{1}{K} \times \cdots \times \frac{1}{K} \). Subsequently, the value of \( g \) in this cube is set equal to the value of \( f \) in one corner of the cube.

Using the same approach one can easily obtain a bound on the number of nodes required to approximately solve a classification problem with a 3LP. Let \( V \subseteq [0,1]^N \) be a subset of which the boundary is a differentiable surface with bounded first derivative and bounded surface area, then there exists a \( \theta \)-3LP with \( N(K+1) \) and \( K^N \) nodes in the first and second hidden layer, respectively, represented by the function \( g : [0,1]^N \rightarrow \mathbb{R} \) such that

\[
\|V - g^{\gamma}(1)\| = \int_{V \setminus g^{\gamma}(1)} dx = O\left(\frac{1}{K}\right).
\]

As an example we have approximated the circle with a partition of squares; see Figure 2.9a. It gives what might be called a first-order approximation.

Cybenko considers second-order approximations. If one also assumes that the surface of the considered subset has a bounded second derivative, it can be shown that a \( \theta \)-3LP exists with \( K^{N+1} \) hidden nodes in the first hidden layer, represented by \( g \), such that \( \|V - g^{\gamma}(1)\| = O\left(\frac{1}{K^2}\right) \); see Cybenko [1990]. Although Cybenko does not give the required number of nodes in the second hidden layer, we think that this number does not depend on the required precision of the approximation, but only on the structure of the given set. In Figure 2.9b a second-order approximation of the circle is given.

Finally, we discuss a recent result of Barron [1991], that gives an explicit bound on the number of hidden nodes that is required in order to approximate a certain function with a given precision by a 2LP.

**Proposition 2.3.10 (Barron [1991]).** Let \( f : \mathbb{R}^N \rightarrow \mathbb{R} \) be a continuous and integrable function whose Fourier transform \( \hat{f} \) satisfies \( \int_{\mathbb{R}^N} |\hat{f}(\alpha)| d\alpha \leq c \), and let \( \gamma \) be an arbitrary sigmoidal function. Then for every \( r > 0 \), every \( K \in \mathbb{N} \), and every \( \nu > 2c \), there exists a \( (\gamma, \beta) \)-2LP, with \( N \) inputs, \( K \) \( \gamma \)-nodes in the hidden layer...
2.4 Concluding remarks

Figure 2.9: An example of a first-order (a) and a second-order (b) piece-wise linear approximation of a circle.

and one $\beta$-output, represented by $g : \mathbb{R}^N \to \mathbb{R}$, such that

$$\int_{B_r} |f(x) - g(x)|^2 \, dx \leq \frac{\nu^2}{K} \int_{B_r} \, dx.$$

With this powerful result we end our discussion on the approximative capabilities of MLPs.

2.4 Concluding remarks

In this chapter we have informally introduced the multi-layered perceptrons (MLPs), a computational model based on a layered network of interconnected nodes. We have investigated finite classification problems and showed that the capabilities and limitations of 1LPs and MLPs for solving these problems are more or less complementary. If a 1LP can solve a given finite classification problem, a description of such a 1LP can be found in polynomial time. However, most finite classification problems cannot be solved by a 1LP. On the other hand, all classification problems can be solved by some MLP, and even by some 2LP, but finding a description of a 2LP with a given number of hidden nodes that solves a given classification problem is $\mathcal{NP}$-hard.

Using counting arguments based on the theory of dichotomies, some general bounds could be derived that give a rough estimate on the total number of nodes required to be able to solve arbitrary finite classification problems. Similar results were obtained by considering approximation results for real-valued MLPs, in particular 2LPs. However, the essential question how many nodes are required to solve a given classification problem with an LP with a certain number of layers, has been left unanswered.

In Chapter 4 we try to obtain an answer for this question using the similarity of MLPs and circuits. In the same way as is done for circuits, we introduce a
number of complexity classes for MLPs and show their relation. In Chapter 5 we use a geometrical approach to the study the complexity of MLPs with real valued inputs and binary outputs, required for solving infinite classification problems. In Chapter 6 we use the results obtained in Chapter 5 to obtain some answers about the required complexity of MLPs for solving combinatorial optimization problems, by reformulation a combinatorial optimization problem as a classification problem. The results are applied to five combinatorial optimization problems in Chapter 7. To be able to discuss these problems in Chapter 4, we introduce these five problems in the following chapter.
Chapter 3

Combinatorial Optimization

3.1 Introduction

In this chapter we introduce our formulation of a real-valued combinatorial optimization problem, and introduce a real-valued version of five well-known combinatorial optimization problems, SORTING, MINIMUM COST SPANNING TREE, SHORTEST NETWORK PATH, SHORTEST NETWORK ROUTE, and DISCRETE DYNAMIC LOT SIZING, that are used in Chapters 4 and 7. For an elaborate discussion of the problems associated with combinatorial optimization we refer to the books by Nemhauser & Wolsey [1988], Papadimitriou & Steiglitz [1982], and Schrijver [1986].

3.2 A formulation for real-valued combinatorial optimization problems

Below, we present the formulation of a combinatorial optimization problem that is used in Chapter 6. The set of instance defining parameters corresponds to the data that can be viewed as the input of the problem. We allow this data to be real-valued.

Definition 3.2.1. A combinatorial optimization problem is represented by a 3-tuple \((I, F, c)\), where (i) for some \(N \in \mathbb{N}\), \(I \subseteq \mathbb{R}^N\) denotes a set of instance defining parameters, (ii) for some \(K \in \mathbb{N}\) and for all \(x \in I\), \(F(x) \subseteq \mathbb{R}^K\) denotes a finite set of feasible solutions for the instance defined by \(x\), and (iii) for all \(x \in I\), \(c(\cdot; x) : F(x) \rightarrow \mathbb{R}\) denotes a cost function on the set of feasible solutions for the instance defined by \(x\). For a given instance \((F(x), c(\cdot; x))\) defined by some \(x \in I\), the problem is to find a feasible solution with minimal cost, i.e., we must find an \(y \in F(x)\), such that \(c(y; x) \leq c(z; x)\), for all \(z \in F(x)\). It is assumed that \(F(x) \neq \emptyset\), for all \(x \in I\), which guarantees that for every instance this problem is solvable.

A combinatorial optimization problem can also be defined using the notation of Chapter 2. With the above notation this can be done as follows.

Definition 3.2.2. (COMBINATORIAL OPTIMIZATION PROBLEM (COP))

**INPUT:** A subset \(I \subseteq \mathbb{R}^N\), a vector \(x \in I\), and a finite subset \(F(x) \subseteq \mathbb{R}^K\), for some \(N, K \in \mathbb{N}\).

**OUTPUT:** \(y \in F(x)\), such that \(c(y; x) \leq c(z; x)\), for all \(z \in F(x)\).
As an example we consider the traveling salesman problem, where, given a set of cities and their relative distances, the problem is to find a tour that visits every city exactly once and returns to its starting point, such that the length of the tour is minimal. Instances of the traveling salesman problem can be encoded by giving a square matrix, in which the i-j-th entry denotes the distance between city i and city j. The number of cities of an instance determines the width of the matrix. A solution can be denoted by listing the cities in the order of their appearance in the tour. Given a distance matrix and a solution the length of the tour corresponding to this tour can be easily calculated. Below we present the traveling salesman problem using the above formulation.

**Definition 3.2.3. (Traveling Salesman)**

Let \( x \in \mathbb{R}^{n \times n} \) be a \( n \times n \) matrix, for some \( n \in \mathbb{N} \), where \( x_{ij} \) represents the distance between city i and city j. Let \( y_{ij} \in \{0,1\} \) be a 0-1 variable, where \( y_{ij} = 1 \), if the tour visits city i at the j-th position, and \( y_{ij} = 0 \) otherwise. Then the problem can be formulated as a tuple \( (I,F,c) \), with

\[
I = \{ x \in \mathbb{R}^{n \times n} \mid x \geq 0 \}, \\
F = \{ y \in \{0,1\}^{n \times n} \mid \sum_{i=1}^{n} y_{ii} = \sum_{j=1}^{n} y_{ij} = 1 \}, \\
c(y;x) = \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} y_{ij} (y_{ij+1}),
\]

where for a given \( x \in I \) the problem is to find a \( y \in F \) that minimizes \( c(y;x) \).

Note that this formulation satisfies corresponds to the general formulation with \( N = K = n^2 \). We have formulated traveling salesmen such that the solutions are 0-1 vectors. This is because we are going to consider MLPs with hard-limiting response functions for solving combinatorial optimization problems; see Chapter 6.

In Chapter 4, we simultaneously consider the real-valued version of a combinatorial optimization problem, as defined above, and the integer-valued version of the same problem. In order to distinguish the two versions, we add the adjective **real** to the real-valued problem and the adjective **integer** to the integer-valued version.

The integer version can be obtained from the real version by restricting the instance parameters to \( \mathbb{B}^N \). In some cases, a finite subset of integer-valued vectors is allowed. In the other chapters we consider the real-valued versions only, and therefore omit the adjective **real**.

In the following section we introduce five well-known combinatorial optimization problems in a 0-1 formulation corresponding to the above introduced format.

### 3.3 The five problem formulations

The first introduced problem is the well-known sorting problem, which is usually stated as follows.

**Definition 3.3.1. (Sorting)**

Given an array of n real numbers, for some \( n \in \mathbb{N} \), the problem is to find the absolute sequence in which the numbers have to be placed in order to obtain a sorted list. Let
\( \pi(i) \) denote the index of the number at position \( i \) in the sorted list. Then the problem can be formulated mathematically as follows.

- Given \( n \in \mathbb{N} \) and \( x_1, \ldots, x_n \in \mathbb{R} \).
- Find a mapping \( \pi : \{1, \ldots, n\} \rightarrow \{1, \ldots, n\} \) that satisfies the following conditions.
  
  (i) \( \pi \) is a permutation.
  
  (ii) \( x_{\pi(i)} \leq x_{\pi(i+1)} \), for all \( i = 1, \ldots, n-1 \).

In Chapter 7 we prove that SORTING can be presented using the following 0-1 formulation of a COP.

**Definition 3.3.2.** *(0-1 COP formulation of sorting)*

Let \( x_1, \ldots, x_n \in \mathbb{R} \) be a set of \( n \) real numbers, for some \( n \in \mathbb{N} \). Let \( y_{ij} \in \{0,1\} \) be a 0-1 variable, where \( y_{ij} = 1 \), if the number \( x_j \) is to be put on the \( i \)th position, and \( y_{ij} = 0 \) otherwise. Then the problem can be formalized as a tuple \((I,F,c)\), with

\[
I = \mathbb{R}^n,
\]

\[
F = \{ y \in \{0,1\}^{n \times n} \mid \sum_{i=1}^{n} y_{ij} = \sum_{j=1}^{n} y_{ij} = 1 \},
\]

\[
c(y;x) = \sum_{i=1}^{n} \sum_{j=1}^{n} (n+1-i) y_{ij} x_j,
\]

where for a given \( x \in I \) the problem is to find a \( y \in F \) that minimizes \( c(y;x) \).

The second problem discussed is the minimum cost spanning tree problem.

**Definition 3.3.3.** *(Minimum cost spanning tree)*

Given a completely connected undirected graph \( G = (V,E) \) on \( V = \{1, \ldots, n\} \) with a cost \( x_e \) associated to edge \( e \) for all \( e \in E \); note that \( |E| = \frac{1}{2}n(n-1) \). The problem is to find a spanning tree of \( G \) with minimal total cost. Let \( y_e \in \{0,1\} \) be a 0-1 variable such that \( y_e = 1 \), if the edge \( e \) is part of the spanning tree, and \( y_e = 0 \) otherwise. Then the problem can be represented as the tuple \((I,F,c)\) given by

\[
I = \mathbb{R}^{\frac{1}{2}n \times (n-1)},
\]

\[
F = \{ y \in \{0,1\}^{\frac{1}{2}n \times (n-1)} \mid y \text{ represents a spanning tree on } \{1, \ldots, n\} \},
\]

\[
c(y;x) = \sum_{e \in E} y_e x_e.
\]

The next problem we consider is the shortest path problem. In general this is the problem of finding the shortest path between two given nodes in a graph, where all edges have a certain non-negative length. For the sake of notational convenience we restrict ourselves to network graphs. Then the problem can be formulated as follows.

**Definition 3.3.4.** *(Shortest network path)*

Given a directed network-graph with a source node \( s \), a terminal node \( t \), and \( n_s \) levels of \( n_t \) nodes; see Figure 3.1. There is an arc from \( s \) to each node of the first level, there is an arc from each node of the last level to \( t \), and there are arcs between the nodes of subsequent levels. With each arc a non-negative length is associated.
The problem is to find a path through the network from $s$ to $t$ that has minimal total length.

To come to a 0-1 formulation we assume without loss of generality that the lengths of the arcs incident with the nodes $s$ and $t$ have a length zero. Let $y_{ik} \in \{0,1\}$ be a 0-1 variable, such that $y_{ik} = 1$, if the path passes through node $i$ of level $k$, and $y_{ik} = 0$ otherwise. Finally, we let $x_{ijk} \geq 0$ denote the length of the arc between node $i$ of level $k$ and node $j$ of level $k+1$. Then the problem can be represented by the tuple $(I, P, c)$ given by

$$I = \mathbb{R}^{n_1 \times n_2 \times (n_2-1)}_{\geq 0},$$
$$P = \{y \in \{0,1\}^{n_1 \times n_2} | \sum_{i=1}^{n_1} y_{ik} = 1\},$$
$$c(y;x) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_2-1} y_{ik} x_{ijk}.$$

Below, we introduce a variant of the many-to-one shortest path problem, which we call **shortest network route**. Consider the network given in Figure 3.1 and assume one wants to determine the shortest path from every node in the network to the terminal node $t$. An elementary property of the collection of shortest paths is that they form a spanning in-tree of the network; an in-tree is a tree with each node having out degree 1. We therefore consider the problem of finding a collection of paths from every node to the terminal node that forms a spanning in-tree and has minimal total length. We call a collection of paths that forms a spanning in-tree a route. Then the problem becomes to find the shortest route in a network, which can be formulated as follows.

**Definition 3.3.5. (Shortest network route)**

Let $y_{ijk} \in \{0,1\}$ be a 0-1 variable, such that $y_{ijk} = 1$, if the route goes from node $i$ of level $k$ to node $j$ of level $k+1$ and $y_{ijk} = 0$ otherwise, where $i, j = 1, \ldots, n_1$ and $k = 1, \ldots, n_2 - 1$. Furthermore, we introduce the auxiliary 0-1 variable $y_{s,0}$ such
that \( y_{x,0} = 1 \) if the route goes from \( s \) to node \( j \) of level \( k = 0 \) and \( y_{x,0} = 0 \) otherwise. Let \( x_{ij} \geq 0 \) denote the length of the arc between node \( i \) of level \( k \) and node \( j \) of level \( k+1 \). Then the length \( w(i,k) \) of the path that starts at node \((i,k)\), ends at the terminal \( t \) and travels using the route set by \( y \) can be recursively defined by

\begin{enumerate}[(i)]
\item \( w(i, n_2) = 0 \), for all \( i = 1, \ldots, n_1 \),
\item \( w(i, k) = \sum_{j=1}^{n_1} y_{ijk}(x_{ij} + w(j, k+1)) \), for all \( i = 1, \ldots, n_1 \) and \( k = 1, \ldots, n_2-1 \), and
\item \( w(s,0) = \sum_{j=1}^{n_2} y_{ij0} w(j,1) \).
\end{enumerate}

Then the problem can be represented by the tuple \((I, F, c)\) given by

\[ I = \mathbb{R}^{n_1 \times n_2 \times (n_2-1)}; \]
\[ F = \{ y \in \{0,1\}^{n_1 \times (n_2-1)} \times \{0,1\}^{n_1 \times n_2} | \sum_{j=1}^{n_1} y_{ijk} = 1 \}; \]
\[ c(y; z) = w(s,0) + \sum_{i=1}^{n_1} \sum_{k=1}^{n_2-1} w(i,k). \]

Finally, we discuss the discrete dynamic lotsizing problem. The general dynamic lotsizing problem is the problem of satisfying at minimum cost the known demands for a specific commodity in a number of consecutive production periods, which is called the planning horizon. It is possible to store units of this commodity to satisfy demands in later periods, but backlogging is not allowed. The costs consist of production and holding costs. The production costs consist of two components, a cost per unit commodity produced, usually called the marginal production cost, and a fixed setup cost that is incurred whenever production occurs in a particular period. The holding costs are linear in the inventory level at the end of the period.

Both the inventory at the beginning and at the end of the planning horizon are assumed to be zero.

**Definition 3.3.6. (Dynamic Lotsizing)**

Let \( n \in \mathbb{N} \) be the length of the planning horizon, and let \( d_i, p_i, r_i \), and \( h_i \) denote demand, marginal production cost, setup cost and unit holding cost in period \( i \), \( i = 1, \ldots, n \), respectively. Let \( u_i \) be the number of units produced in period \( i \), let \( s_i \) be the number of units in stock at the end of period \( i \), and let \( y_i = 1 \), in case that a setup occurs in period \( i \), and \( y_i = 0 \), otherwise. Then the problem can be formulated as

\[
\begin{align*}
\min \quad & \sum_{i=1}^{n} (p_i u_i + r_i y_i + h_i s_i), \\
\text{s.t.} \quad & s_{i+1} + u_i - d_i = s_i, \quad i = 1, \ldots, n, \\
& s_0 = s_n = 0, \\
& u_i \leq y_i \sum_{j=i}^{n} d_j, \quad i = 1, \ldots, n, \\
& u_i \geq 0, s_i \geq 0, \quad i = 1, \ldots, n, \\
& y_i \in \{0,1\}, \quad i = 1, \ldots, n.
\end{align*}
\]

The above formulation is a mixed-integer linear program formulation, but the problem can be solved in polynomial time; see [Evans, 1985] for an efficient \( O(n^2) \) implementation, and [Wagelmans, Van Hoesel & Kolen, 1989] for an \( O(n \log n) \) implementation. The problem becomes much harder if some kind of resource constraints
like \( u_i \leq c_i, \ i = 1, \ldots, n \), are added; see Bitran & Yanasse, 1982; Florian, Lenstra & Rinnooy Kan, 1980] for an analysis of the complexity of the problem in various cases. Here, we focus on the special case of the above problem that was introduced by Wagner & Whitin [1958]. They assumed identical marginal production costs, \( p_i = p \), and non-negative unit holding costs, \( h_i \geq 0 \). For notational convenience, we additionally assume that setup costs and holding costs are also identical, \( r_i = r \), \( h_i = h \). In this special case the problem can be solved in linear time; see Wagelmans, Van Hoesel & Kolen [1989].

Next, we use the properties stated by Wagner & Whitin in order to obtain a tractable formulation. The first property is that \( \sum_{i=1}^{n} u_i = \sum_{i=1}^{n} d_i \), which follows by summing the first constraint of (3.1) and using that \( s_0 = n = 0 \). Substitution in the object function then shows, using \( p_i = p \), that the variables \( u_i \) can be eliminated from the formulation. This yields the following reformulation of (3.1):

\[
\begin{align*}
\min \quad & \sum_{i=1}^{n} (r(1-y_i) + h s_i), \\
\text{s.t.} \quad & 0 \leq s_i + d_i - s_{i-1} \leq y_i \sum_{j=1}^{n} d_j, \quad i = 1, \ldots, n, \\
\quad & s_0 = s_n = 0, \\
\quad & s_i \geq 0, \quad i = 1, \ldots, n, \\
\quad & y_i \in \{0, 1\}, \quad i = 1, \ldots, n.
\end{align*}
\]

(3.2)

The next property which has been proved by Wagner & Whitin, states that in an optimal solution of (3.2) a setup occurs in a period only if the inventory level at the end of the previous period is zero, i.e., \( y_i s_{i-1} = 0 \), for all \( i = 1, \ldots, n \). This implies that in every period \( s_i = \sum_{j=i}^{n} d_j \), for some \( k \geq i \). Using this result, Wagner & Whitin showed that (3.2) can be formulated as a shortest path problem. We present this result using the notation of the shortest path problem given above.

**Proposition 3.3.1 (Wagner & Whitin [1958]).** Let the numbers \( x_{ki} \in \mathbb{R}^+ \cup \{\infty\} \), for all \( l, k = 0, \ldots, n \) and \( i = 0, \ldots, n-1 \), be defined by

\[
x_{ki} = \begin{cases} 
  r, & l = i, k \geq i + 1, \quad i = 0, \ldots, n-1, \\
  h \sum_{j=i+1}^{k} d_j, & l = k, k \geq i + 1, \quad i = 0, \ldots, n-1, \\
  \infty, & \text{otherwise.}
\end{cases}
\]

Then (3.2) is equivalent to the problem of finding \( y \in \{0, 1\}^{n+1} \) with \( \sum_{k=0}^{n} y_{ki} = 1 \), for all \( i = 0, \ldots, n \), that minimizes

\[
c(y; x) = \sum_{i=0}^{n} \sum_{k=0}^{n-1} \sum_{i=0}^{n-1} y_{i+1} x_{ki}.
\]

Although, the result of Proposition 3.3.1 is useful for studying the MILP-complexity of DYNAMIC LOT SIZING, we mostly use the more compact formulation derived below. Based on the zero-inventory property of Wagner & Whitin, we come to the following equality for the stock level. For all \( i = 1, \ldots, n \),

\[
s_i = \sum_{j=i+1}^{n} \left( \prod_{k=i+1}^{j} (1-y_k) \right) d_j.
\]

(3.3)

One can verify that (3.3) satisfies the constraints of (3.2) provided that \( y_1 = 1 \). Substitution of (3.3) in the object function of (3.2) yields the following equivalent
3.4 Concluding remarks

formulation.

\[
\min \sum_{i=1}^{n} \left[ r y_i + h d_i \sum_{j=1}^{i} \left( \prod_{k=j}^{i} (1 - y_k) \right) \right],
\]

s.t. \( y_1 = 1, \) \( y_i \in \{0,1\}, \) \( i = 2, \ldots, n. \) \hfill (3.4)

Denote \( z_i = h d_i / r, \) eliminate \( y_1, \) renumber the remaining variables from 1 to \( n - 1, \)
instead of 2 to \( n, \) and replace \( n \) by \( n + 1. \) Then we obtain the following formulation
of DDLR.

\textbf{Definition 3.3.7. (Discrete Dynamic Lotsizing)}

\textit{Let} \( x_1, \ldots, x_n \geq 0 \) \textit{be the demand that has to be satisfied for the next} \( n \) \textit{periods,}
\textit{for some} \( n \in \mathbb{N}. \) \textit{Except for the setup in period 0, a setup occurs in period} \( i, \) \textit{if}
\( y_i = 1, \) \textit{and does not occur, if} \( y_i = 0. \) \textit{The cost of a setup is} \( L, \) \textit{the cost of storing a}
\textit{quantity} \( x \) \textit{for one period is} \( z, \) \textit{and there are no other costs. Then the problem can}
\textit{be represented as the tuple} \((I,F,c)\) \textit{given by}

\[
\begin{align*}
I &= \{ x \in \mathbb{R}^n | x_i \geq 0 \}, \\
F &= \mathbb{B}^n, \\
c(y;x) &= \sum_{i=1}^{n} \sum_{j=1}^{i} \left( \prod_{k=j}^{i} (1 - y_k) \right) x_i + \sum_{i=1}^{n} (1 - y_i).
\end{align*}
\hfill (3.5)
\]

\textit{where for a given} \( x \in I \) \textit{the problem is to find a} \( y \in F \) \textit{that minimizes} \( c(y;x). \)

3.4 Concluding remarks

In this chapter we have presented a formulation of combinatorial optimization problems with real-valued data. We used a 0-1 formulation because this is representation
used in Chapter 6, where we consider MLPs with real-valued inputs and hard-limiting response functions for solving combinatorial optimization problems. Furthermore, we have introduced five well-known combinatorial optimization problems
\textit{Sorting, Minimum Cost Spanning Tree, Shortest Network Path, Shortest Network Route, and Discrete Dynamic Lotsizing,} in the proposed 0-1
formulation. These problems are considered in detail in Chapters 4 and 7.
Chapter 4

Complexity

4.1 Introduction

In Chapter 2, we have introduced the MLPs and studied some of their complexity aspects from a neural network point of view. This chapter looks at MLPs from a different angle. It considers them as a model for parallel computation, whose capabilities can be determined by comparing it with other existing models for parallel computation. Because of the structure of MLPs, the capabilities and complexity of MLPs are most easily compared with those of Boolean circuits; see Wegener [1987] for an introduction to the theory of Boolean circuits. Comparisons of Boolean circuits with other computational models, like those based on Turing machines and PRAMs, have shown that Boolean circuits are universal; see for instance Johnson [1990] and Karp & Ramachandran [1990]. This means that Boolean circuits are in fact equivalent with those other models. Furthermore, these studies have given insight in the complexity of Boolean circuits for solving specific problems. Hence, by comparing MLPs to Boolean circuits we can obtain insight in the complexity of MLPs for solving specific problems. Here we are particularly interested in the complexity of MLPs for solving combinatorial optimization problems, as introduced in Chapter 3.

The strategy used in this chapter is as follows. First, we introduce a general framework for so-called uni-directional computational models that contains both Boolean circuits and MLPs as special cases; see Section 4.2. A uni-directional computational model is a network of interconnected nodes that is capable of performing a well-defined set of loop-free calculations. For both Boolean circuits and MLPs a number of subtypes will be distinguished. For Boolean circuits emphasis is given to the different types of weighted threshold circuits, since they can also be viewed as certain types of MLPs, in which the input values are restricted to binary values and the response function is the threshold function.

Secondly, we introduce a number of complexity classes for uni-directional computational models, as has been done for Boolean circuits; see Chandra, Stockmeyer & Vishkin [1984] and Section 4.3. These complexity classes contain problems that can be solved by a restricted set of uni-directional computational models. The re
restrictions apply to the number of layers, the number of nodes, the type of the node functions used, the set of input values, the set of output values, the fan-in of the nodes, and the accuracy required by the computation of the response function.

Finally, we discuss some results with respect to the introduced complexity classes, most of which are available from literature. We start with some relevant results concerning Boolean circuits. A new set of results concern circuits with their fan-in bounded by a polynomial in the logarithm of the number of inputs. Subsequently, we discuss some results for MLPs that have Boolean inputs and sigmoidal response functions. This enables us to compare the capabilities of MLPs to those of the weighted threshold circuits. We end by formulating our own results on the complexity of MLPs that have real-valued inputs and hard-limiting response functions, that solve the five combinatorial optimization problems introduced in Chapter 3, and are based on the results obtained in the Chapters 5, 6, and 7.

4.2 Uni-directional computational models

In this section we introduce the class of uni-directional computational models as a general framework for both Boolean circuits and feedforward neural networks. Our approach is based upon the approaches followed by Judd [1991] and Farber [1991].

A uni-directional computational model is a directed network of interconnected nodes that is capable of performing a certain well-defined loop-free computation. Formally, a uni-directional computational model can be defined as follows.

Definition 4.2.1. Let \( \mathcal{F} \) be a set of functions, \( \mathcal{F} \subseteq \bigcup_{N \in \mathbb{N}} \{D^N \rightarrow R\} \), for some \( R \subseteq D \subseteq \mathbb{R} \). A uni-directional computational model (UCM) with node function set \( \mathcal{F} \) is a 5-tuple given by \( M = (X, Y, Z, E, G) \), where

\[
\begin{align*}
X, Y, \text{ and } Z & \text{ are finite ordered sets,} \\
X \cap Z & = \emptyset, \\
Y & \subseteq Z, \\
(\{1\} \cup X \cup Z, E) & \text{ is a directed, acyclic graph,} \\
G & : Z \rightarrow \mathcal{F},
\end{align*}
\]

and \( 1 \) denotes a fixed vertex labelled 1.

The set \( Z \) represents a set of nodes, each of which computes a function determined by \( G \). A node \( z \in Z \) is referred to as an \( G(z) \)-node. \( Y \) represents a set of output nodes and \( Z \setminus Y \) represents the set of hidden nodes. The vertices in \( X \) correspond to the inputs of the UCM. The vertex \( 1 \) represents a constant input, but is not counted as a true input of the UCM. The inputs \( X \) of the UCM are the only nodes that can be set to a value, with the set of possible input values determined by \( D \). The output nodes \( Y \) are the only nodes from which a value can be obtained, the set of possible output values determined by \( R \). The set \( E \) represents the connections between the hidden nodes, the output nodes, and the inputs. We call the graph \( (\{1\} \cup X \cup Z, E) \) the interconnection graph of \( M \). The fan-in of a node denotes the in-degree of the node with respect to the connection graph. Similarly, the fan-out of a node denotes
the out-degree of the node. We assume that the inputs have zero fan-in and the outputs have zero fan-out.

![Diagram](image)

Figure 4.1: Circuits that compute the EOR of two Boolean inputs. The constant input 1 is omitted for convenience. In (a) a classical circuit, in (b) an alternating circuit; see Section 4.2.1 for a definition of an alternating circuit.

If $D = \mathcal{R} = \mathcal{B}$, the node functions have binary inputs and outputs and the definition of a UCM corresponds to the usual definition of a Boolean circuit. As an example, consider the circuit presented in Figure 4.1a. This circuit corresponds to a UCM given by $M = (X, Y, Z, E, G)$, where

- $Z = \{z_1, z_2, z_3, z_4\}$,
- $X = \{x_1, x_2\}$,
- $Y = \{z_4\}$,
- $E = \{(x_1, z_1), (x_1, z_2), (x_2, z_1), (x_2, z_2), (x_2, z_3), (z_1, z_4), (z_3, z_4)\}$,
- $G(z_1) = \text{OR}$, $G(z_2) = \text{AND}$, $G(z_3) = \text{NOT}$, $G(z_4) = \text{AND}$.

The constant input 1 is not used and therefore omitted. We say that the circuit $M$ computes the exclusive-or function, because if we offer two Boolean values $d_1, d_2 \in \mathcal{B}$ to the inputs $x_1$ and $x_2$, respectively, and extend these values in the obvious way through the connection graph, the value obtained at the output equals

$$ \text{AND(OR}(d_1, d_2), \text{NOT(AND}(d_1, d_2))) \equiv \text{EOR}(d_1, d_2).$$

The node functions used in the above example are all symmetric functions, i.e., we have that $G(d_1, \ldots, d_k) = G(d_{\pi(1)}, \ldots, d_{\pi(k)})$, for all permutations $\pi : \{1, \ldots, L\} \to \{1, \ldots, L\}$. This enables a straightforward description of the function that is computed by the above circuit. We also consider UCMs that use node functions which are not symmetric, for instance the weighted threshold functions; see Section 4.2.1.
In those cases a convention is required for the order of the inputs of the node functions. We assume that the order of the inputs of a node function corresponds to the order of the incoming connections of that node. The order of the incoming connections of a node is defined as follows. First, comes the connection to the constant input 1, then the connections to the inputs of the UCM, ordered by the number of the input, and, finally, the connections to the other nodes, ordered by the number of the nodes. This yields the following characterization of the function that is computed by a UCM.

**Definition 4.2.2.** Let \( M = (X, Y, Z, E, G) \) be a UCM with \( X = \{ x_1, \ldots, x_N \} \), \( Y = \{ y_1, \ldots, y_K \} \), and \( Z = \{ z_1, \ldots, z_T \} \). For each \( d \in D^N \), define \( v_d : Z \rightarrow \mathbb{R} \) as follows. Let \( z \in Z \) and choose \( p, q \in N_0 \), and \( \mu : \{ 1, \ldots, p \} \rightarrow \{ 1, \ldots, N \} \), \( \kappa : \{ 1, \ldots, q \} \rightarrow \{ 1, \ldots, T \} \), \( \mu, \kappa \) both increasing, such that
\[
\{ x_{\mu(1)}, \ldots, x_{\mu(p)} \} \cup \{ x_{\kappa(1)}, \ldots, x_{\kappa(q)} \} = \{ z' \in X \cup Z \mid (z', z) \in E \}.
\]

Then we define
\[
v_d(z) = \begin{cases} 
G(z)(1, d_{\mu(1)}, \ldots, d_{\mu(p)}, v_d(x_{\mu(1)}), \ldots, v_d(x_{\mu(q)})), & \text{if } (1, z) \in E, \\
G(z)(d_{\mu(1)}, \ldots, d_{\mu(p)}, v_d(z_{\mu(1)}), \ldots, v_d(z_{\mu(q)})), & \text{if } (1, z) \notin E.
\end{cases}
\]

We call \( v_d(z) \) the value of node \( z \) on input \( d \). The output of \( M \) on input \( d \) is defined to be \( v_d(y_1), \ldots, v_d(y_K) \). An \( N \)-input UCM given by \( M \) is said to compute the function \( f : D^N \rightarrow \mathbb{R}^K \), if for all \( d \in D^N \), the output of \( M \) on input \( d \) is \( f(d) \).

Informally, we say that \( y = f(x) \).

The above definition of the value \( v_d(z) \) of node \( z \) is not a proper definition, because it contains a reference to \( v_d(z') \) for some \( z' \). This can be corrected by defining \( v_d(z) \) recursively with respect to the level of \( z \), which is defined in Definition 4.2.3 below.

Note that this implies that the computation of a UCM is synchronized, i.e., the computation of the nodes in level \( t \) starts only after the computation of the nodes in level \( t - 1 \) has been completed.

**Definition 4.2.3.** Let \( M = (X, Y, Z, E, G) \) be a UCM. For each \( z \in X \cup Z \) we define the level of \( z \), denoted as \( \text{level}(z) \), to be the maximum number of nodes in \( Z \) on any path from an input to \( z \). Thus, iteratively, \( \text{level}(z) = 0 \), if \( z \in X \cup \{ 1 \} \), and \( \text{level}(z) = 1 + \max \{ \text{level}(z') \mid z' \in X \cup Z, (z', z) \in E \} \), if \( z \in Z \). The depth of \( M \) is defined as the maximum of \( \text{level}(z) \), \( z \in X \cup Z \). If the depth of \( M \) is \( m \), the nodes of \( M \) can be partitioned into \( m \) layers. A node \( z \in Z \) is said to be in layer \( i \) if \( \text{level}(z) = i \). The size of layer \( i \) is the number of nodes in layer \( i \). The size of \( M \) is defined to be \( |Z| \), the number of nodes.

Alternatively, the size of a UCM may be defined to be \( |E| \), i.e., the number of connections or wires. This does not make any difference for the results in this chapter that consider UCMs with size bounded by a polynomial in the number of inputs. This is because the two sizes are polynomially related, which follows from \( |Z| \leq |E| \leq |Z|(|Z| + N + 1) \), where \( N = |X| \) is the number of inputs. However, choosing for the number of wires instead of the number of nodes might cause some differences in those cases where one considers UCMs with constant size or with size bounded by a polynomial in the logarithm of the number of inputs.

Without loss of generality we may assume that the \( m \)-th layer contains outputs only. Furthermore, we make the following assumptions about the structure of the
4.2 Uni-directional computational models

1. The output nodes are all in the same layer, i.e., level(y) = m, for all y ∈ Y, where m is the depth of M. This implies that we can speak of the output layer. The remaining layers are usually called the hidden layers.

2. There are connections between nodes in subsequent layers only, i.e., (x', z) ∈ E, only if level(x) = level(x') + 1, for all x, z' ∈ {1} ∪ X ∪ Z.

Imposing these conditions upon the UCMs considered essentially means that we study UCMs in the class of strictly feed-next networks opposed to the more general feed-forward networks; see also Muruga [1971]. This is not very restrictive, since one can easily verify that in case that the function set contains the identity function, every UCM can be converted into a UCM that satisfies the above conditions with the same number of layers, a polynomial increase in the number of nodes, and the same maximum fan-in and fan-out. In the following section we consider the subclass of UCMs known as circuits. For function set for circuits that are complete, it can be shown that this set contains either the identity function or the NOT-function [Wegener, 1987]. In the latter case the restriction to feed-next networks implies at most a doubling of the depth.

The above conditions also imply that only nodes in the first layer of a UCM are allowed to be connected to the constant input 1. Especially for the feed-forward networks introduced in Section 4.2.2, it is convenient to have a constant input available in every layer. Our approach to circumvent this difficulty is to use the constant function 1 in all of the considered node function sets.

4.2.1 Boolean circuits

We already mentioned that choosing D = R = B in the definition of a UCM corresponds to the usual definition of a Boolean circuit. We name a circuit by the node function set that is used.

Definition 4.2.4. A UCM is called an F-circuit, if D = R = B, i.e., if the node function set is F ⊆ \bigcup_{\mathcal{N}_\text{EN}} (B^2 \rightarrow B).

In Figure 4.1.a, we have shown an example of an N-circuit that uses the node function set \mathcal{N} = \{\text{AND}^2, \text{OR}^2, \text{NOT}, 1\}. Here the superscript 2 is added to indicate that the considered AND and OR functions have 2 inputs. The node function set is denoted by \mathcal{N}', to associate it with \mathcal{N}, the class which is named after Nick Pippenger and consists of problems that can be solved with circuits that have fan-in at most 2; see also Section 4.3. The \mathcal{N}-circuits are important because \mathcal{N} is representative for the class of problems that can be solved with circuits that have bounded fan-in, which is often an important constraint in VLSI design; see also Wegener [1987]. There is a large amount of literature that deals with theory of the complexity of \mathcal{N}-circuits; see for instance Wegener [1987] and Dunne [1988].

On the other hand, we consider circuits with nodes that may have arbitrary large fan-in. The best studied node function set in this case is the one denoted by \mathcal{A} = \bigcup_{\mathcal{N}_\text{EN}} (\text{AND}^N, \text{OR}^N, \text{NOT}, 1). Every A-circuit can be converted into an alternating
circuit, with the same depth and a polynomial increase of size, which explains the
name of this node function set; see Parberry [1991]. An alternating circuit is a circuit
in which all of the nodes in any given layer compute the same function, the nodes at
level 1 are NOT-nodes, and the layers, apart from the first one, alternate between
AND-nodes and OR-nodes. Figure 4.1b presents an example of an alternating circuit
that computes the parity function. As usual, AC denotes the class of problems that
can be solved by A-circuits; see also Section 4.3.

It is well-known that every Boolean function can be computed by some N-circuit,
and, since A is a proper subset of A, the same holds for A-circuits. Thus the node function
sets A and A are complete.

**Definition 4.2.5.** A node function set \( F \subseteq \bigcup_{N \in \mathbb{N}} \{ \mathbb{B}^N \to \mathbb{B} \} \) is called complete, if
every Boolean function can be computed by some F-circuit.

In addition to A and A, we introduce nine more complete node function sets for
the construction of circuits. To this end we define the following seven functions.

**Definition 4.2.6.** The disjunction AND\(^N\), conjunction OR\(^N\), majority function
MAJ\(^N\), congruent-modulo-p function C\(^N\)_p, exactly-p function E\(^N\)_p,
threshold-p function T\(^N\)_p, and weighted threshold function WT\(^N\)_w, are for every \( N \in \mathbb{N}, p \in \mathbb{N}, x \in \mathbb{B}^N, \) and \( w \in \mathbb{R}^N \), defined by

\[
\begin{align*}
\text{AND}^N(x) &= 1 \iff \sum_{i=1}^{N} x_i = N, \\
\text{OR}^N(x) &= 1 \iff \sum_{i=1}^{N} x_i \geq 1, \\
\text{MAJ}^N(x) &= 1 \iff \sum_{i=1}^{N} x_i \geq N/2, \\
C^N_p(x) &= 1 \iff \sum_{i=1}^{N} x_i \equiv 0 \pmod{p}, \\
E^N_p(x) &= 1 \iff \sum_{i=1}^{N} x_i = p, \\
T^N_p(x) &= 1 \iff \sum_{i=1}^{N} x_i \geq p, \\
\text{WT}^N_w(x) &= 1 \iff \sum_{i=1}^{N} w_i x_i \geq 0,
\end{align*}
\]

respectively.

The node function sets considered here for constructing circuits are defined as folows.

**Definition 4.2.7.** The node function sets corresponding to circuits with bounded
fan-in, alternating circuits, congruent-modulo-p circuits, all congruents circuits,
majority circuits, threshold circuits, polynomial integer weighted threshold circuits,
super polynomial integer weighted threshold circuits, exponential integer weighted
threshold circuits, integer weighted threshold circuits, and real weighted threshold
circuits, are defined by

\[
\begin{align*}
\mathcal{N} &= \{ \text{AND}^3, \text{OR}^3, \text{NOT}, 1 \}, \\
\mathcal{A} &= \bigcup_{N \in \mathbb{N}} \{ \text{AND}^N, \text{OR}^N, \text{NOT}, 1 \}, \\
p\mathcal{A} &= \mathcal{A} \cup \bigcup_{N \in \mathbb{N}} \{ C^N_p \},
\end{align*}
\]
\[ \mathcal{AC} = \mathcal{A} \cup \bigcup_{N \in \mathbb{N}, p \in \mathbb{N}} \{ C^N_p \}, \]
\[ \mathcal{E} = \mathcal{A} \cup \bigcup_{N \in \mathbb{N}, p \in \mathbb{N}} \{ E^N_p \}, \]
\[ \mathcal{M} = \mathcal{A} \cup \{ \text{MAJ}^N \}, \]
\[ T = \mathcal{A} \cup \bigcup_{N \in \mathbb{N}, p \in \mathbb{N}} \{ T^N_p \}, \]
\[ \mathcal{PWT} = \mathcal{A} \cup \bigcup_{N \in \mathbb{N}, w \in \mathbb{R}^N} \{ \text{WT}^N_w \mid \log \|w\|_\infty = O(\log N) \}, \]
\[ \mathcal{SPWT} = \mathcal{A} \cup \bigcup_{N \in \mathbb{N}, w \in \mathbb{R}^N} \{ \text{WT}^N_w \mid \log \|w\|_\infty = \log^\omega(N) \}, \]
\[ \mathcal{EIWT} = \mathcal{A} \cup \bigcup_{N \in \mathbb{N}, w \in \mathbb{R}^N} \{ \text{WT}^N_w \mid \log \|w\|_\infty = N^{O(1)} \}, \]
\[ \mathcal{ITWT} = \mathcal{A} \cup \bigcup_{N \in \mathbb{N}, w \in \mathbb{R}^N} \{ \text{WT}^N_w \}, \]
\[ \mathcal{RTWT} = \mathcal{A} \cup \bigcup_{N \in \mathbb{N}, w \in \mathbb{R}^N} \{ \text{WT}^N_w \}, \]
respectively, where \( \|w\|_\infty = \max \{ w_i \} \).

Note that it is useless to consider node function sets of real-valued weighted thresholds with bounds on the size of the real-valued weights used, since \( \text{WT}^N_{\alpha w} = \text{WT}^N_w \), for all \( \alpha > 0 \). However, it might be interesting to consider other types of constraints on the real-valued weights. We mention two possibilities of such constraints; see also Shawe-Taylor, Anthony & Kern [1992].

- The number of non-zero bits of the weights is bounded by a polynomial in \( N \).
- An \( N \)-bit approximation of the weights can be computed in a time bounded by a polynomial in \( N \).

In Section 4.3 we use the node function sets given above to define a number of complexity classes. One set of classes is named by putting a \( \mathcal{C} \) behind the corresponding node function set. In this way we obtain for example \( \mathcal{AC}, \mathcal{ACC} \) and \( TC \). Therefore, the names for the above node function sets have been chosen such that the name of the resulting complexity class corresponds to the one used in literature; see for instance [Dutte, 1988; Johnson, 1990; Karp & Ramachandran, 1990; Shmoys & Tardos, 1990; Wegener, 1987]. An exception is our choice for the threshold function and the majority function, which are in some papers interchanged. On the other hand the above names are an attempt to construct the names in a unifying manner. Consequently, we use \( p-\mathcal{AC}^k \) instead of the usual \( \mathcal{AC}^k(p) \); see Section 4.3. A drawback of our notation is that \( \mathcal{AC} \) stands for the class of problems that can be solved with \( \mathcal{A} \)-circuits, as well as for one of the node function sets. Usually, this does not lead to confusion. The complexity classes introduced in Section 4.3 are used to study the relative power of the circuits defined in this section and to compare them to the power of certain types of feedforward neural networks defined in the following section.
4.2.2 Feedforward neural networks

A second subclass of the UCMs is obtained by choosing \( D = \mathbb{R} \) and arbitrary \( R \subseteq \mathbb{R} \). A UCM in this class is called a feedforward neural network. Among the feedforward neural networks there is also a number of subclasses that can be distinguished. One of the best studied is the class of multi-layered perceptrons (MLPs). These have in common that the node functions all consist of an arbitrary, real-valued, response function applied to a weighted sum of the inputs of the node functions. We differentiate between the various MLP-models by the type of response functions used.

**Definition 4.2.8.** Let \( WL^N_w : \mathbb{R}^N \rightarrow \mathbb{R} \) be defined by \( WL^N_w(x) = w \cdot x \), for all \( x \in \mathbb{R}^N \), \( w \in \mathbb{R}^N \), and \( N \in \mathbb{N} \), and let \( \Sigma \) be a set of functions from \( \mathbb{R} \) to \( \mathbb{R} \). A UCM is called a \( \Sigma \)-MLP, if the node function set is

\[
\mathcal{F}_C = \{1\} \cup \bigcup_{w \in \mathbb{R}^N} \bigcup_{N \in \mathbb{N}} \{ \gamma \circ WL^N_w \}.
\]

(4.1)

If \( \Sigma \) consists of a single function \( \gamma : \mathbb{R} \rightarrow \mathbb{R} \) we speak of \( \gamma \)-MLPs.

In Chapter 2 we gave a more constructive definition of \( \Sigma \)-MLPs. That the two definitions are equivalent follows directly from the following result.

**Proposition 4.2.1.** The set of vector functions that can be computed by a \( \Sigma \)-MLP with \( N \) inputs, \( K \) outputs and depth \( m \), equals \( \Sigma \)-\( R_{n,K} \) defined in Definition 2.3.1.

The class of response functions is usually restricted in one way or another. Except that most authors consider non-linear \([0,1]\)-functions, the restrictions and the names used for the response functions vary from author to author. In Table 2.1 of Chapter 2, we have presented some of the most used restrictions and names. There are two sigmoidal functions that deserve special attention, because they are considered in the majority of the literature on feedforward neural networks.

**Definition 4.2.9.** The sigmoid or logistic function \( \sigma \), and the hard-limiting or step function \( \theta \), are for all \( x \in \mathbb{R} \) defined by

\[
\sigma(x) = \frac{1}{1 + \exp(-x)},
\]

and

\[
\theta(x) = \begin{cases} 
1, & \text{if } x \geq 0, \\
0, & \text{if } x < 0,
\end{cases}
\]

respectively.

The sigmoid function is often used in the backpropagation algorithm given by Rumelhart, Hinton & Williams [1986], and satisfies all of the constraints listed in Table 2.1. The hard-limiting function \( \theta \) satisfies \( \theta(+\infty) = 1 \), \( \theta(-\infty) = 0 \), and is non-decreasing. A consequence of the “feed-next” assumption is that \( \theta \)-MLPs have a special first layer, since this is the only layer with real-valued inputs.

Similarly as done for circuits we define some special node function sets. Note that in this case it does make sense to consider bounds on the size of the real-valued weights, since for instance \( \sigma \circ WT^N_w \neq \sigma \circ WT^N_w \), for \( \alpha > 0 \) in general.

**Definition 4.2.10.** Let \( \gamma : \mathbb{R} \rightarrow \mathbb{R} \) be a response function. Then the node function sets corresponding to polynomial integer weighted \( \gamma \)-MLPs, super polynomial integer...
4.3 Complexity classes

weighted $\gamma$-MLPs, exponential integer weighted $\gamma$-MLPs, integer weighted $\gamma$-MLPs, polynomial real weighted $\gamma$-MLPs, super polynomial real weighted $\gamma$-MLPs, and real weighted $\gamma$-MLPs, are defined by

$$\gamma\text{-PTWL} = \{1 \cup \bigcup \bigcup N \in \mathbb{N} \cup \{\gamma \circ WL_{\infty}^{N} \mid \log \|w\|_{\infty} = O(\log N)\},$$

$$\gamma\text{-STIWLC} = \{1 \cup \bigcup \bigcup N \in \mathbb{N} \cup \{\gamma \circ WL_{\infty}^{N} \mid \log \|w\|_{\infty} = O(1)N\},$$

$$\gamma\text{-ETWLC} = \{1 \cup \bigcup \bigcup N \in \mathbb{N} \cup \{\gamma \circ WL_{\infty}^{N} \mid \log \|w\|_{\infty} = N^{O(1)}\},$$

$$\gamma\text{-IWLC} = \{1 \cup \bigcup \bigcup N \in \mathbb{N} \cup \{\gamma \circ WL_{\infty}^{N}\},$$

respectively.

In this section and the previous one, we have introduced node function sets for the subclasses of UCMs denoted as MLPs and Boolean circuits, respectively. In the following section these node function sets are used to define several complexity classes for the considered subclasses of UCMs.

4.3 Complexity classes

In the previous section we have introduced UCMs as a general framework for both Boolean circuits and MLPs. In this section we introduce a number of complexity classes for UCMs. Our approach is based on the approach of Chandra, Stockmeyer & Vishkin [1984], who introduce a number of complexity classes for Boolean circuits. The introduced complexity classes contain problems that can be solved with a restricted set of UCMs. For each complexity class the set of restrictions is different. Placing a problem in a certain class or showing that a problem is not a member of a certain class gives information of the complexity of that problem with respect to UCMs.

The outline of this section is as follows. First, we introduce a general framework for a complexity class for UCMs. Secondly, this framework is used to introduce a number of complexity classes for Boolean circuits. The majority of the introduced classes already exists in literature. The new classes concern Boolean circuits that have their fan-in bounded by a polynomial in the logarithm of their input size. With the introduced complexity classes we formulate a number of results about the complexity of Boolean circuits. Except for the results that concern the newly introduced classes, these results are available from literature. We focus on the results related to circuits of bounded depth and particularly to the weighted threshold circuits, since these are most relevant to the complexity of MLPs. Thirdly, we introduce a number of
complexity classes for MLPs similar to those for circuits, and use these to formulate some results on MLP-complexity that have recently appeared in literature. Finally, we show the connection between the complexity results given in this chapter and the results obtained in the Chapters 5, 6, and 7. This is done by formulating the results obtained in Chapter 7 on the five combinatorial optimization problems introduced in Chapter 3, in terms of the complexity classes introduced in this section.

4.3.1 UCM-complexity

Before we can define the UCM-complexity of a problem by introducing complexity classes, we have to define our notion of a problem, and we have to define what we understand by solving a problem with a UCM. Following the approach of Chandra, Stockmeyer & Vishkin [1984], we define a problem to be a set of vector functions, which may be real-valued. The set of functions is assumed to be infinite and each function in the set has a different number of inputs. Without loss of generality we assume that for all \( N \in \mathbb{N} \), the set contains a function with \( N \) inputs. This yields the following definition.

**Definition 4.3.1.** Let \( \mathcal{PD}, \mathcal{PR} \subseteq \mathbb{R} \). A problem is a set \( \{f_1, f_2, f_3, \ldots \} \), such that for every \( N \in \mathbb{N} \) there exists a \( K \in \mathbb{N} \) with \( f_N : \mathcal{PD}^N \rightarrow \mathcal{PR}^K \).

The sets \( \mathcal{PD} \) and \( \mathcal{PR} \) stand for problem domain and problem range, respectively. The usual shorthand notation for a problem is \( (f_N : \mathcal{PD}^N \rightarrow \mathcal{PR}^K(N))_{N \in \mathbb{N}} \), where \( K(N) \) is used to emphasize that \( K \) depends on \( N \), or simply \( \{f_N\}_{N \in \mathbb{N}} \), in case that \( \mathcal{PD} \) and \( \mathcal{PR} \) are known.

Usually, we define the same problem more loosely as follows.

**Definition 4.3.2.** Let \( \mathcal{PD}, \mathcal{PR} \subseteq \mathbb{R} \) and \( f_N : \mathcal{PD}^N \rightarrow \mathcal{PR}^K \), for some \( N \in \mathbb{N} \) and \( K \in \mathbb{N} \). Then the problem is given by

**INPUT:** \( x \in \mathcal{PD}^N \).

**OUTPUT:** \( f_N(x) \in \mathcal{PR}^K \).

Note that in Definition 4.3.2 we implicitly assume that \( N \) runs over all integers.

The formulation used in Definition 4.3.1 implies that for each input \( d \in \mathcal{PD}^N \) there is a unique output \( f_N(d) \). This is often true for arithmetical problems like integer addition and multiplication, which in the style of Definition 4.3.2, can be formulated as follows.

**Definition 4.3.3.** (ADDITION (MULTIPLICATION))

**INPUT:** Two \( N \)-bit binary numbers.

**OUTPUT:** Their sum (product).

It is straightforward to formulate ADDITION according to Definition 4.3.1. Let \( \mathcal{PD} = \mathcal{PR} = \mathbb{B}, K(N) = \lfloor N/2 \rfloor + 1 \), and define \( f_N : \mathbb{B}^N \rightarrow \mathbb{B}^{K(N)} \) by \( f_{2N-1}(d) \equiv 0 \), and

\[
f_N(d) = \tau \iff \tau \in \mathbb{B}^{N+1} \land \sum_{i=1}^{N}(d_i + d_{i+N})2^i = \sum_{i=1}^{N+1} \tau_i 2^i.
\]
4.3 Complexity classes

Then \( \text{ADDITION} = \{ f_N : \mathbb{B}^N \to \mathbb{B}^{K(N)} \}_{N \in \mathbb{N}} \). For other examples of problems with a unique solution we refer to [Chandra, Stockmeyer & Vishkin, 1984]. Note also that the node function sets given in the previous section can be considered as problems of the above kind.

In many cases a problem may not have a unique solution; see for instance the various classification and loading problems discussed in Chapter 2. One way to circumvent this difficulty is to consider the decision variant of such problems. In the decision variant of a problem the output is a single 0-1 variable denoting whether or not the original problem has a solution. The variant of the problem in which one asks for a solution of the original problem is usually called the search variant of the problem. As an example we consider the problem of finding an Euler cycle in a graph. Its decision variant can be given as follows.

**Definition 4.3.4. (Euler Cycle; Decision Variant)**

**INPUT:** The adjacency matrix of an undirected graph on \( N \) vertices.

**OUTPUT:** Is there a cycle that traverses every edge exactly once?

An alternative way of obtaining a unique solution is to add extra constraints to the output. A well-known method is to ask for the lexicographically first solution, where in this case the ordering applies to a numbering of the edges of the graph.

**Definition 4.3.5. (Euler Cycle; Search Variant)**

**INPUT:** The adjacency matrix of an undirected graph on \( N \) vertices.

**OUTPUT:** The lexicographically first cycle that traverses every edge exactly once, if it exists. “Infeasible”, otherwise.

In this thesis we use the second approach, i.e., we consider search variants that have been made unique through the addition of extra constraints. However, the addition of constraints must be done carefully, because it may affect the complexity of the problem. We return to this issue in Chapter 6.

Now that we have defined problems, we can think of how to solve them. For this we have introduced the UCMs. To be more precise, we consider families of UCMs that solve a given problem. This is defined as follows.

**Definition 4.3.6.** Let \( \mathcal{PD}, \mathcal{PR} \subseteq \mathbb{R} \). Let \( \mathcal{F} \subseteq \bigcup_{N \in \mathbb{N}} \{ f^N : \mathcal{D}^N \to \mathcal{R} \} \) be a node function set with \( \mathcal{D} \supseteq \mathcal{PD} \) and \( \mathcal{R} \supseteq \mathcal{PR} \). Then the the problem \( \{ f_N : \mathcal{PD}^N \to \mathcal{PR}^{K(N)} \}_{N \in \mathbb{N}} \) is said to be solvable by a family \( (M_N)_{N \in \mathbb{N}} \) of UCMs that use the node function set \( \mathcal{F} \), if for every \( N \), there exists a UCM \( M_N \) in the family, with \( N \) inputs and \( K(N) \) outputs, that can compute \( f_N \).

In the above definition we say that there exists a UCM \( M_N \) that can compute \( f_N \). More precisely, it should be formulated that there exists a UCM \( M_N \) that computes a function \( f_N : \mathcal{PD}^N \to \mathcal{PR}^{K(N)} \), such that \( f_N(z) = f_N(z) \), for all \( z \in \mathcal{PD}^N \), i.e., \( f_N \) is the restriction of \( f_N \) to \( \mathcal{PD}^N \).

Once we know that a problem can be solved by a family of UCMs, we want to know the complexity of the problem, i.e., the number of layers, the number of nodes, the size of the weights, the maximum fan-in, etc. A useful approach is to categorize
problems with respect to their UCM-complexity. This is done by the introduction of
complexity classes for UCMs, similarly as done for Boolean circuits; see for instance
Chandra, Stockmeyer & Vishkin [1984]. Each complexity class contains problems
that can be solved by families of UCMs that satisfy a certain set of restrictions. The
restrictions used below apply to the depth, size, node function set, set of possible
inputs, set of possible outputs, and maximum fan-in of the considered UCMs.

**Definition 4.3.7.** Let $PD, PR \subseteq R$, let $F \subseteq \bigcup_{n \in \mathbb{N}} (D^n \to R)$ be a node function
set with $D \supseteq PD$ and $R \supseteq PR$, and let $D$, $S$, and $FI$ be functions from $N$ to $R^+$. Then the complexity class denoted by

$$UCM[D \ | \ S \ | \ F \ | \ PD \ | \ PR \ | \ FI],$$

contains all problems $\{f_N : PD^N \to PR^{N(N)}\}_{N \in \mathbb{N}}$ such that for every $N \in \mathbb{N}$,
$f_N$ can be computed by a UCM with node function set $F$, $N$ inputs, $K(N)$ outputs,
depth $\leq D(N)$, size $\leq S(N)$, and maximum fan-in $\leq FI(N)$.

The above definition does not contain an explicit restriction for the weights used in
for instance weighted threshold circuits. Instead we have chosen to restrict the
weights through a limitation of the node function set used. This is because we think
that it gives the clearest formulation, as not all node function sets have weights, and
it allows for a uniform naming of the different complexity classes. On the other hand
the restriction on the fan-in is given explicitly, while it can also be given implicitly
through a restriction on the node function set. The reason for this choice is that it is
in accordance with existing notation. Note that we do not consider a restriction
on the maximum fan-out of the considered UCMs because it falls outside the scope
of this thesis; see [Maroga, 1971] for more about feedforward networks with limited
fan-out, and [Wegener, 1987] for more about Boolean circuits with limited fan-out.

The restrictions to the depth, size, and maximum fan-in are given by the functions
$D$, $S$, and $FI$, respectively. For every choice of one of these functions Definition 4.3.7
yields a new complexity class. This amount of detail is undesirable, as one is usually
interested in an order estimate only. We therefore allow $D$, $S$, and $FI$, to be sets of
functions also. If $D'$ is a set of functions then by definition

$$UCM[D' \ | \ S \ | \ F \ | \ PD \ | \ PR \ | \ FI] = \bigcup_{D \in D'} UCM[D \ | \ S \ | \ F \ | \ PD \ | \ PR \ | \ FI],$$

and similarly for the size and fan-in restrictions, and combinations of these restrictions.
We consider the following sets of functions, where $k$ may be any non-negative
real-valued constant.

**Definition 4.3.8.** For all $k \geq 0$

- $\text{CONST} = \{f : N \to R^+ \mid f(N) = O(1)\}$,
- $\log^k = \{f : N \to R^+ \mid f(N) = O(\log^k N)\}$,
- $\text{POLYLOG} = \{f : N \to R^+ \mid f(N) = O(\log^k N)\}$,
- $\text{POLY} = \{f : N \to R^+ \mid \log f(N) = O(\log N)\}$,
- $\text{SUPERPOLY} = \{f : N \to R^+ \mid \log f(N) = O(\log N)\}$,
- $\text{EXP} = \{f : N \to R^+ \mid f(N) = N^{O(1)}\}$,
- $\infty = \{f : N \to R^+\}$. 
4.3 Complexity classes

So, as an example, the complexity class

$$\text{UCM} | \text{POLYLOG} | \text{POLY} | \mathcal{F} | \text{PD} | \mathcal{PR} | \infty$$

denotes the class of problems \( \{ f_N \}_{N \in \mathbb{N}} \) for which \( f_N \) can be computed by a UCM with node function set \( \mathcal{F} \), \( N \) inputs, depth \( \leq c_1 \log^{k_1} N \), size \( \leq c_2 N^{k_2} \), and unbounded fan-in, for some \( c_1, c_2 > 0 \) and \( k_1, k_2 \geq 0 \).

The definition of a complexity class as given above, only requires the existence of a UCM for each \( f_N \). It does not specify how each UCM is to be constructed for each \( N \), nor at what cost this can be done, i.e., the what the complexity is of the construction. Solving a problem by a family of UCMs is therefore called a "nonuniform" method for solving a problem; see for more on the difference between uniform and nonuniform computational models Johnson [1990] and Shmoys & Tardos [1990]. A uniform family of UCMs is one in which the UCM with \( N \) inputs can be generated automatically, given \( N \). Different models of uniformity can be distinguished, which differ with respect to the complexity of the generation mechanism that is allowed; see Johnson [1990] for a description of the most frequently used models. As a rule, we consider here nonuniform families of UCMs. However, at some places where the results for uniform families differ significantly this is notified.

We end this section with some final remarks that mainly concern the complexity classes introduced in the following sections.

1. We do not consider the concept of reduction, mainly because we have not been able to generalize it to UCMs. At some places we make some remarks with respect to constant-depth or projection reducibility; see for an explanation of these notions Chandra, Stockmeyer & Vishkin [1984], Dunne [1988] or Wegener [1987].

2. In general we do not consider other models for sequential and parallel computation like Turing machines and PRAMs. For an overview of the relation between the various models we refer to [Johnson, 1990].

3. We consider deterministic computations and deterministic complexity classes only.

For more on circuit complexity we refer to the excellent overviews by Johnson [1990], Karp & Ramachandran [1990], and Shmoys & Tardos [1990], the books by Wegener [1987], Dunne [1988], and Parberry [1987], and the papers by Chandra, Stockmeyer & Vishkin [1984], Parberry [1991], and Parberry [1990].

4.3.2 Circuit complexity

In this section we use the framework formulated in the previous section to discuss some results on the complexity of Boolean circuits. We focus on the results that are relevant in connection with our study of the complexity of MLPs. This implies that we mainly look at Boolean circuits of bounded depth, in particular of logarithmic depth or less, and at circuits that use the threshold node function set or one of the weighted threshold node function sets. The results considered beyond this central theme are presented to be able to compare the capabilities of threshold circuits with those of more familiar models of computation.
The main order in which we present the results is determined by the depth of the considered circuits, and can be subdivided into three parts. Firstly, we consider circuits of logarithmic depth. Secondly, we increase depth and consider circuits of polylogarithmic depth or more. Thirdly, we study some results for circuits of constant and fixed depth. A problem is solved by a family of circuits of constant depth, if the depth of the circuits is bounded by an arbitrary constant. A problem is solved by a family of circuits of fixed depth, if the depth of the circuits is bounded by a beforehand chosen constant. Except for general results we also examine the complexity of the integer versions of the five problems introduced in Chapter 3.

Logarithmic depth circuits
A Boolean circuit is a UCM with \( D = R = B \), i.e., it is allowed to use binary node functions only. Consequently, circuits can only be used for solving binary problems, i.e., \( PD = PR = B \). Below, we introduce a first set of circuit complexity classes for an arbitrary binary node function set \( F \). These two classes contain problems that can be solved by circuits with a polynomial number of nodes, unbounded fan-in, and depth bounded by a polynomial of the logarithm of the number of inputs, or bounded by a fixed power of the logarithm of the number of inputs, respectively.

**Definition 4.3.9.** Let \( F \) be a node function set corresponding to a circuit. Then the complexity classes \( FC \) and \( FC^k \) are defined by

\[
\begin{align*}
FC &= UCMPOLYLOG|POLY|F|B|B|\infty], \\
FC^k &= UCMPOLYLOG|POLY|F|B|B|\infty],
\end{align*}
\]

for all \( k \geq 0 \).

For \( F \) one can substitute any of the node function sets introduced in Section 4.2.1. This yields for instance the complexity classes \( pAC, EC, TC^k \), and \( SPIWTCC^k \). The well-known classes \( AC^k \) and \( NC^k \) are among the best studied complexity classes for circuits, representing problems that can be solved by \( \mathcal{A} \)-circuits of depth \( O(\log^k N) \), polynomial size, unbounded fan-in, and bounded fan-in, respectively. Note that \( NC \) can also be defined as

\[
NC = UCMPOLYLOG|POLY|A|B|B|\text{CONST}],
\]

and similarly for \( NC^k \). Except for the \( NC \)-classes, the complexity classes defined in Definition 4.3.9 correspond to Boolean circuits with unbounded fan-in. However, since the number of nodes of these circuits is bounded by a polynomial in the number of inputs, the fan-in is also bounded by a polynomial in the number of inputs. Next, we introduce a set of classes in which the fan-in is polylogarithmic in the number of inputs, which fills the gap between the classes with constant fan-in and the classes with polynomial fan-in.

**Definition 4.3.10.** Let \( F \) be a node function set corresponding to a circuit. Then the complexity classes \( FB \) and \( FB^k \) are defined by

\[
\begin{align*}
FB &= UCMPOLYLOG|POLY|F|B|B|POLYLOG], \\
FB^k &= UCMPOLYLOG|POLY|F|B|B|POLYLOG],
\end{align*}
\]

for all \( k \geq 0 \).

The classes introduced in Definition 4.3.10 are of interest due to the results presented in Theorem 4.3.1 and Theorem 4.3.2 given below. These theorems describe
the relation between the various classes corresponding to Boolean circuits of depth bounded by $O(\log^k N)$, that use the different node function sets introduced for Boolean circuits.

**Theorem 4.3.1.** For all $k \geq 0$

$$\mathcal{N}^k \subseteq \mathcal{A}^k \subseteq \mathcal{P}\mathcal{A}^k \subseteq \mathcal{A}\mathcal{C}^k \subseteq \mathcal{E}\mathcal{C}^k = \mathcal{M}\mathcal{C}^k = T\mathcal{C}^k \subseteq N\mathcal{C}^{k+1},$$

and

$$\mathcal{N}^k \subseteq \mathcal{A}^k \subseteq \mathcal{P}\mathcal{A}^k \subseteq \mathcal{A}\mathcal{C}^k \subseteq \mathcal{E}\mathcal{C}^k = \mathcal{M}\mathcal{B}^k = T\mathcal{B}^k \subseteq \mathcal{A}^k.$$

The above result can be proven using the concept of constant-depth reducibility; see [Chandra, Stockmeyer & Vishkin, 1984; Wegener, 1987]. In fact Chandra, Stockmeyer & Vishkin [1984] show that $\mathcal{A}$, $\mathcal{P}\mathcal{A}$, $\mathcal{E}$, $\mathcal{M}$, and $\mathcal{T}$, viewed as problems satisfy the same relation as their $C$-classes given above, with $\subseteq$ replaced by $\subseteq_{\text{cut}}$, denoting constant-depth reducible. However, we do not use constant-depth reducibility, since it is based on uniform $A$-circuits, which complicates the argumentation. Instead we use the following lemma.

**Lemma 4.3.1.** Let $F$ be a node function set corresponding to a circuit, and let $F = \{f_N : B^N \to B^{K(N)}\}_{N \in \mathbb{N}}$ be a problem in $\mathcal{F}^k$. Then for all $k \geq 0$, we have $\mathcal{A}^k \subseteq \mathcal{F}^k$ and $\mathcal{M}^k \subseteq \mathcal{F}^k$, where $H$ is the node function set defined by

$$H = \bigcup_{N \in \mathbb{N}} \bigcup_{i=1}^{K(N)} \{ (f_N)_i \}.$$  \hspace{-2cm} (4.2)

**Proof.** We prove the first result. Changes that have to be made in order to change the proof into a proof for the second result are placed between parenthesis.

Let the problem $H = \{h_N\}_{N \in \mathbb{N}}$ be in $\mathcal{A}^k$ ($\mathcal{M}^k$). Then for every $N \in \mathbb{N}$, $h_N$ can be computed by a UCM given by $M = (X, Y, Z, E, G)$ with $N$ inputs, size $= p(N)$, depth $\leq c \cdot \log^k N$, unbounded fan-in (maximum fan-in $\leq \hat{p}(\log N)$), for some polynomial $p$ (and $\hat{p}$) and constant $c$, that uses the node function set $H$ defined by (4.2). Consider a certain node $x \in Z$ of this UCM $M$, and assume that it is a $(f_r)_r$-node, for some $r \leq p(N)$ ($r \leq \hat{p}(\log N)$) and $i \leq K(r)$. Since $F \in \mathcal{F}^k$, $f_r : B^r \to B^{K(r)}$ is computed by a UCM $M'$ with $r$ inputs, size $= \hat{p}(r)$, and depth $\leq \hat{c}$, for some polynomial $\hat{p}(r)$ and constant $\hat{c}$. (Obviously, in this UCM every node has fan-in $\leq \text{size} + r \leq \hat{p}(r) + r$.)

If we replace node $x$ in $M$ by the UCM $M_1$, of which we use the $i$-th output only, the function computed by the obtained UCM is still $h_N$. If we replace all nodes $x \in Z$ in $M$ in this way, we obtain an UCM $M = (X, Y, Z, E, G)$ that computes $h_N$ and has $N$ inputs, size $\leq p(N) \hat{p}(p(N))$, depth $\leq \hat{c} \log^k N$, (fan-in $\leq \hat{p}(\hat{p}(\log N)) + \hat{p}(\log N)$), and that uses the node function set $F$. Hence, $H \in \mathcal{A}^k$ ($H \in \mathcal{F}^k$).

Using Lemma 4.3.1 and some circuit complexity results from [Wegener, 1987], the proof of Theorem 4.3.1 is straightforward.

**Proof of Theorem 4.3.1.** Since $\mathcal{N} \subseteq \mathcal{A} \subseteq \mathcal{P}\mathcal{A} \subseteq \mathcal{A}\mathcal{C}$ and $\mathcal{M} \subseteq \mathcal{T}$, we directly find that $\mathcal{N}^k \subseteq \mathcal{A}^k \subseteq \mathcal{P}\mathcal{A}^k \subseteq \mathcal{A}\mathcal{C}^k$ and $\mathcal{M}^k \subseteq \mathcal{T}^k$. Next, we observe that
for all \( N \in \mathbb{N} \), \( p = 1, \ldots, N \), and \( x \in \mathcal{B}^N \)

\[
C_p^N(x) = \bigvee_{i=0}^{\lfloor \frac{N}{p} \rfloor} E^N_i(x),
\]

\[
T_p^N(x) = \bigvee_{i=p}^{N} E^N_i(x),
\]

\[
E^N_p(x) = T_p^N(x) \land \neg T_{p+1}^N(x),
\]

\[
T_p^N(x) = \text{MAJ}^N_p(x, y),
\]

where \( y \in \mathcal{B}^N \) is such that \( E^N_{N-p}(y) = 0 \). This proves that

\[
\{(C_1^N(x), \ldots, C_N^N(x))\}_{N \in \mathbb{N}} \in \mathcal{EC}^0,
\]

\[
\{(T_1^N(x), \ldots, T_N^N(x))\}_{N \in \mathbb{N}} \in \mathcal{EC}^0,
\]

\[
\{(E_1^N(x), \ldots, E_N^N(x))\}_{N \in \mathbb{N}} \in \mathcal{T}^0,
\]

\[
\{(T_1^N(x), \ldots, T_N^N(x))\}_{N \in \mathbb{N}} \in \mathcal{MC}^0,
\]

and, hence, using Lemma 4.3.1, it follows that \( \mathcal{ACC}^k \subseteq \mathcal{EC}^k \), \( \mathcal{TC}^k \subseteq \mathcal{EC}^k \), \( \mathcal{EC}^k \subseteq \mathcal{TC}^k \) and \( \mathcal{TC}^k \subseteq \mathcal{MC}^k \). With the above \( \mathcal{MC}^k \subseteq \mathcal{TC}^k \) this yields \( \mathcal{ACC}^k \subseteq \mathcal{EC}^k = \mathcal{MC}^k = \mathcal{TC}^k \).

Similarly, it follows that

\[
\mathcal{NB}^k \subseteq \mathcal{AB}^k \subseteq \mathcal{pAB}^k \subseteq \mathcal{ACB}^k \subseteq \mathcal{EB}^k = \mathcal{MB}^k = \mathcal{TB}^k.
\]

Note that \( \mathcal{NB}^k = \mathcal{NC}^k \) by definition. This implies that it remains to prove that \( \mathcal{TC}^k \subseteq \mathcal{NC}^{k+1} \) and \( \mathcal{TB}^k \subseteq \mathcal{AC}^k \). Both results are based upon powerful results obtained in circuit complexity theory.

Firstly, it can be shown that \( \{(T_1^N(x), \ldots, T_N^N(x))\}_{N \in \mathbb{N}} \in \mathcal{NC}^1 \); see for instance Wegener [1987]. Then \( \mathcal{TC}^k \subseteq \mathcal{NC}^{k+1} \) follows similarly as in the proof of Lemma 4.3.1.

Secondly, it can be shown that \( \{(T_1^N(x), \ldots, T_N^N(x))\}_{N \in \mathbb{N}} \in \mathcal{AC}^0 \), if \( f(N) = O(\log^r N) \), for some fixed \( r \geq 0 \); see Wegener, 1987, Theorem 11.2.2. Then \( \mathcal{TB}^k \subseteq \mathcal{AC}^k \) follows similarly as in the proof of Lemma 4.3.1. \( \square \)

Theorem 4.3.1 compares threshold circuits with circuits that use more elementary node functions. The result is that the threshold circuits are among the most powerful circuits distinguished in Theorem 4.3.1. In the following theorem it is shown that adding weights to the thresholds does not increase their capabilities.

**Theorem 4.3.2.** For all \( k \geq 0 \)

\[
\mathcal{TC}^k = \mathcal{PIWTC}^k = \mathcal{SPIWTC}^k = \mathcal{EIWT}^k = \mathcal{IWTC}^k = \mathcal{RWTC}^k,
\]

and

\[
\mathcal{TB}^k = \mathcal{PIWTB}^k = \mathcal{SPIWTB}^k = \mathcal{EIWTB}^k = \mathcal{IWTB}^k = \mathcal{RWWTB}^k.
\]

**Proof.** Firstly, it is obvious that \( \mathcal{TC}^k \subseteq \mathcal{PIWTC}^k \subseteq \mathcal{SPIWTC}^k \subseteq \mathcal{EIWT}^k \subseteq \mathcal{IWTC}^k \subseteq \mathcal{RWTC}^k \), and similarly for the same expression with \( \mathcal{C} \) replaced by \( \mathcal{B} \).

Secondly, by Proposition 2.2.4 it follows that weight-vectors with real values can be replaced by weight-vectors with integer weights. Hence, \( \mathcal{RWTC}^k \subseteq \mathcal{IWT}^k \) and \( \mathcal{RWWTB}^k \subseteq \mathcal{IWTB}^k \).

Thirdly, we already mentioned in Chapter 2 the result by Muroga, Toda & Takaue
[1960], who showed that an integer weight-vector of arbitrary weight can be replaced by an integer weight vector of size at most $O(N^N) = 2^{O(N^2)}$, if $N$ is the number of inputs of the threshold-node corresponding to the weight; see also Proposition 2.2.5 and below. This yields $\text{TWTC}^k \subseteq \text{EIWTC}^k$ and $\text{TW TB}^k \subseteq \text{SPIW TB}^k$.

Finally, Chandra, Stockmeyer & Vishkin [1984] have shown that the problem of adding $N$ integers of $N$ bits each is in $\text{TC}^0$. Hence, similarly as in the proof of Lemma 4.3.1 one can show that $\text{EIWTC}^k \subseteq \text{TC}^k$ and $\text{SPIW TB}^k \subseteq \text{TB}^k$. □

Except for the $\text{TWTC}^k = \text{RWTC}^k$ and $\text{TW TB}^k = \text{RW TB}^k$ equalities, the results given in Theorem 4.3.1 and Theorem 4.3.2 hold for polynomial-time uniform circuits as well as for nonuniform circuits. Using the concept of constant-depth reducibility the same result can be shown for the further restricted class of $\text{LOGSPACE}$ or $\text{NC}^1$-uniform circuits; see Chandra, Stockmeyer & Vishkin [1984] and Wegener [1987]. Definitions of $\mathcal{P}$-uniform circuits and $\text{NC}^1$-uniform circuits are also given in [Johnson, 1990].

Above logarithmic

The results of Theorem 4.3.1 and Theorem 4.3.2 can be extended to the classes of polylog depth. In fact all the introduced polylog classes collapse onto one class; see Corollary 4.3.1, which mentions the most relevant polylog classes only. This follows directly from Theorem 4.3.1 and Theorem 4.3.2, using $\mathcal{FC} = \cup_{k \geq 0} \mathcal{FC}^k$ and $\mathcal{FB} = \cup_{k \geq 0} \mathcal{FB}^k$.

Corollary 4.3.1.

$$\text{NC} = \mathcal{AB} = \text{TB} = \mathcal{AC} = \text{TC} = \text{TWTC} = \text{RWTC}.$$ (4.3)

Except for the last equality, the equalities in (4.3) hold for $\mathcal{P}$-uniform families as well as for nonuniform families.

If we allow the depth of the considered circuits to be unbounded, it is bounded by the number of nodes, i.e., polynomial. We introduce the following notation.

Definition 4.3.11. Let $\mathcal{F}$ be a node function set corresponding to a circuit. Then the complexity class $\mathcal{FC}^\infty$ is defined by

$$\mathcal{FC}^\infty = \text{UCM}[\infty \mid \text{POLY} \mid \mathcal{F} \mid \mathcal{B} \mid \mathcal{B} \mid \infty].$$

It is well-known that uniform $\mathcal{AC}^\infty$ equals $\mathcal{P}$; see for instance Parberry [1991]. Furthermore, nonuniform $\mathcal{AC}^\infty$ is usually denoted as $\mathcal{P}/\text{POLY}$; see for instance Johnson [1990]. We thus have in the uniform case

$$\text{TWTC} = \mathcal{AC} \subseteq \mathcal{AC}^\infty = \mathcal{P} \subseteq \text{NP},$$

and in the nonuniform case

$$\text{RW TC} = \mathcal{AC} \subseteq \mathcal{AC}^\infty = \mathcal{P}/\text{POLY}.$$
Probably the best-known conjecture of this type is that \( \mathcal{NP} \not\subseteq \mathcal{P} \), with the candidates for \( \mathcal{P} \setminus \mathcal{NP} \) membership the \( \mathcal{NP} \)-complete problems; see Garey & Johnson [1979]. The relation between \( \mathcal{P}/\text{poly} \) and \( \mathcal{NP} \) is still unclear. It is known that \( \mathcal{P}/\text{poly} \not\subseteq \mathcal{NP} \), because \( \mathcal{P}/\text{poly} \) contains undecidable or non-recursively computable problems; see for references Shmoys & Tardos [1990]. Furthermore, it is conjectured that \( \mathcal{NP} \not\subseteq \mathcal{P}/\text{poly} \); see Johnson [1990] and Shmoys & Tardos [1990]. This implies that it is unlikely to find a \( \mathcal{RWT} \)-circuit of polynomial depth and polynomial size that solves a given \( \mathcal{NP} \)-complete problem, and even more unlikely to find uniform \( \mathcal{TWT} \)-circuits of polylog depth and polynomial size for such a problem.

**Sublogarithmic depth**

Next, we consider some results on constant depth circuits and fixed depth circuits, with their number of nodes bounded by a polynomial, a power of the logarithm, or a constant function, of the number of inputs, respectively. We start with a result on constant depth, polynomial size circuits.

Although useful, the results so far did not yield any separation, i.e., it is still possible by what we discussed so far that \( \mathcal{NC}^0 \) is equal to \( \mathcal{NP} \). In Theorem 4.3.3 below we discuss a number of recent separation results on the constant depth level.

**Theorem 4.3.3** (Furst, Saxe & Sipser [1984], Smolensky [1987]).

For all prime numbers \( p \), we have

\[
\mathcal{NC}^0 \subseteq \mathcal{T}B^0 \subseteq \mathcal{AC}^0 \subseteq \text{p-AC}^0 \subseteq \text{ACC}^0.
\]

**Proof.** Firstly, if \( \{f_N\}_{N \in \mathbb{N}} \) is a problem in \( \mathcal{NC}^0 \), \( \mathcal{TB}^0 \), or \( \mathcal{AC}^0 \), then one easily sees that \( f_N \) depends on at most a constant number of inputs, a polylog number of inputs, or all \( N \) inputs, respectively. This is easily shown to imply that \( \mathcal{NC}^0 \subseteq \mathcal{T}B^0 \subseteq \mathcal{AC}^0 \).

Secondly, \( \mathcal{AC}^0 \subseteq \text{p-AC}^0 \) is a landmark result of Furst, Saxe & Sipser [1984]. They have proved that \( \mathcal{PAR}^N = \{ \text{PAR}^N \}_{N \in \mathbb{N}} \not\subseteq \mathcal{AC}^0 \), where \( \text{PAR}^N(x) \) is the parity function which equals 1, if the sum of its inputs is odd. Since \( \text{PAR}^N(x) = \text{NOT}(C_N^0(x)) \), this proves \( \mathcal{AC}^0 \subseteq 2\cdot\mathcal{AC}^0 \), which was later generalized to all primes.

Thirdly, Smolensky [1987] that \( q\cdot\mathcal{AC}^0 \subseteq \text{p-AC}^0 \), if \( p \) is prime and \( q \neq p^k \). Hence, \( p\cdot\mathcal{AC}^0 \subseteq \text{ACC}^0 \), if \( p \) is prime. Note that \( q\cdot\mathcal{AC}^0 = \text{p-AC}^0 \), if \( p \) is prime and \( q = p^k \); see also Chandra, Stockmeyer & Vishkin [1984].

\( \square \)

**Remarks.**

1. The result by Furst, Saxe & Sipser was improved by Yao [1985], who gave the first exponential lower bound. His complicated proof was much simplified and improved by Hastad [1986]; see also Wegener [1987].

2. The techniques for proving that a certain problem is not an element of \( 2\cdot\mathcal{AC}^0 \) were developed by Razborov in a series of papers; see Razborov [1985a], Razborov [1985b], and Razborov [1987]. Using these techniques he showed for instance that \( \{ \text{MAJ}^N \}_{N \in \mathbb{N}} \not\subseteq 2\cdot\mathcal{AC}^0 \); see also Dunne [1988].

3. Using the concept of constant depth reducibility other problems can be excluded from membership of \( \mathcal{AC}^0 \) and \( 2\cdot\mathcal{AC}^0 \); see Chandra, Stockmeyer & Vishkin [1984] for a number of examples.
4.3 Complexity classes

The separation results given in Theorem 4.3.3 are the most powerful separation results known until now. It is for instance not known whether or not $\mathcal{AC}^0 \neq \mathcal{NP}$: see also Johnson [1990]. Furthermore, the above results show that the threshold node function set is among the most powerful node function sets considered for constant-depth and polylog-depth circuits. Chandra, Stockmeyer & Vishkin have shown that problems like INTEGER MULTIPLICATION, INTEGER MULTIPLE ADDITION, and INTEGER SORTING all belong to $\mathcal{TC}^0$; see Chandra, Stockmeyer & Vishkin [1984] and also Siu & Bruck [1990]. Hence, also a problem like INTEGER MATRIX MULTIPLICATION belongs to $\mathcal{TC}^0$. This already indicates that $\mathcal{TC}^0$ is a relatively rich class of problems. Furthermore, Reif has shown that $\mathcal{TC}^0$ contains even more difficult problems like multiplying $N$ numbers of $N$ bits each, INTEGER DIVISION, and problems that can be represented by some power series; see Reif [1987] and also Beame, Cook & Hoover [1984], who originally showed how to solve these problems in $\mathcal{NC}^1$. In [Parberry & Schnitger, 1988; Parberry & Schnitger, 1989] the relation between $\mathcal{TC}^0$ and other parallel computational models is studied, one of which is the Boltzmann machine.

Before the constant-depth separation results where found, some separation results were known on the lower level of fixed depth circuits. We distinguish the following complexity classes corresponding to fixed depth circuits.

**Definition 4.3.12.** Let $\mathcal{F}$ be a node function set corresponding to a circuit. Then the complexity classes $\mathcal{FC}^0_d$, $\mathcal{FC}^{0,l}_d$, and $\mathcal{FC}^{0,s}_d$ are defined by

$$\mathcal{FC}^0_d = \text{UCM}[d | \text{POLY} | \mathcal{F} | \mathcal{B} | \mathcal{B} | \infty],$$

$$\mathcal{FC}^{0,l}_d = \text{UCM}[d | \log^l | \mathcal{F} | \mathcal{B} | \mathcal{B} | \infty],$$

$$\mathcal{FC}^{0,s}_d = \text{UCM}[d | s | \mathcal{F} | \mathcal{B} | \mathcal{B} | \infty],$$

for all $l \geq 0$ and $d, s \in \mathbb{N}$.

The polylog and constant size circuits defined above are used in the following section.

Many problems in $\mathcal{AC}^0$ can be shown to be outside $\mathcal{AC}^0_d$; see [Dunne, 1988; Wegener, 1987] for some references. Of the same kind is also the result of Minsky & Papert who show that PARITY cannot be solved by a depth 3 circuit of polynomial size with AND functions in the first layer and a threshold function in the output layer; see Theorem 2.2.3 and Minsky & Papert [1969]. For $\mathcal{AC}^0_d$ with $d \geq 3$, far less results are known. An important breakthrough was the separation result of Sipser, who proved the following; see also Sipser [1983] and Dunne [1985].

**Proposition 4.3.1 (Sipser [1983]).**

$$\mathcal{AC}^0_1 \subset \mathcal{AC}^0_2 \subset \mathcal{AC}^0_3 \subset \cdots.$$

For the class $\mathcal{TC}^0_d$ separation has been demonstrated up to $d = 3$ only. This result is due to Hajnal, Maass, Pudlak, Szegedy & Turan [1987], who also use a problem based on PARITY; see Chapter 2 for more on PARITY and a proof that PARITY $\notin \mathcal{TC}^0_2$.

**Theorem 4.3.4 (Hajnal, Maass, Pudlak, Szegedy & Turan [1987]).**

$$\mathcal{TC}^0_1 \subset \mathcal{TC}^0_2 \subset \mathcal{TC}^0_3.$$

**Proof.** First, $\{\text{PAR}^N\}_{N \in \mathbb{N}} \notin \mathcal{TC}^0_1$, as every threshold node is monotone in every variable. Since $\{\text{PAR}^N\}_{N \in \mathbb{N}} \in \mathcal{TC}^0_2$, it follows that $\mathcal{TC}^0_2 \subset \mathcal{TC}^0_3$. 


Secondly, using a remarkable simple proof Hajnal, Maass, Pudlak, Szegedy & Turan have shown that \( \{ \text{IP}^N \}_{N \in \mathbb{N}} \not\in \mathcal{T}C_2^0 \), where \( \text{IP}^N(x,y) = \text{PAR}^N(x_1, y_1, \ldots, x_N y_N) \) is the inner-product modulo-2 function. Furthermore, since \( \{ \text{PAR}^N \}_{N \in \mathbb{N}} \in \mathcal{T}C_2^0 \), it follows that \( \{ \text{IP}^N \}_{N \in \mathbb{N}} \in \mathcal{T}C_2^0 \), and, hence, \( \mathcal{T}C_2^0 \subseteq \mathcal{T}C_3^0 \). \( \square \)

Remarks.

1. Hajnal, Maass, Pudlak, Szegedy & Turan [1987] in fact showed that \( \{ \text{IP}^N \}_{N \in \mathbb{N}} \) is not a member of \( \mathcal{P} \mathcal{I} \mathcal{W} \mathcal{T}C_2^0 \). Furthermore, they mention that the proof can be extended to threshold circuits having arbitrary weights on the first level.

2. Krause & Waack [1991] extended the result on the lower bound for \( \{ \text{IP}^N \}_{N \in \mathbb{N}} \), to depth 2 circuits with arbitrary symmetric nodes at the bottom level and a threshold node at the top. Furthermore, they proved lower bounds for such circuits with \( C_p^N \) nodes at the top.

3. The result for \( \{ \text{IP}^N \}_{N \in \mathbb{N}} \) also proves that \textsc{multiplication} and \textsc{connectivity} are not in \( \mathcal{T}C_2^0 \), since \textsc{inner-product-mod-2} is projection reducible to these problems; see Hajnal, Maass, Pudlak, Szegedy & Turan [1987].

Because Lemma 4.3.1 does not hold for fixed depth classes, the results of Theorem 4.3.2 do not automatically carry over to the threshold classes of fixed depth. However, the steps of the proof of Theorem 4.3.2 that do not use Lemma 4.3.1 hold also for fixed depth. Furthermore, one easily shows that \( \mathcal{P} \mathcal{I} \mathcal{W} \mathcal{T}C_2^d \subseteq \mathcal{T}C_2^d \) by copying nodes.

**Corollary 4.3.2.**

\[
\mathcal{T}C_2^d = \mathcal{P} \mathcal{I} \mathcal{W} \mathcal{T}C_d^d \subseteq \mathcal{S} \mathcal{P} \mathcal{I} \mathcal{W} \mathcal{T}C_d^0 \subseteq \mathcal{E} \mathcal{I} \mathcal{W} \mathcal{T}C_d^0 = \mathcal{I} \mathcal{W} \mathcal{T}C_d^0 = \mathcal{R} \mathcal{W} \mathcal{T}C_d^0.
\]

We conjecture that the inclusions are strict. The case \( \mathcal{S} \mathcal{P} \mathcal{I} \mathcal{W} \mathcal{T}C_d^0 \subseteq \mathcal{E} \mathcal{I} \mathcal{W} \mathcal{T}C_d^0 \), follows from Theorem 2.2.4.

Finally, we introduce some complexity classes in which the number of nodes is unbounded. The corresponding computational models fall into the class of models for unbounded parallelism.

**Definition 4.3.13.** Let \( \mathcal{F} \) be a node function set corresponding to a circuit. Then the complexity classes \( \mathcal{F} \mathcal{C}^{\ast, \infty} \), \( \mathcal{F} \mathcal{C}^{\ast, \infty} \), and \( \mathcal{F} \mathcal{C}^{0, \infty} \), are defined by

\[
\begin{align*}
\mathcal{F} \mathcal{C}^{\ast, \infty} &= \text{UCM}[\text{POLYLOG}] \infty \mid |\mathcal{F}| |\mathcal{B}| |\mathcal{B}| \infty, \\
\mathcal{F} \mathcal{C}^{2, \infty} &= \text{UCM}[\log^2] \infty \mid |\mathcal{F}| |\mathcal{B}| |\mathcal{B}| \infty, \\
\mathcal{F} \mathcal{C}^{0, \infty} &= \text{UCM}[d] \infty \mid |\mathcal{F}| |\mathcal{B}| |\mathcal{B}| \infty,
\end{align*}
\]

for all \( k \geq 0 \) and \( d \in \mathbb{N} \).

Note that to remain consistent with the previously defined \( \mathcal{F} \mathcal{C} \) and \( \mathcal{F} \mathcal{C}^k \) classes, we should write \( \mathcal{F} \mathcal{C}^{\infty} \) instead of \( \mathcal{F} \mathcal{C}^{\ast, \infty} \). However, the former is easily confused with \( \mathcal{F} \mathcal{C}^{\infty} \) and therefore replaced by the latter. There exists a clear relation between \( \mathcal{A} \mathcal{C}^{\ast, \infty} \) and the standard class of unbounded parallelism denoted by POLYLOGSPACE; see [Johnson, 1990; Parberry, 1987] for a definition of POLYLOGSPACE. This relation can be expressed by

\[
\mathcal{A} \mathcal{C} \subseteq \mathcal{A} \mathcal{C}^{\ast, \infty} \subseteq \text{POLYLOGSPACE}.
\]
4.3 Complexity classes

The relation between $\text{POLYLOGSPACE}$ and $\mathcal{P}$ is still unclear. It has been conjectured that $\text{POLYLOGSPACE} \not\subseteq \mathcal{P}$ and $\mathcal{P} \not\subseteq \text{POLYLOGSPACE}$, but all that is known is that $\mathcal{P}$ is not equal to $\text{POLYLOGSPACE}$; see Johnson [1990] and Parberry [1987].

Complexity of integer versions of case studies

We end this section by studying the complexity of the integer versions of the five problems that we introduced in Chapter 3.

Theorem 4.3.5.

\[
\begin{align*}
\text{INTEGER SORTING} & \in \mathcal{T}C^0, \\
\text{INTEGER MINIMUM COST SPANNING TREE} & \in \mathcal{T}C^3, \\
\text{INTEGER SHORTEST NETWORK PATH} & \in \mathcal{T}C^1, \\
\text{INTEGER SHORTEST NETWORK ROUTE} & \in \mathcal{T}C^1, \\
\text{INTEGER DISCRETE DYNAMIC LOTSIZING} & \in \mathcal{T}C^3.
\end{align*}
\]

Proof. That INTEGER SORTING is in $\mathcal{T}C^0$ follows directly from the results of Chandra, Stockmeyer & Vishkin [1984]; see also [Preparata, 1978; Kronsjö, 1985; Kindervater, 1989]. Furthermore, it is well known that the INTEGER ALL-PAIRS SHORTEST PATH problem has the same circuit complexity as the computation of the $N$-th power of an $N \times N$ matrix, where $N$ is the number of vertices; see for instance Karp & Ramachandran [1990] or Kindervater [1989]. Maggs & Plotkin [1988] have shown that the same holds for INTEGER MINIMUM COST SPANNING TREE. Since MATRIX MULTIPLICATION is in $\mathcal{T}C^0$, it follows easily that MATRIX POWERING is in $\mathcal{T}C^1$. Now the result follows because INTEGER SHORTEST NETWORK PATH, INTEGER SHORTEST NETWORK ROUTE, and INTEGER DISCRETE DYNAMIC LOTSIZING are all special cases of INTEGER ALL-PAIRS SHORTEST PATH. \qed

It is well-possible that in addition to INTEGER SORTING also one of the other considered problems is in $\mathcal{T}C^0$, but we do not know of any proof. Furthermore, we note that INTEGER MINIMUM COST SPANNING TREE and INTEGER SHORTEST NETWORK ROUTE are projection reducible from CONNECTIVITY, which implies that these problems are not in $\mathcal{T}C^2$, see Chandra, Stockmeyer & Vishkin [1984]. Finally, we note that there is a direct correspondence between the above $\mathcal{T}C^1$ solutions for INTEGER MINIMUM COST SPANNING TREE and INTEGER ALL PAIRS SHORTEST PATH, and the $\mathcal{AC}^1$ solution of computing the transitive closure of a matrix. The latter follows from the fact that BOOLEAN MATRIX MULTIPLICATION is in $\mathcal{AC}^0$; see Karp & Ramachandran [1990].

4.3.3 MLP-complexity

In this section we introduce a number of complexity classes for MLPs, similarly as we did for Boolean circuits in the previous section. The essential difference is that the node function set is now allowed to be defined for the whole set of real-valued numbers $\mathbb{R}$, instead of being restricted to $\mathbb{B}$ as in the case of circuits. We consider node function sets corresponding to $\Sigma$-MLPs, where $\Sigma$ is an arbitrary set of functions from $\mathbb{R}$ to $\mathbb{R}$; see also Section 4.2.2.

We start this section with a formulation of some results of Maass, Schnitger &
Sontag [1991], who consider $\Sigma$-MLPs for solving binary problems, because of its overlap with the previous section. To this end Definition 4.3.7, that defined a general UCM complexity class, is extended with one more restriction. In the second part of this section we introduce some complexity classes for MLPs with real-valued inputs, discuss their relation with the complexity classes for Boolean circuits, and formulate the results that we present in Chapter 7 in terms of the introduced complexity classes for MLPs.

**Robust MLPs**

Maass, Schnitger & Sontag consider $\Sigma$-MLPs that solve a given problem with some robustness, i.e., they require that the computation at the output node is insensitive to infinitesimal changes in the value of any node in the preceding layer. For this reason they define the notion of computing with a given positive separation.

**Definition 4.3.14** (Maass, Schnitger & Sontag [1991]).

A $\Sigma$-MLP $M$ with one output node, computes a Boolean function $f : \mathbb{B}^N \rightarrow \mathbb{B}$ with separation $\varepsilon > 0$, if there is some $t \in \mathbb{R}$ such that for any input $x \in \mathbb{B}^N$ the output node outputs a value which is at least $t + \varepsilon$, if $f(x) = 1$, and at most $t - \varepsilon$, otherwise.

The definition of computing with separation can be easily extended to $\Sigma$-MLPs with more than one output node.

Obviously, computing with a positive separation is not sufficient to achieve robust computation, as any $\theta$-output computes with separation $\frac{\varepsilon}{2}$. Maass, Schnitger & Sontag therefore consider response functions $\gamma \in \Sigma$ that are Lipschitz bounded. The most used examples of such smooth functions are functions that are bounded, differentiable, and have a bounded first derivative. An example of such a function is the sigmoid response function $\sigma$. A nice feature about $\Sigma$-MLPs that use Lipschitz bounded response functions and compute with positive separation is the following. If the considered $\Sigma$-MLP has $N$ inputs, polynomial size, polynomially bounded weights, and computes a given binary function with separation larger than or equal to $[\text{POLY}(N)]^{-1}$, then all the nodes have tolerance larger than or equal to $[\text{POLY}(N)]^{-1}$, i.e., their values may fluctuate at least $[\text{POLY}(N)]^{-1}$, without affecting the output of the MLP.

To formulate the results of Maass, Schnitger & Sontag we introduce the following set of complexity classes.

**Definition 4.3.15.** Let $\Sigma$ be a set of Lipschitz bounded functions from $\mathbb{R}$ to $\mathbb{R}$, let $\mathcal{F} = \mathcal{F}_\Sigma \subset \cup_{n \in \mathbb{N}}(\mathbb{R}^n \to \mathbb{R})$, be the corresponding node function set, and let $D, S, F_1$, and $A$ be functions from $\mathbb{N}$ to $\mathbb{R}^+$. Then the complexity class denoted by

$$\text{SEP}[D \mid S \mid \mathcal{F} \mid \mathbb{B} \mid \mathcal{F}_1 \mid A],$$

contains all problems $\{f_N : \mathbb{B}^N \to \mathbb{B}^{K(N)}\}_{N \in \mathbb{N}}$, such that for every $N \in \mathbb{N}$, $f_N$ can be computed with separation $\geq 1/A(N)$ by a $\Sigma$-MLP with node function set $\mathcal{F}$, $N$ inputs, $K(N)$ outputs, depth $\leq D(N)$, size $\leq S(N)$, and maximum fan-in $\leq F_1(N)$.

If a binary function is computed with separation by a $\Sigma$-MLP, strictly speaking it is not computed by that $\Sigma$-MLP. This can be remedied by adding one extra layer with $\theta$-nodes. Thus we have

$$\text{SEP}[D \mid S \mid \mathcal{F} \mid \mathbb{B} \mid \mathcal{F}_1 \mid A] \subseteq \text{UCM}[D + 1 \mid 2S \mid \mathcal{F} \cup \theta \mid \text{PTWL} \mid \mathbb{B} \mid \mathcal{F}_1].$$
If the response functions in $\Sigma$ are monotone this extra layer is superfluous.

The following complexity classes are defined similarly as those defined for Boolean circuits.

**Definition 4.3.16.** Let $\mathcal{F}$ be a node function set corresponding to a MLP. The complexity classes $\mathcal{F}_S$, $\mathcal{F}_S^k$, $\mathcal{F}_S^0$, $\mathcal{F}_S^{0,0}$, and $\mathcal{F}_S^{0,0}$, are defined by

\[
\begin{align*}
\mathcal{F}_S &= \text{SEP}[\text{POLYLOG} | \mathcal{F} | \text{R} | \text{R} | \text{R} | \text{POLY}], \\
\mathcal{F}_S^k &= \text{SEP}[\text{POLYLOG} | \mathcal{F} | \text{R} | \text{R} | \text{POLY}], \\
\mathcal{F}_S^0 &= \text{SEP}[d | \text{POLY} | \mathcal{F} | \text{R} | \text{R} | \text{POLY}], \\
\mathcal{F}_S^{0,0} &= \text{SEP}[d | \text{POLY} | \mathcal{F} | \text{R} | \text{R} | \text{POLY}], \\
\mathcal{F}_S^{0,0} &= \text{SEP}[d | \text{POLY} | \mathcal{F} | \text{R} | \text{R} | \text{POLY}],
\end{align*}
\]

for all $k, l \geq 0$ and $d, s \in \mathbb{N}$.

Firstly, we consider a result of Maass, Schnitger & Sontag for $\gamma$-MLPs of fixed size and, consequently, fixed depth. Note that it separates two complexity classes similarly as discussed for Boolean circuits.

**Proposition 4.3.2 (Maass, Schnitger & Sontag [1991]).** Let $\gamma : \mathbb{R} \to \mathbb{R}$ be a Lipschitz bounded, sigmoidal response function satisfying the conditions of Maass, Schnitger & Sontag presented in Table 2.1. Then for all $d \in \mathbb{N}$

\[
\mathcal{R}WTC_2^{d,0} \subseteq \gamma \cdot \mathcal{R}WLS_2^{d,0}
\]

and

\[
\mathcal{R}WTC_2^{d,0} \subset \gamma \cdot \mathcal{R}WLS_2^{d,0}.
\]

The strict inclusion for $\gamma$-MLPs of depth 2 is obtained from

\[
\{\text{BOR}([\text{Maj}^N(x), \text{Maj}^N(y)]) | x, y \in \gamma \cdot \mathcal{R}WLS_2^{d,0} \setminus \mathcal{R}WTC_2^{d,0}\},
\]

which Maass, Schnitger & Sontag prove using an extension of the techniques developed in [Hajnal, Maass, Pudlak, Szegedy & Turan, 1987]. Some interesting related papers are one by Toda, Funahashi & Usui [1991], who discuss the possibility that multiplication is in $\gamma \cdot \mathcal{R}WLS_2^{d,0}$, for $\gamma$ that satisfy the conditions of Maass, Schnitger & Sontag presented in Table 2.1, and one by Stork & Allen [1992], who prove that

\[
\{\text{PAR}^N \} | x \in \gamma \cdot \mathcal{P}WLS_2^{d,0},
\]

where $\gamma(x) = x - \frac{1}{2} \cos(\pi x)$.

Secondly, we discuss some results of Maass, Schnitger & Sontag for constant depth $\gamma$-MLPs, in case that $\gamma$ satisfies an additional constraint.

**Proposition 4.3.3 (Maass, Schnitger & Sontag [1991]).** Let $\gamma$ be a non-linear $[0, 1]$-function that is non-decreasing, Lipschitz bounded, and satisfies

\[
\forall k \in \mathbb{N} \exists ! \in \mathbb{N} \forall x > 0 : \max(\gamma(-x^k), 1 - \gamma(x^k)) \leq x^{-(k+1)}.
\]

Then for all $d \in \mathbb{N}$

\[
\mathcal{T}C_2^d = \gamma \cdot \mathcal{P}WLS_2^d
\]

and

\[
\mathcal{T}C_2^d = \gamma \cdot \mathcal{R}WLS_2^d.
\]
Similar but weaker results have been obtained in [Obadovic & Parberry, 1990a; Obadovic & Parberry, 1990b] and in [Shaw-Taylor, Anthony & Kern, 1992]. Related is also the work on reliable threshold circuits; see Parberry [1990], Berman, Parberry & Schnitger [1991], and Hajnal, Maass, Pudlak, Szegedy & Turan [1987].

**Obtained results**

So far we have studied binary problems only. In the remainder of this thesis we consider problems of the form \( \{f_N : \mathbb{R}^N \to \mathbb{B}^{k(N)}\}_{N \in \mathbb{N}} \). The corresponding complexity classes are given by the following definition.

**Definition 4.3.17.** Let \( \mathcal{F} \) be a node function set corresponding to a \( \Sigma \)-MLP. The complexity classes \( \mathcal{F} \mathcal{N}, \mathcal{F} \mathcal{N}^\infty, \mathcal{F} \mathcal{N}^k, \mathcal{F} \mathcal{N}^0, \mathcal{F} \mathcal{N}^{k,\infty}, \mathcal{F} \mathcal{N}^{0,\infty}, \mathcal{F} \mathcal{N}^{0,k}, \mathcal{F} \mathcal{N}^{0,0}, \mathcal{F} \mathcal{N}^{0}\), and \( \mathcal{F} \mathcal{N}^{0,s} \), are defined by

\[
\begin{align*}
\mathcal{F} \mathcal{N} &= \text{UCM}(\text{POLYLOG} | \text{POLY} | \mathcal{F} | \mathbb{R} | \mathbb{B} | \mathcal{F}), \\
\mathcal{F} \mathcal{N}^\infty &= \text{UCM}(\text{POLYLOG} | \infty | \mathcal{F} | \mathbb{R} | \mathbb{B} | \mathcal{F}), \\
\mathcal{F} \mathcal{N}^k &= \text{UCM}(|\text{POLY} | \mathcal{F} | \mathbb{R} | \mathbb{B} | \mathcal{F}), \\
\mathcal{F} \mathcal{N}^\infty &= \text{UCM}(|\text{POLY} | \mathcal{F} | \mathbb{R} | \mathbb{B} | \mathcal{F}), \\
\mathcal{F} \mathcal{N}^{k,\infty} &= \text{UCM}(\text{LOG} | \infty | \mathcal{F} | \mathbb{R} | \mathbb{B} | \mathcal{F}), \\
\mathcal{F} \mathcal{N}^{0,\infty} &= \text{UCM}(\text{LOG} | \infty | \mathcal{F} | \mathbb{R} | \mathbb{B} | \mathcal{F}), \\
\mathcal{F} \mathcal{N}^{0,k} &= \text{UCM}(\text{LOG} | \infty | \mathcal{F} | \mathbb{R} | \mathbb{B} | \mathcal{F}), \\
\mathcal{F} \mathcal{N}^{0,0} &= \text{UCM}(\text{LOG} | \infty | \mathcal{F} | \mathbb{R} | \mathbb{B} | \mathcal{F}), \\
\mathcal{F} \mathcal{N}^{0,s} &= \text{UCM}(\text{LOG} | \infty | \mathcal{F} | \mathbb{R} | \mathbb{B} | \mathcal{F}),
\end{align*}
\]

for all \( k, l \geq 0 \) and \( d, s \in \mathbb{N} \).

We can use the results on circuit complexity to obtain results on MLP-complexity, by restricting the set of inputs of the considered MLP to binary values. Let \( \mathcal{F} \) be a set of functions, then we use \( \mathcal{F}_b \) to denote the set of functions that is obtained by restricting the functions in \( \mathcal{F} \) to binary inputs. Furthermore, this operator is extended in the obvious elementwise way to supersets of functions. Using the definition of UCM(\( \cdot \)), we directly find the following result.

**Corollary 4.3.3.**

\[
\text{UCM}(D | S | \mathcal{F} | \mathbb{R} | \mathbb{B} | \mathcal{F}^b) \subseteq \text{UCM}(D | S | \mathcal{F}_b | \mathbb{B} | \mathcal{F}^b).
\]

Applying this result to one of the node function sets for \( \theta \)-MLPs, we derive

\[
\text{UCM}(D | S | \theta \cdot \text{PIWLC} | \mathbb{R} | \mathbb{B} | \mathcal{F}^b) \subseteq \text{UCM}(D | S | \text{PIWTC} | \mathbb{B} | \mathcal{F}^b),
\]

Choosing \( D = \text{LOG}^k \), for some \( k \geq 0 \), and \( S = \text{POLY} \) in (4.4), yields for instance

\[
\text{\theta-PIWLC} \subseteq \text{PIWTC} = \mathcal{T},
\]

and similarly for other node function sets and other complexity classes. The result (4.5) gives a necessary condition for a problem to be solvable by a family of \( \theta \)-MLPs, with a polynomial number of nodes, a depth bounded by \( O(\text{LOG}^k N) \), and its weights bounded by a polynomial in \( N \), where \( N \) is the number of inputs, namely that the binary variant of the considered problem is in \( \mathcal{T} \). This is the reason for the selection of the five test problems \text{REAL SORTING}, \text{REAL MINIMUM COST SPANNING TREE}, \text{REAL SHORTEST NETWORK PATH}, \text{REAL SHORTEST NETWORK ROUTE}, and \text{REAL DISCRETE DYNAMIC LOTSIZING}, as their integer versions are all in \( \mathcal{T} \); see Theorem 4.3.5.
4.4 Concluding remarks

In the following theorem we present our results on the MLP-complexity of the above five problems, using the complexity classes introduced in this chapter. These results are derived in Chapter 7 using the theory that is developed in Chapters 5 and 6.

Theorem 4.3.6. The problems REAL SORTING, REAL MINIMUM COST SPANNING TREE, REAL SHORTEST NETWORK PATH, REAL SHORTEST NETWORK ROUTE, and REAL DISCRETE DYNAMIC LOTSIZING, can be solved by a $\theta$-SLP with an exponential number of nodes, i.e., they are all member of

$$\text{UCM}\left[3|\exp|\theta\cdot\Pi^2\text{W}^L|R|B|\infty\right] \subseteq \theta\cdot\text{RWLN}^0_{3\infty}.$$

Furthermore, REAL SORTING and REAL MINIMUM COST SPANNING TREE can be solved by a $\theta$-MLP with a polynomial number of nodes, whereas REAL SHORTEST NETWORK PATH, REAL SHORTEST NETWORK ROUTE, and REAL DISCRETE DYNAMIC LOTSIZING, cannot be solved by a $\theta$-MLP with a polynomial number of nodes. More specifically, we have

$$\begin{align*}
\text{REAL SORTING} & \in \theta\cdot\Pi^2\text{W}^L 0^3, \\
\text{REAL MINIMUM COST SPANNING TREE} & \in \theta\cdot\Pi^2\text{W}^L 1^3, \\
\text{REAL SHORTEST NETWORK PATH} & \notin \theta\cdot\text{RWLN}^\infty, \\
\text{REAL SHORTEST NETWORK ROUTE} & \notin \theta\cdot\text{RWLN}^\infty, \\
\text{REAL DISCRETE DYNAMIC LOTSIZING} & \notin \theta\cdot\text{RWLN}^\infty, \\
\text{REAL DISCRETE DYNAMIC LOTSIZING cannot be solved by a } \theta\text{-2LP, whichever the number of nodes used, i.e.,} \\
\text{REAL SORTING} & \notin \theta\cdot\text{RWLN}^\infty, \\
\text{REAL DISCRETE DYNAMIC LOTSIZING} & \notin \theta\cdot\text{RWLN}^\infty 0^3.
\end{align*}$$

Proof. See Chapter 7 of this thesis. \hfill \Box

The results given in Theorem 4.3.6 for REAL SORTING and REAL DISCRETE DYNAMIC LOTSIZING imply the following separation result.

Corollary 4.3.4.

$$\theta\cdot\text{RWLN}^\infty 0^2 \subset \theta\cdot\text{RWLN}^\infty 3^\infty.$$ 

This ends our study of the complexity of MLPs based on their relation with Boolean circuits. In Chapter 5 we study the complexity of MLPs by examining their capabilities for solving combinatorial classification problems, based on a geometrical approach. The results are used in Chapter 6 to obtain general results on the complexity of MLPs for solving combinatorial optimization problems, which is applied in Chapter 7 to the five problems mentioned in Theorem 4.3.6.

4.4 Concluding remarks

In this chapter we have considered the complexity of MLPs by relating them to Boolean circuits, a parallel computational model which complexity is thoroughly studied. This is done by introducing a general framework for uni-directional computational models (UCMs) that captures both Boolean circuits and MLPs as special cases. Generalizing the standard approach for Boolean circuits, we introduce the concept of complexity classes for UCMs; see also Johnson [1990], Chandra, Stockmeyer & Vishkin [1984], and Wegener [1987]. Every complexity class contains all
problems that can be solved by a restricted set of UCMs. The restrictions that we consider, apply to the number of layers, the number of nodes, the type of node functions used, the set of input values, the set of output values, the fan-in of the nodes, and sometimes the accuracy of the computation of the nodes: the last restriction based on the work of Maass, Schnitger & Sontag [1991]. The well-known complexity classes for Boolean circuits can be recovered by restricting the input and output values to 0 and 1, and choosing an appropriate node function set consisting of Boolean functions, of the many that are available from literature. We have presented some of the most important old results on the complexity of Boolean circuits, that are expressed in terms of the introduced complexity classes. One set of new results is obtained for Boolean circuits that have their fan-in bounded by a polynomial in the logarithm of the number of inputs of the considered circuit. These results are inspired by the results obtained by Shawe-Taylor, Anthony & Kern [1992]. One of their results is a weaker version of

$$\mathcal{NC}^k \subseteq \text{UCM}[\log^k \text{ poly}] \text{ WTT} \{B | B| \log \} \subseteq \mathcal{AC}^k.$$  \hspace{1cm} (4.6)

However, it is fairly trivial to prove (4.6) since any Boolean function with $O(\log N)$ inputs can be computed by an $\mathcal{AC}$-circuit of polynomial size and constant depth, namely by using its disjunctive normal form (DNF); see for instance Wegener [1987]. The proof can be completed using an argument similar as given in Lemma 4.3.1.

In our discussion of the results for Boolean circuits, emphasis is given to the different types of threshold circuits, because these models are the most closely related to MLPs, in particular to $\theta$-MLPs. Because we are interested in MLPs with constant depth or less, i.e., fixed depth, we mainly look at circuits with constant depth or less, and circuits with their depth bounded by a polynomial in the logarithm of the number of inputs. Comparison of the relative power of the different types of circuits showed that the unweighted threshold circuits are among the most powerful circuits. Furthermore, adding weights does not increase the capabilities of threshold circuits, whatever the size of the weights allowed, possibly except for the case that real-valued weights are used.

The next step was to compare threshold circuits to MLPs. This can be done by restricting the inputs of the considered MLPs to $B$. Obviously, this transforms $\theta$-MLPs into weighted threshold circuits, which gives a direct view of the relative power of $\theta$-MLPs. Maass, Schnitger & Sontag [1991] go one step further and compare threshold circuits to $\gamma$-MLPs, where $\gamma$ is a sigmoid function that must satisfy certain conditions, using a convention how to interpret the output value; see also Table 2.1. Note that the standard sigmoidal $\sigma$ satisfies their conditions. They show some results that indicate that $\gamma$-MLPs are equally powerful as threshold circuits, except for $\gamma$-MLPs of fixed size and fixed depth, that outperform their threshold counterparts. This last result belongs to the class of separation results, which show that a certain problem can be solved by one computational model and not by another model. Most separation results concern Boolean circuits or MLPs of fixed depth. Exceptions are the groundbreaking results based on the work of Sipser and Razborov; see Theorem 4.3.3. In many of the separation results an important role is played by PARITY, which goes back to the well-known result of Minsky & Papert [1969]; see also Chapter 2. This also holds for our own separation result, which shows that REAL SORTING and REAL DISCRETE DYNAMIC LOTSIZING can be solved by a $\theta$-3LP,
but not by a $\theta$-2LP; see Chapter 7. The underlying theory is proved in Chapter 5, and applied to combinatorial optimization problems in general in Chapter 6.

We have introduced the MLP-model as a computational model of a network of interconnected nodes, in which at every node the weighted sum of an arbitrary number of real-valued inputs can be computed in constant time. One may question if this is a realistic model. Vergis, Steiglitz & Dickinson [1986] have argued that the model is not feasible using present day techniques, that are based on systems which can be described by well-behaved ordinary differential equations. To this end they show that any analog computer based on such a system can be simulated efficiently by a digital computer, in the sense that the time required by the digital computer to simulate the analog computer is bounded by a polynomial function of the resources used by the analog computer. For a digital computer the above assumption is certainly not doable, even for integer-valued inputs and weights; see also Wegener [1987] and Chapter 4. However, Vergis, Steiglitz & Dickinson [1986] leave open the possibility that the assumption can be met by an analog computer based on some strongly nonlinear behavior, perhaps arising from quantum mechanical mechanisms. This is corroborated by the many promising experiments with optical implementations of multi-layered perceptrons, which indicate that there may exist an optical implementation of the introduced model; see also [Hecht-Nielsen, 1990; Hecht-Nielsen, 1987].
Chapter 5

The MLP Complexity of CCPs

5.1 Introduction

In this chapter we examine some complexity aspects of classification problems, with respect to the number of layers and the number of nodes required by multi-layered perceptrons using the hard-limiting response function, that solve these problems; see Chapter 2 for an introduction and a definition of these so-called $\theta$-MLPs. This chapter is an intermediate chapter between Chapter 4, that considers the complexity of problems in general, and Chapter 6, that deals with the class of combinatorial optimization problems which were introduced in Chapter 3.

Before we can explain in greater detail the topics discussed in this section we have to define our notion of a combinatorial classification problem (CCP).

Definition 5.1.1. A combinatorial classification problem is represented by a $\mathcal{S}$-tuple $(\Omega, L, \Gamma)$, where, (i) for some $N \in \mathbb{N}$, $\Omega \subseteq \mathbb{R}^N$ denotes a set of objects that must be classified, (ii) for some $K \in \mathbb{N}$, $L \subseteq \mathcal{B}_K$ denotes a set of labels, and (iii) $\Gamma$ denotes a collection of subsets of $\Omega$, one for each label, expressed by $\Gamma = \{\Omega_l \subseteq \Omega | l \in L\}$. For a given object $x \in \Omega$, the problem is to find a label $l \in L$ such that $x \in \Omega_l$. It is assumed that $\bigcup_{l \in L} \Omega_l = \Omega$, which guarantees that for each $x \in \Omega$ this problem can be solved.

The above definition of a CCP does not confirm to the general definition of a problem given in Chapter 4. This is firstly, because $N$ does not explicitly occur as a running variable. Following the definition of a problem given in Chapter 4, we should index $\Omega$, $L$, and $\Gamma$ with $N$ and let it run over all positive integers; see Definition 4.3.1. However, in this chapter we omit any explicit indexing with $N$ because it complicates the notation. Instead we use $N \in \mathbb{N}$ as an implicit index for most of the presented results.

A second difficulty is that we might need to add constraints to guarantee that for every $x \in \Omega$ there is a unique solution corresponding to a unique label $l \in L$ for which $x \in \Omega_l$. This is the subject of Section 5.7, which considers general CCPs. In the vast part of this chapter we consider a special class of CCPs that have a unique solution by definition; see below for a definition of this class.
The objective of this chapter is to examine whether $\theta$-MLPs, i.e., multi-layered perceptrons that use the hard-limiting response function, can be used to solve CCPs, and to examine the complexity of $\theta$-MLPs that solve CCPs. More specifically, this chapter considers the following two questions. Given a CCP, what is the minimal number of layers and what is the minimal number of first-layer nodes required by any $\theta$-MLP that solves this CCP. Before we can consider these questions, we have to formulate what we mean by solving a CCP with a $\theta$-MLP. Note that this can be done, although there can in general be more than one solution to a CCP.

**Definition 5.1.2.** A CCP given by $(\Omega, L, \Gamma)$ with $\Omega \subseteq \mathbb{R}^N$ and $L \subseteq \mathbb{B}^N$, for some $N, K \in \mathbb{N}$, is solved by a $\theta$-MLP with $N$ inputs and $K$ outputs represented by the function $f : \mathbb{R}^N \rightarrow \mathbb{B}^K$, if $f(x) \in L$ and $x \in \Omega_f(x)$, for all $x \in \Omega$.

In this chapter we only consider MLPs that use the hard-limiting response function denoted by $\theta$. Therefore, we mostly use the abbreviation MLP instead of $\theta$-MLP in the remainder of this chapter.

As mentioned above, we consider a special kind of CCPs in the major part of this chapter. In these problems one has to classify two disjoint subsets, i.e., $L = \mathbb{B}$, corresponding to the case that $K = 1$ and $\Gamma = \{\Omega_0, \Omega_1\}$, with $\Omega_0 \cap \Omega_1 = \emptyset$. We call these problems binary classification problems (BCPs). Usually, we consider BCPs that satisfy $\Omega = \mathbb{R}^N$. We call these type of problems complete classification problems, in contrast to the incomplete classification problems where $\Omega \subset \mathbb{R}^N$. Complete classification problems can be described by one subset of $\mathbb{R}^N$ only, which we take to be $\Omega_1$, the subset corresponding to the vectors that have to be mapped to the label 1. Thus the BCP defined by $(\mathbb{R}^N, \mathbb{B}, \{V, V^*\})$, for some subset $V \subseteq \mathbb{R}^N$, will be denoted by $V$ only. Furthermore, instead of saying that an MLP solves $(\mathbb{R}^N, \mathbb{B}, \{V, V^*\})$, we say that it classifies $V$. Consequently, the classification of a subset with an MLP can be defined as follows.

**Definition 5.1.3.** A subset $V \subseteq \mathbb{R}^N$ is classified with an MLP represented by $f : \mathbb{R}^N \rightarrow \mathbb{B}$, if

$$f(x) = \begin{cases} 1 & \text{if } x \in V, \\ 0 & \text{if } x \notin V. \end{cases}$$

Thus $V$ is classified with an MLP represented by the function $f$, if $f$ is equal to the characteristic function of $V$, i.e., $f = \chi_V$.

The questions studied in the main part of this chapter can now be formulated as follows. Given a subset $V \subseteq \mathbb{R}^N$, what is the minimal number of layers and what is the minimal number of first-layer nodes required by any MLP that classifies $V$. For issues concerning the question about the minimal number of layers, we introduce the collection of subsets that can be classified with an MLP that has $m$ layers, for some $m \in \mathbb{N}$.

**Definition 5.1.4.** Let $N \in \mathbb{N}$ and $m \in \mathbb{N}$. Then the collection of subsets of $\mathbb{R}^N$ that can be classified with an MLP with $m$ layers is denoted by $C_{m,N}$.

Usually, we abbreviate $C_{m,N}$ to $C_m$, except for those cases where $N$ is explicitly needed.
5.1 Introduction

An alternative but equivalent formulation of the above questions is the following. Given an integer \( N \) and a function from \( \mathbb{R}^N \) to \( \mathcal{B} \), what is the minimal number of layers and what is the minimal number of first-layer nodes required by any MLP that computes this function. Answers to these questions can be used to obtain bounds for some of the complexity classes introduced in Chapter 4. However, this is not the main objective of this chapter. Instead, we focus on results that are useful for Chapter 6, where we consider the subclass of classification problems that arise from combinatorial optimization.

In Chapter 2 we introduced \( \theta_{m,N,K} \) to denote the set of functions that can be computed by an MLP with \( m \) layers, \( N \) inputs, \( K \) outputs, and that uses the hard-limiting response function denoted by \( \theta \). For convenience we repeat this definition.

**Definition 5.1.5.** Let \( A_N \) denote the set of all affine functions from \( \mathbb{R}^N \) to \( \mathbb{R} \) defined by

\[
A_N = \{ f : \mathbb{R}^N \to \mathbb{R} \mid f(x) = a \cdot x + b, x \in \mathbb{R}^N, a \in \mathbb{R}^N, b \in \mathbb{R} \}.
\]

Then the set \( R_{m,N,K} \subseteq \mathbb{R}^N \to \mathcal{B}^K \), of all functions that can be computed by an MLP with \( m \) layers, \( N \) inputs, and \( K \) outputs, is defined iteratively by

\[
R_{1,N,K} = \{ (\theta \circ f_1, \ldots, \theta \circ f_K) \mid f_i \in A_N, i = 1, \ldots, K \},
\]

and, for \( m > 1 \),

\[
R_{m,N,K} = \{ g \circ h \mid g \in R_{l,N,L}, h \in R_{m-1,N,L}, L \in \mathbb{N} \}.
\]

In this chapter we mostly consider \( R_{m,N,1} \), which is usually abbreviated to \( R_m \), if \( N \) is known, and to \( R_{m,N} \), if \( N \) is explicitly needed. The results in this chapter are characterizations of \( R_m \). However, as already indicated, we usually give the characterizations using the classification terminology that we introduced above. Results formulated in terms of \( C_m \) can be easily translated into results for \( R_m \), since \( R_{m,N} = \{ X_V \mid V \in C_{m,N} \} \). To be able to formulate this relation reversely, we introduce the following notation for the inverse of the characteristic function.

**Definition 5.1.6.** Let \( f : \mathbb{R}^N \to \mathcal{B} \). Then the set \( \mathcal{J}(f) \subseteq \mathbb{R}^N \) is defined by

\[
\mathcal{J}(f) = \{ x \in \mathbb{R}^N \mid f(x) = 1 \}.
\]

It is obvious that \( V \subseteq \mathbb{R}^N \) can be classified with an MLP, if and only if \( V = \mathcal{J}(f) \) for some \( f \in R_{m,N} \), i.e., \( C_{m,N} = \{ \mathcal{J}(f) \mid f \in R_{m,N} \} \). We prefer to use \( \mathcal{J} \) in formulations concerning \( R_{m,N} \) or \( C_{m,N} \), and not \( X \). In the remainder of this chapter we present characterizations of \( C_m \) in the following order.

Section 5.2 introduces the remaining basic elements of the notations used in this chapter, which mainly concerns notations for halfspaces, polyhedra and related structures. Furthermore, it introduces some preliminary results.

In Section 5.3 we discuss \( C_m \), for \( m \neq 2 \). After a discussion of the trivial \( C_1 \), we derive a necessary and sufficient condition for a subset to be classifiable with an MLP, for \( m \geq 3 \). Since this condition is equal for all \( m \geq 3 \), this shows that the minimal number of layers required to classify an arbitrary classifiable subset is 3 or less. This section also contains a construction of a 3LP that classifies a given classifiable subset, and it contains some preliminary results used in the subsequent sections.
Section 5.4 discusses the minimal number of first-layer nodes that is required for the classification of a given subset. There are two reasons for putting this section between the section on $C_m$, $m \neq 2$, and the section on $C_2$.

- In Section 5.4 the number of layers is arbitrary, as in the main part of Section 5.3. Furthermore, the derived results may be viewed as necessary conditions for classification with MLPs that have a limited number of nodes.
- Some of the notions and results given in Section 5.4 are used in Section 5.5.

The results of Section 5.4 present a lower bound on the minimal number of first-layer nodes, which is shown to be tight for the special case of subsets that correspond to full-dimensional polyhedrons.

In Section 5.5 we extensively discuss $C_2$. Firstly, we derive a number of necessary conditions for a subset to be classifiable with a 2LP. Secondly, we derive a set of sufficient conditions for a subset to be classifiable with a 2LP. Although the necessary conditions and sufficient conditions can be shown to coincide for a special class of subsets, they do not match in general. The necessary conditions are of a local nature, whereas the sufficient conditions are of a global nature, based on a decomposition of the considered subset into a number of basic subsets. The basic subsets and the decomposition are considered in detail in two separate subsections.

Finally, this chapter ends in Section 5.7 with a brief discussion of the implications for arbitrary combinatorial classification problems.

5.2 Preliminaries

Before we can start the discussion we need some additional notations, which forms the subject of this section. Furthermore, the section contains some basic theory on polyhedrons and some other preliminary results.

We consider subsets $V \subseteq \mathbb{R}^N$ for some fixed $N \in \mathbb{N}$. Below we define the complement, interior and the closure of $V$, the latter two as induced by the standard, Euclidean topology. We start by using the Euclidean distance to define the notion of an open ball.

**Definition 5.2.1.** The open ball $B(x_0, \delta)$ with center $x_0 \in \mathbb{R}^N$ and radius $\delta > 0$ is defined by
\[
B(x_0, \delta) = \{x \in \mathbb{R}^N \mid \|x - x_0\|_2 < \delta\}.
\]

We use $B$ to denote any $B(x_0, \delta)$ for some $x_0 \in \mathbb{R}^N$ and $\delta > 0$.

**Definition 5.2.2.** Let $V \subseteq \mathbb{R}^N$. The complement of $V$ is denoted as $V^\complement$ and is given by
\[
V^\complement = \mathbb{R}^N \setminus V.
\]

The interior of $V$ is denoted as $V^\circ$ and is given by
\[
V^\circ = \bigcup \{B \mid B \text{ is an open ball and } B \subseteq V\}.
\]

The closure of $V$ is denoted as $\overline{V}$ and is given by
\[
\overline{V} = \bigcap \{B^\complement \mid B \text{ is an open ball and } B \subseteq V^\complement\}.
\]
More informally, $V^\circ$ is everything of $V$ except its bounds, and $\overline{V}$ is everything of $V$ augmented with its bounds. One easily verifies that $\overline{V} = ((V^\circ)^*)^*$.

Some further notations are the following. We sometimes use $V^{(+1)}$ and $V^{(-1)}$ to denote $V$ and $V^*$, respectively. Finally, we sometimes use $\lambda V$, for some $\lambda \in \mathbb{R}$, to denote the set defined by

$$\lambda V = \{\lambda x \mid x \in V\}.$$  

As a special case we use $-V$ as an abbreviation for $(-1)V$. Note that if $B = B(x_0, \delta)$ is an open ball, then $\lambda B = B(\lambda x_0, |\lambda|\delta)$ is also an open ball.

As we have seen in Chapter 2, an MLP with real-valued inputs that uses the hard-limiting response function $\theta$, introduces a number of hyperplanes that subdivide the input space. Each hyperplane corresponds to an affine halfspace, for which we introduce some special notation. We distinguish between closed and open affine halfspaces $W$. For completeness we present below the complement $W^*$ of $W$, its interior $W^\circ$, its closure $\overline{W}$, as well as two additional sets $W^-$ and $W^-$. The set $W^\circ$ denotes the set of border points of $W$ and is in general given by $\overline{W} \cap (\overline{W}^*)$. Because $W^\circ$ is an affine halfspace, $W^\circ$ corresponds to the hyperplane that bounds $W$. The set $W^-$ denotes the set obtained by mirroring $W$ in one of its border points. It equals $(\overline{W})^*$, if $W$ is a closed affine halfspace, and equals $(\overline{W})^*$, if $W$ is an open affine halfspace.

**Definition 5.2.3.** Let $W = \{x \in \mathbb{R}^N \mid a \cdot x + b \geq 0\}$ for some $a \in \mathbb{R}^N \setminus \{0\}$ and $b \in \mathbb{R}$. Then the sets $W^\circ$, $W^*$, $\overline{W}$, $W^-$, and $W^-$ are given by

$$W^\circ = \{x \in \mathbb{R}^N \mid a \cdot x + b < 0\},$$

$$W^* = \{x \in \mathbb{R}^N \mid a \cdot x + b > 0\},$$

$$\overline{W} = \{x \in \mathbb{R}^N \mid a \cdot x + b \geq 0\},$$

$$W^- = \{x \in \mathbb{R}^N \mid a \cdot x + b \leq 0\},$$

$$W^- = \{x \in \mathbb{R}^N \mid a \cdot x + b = 0\},$$

respectively.

**Definition 5.2.4.** Let $W = \{x \in \mathbb{R}^N \mid a \cdot x + b > 0\}$ for some $a \in \mathbb{R}^N \setminus \{0\}$ and $b \in \mathbb{R}$. Then the sets $W^\circ$, $W^*$, $\overline{W}$, $W^-$, and $W^-$ are given by

$$W^\circ = \{x \in \mathbb{R}^N \mid a \cdot x + b \leq 0\},$$

$$W^* = \{x \in \mathbb{R}^N \mid a \cdot x + b > 0\},$$

$$\overline{W} = \{x \in \mathbb{R}^N \mid a \cdot x + b \geq 0\},$$

$$W^- = \{x \in \mathbb{R}^N \mid a \cdot x + b < 0\},$$

$$W^- = \{x \in \mathbb{R}^N \mid a \cdot x + b = 0\},$$

respectively.

Halfspaces can be used to form more complex sets by intersection and unification. We name some special collections in the two definitions below. First, we use closed halfspaces only.

**Definition 5.2.5.** The collection of closed affine halfspaces $H$, the collection of polyhedra $P$, and the collection of unions of polyhedra $U$, are defined by

$$H = \{V \subseteq \mathbb{R}^N \mid \exists a \in \mathbb{R}^N \setminus \{0\} \exists b \in \mathbb{R} : V = \{x \in \mathbb{R}^N \mid a \cdot x + b \geq 0\}\}.$$
\[ P = \{ V \subseteq \mathbb{R}^N \mid V = \cap_{i=1}^{L_i} W_i, W_i \in H, K \in N_0 \}, \]
\[ U = \{ V \subseteq \mathbb{R}^N \mid V = \cup_{i=1}^{L_i} V_i, V_i \in P, L \in N_0 \}, \]
respectively.

The subsets that are in the above collections \( H, P, \) and \( U, \) are closed, i.e., their border points belong to the subsets. In the following definition we define three more collections \( \tilde{H}, \tilde{P}, \) and \( \tilde{U} \) in which the subsets do not have to be closed.

**Definition 5.2.6.** The collection of open and closed affine halfspaces \( \tilde{H}, \) the collection of pseudo polyhedra \( \tilde{P}, \) and the collection of unions of pseudo polyhedra \( \tilde{U}, \) are defined by

\[
\tilde{H} = \{ V \subseteq \mathbb{R}^N \mid V \in H \lor V^* \in H \},
\]
\[
\tilde{P} = \{ V \subseteq \mathbb{R}^N \mid V = \cap_{i=1}^{L_i} W_i, W_i \in H, K \in N_0 \},
\]
\[
\tilde{U} = \{ V \subseteq \mathbb{R}^N \mid V = \cup_{i=1}^{L_i} V_i, V_i \in \tilde{P}, L \in N_0 \},
\]
respectively.

A polyhedron \( V \in P \) is the intersection of a finite collection of closed affine halfspaces. Therefore, all its bounds, usually called faces, belong to the set. A pseudo polyhedron \( V \in \tilde{P} \) is the intersection of a finite collection of closed or open affine halfspaces and can have faces belonging to the set and faces belonging to its complement \( V^* \). The collection \( \tilde{U} \) can be viewed as the collection of all subsets of \( \mathbb{R}^N \) that have a finite number of piece-wise linear bounds.

We give some elementary results for the introduced collections that are used in the remainder of this chapter.

**Lemma 5.2.1.** Let \( V \in \tilde{U} \). Then \( V^* \in \tilde{U} \).

**Proof.** One easily verifies that

\[
\left( \bigcup_{i=1}^{K} \bigcap_{j=1}^{L_i} W_{ij} \right)^* = \bigcap_{i=1}^{K} \bigcup_{j=1}^{L_i} W_{ij}^*,
\]

\[
= \bigcup_{j_1=1}^{L_1} \cdots \bigcup_{j_K=1}^{L_K} \bigcap_{i=1}^{K} W_{ij}^*,
\]

from which the result directly follows. \( \square \)

Next, we show that the closure of a pseudo polyhedron is a polyhedron. Recall that a pseudo polyhedron is a polyhedron with a number of "missing" faces, which implies that the result is intuitively clear. Note however that \( \cap_i W_i \neq \cap_i W_i \) in general.

**Lemma 5.2.2.** Let \( V \in \tilde{P} \). Then \( \overline{V} \in P \).

**Proof.** Let \( V \in \tilde{P} \). If \( V = \emptyset \), then \( \overline{V} = \emptyset \in P \).

Assume that \( V \neq \emptyset \). Since \( V \in \tilde{P} \) we have that \( V = \cap_{i=1}^{K} W_i \), for some \( K \in N \) and \( W_i \in H \). Let \( W_i = \{ x \mid a_i \cdot x + b_i \geq 0 \}, i \in I_1, \) and \( W_i = \{ x \mid a_i \cdot x + b_i > 0 \}, i \in I_2, \) for some \( a_i \in \mathbb{R}^N, b_i \in \mathbb{R}, i \in \{1, \ldots, K\} = I_1 \cup I_2, \) and define

\[
W = \bigcap_{i=1}^{K} \{ x \mid a_i \cdot x + b_i \geq 0 \}.
\]
Since \( V \subseteq W \), we have that \( \overline{V} \subseteq \overline{W} = W \). In order to prove that \( W \subseteq \overline{V} \), we take \( x \in W \setminus V \) and \( \varepsilon > 0 \). Then \( a_i \cdot x + b_i = 0 \), for \( i \in I_3 \subseteq I_2 \) and \( a_i \cdot x + b_i > 0 \), for \( i \in I_2 \setminus I_3 \). Let \( \delta > 0 \) be such that \( a_i \cdot y + b_i > 0 \), for all \( i \in I_3 \setminus I_2 \) and \( y \in \mathbb{R}^N \) with \( \|x - y\| < \varepsilon \).

Because \( V \neq \emptyset \) we have that \( x \in V \), for some \( z \in \mathbb{R}^N \). Let \( \lambda = \min(\varepsilon, \delta, 1)(\|z\| + 1)\|z\|^{-1} \) and define \( y = (1 - \lambda)x + \lambda z \). Then \( \|x - y\| < \varepsilon \), and it remains to show that \( y \in V \).

Firstly, since \( x, z \in W \), \( \lambda \in (0, 1] \), and \( W \) is convex we find that \( y \in W \), which implies that \( a_i \cdot y + b_i \geq 0 \), for all \( i \in I_1 \). Secondly, we have that

\[
    a_i \cdot y + b_i = (1 - \lambda)(a_i \cdot x + b_i) + \lambda(a_i \cdot z + b_i) = \lambda(a_i \cdot x + b_i) > 0,
\]

for all \( i \in I_3 \). Finally, \( \|x - y\| \leq \delta \) implies that \( a_i \cdot y + b_i > 0 \), for all \( i \in I_3 \setminus I_2 \). \( \Box \)

Since we do have \( \bigcup V_i = \bigcup \overline{V_i} \), for finite collections of subsets, we directly find the following result.

**Corollary 5.2.1.** Let \( V \subseteq U \). Then \( \overline{V} \subseteq U \).

In Chapter 2 we used a variant of Farkas’ Lemma to prove a necessary and sufficient condition for a finite classification problem to be solvable by a 1LP; see Theorem 2.2.1. Below we give a reformulation of this result that is useful in this chapter.

**Proposition 5.2.1.** Let \( N \in \mathbb{N} \) and \( \Omega_1, \Omega_2 \subseteq \mathbb{B}^N \) with \( \Omega_1 \cap \Omega_2 = \emptyset \). Then there exist \( a \in \mathbb{R}^N \) and \( b \in \mathbb{R} \) such that

\[
    \begin{cases} 
    ax + b \geq 0 & \text{if } x \in \Omega_1 \\
    ax + b < 0 & \text{if } x \in \Omega_2, 
    \end{cases}
\]

if and only if there do not exist \( \lambda \geq 0, z \in \mathbb{B}^N \), not all zero, that satisfy

\[
    \sum_{x \in \Omega_1} \lambda_x z = \sum_{x \in \Omega_2} \lambda_x z = \sum_{x \in \Omega_2} \lambda_x z.
\]

Since polyhedrons play an important role in the remainder of this chapter, we discuss an alternative representation of polyhedrons due to Minkowski, that is used in Section 5.6; see also Nemhauser & Wolsey [1988] and Schrijver [1986]. We use the formulation presented in [Schrijver, 1986], which requires the definition of the convex-hull and the cone of a given subset.

**Definition 5.2.7.** Let \( V \subseteq \mathbb{R}^N \). The convex-hull of \( V \) is denoted by \( \text{conv.hull}(V) \) and is given by

\[
    \text{conv.hull}(V) = \{ x \in \mathbb{R}^N \mid x = \sum_{i=1}^r \lambda_i v_i, v_i \in V, \lambda_i \geq 0, \sum_{i=1}^r \lambda_i = 1, r \geq 1 \}.
\]

**Definition 5.2.8.** Let \( V \subseteq \mathbb{R}^N \). The cone of \( V \) is denoted by \( \text{cone}(V) \) and is given by

\[
    \text{cone}(V) = \{ x \in \mathbb{R}^N \mid x = \sum_{i=1}^r \lambda_i v_i, v_i \in V, \lambda_i \geq 0, r \geq 0 \}.
\]

The convex-hull of a subset is the smallest convex set containing that subset. Similarly, the cone of a subset is the smallest convex cone containing that subset. A
convex cone is a nonempty set of vectors $C$ satisfying $\lambda x + \mu y \in C$, for all $x, y \in C$ and $\lambda, \mu \geq 0$. The well-known result of Minkowski states that every polyhedron can be written as the sum of a convex-hull and a cone of two finite sets; for a proof of this result see Schrijver [1986].

**Proposition 5.2.2 (Minkowski).** Let $V \subseteq \mathbb{R}^n$. Then $V \subseteq P'$, if and only if

$$V = \text{conv.hull}(S) + \text{cone}(T),$$

for some finite subsets $S, T \subseteq \mathbb{R}^n$.

Note that by definition we have that $V_1 + V_2 = \{x_1 + x_2 \mid x_1 \in V_1, x_2 \in V_2\}$, for $V_1, V_2 \subseteq \mathbb{R}^n$, $\emptyset + V_2 = \emptyset$, $\text{conv.hull}(\emptyset) = \emptyset$, and $\text{cone}(\emptyset) = \text{cone}(\{0\}) = \{0\}$. This implies that the result of Proposition 5.2.2 is valid for the empty polyhedron also.

We end this section with some remarks about affine subspaces. To this end we first have to define a linear subspace.

**Definition 5.2.9.** The subset $T \subseteq \mathbb{R}^n$ is called a linear subspace if $\lambda x + \mu y \in T$, for all $x, y \in T$ and $\lambda, \mu \in \mathbb{R}$.

**Definition 5.2.10.** The subset $T \subseteq \mathbb{R}^n$ is called an affine subspace if $T = x_0 + T'$, for some linear subspace $T' \subseteq \mathbb{R}^n$ and $x_0 \in \mathbb{R}^n$.

The following is a well-known result about linear subspaces; see also Nemhauser & Wolsey [1988].

**Proposition 5.2.3.** Let $T \subseteq \mathbb{R}^n$. Then $T$ is a linear subspace, if and only if there exist a $K \times N$ matrix $B$, for some $K \leq N$, such that $T = \{x \in \mathbb{R}^N \mid x = uB, u \in \mathbb{R}^K\}$.

This implies the following result.

**Corollary 5.2.2.** Let $T \subseteq \mathbb{R}^N$. Then $T$ is an affine subspace, if and only if there exist a $K \times N$ matrix $B$, for some $K \leq N$, and an $x_0 \in \mathbb{R}^N$ such that $T = \{x \in \mathbb{R}^N \mid x = x_0 + uB, u \in \mathbb{R}^K\}$.

Finally, we define the dimension of an affine subspace.

**Definition 5.2.11.** Let $T \subseteq \mathbb{R}^N$ be an affine subspace, say $T = \{x \in \mathbb{R}^N \mid x = x_0 + uB, u \in \mathbb{R}^K\}$, for some $K \times N$ matrix $B$, $K \leq N$, and $x_0 \in \mathbb{R}^N$. Then the dimension of $T$, denoted by $\dim(T)$, is given by the rank of the matrix $B$.

See Nemhauser & Wolsey [1988] for a definition of the rank of a matrix.

### 5.3 Characterizations of $C_m$

In this section we give necessary and sufficient conditions for a subset to be classifiable with an mLP, for $m \neq 2$. The case $m = 2$ is discussed in Section 5.5.1 and Section 5.5.3, where we present necessary conditions and sufficient conditions for a subset to be classifiable with a 2LP, respectively. However, part of the results given in this section apply to the case $m = 2$ as well and are used in the above mentioned sections.
5.3 Characterisations of $C_m$

We start with the case $m = 1$. It is well-known that with a 1LP functions representing single halfspaces and the two constant functions $f \equiv 0$ and $f \equiv 1$ can be formed; see also Chapter 2. More precisely, we have the following result.

**Theorem 5.3.1.** $C_1 = H \cup \{0, R^N\}$.

**Proof.** We only prove that $C_1 \subseteq H \cup \{0, R^N\}$; the proof of the reverse is similar.

Let $V \in C_1$, then $V = \mathcal{F}(f)$, for some $f \in R_{1,N,1}$. Consequently, $f(x) = \theta(a \cdot x + b)$ for some $a \in R^N$, $b \in R$, and all $x \in R^N$. If $a = 0$, then either $f(x) = 1$, $x \in R^N$, giving $V = R^N$, or $f(x) = 0$, $x \in R^N$, corresponding with $V = \emptyset$. If $a \neq 0$, then $V = \{x \in R^N | a \cdot x + b \geq 0\} \in H$.

Next, we consider the cases $m \geq 3$. We prove the less well-known result that a given subset can be classified with an mLP, for some $m \geq 3$, if and only if the subset is a member of $\bar{U}$. This result has been stated before, but not in such a rigorous formulation; see for instance Gibson & Cowan [1990]. Intuitively, the result is clear, as follows from the following argument. Every set in $\bar{U}$ can be constructed by placing the required number of hyperplanes, and coloring the cells formed with these hyperplanes with two colors representing 0 and 1. The placing of the hyperplanes is done with the first layer of an mLP, the coloring with 0 and 1 corresponds to the construction of a suitable Boolean function by the remaining layers. It is well-known that every Boolean function can be constructed using two layers, for instance using the threshold version of its conjunctive normal form (CNF) or disjunctive normal form (DNF); see Wegener [1987] and also Chapter 4. Hence, a 3LP is sufficient to classify a given set in $\bar{U}$. To show that every subset that can be classified with an mLP is in $\bar{U}$, one can use the above arguments reversely.

Next we present a formal proof of the statement given above. The proof is based upon a number of lemmas, which are proved subsequently. Many of these intermediate results are also used in the following sections.

**Theorem 5.3.2.** Let $m \geq 3$. Then $C_m = \bar{U}$.

**Proof.** Follows directly from the following two lemmas.

**Lemma 5.3.1.** Let $V \in \bar{U}$ and $m \geq 3$. Then $V \in C_m$, i.e., $V$ can be classified with an mLP.

**Lemma 5.3.2.** Let $V \in C_m$, for some $m \geq 1$. Then $V \in \bar{U}$.

Alternatively, the results given by Lemma 5.3.1 and Lemma 5.3.2 can be summarized as follows.

**Corollary 5.3.1.** Let $V \subseteq R^N$. Then there exists an MLP that classifies $V$, if and only if $V \in \bar{U}$.

Before we prove Lemma 5.3.1 we prove two other auxiliary lemmas.

**Lemma 5.3.3.** Let $V \in C_m$, for some $m \geq 2$. Then $V^* \in C_m$.
Proof. Suppose that \( V \in C_m, m \geq 2 \), then \( V = \mathcal{J}(f) \), for some \( f \in R_m = R_{m,N,1} \). Then \( f = g \circ h \) with \( g \in R_{1,K,1} \) and \( h \in R_{m-1,N,K} \), for some \( K \in \mathbb{N} \). Let \( g = \theta \circ \tilde{g} \), then since \( \{h(x) \mid x \in \mathbb{R}^K\} \) is finite, there exists an \( \varepsilon > 0 \) such that for \( \tilde{g} = \tilde{g} + \varepsilon \), we have that \( \tilde{g}(h(x)) \neq 0 \), and \( \tilde{g}(h(x)) > 0 \), if and only if \( g(h(x)) \geq \varepsilon \), for all \( x \in \mathbb{R}^N \). Hence, it follows that

\[
1 - f(x) = 1 - \theta[\tilde{g}(h(x))] = 1 - \theta[\tilde{g}(h(x))] = \theta[-\tilde{g}(h(x))],
\]

for all \( x \in \mathbb{R}^N \). Since \( -\tilde{g} \in \Lambda_K \) this implies that \( V^* = \mathcal{J}(1 - f) \in C_m \). \( \square \)

Note that Lemma 5.3.3 does not hold for \( m = 1 \). For this reason we in general have to distinguish between subsets that are in \( C_m \) or subsets whose complement is in \( C_m \).

Lemma 5.3.4. Let \( m \in \mathbb{N} \) and let \( V_1, \ldots, V_L \subseteq \mathbb{R}^N \) be a collection of subsets with \( V_i \in C_m \) or \( V_i^* \in C_m \) for all \( i = 1, \ldots, L \). Then \( \bigcap_{i=1}^L V_i \subseteq C_{m+1} \).

Proof. Let \( 0 \leq K \leq L \) be such that \( V_i \in C_{m_1} \), for \( i = 1, 2, \ldots, K \), and \( V_i^* \in C_{m_1} \), for \( i = K + 1, \ldots, L \). Define \( f \in R_{m+1,N,1} \) by \( f = g \circ h \), where \( g \in R_{1,L,1} \) is given by \( g(w) = \theta\left(\sum_{i=1}^K w_i - \sum_{i=K+1}^L w_i - K\right) \), and \( h \in R_{m,N,L} \) is given by \( \mathcal{J}(h_i) = V_i \), \( i = 1, \ldots, K \), and \( \mathcal{J}(h_i) = V_i^* \), \( i = K + 1, \ldots, L \). Then we have

\[
x \in \mathcal{J}(f) \iff g(h(z)) = 1 \iff \sum_{i=1}^K h_i(x) - \sum_{i=K+1}^L h_i(x) - K \geq 0 \iff \forall i \leq K : h_i(x) = 1 \land \forall i > K : h_i(x) = 0 \iff \forall 1 \leq i \leq L : x \in V_i,
\]

which completes the proof. \( \square \)

Instead of taking the intersection of a number of subsets one can take the union of a number of subsets. Combining Lemma 5.3.4 and Lemma 5.3.3 we directly obtain the following result.

Corollary 5.3.2. Let \( m \in \mathbb{N} \) and let \( V_1, \ldots, V_L \subseteq \mathbb{R}^N \) be a collection of subsets with \( V_i \in C_m \) or \( V_i^* \in C_m \), for all \( i = 1, \ldots, L \). Then \( \bigcup_{i=1}^L V_i \subseteq C_{m+1} \).

The above results can be used straightforwardly to prove Lemma 5.3.1.

Proof of Lemma 5.3.1. It follows directly from Theorem 5.3.1 and Lemma 5.3.4 that \( \bar{P} \subseteq C_2 \). Using Corollary 5.3.2 yields \( \bar{U} \subseteq C_2 \). The result now follows from \( C_m \subseteq C_{m+1} \), for all \( m \geq 1 \), which is a special case of Lemma 5.3.4. \( \square \)

Based on the above results, one can easily find an explicit construction of a 3LP that classifies a given subset \( V \subseteq \bar{U} \). Assume \( V = \bigcup_{i=1}^{K_m} \bigcap_{j=1}^{L_i} W_{ij} \), for some \( W_{ij} \in H, K, L_i \in \mathbb{N} \). Define the coefficients \( a_i \in \{-1,+1\} \), \( i = 1, \ldots, K \), and the functions \( h_i \in R_{i,N,L_i}, g_i \in R_{i,L_i,K_i}, i = 1, \ldots, K, e \in R_{1,K_i,1} \), and \( f \in R_{m,N,1} \), as follows. Firstly, for all \( i = 1, \ldots, K \) and \( j = 1, \ldots, L_i \), if \( W_{ij} \in H \), then set \( a_{ij} = +1 \) and choose \( h_{ij} \in R_{i,N,1} \) such that \( \mathcal{J}(h_{ij}) = W_{ij} \), if \( W_{ij} \notin H \), then set \( a_{ij} = -1 \) and choose \( h_{ij} \in R_{i,N,1} \) such that \( \mathcal{J}(h_{ij}) = W_{ij}^* \). Secondly, for all \( i = 1, \ldots, K \), let

\[
g_i(w_1, \ldots, w_{L_i}) = \theta \left[ \sum_{j=1}^{K_i} a_{ij} w_{ij} - \sum_{j=1}^{K_i} (a_{ij} + 1)/2 \right].
\]
Thirdly, let

\[ e(z_1, \ldots, z_K) = \theta[-1 + \sum_{i=1}^{K} z_i] \, . \]

Fourthly and finally, let \( f(x) = e(g_1(h_1(x)), \ldots, g_K(h_K(x))) \), for all \( x \in \mathbb{R}^N \). Then it can be verified in steps that \( f(x) = 1 \), if and only if \( x \in V \). The weights of the first, second, and third layer of this 3LP follow from the definition of \( h \), \( g \), and \( e \), respectively. In Figure 5.1 we have displayed the basic structure of the corresponding 3LP, in case that \( K = L = N = 3 \).
Before we prove Lemma 5.3.2 we prove the following lemma.

**Lemma 5.3.5.** Let \( V = \mathcal{J}(f) \), for some \( f \in R_m, m \geq 2, \) \( f = g \circ h, g \in R_{m-1,l,1}, \)
\( h \in R_{1,N,1,1} \), and \( L \in \mathbb{N} \). Then there exist \( K \in \mathbb{N} \) and \( x_1, \ldots, x_K \in \mathbb{R}^N \) such that

\[
V = \bigcup_{i=1}^{K} \bigcap_{j=1}^{L} [\mathcal{J}(h_j)]^{(2h_j(x_i)-1)},
\]

where \([\mathcal{J}(h_j)]^{(i)} = \mathcal{J}(h_j)\) and \([\mathcal{J}(h_j)]^{(i-1)} = \mathcal{J}^*(h_j)\).

**Proof.** For all \( x \in \mathbb{R}^N \) we define the set \( T(x) = \{ y \in \mathbb{R}^N \mid h(y) = h(x) \} \). The proof is completed by showing the following results.

(i) There exist \( K \in \mathbb{N} \) and \( x_1, x_2, \ldots, x_K \in \mathbb{R}^N \) such that \( V = \bigcup_{i=1}^{K} T(x_i) \).

(ii) For every \( x \in \mathbb{R}^N \) we have that \( T(x) = \bigcap_{j=1}^{L} [\mathcal{J}(h_j)]^{(2h_j(x)-1)} \).

The proof of (i) starts by considering the set \( S = \{ z \in h(\mathbb{R}^N) \mid g(z) = 1 \} \). Since \( h(\mathbb{R}^N) = \{ h(z) \mid z \in \mathbb{R}^N \} \subseteq B^L \) is finite, \( S \) is finite and hence, we have that \( S = \{ h(x_1), h(x_2), h(x_K) \} \), for some \( K \in \mathbb{N} \) and \( x_1, x_2, \ldots, x_K \in \mathbb{R}^N \). It remains to be shown that \( x_1, x_2, \ldots, x_K \) satisfy the expression for \( V \) given in (i), which follows straightforwardly from

\[
\begin{align*}
  z \in V & \iff f(z) = g(h(z)) = 1 \\
  & \iff h(z) \in S \\
  & \iff \exists i, 1 \leq i \leq K : h(z) = h(x_i) \\
  & \iff \exists i, 1 \leq i \leq K : x \in T(x_i).
\end{align*}
\]

To prove (ii) we note that

\[
\begin{align*}
  y \in T(x) & \iff h(y) = h(x) \\
  & \iff \forall j, 1 \leq j \leq L : h_j(x) = h_j(y) \\
  & \iff [\exists j, h_j(x) = 1 : y \in \mathcal{J}(h_j)] \land [\forall j, h_j(x) = 0 : y \in \mathcal{J}^*(h_j)] \\
  & \iff y \in \bigcap_{j=1}^{L} [\mathcal{J}(h_j)]^{(2h_j(x)-1)}.
\end{align*}
\]

This completes the proof of (i) and (ii). \( \square \)

It remains to prove Lemma 5.3.2.

**Proof of Lemma 5.3.2.** We show that \( C_\alpha \subseteq \bar{U} \), for all \( m \geq 1 \). For \( m = 1 \), this follows from \( C_1 = H \cup \{ \emptyset, \mathbb{R}^N \} \subseteq \bar{U} \). For \( m \geq 2 \), this follows from Lemma 5.3.5, using that \( \mathcal{J}(h_j), \mathcal{J}^*(h_j) \in H \cup \{ \emptyset, \mathbb{R}^N \} \), for all \( j = 1, \ldots, L \). \( \square \)

The above shows that the classification capabilities of MLPs are equal for \( m \geq 3 \), if we assume that the total number of nodes is unbounded. As follows from the complexity results of Boolean circuits, discussed in Chapter 4, this is not true for MLPs where the total number of nodes is bounded. In the following section we consider MLPs with a bounded number of nodes in the first layer and an unbounded number of nodes in the other layers. Still, these results do not show any separation between MLPs, for \( m \geq 3 \). In Section 5.5.1 we show that the condition given in
Lemma 5.3.1 is not sufficient for solving a binary classification problem by a 2LP, i.e. there exist binary classification problems that can be solved by a 3LP but cannot be solved by a 2LP, which implies that $C_2$ is a proper subset of $C_3$.

We end this section by presenting a necessary condition for a subset to be classifiable with an MLP, which is based on the notion of an affine subspace; see Definition 5.2.10 and below for the definition of an affine subspace and some related properties. We start with the trivial observation that every computable function preceded by an affine transformation is again computable.

**Proposition 5.3.1.** Let $f \in R_{m,N}$, let $K \in N$, let $B$ be a $K \times N$ matrix, let $z_0 \in R^N$, and let the function $g : R^K \rightarrow B$ be defined by $g(u) = f(z_0 + uB)$, for all $u \in R^K$. Then $g \in R_{m,K}$.

Note that we frequently use this result implicitly when we show a figure of a subset that has to be classifiable or has to be unclassifiable; we omit the axes and the scaling because these are irrelevant to the fact whether or not the subset is classifiable with an MLP.

The above result also implies that every intersection of a classifiable subset with an affine subspace is again classifiable. This can be formulated as follows.

**Corollary 5.3.3.** Let $V \in C_{m,N}$ and let $T = \{z \in R^N | z = z_0 + uB, u \in R^K\}$ be an affine subspace, for some $K \times N$ matrix $B$, $K \leq N$, and $z_0 \in R^N$. Then $(V \cap T)_{R^K} \in C_{m,K}$, where $(V \cap T)_{R^K}$ denotes the intersection $V \cap T$ viewed as a subset of $R^K$, which is defined by $(V \cap T)_{R^K} = \{u \in R^K | z_0 + uB \in V\}$.

### 5.4 Minimal number of first-layer nodes

In the previous section we showed that $V \in \tilde{U}$ is a necessary condition for a subset $V$ to be classifiable with an MLP. In this section we derive a lower bound for the number of nodes in the first hidden layer that is required for the classification of $V$ with any MLP. That is, if $V$ is classified with an $m$LP, for some $m \geq 1$, then the $m$LP must have a number of first-layer (hidden) nodes that is at least the lower bound derived in this section. The lower bound is given by the number of hyperplanes that are a part of the surface of $V$. As we will see this result belongs to the class of results that are intuitively trivial but hard to prove. First, we consider the special case that $V$ is a polyhedron, because this will introduce some notations and techniques useful in considering the general case. Furthermore, for a subclass of the polyhedrons the lower bound can be shown to be tight. In our treatment of polyhedrons we use a number of basic results that are discussed in Nemhauser & Wolsey, 1988. In fact we use these results to obtain an upper bound on the minimal number of first layer nodes necessary for the classification of a polyhedron. In the second part of this section we derive the lower bound for arbitrary subsets in $\tilde{U}$, which is equal to the upper bound in some special cases.

We introduce a special symbol for the minimal number of first-layer nodes that is required by any MLP that classifies a given subset.
Definition 5.4.1. Let $V \subseteq \mathbb{R}^N$. Then the minimal number of first-layer nodes that is required by any MLP with two or more layers that classifies $V$, is denoted by $L(V)$ and is given by

$$L(V) = \min \{ L \in \mathbb{N} | V = \mathcal{J}(f), f = g \circ h, g \in R_{m-1,l,i}, h \in R_{1,n,I}, m \geq 2 \}. \quad (5.2)$$

As usual we define the minimum over an empty set to be infinite. Because we want the expression for $L(V)$ given by (5.2) to be as short as possible, it does not consider 1LPs. However, since any 1LP can be extended to a 2LP with the same number of first-layer nodes, it follows that $L(V) = 1$, if $V \in C_1$. This implies that $L(V)$ is equal to the minimal number of first-layer nodes that is required by any MLP that classifies $V$. In fact, one easily shows that $L(V) = 1$, if and only if $V \in \tilde{H} \cup \{ \emptyset, \mathbb{R}^N \}$. Furthermore, it follows from Lemma 5.3.3 that $V$ and $V^*$ require the same number of first-layer nodes, which is also intuitively clear.

Proposition 5.4.1. Let $V \subseteq \mathbb{R}^N$. Then $L(V) = L(V^*)$.

From Theorem 5.3.2 it follows that $L(V) < \infty$, if and only if $V \in \tilde{U}$. More concrete, if $V \in \tilde{U}$ and hence $V = \bigcup_{i=1}^{K_1} \bigcap_{j=1}^{K_2} W_{ij}$, for some $W_{ij} \in \tilde{H}, K_1, K_2 \in \mathbb{N}$, then the construction of a 3LP that classifies $V$, given in Section 5.3, yields that $L(V) \leq \sum_{i=1}^{K_2} L_i$. There is no guarantee that this bound is tight. We therefore search for lower bounds for $L(V)$. In some special cases the derived lower bound is shown to be tight, in which case we have found an expression for the minimal number of first-layer nodes that is required for the classification of the considered subset. One such a special case is to be found within the collection of polyhedrons, which are treated first. Note that because of Proposition 5.4.1, all results derived for polyhedrons are valid for the complements of polyhedrons also.

5.4.1 Polyhedrons

In what follows, $V \subseteq \mathbb{R}^N$ is a polyhedron, i.e., $V = \bigcap_{i=1}^{K} W_i$, for some $W_i \in H, i = 1, \ldots, K$, and $K \in \mathbb{N}$. We may assume without loss of generality that all the defining halfspaces are different, i.e., $W_i \neq W_j$, for all $i \neq j$. In Section 5.3 it is noted that polyhedrons can be classified with a 2LP; see the proof of Lemma 5.3.1. Furthermore, it follows from the proof of Lemma 5.3.4 that the polyhedron $V$ can be classified with a 2LP that has $K$ nodes in the first hidden layer, and, hence, $L(V) \leq K$. To find the minimal number of first layer nodes, we search for the minimal representation of $V$ as an intersection of halfspaces. Hence, the objective is to delete the unnecessary halfspaces in the definition of $V$. To this end we define the notion of a halfspace necessary for the description of $V$.

Definition 5.4.2. The halfspace $W \in H$ is necessary for the description of $V$ if $V \subseteq W$ and if there exists an $x \in W^*$ such that $x \in W_i$, for all $i \in \{1, \ldots, K\}$ with $W_i \neq W$. 

We introduced the notion of a halfspace $W$ being necessary for the description of $V$, because Definition 5.4.2 does not explicitly demands that $W$ is part of the description of $V$. We have chosen for this formulation, because it is notionally convenient in many of the forthcoming results. However, we show below that a halfspace necessary for the description of a polyhedron, must always be part of the
5.4 Minimal number of first-layer nodes

definition of the considered polyhedron, which implies that it is in fact necessary in
the description of that polyhedron; see also Nemhauser & Wolsey [1988].

Lemma 5.4.1. Let \( W \in H \) be necessary for the description of \( V \). Then \( W = W_i \),
for some \( i \in \{1, \ldots, K\} \).

Proof. If \( W \neq W_i \), for all \( i = 1, \ldots, K \), then, since \( W \) is necessary for the description
of \( V \), there exists an \( x \in W^* \) with \( x \in \bigcup_{i=1}^{K} W_i = V \), contradicting \( V \subseteq W \).
\(\square\)

Consequently, all halfspaces that are not part of the definition of \( V \) are not nec-
essary for the description of \( V \). In general, unnecessary halfspaces are usually called
redundant halfspaces.

Definition 5.4.3. The halfspace \( W \) is redundant for the description of \( V \), if it is
not necessary for the description of \( V \).

Below we give an alternative formulation of redundant halfspaces.

Lemma 5.4.2. The halfspace \( W \) is redundant for the description of \( V \), if and only
if \( \bigcap \{W_i \mid W_i \neq W\} = V \).

Proof. If \( W \) is redundant, then \( W \) is not necessary. Hence, \( V \subseteq W \), or, if \( x \in W_i \),
for all \( i \in \{1, \ldots, K\} \) with \( W_i \neq W \), then \( x \in W \). In the first case we have that
\( W \neq W_i \), for all \( i \in \{1, \ldots, K\} \), which implies that \( \bigcap \{W_i \mid W_i \neq W\} = V \). In the
second case we find the second inclusion of \( V \subseteq \bigcap \{W_i \mid W_i \neq W\} \).

If \( W \) is not redundant, then there exists an \( x \in W^* \subseteq V^* \) such that \( x \in W_i \), for all
\( i \in \{1, \ldots, K\} \) with \( W_i \neq W \). Consequently, \( \bigcap \{W_i \mid W_i \neq W\} \neq V \).
\(\square\)

The above definitions of necessary and redundant halfspaces are inspired by the
following result.

Theorem 5.4.1. Let \( V^* \neq \emptyset \) and let \( W_1, \ldots, W_L \) be necessary and \( W_{L+1}, \ldots, W_K \)
be redundant for the description of \( V \), for some \( 1 \leq L \leq K \). Then \( V = \bigcap_{i=1}^{L} W_i \) is
the unique minimal representation of \( V \) as an intersection of halfspaces.

This is exactly the kind of result that we aim at, in order to find the minimal number
of first-layer nodes to classify \( V \). In fact we prove the following result, but the proof
must be postponed until the end of Section 5.4.2.

Theorem 5.4.2. Let \( V \subseteq \mathbb{R}^N \) be a polyhedron with \( V^* \neq \emptyset \). Then
\( L(V) = \{W \in H \mid W \) is necessary for the description of \( V\} \).

Although Theorem 5.4.1 is intuitively trivial, we do not know of any short proof.
Below we show that Theorem 5.4.1 follows from a similar result, which is based on
some basic polyhedral theory concerning facets. To this end we need quite a large
number of definitions and intermediate results. Except for some minor notational
differences, Definition 5.4.4 through 5.4.10 and Proposition 5.4.2 through 5.4.5 have
been copied from Chapter I.4 of the book by Nemhauser & Wolsey [1988].

Definition 5.4.4. A set of points \( x^1, \ldots, x^k \in \mathbb{R}^N \) is called linearly independent, if
the unique solution of \( \sum_{i=1}^{k} \lambda_i x^i = 0 \) is \( \lambda_i = 0 \), \( i = 1, \ldots, k \).
**Definition 5.4.5.** A set of points \( x^1, \ldots, x^k \in \mathbb{R}^N \) is called affinely independent, if the set \((x^1, 1), \ldots, (x^k, 1) \in \mathbb{R}^{N+1}\) is linearly independent.

**Definition 5.4.6.** The polyhedron \( V \subseteq \mathbb{R}^N \) is of dimension \( k \), denoted by \( \dim(V) = k \), if the maximum number of affinely independent points in \( V \) is \( k + 1 \).

In Section 5.2 we defined an affine subspace and its dimension; see Definitions 5.2.10 and 5.2.11. A well-known result states that each affine subspace is a polyhedron, and one can easily show that the definition of the dimension of an affine subspace corresponds to the general definition of the dimension of a polyhedron when applied to affine subspaces; see for instance Nemhauser & Wolsey [1988]. In case that an affine subspace is not equal to the whole space \( \mathbb{R}^N \), its dimension is less than \( N \). However, in this section we are particularly interested in polyhedrons that have a dimension that equals \( N \). This type of polyhedrons is given a special name.

**Definition 5.4.7.** The polyhedron \( V \subseteq \mathbb{R}^N \) is full-dimensional if \( \dim(V) = N \).

**Proposition 5.4.2.** The polyhedron \( V \subseteq \mathbb{R}^N \) is full-dimensional, if and only if \( V^\circ \neq \emptyset \).

**Definition 5.4.8.** The halfspace \( W \subseteq H \) is said to correspond to a valid inequality for \( V \) if \( V \subseteq W \).

Note that a necessary halfspace always corresponds to a valid inequality.

**Definition 5.4.9.** Let \( W \subseteq H \) correspond to a valid inequality for the polyhedron \( V \subseteq P \). Then \( F = V \cap W^\circ \) is called a face of \( V \), and we say that \( W \) represents \( F \). A face \( F \) is said to be proper if \( F \neq \emptyset \) and \( F \neq V \).

**Definition 5.4.10.** A face \( F \) of the polyhedron \( V \) is called a facet if \( \dim(F) = \dim(V) - 1 \).

**Proposition 5.4.3.** Let \( F \) be a facet of \( V \). Then there is an \( i \in \{1, \ldots, K\} \) such that \( W_i \) represents \( F \).

The following result states that not all halfspaces representing a facet can be removed from the description of the polyhedron. For the proof of this result one makes use of Farkas’ lemma; see Proposition 2.2.2.

**Proposition 5.4.4.** Let \( F \) be a facet of \( V \) and let the halfspaces \( W_1, \ldots, W_L \) represent \( F \), for some \( 1 \leq L \leq K \). Then \( V \subseteq \bigcap_{l=1}^K W_l \).

Using Proposition 5.4.3, Definition 5.4.7, Definition 5.4.10, Proposition 5.4.4, and Lemma 5.4.2, in this order, one finds the following result.

**Corollary 5.4.1.** Let \( V \) be full-dimensional and let \( F \) be a facet of \( V \). Then there is exactly one \( i \in \{1, \ldots, K\} \) such that \( W_i \) represents \( F \), and this halfspace \( W_i \) is necessary for the description of \( V \).

The following result gives a condition for a necessary halfspace to represent a facet.

**Proposition 5.4.5.** Let \( W \subseteq H \) be necessary for the description of \( V \) and let \( W^\circ \cap V \neq \emptyset \). Then \( W \) represents a facet of \( V \).
5.4 Minimal number of first-layer nodes

Using that $V^o = \cap_{i=1}^K W_i^o$, combined with Lemma 5.4.1 and Proposition 5.4.2, we obtain the following result.

Corollary 5.4.2. Let $V$ be full-dimensional and let $W \in H$ be necessary for the description of $V$. Then $W$ represents a facet of $V$.

Now we come to the main result.

Theorem 5.4.3. Let $V$ be a full-dimensional polyhedron, let $V = \cap_{i=1}^K W_i$, for some $W_i \in H$, $W_i \neq W_j$ if $i \neq j$, and $K \in \mathbb{N}$, let $F_i$, $i = 1, \ldots, L$, denote the facets of $V$, and let $W_i$ represent $F_i$, $i = 1, \ldots, L$, for some $1 \leq L \leq K$. Then $V = \cap_{i=1}^L W_i$ is the unique minimal representation of $V$ as an intersection of halfspaces.

Proof. Let $V_j = \cap_{i=1}^j W_i$. We show by induction that $V_j = V$, for $j = K, K - 1, \ldots, L$. Obviously, $V_K = V$. Assume that $V_j = V$, for some $L < j \leq K$. Since $W_i \neq W_i$ for all $i < j$, it follows from Corollary 5.4.1 that $V_j$ does not represent a facet of $V = V_j$. Hence, Corollary 5.4.2 implies that $W_j$ is redundant for the description of $V_j$. Using Lemma 5.4.2 it follows that $V_{j-1} = V_j = V$.

Corollary 5.4.1 implies that the given representation is minimal and unique. □

It is now straightforward to prove Theorem 5.4.1

Proof of Theorem 5.4.1. Follows from Proposition 5.4.2, Corollary 5.4.1, Corollary 5.4.2, and Theorem 5.4.3. □

Furthermore, we may conclude the following.

Corollary 5.4.3. Let $V \subset \mathbb{R}^N$ be a full-dimensional polyhedron. Then $L(V)$ is less than or equal to the number of facets of $V$, i.e.,

$$L(V) \leq |\{W \in H \mid W \text{ represents a facet of } V\}|.$$ \hspace{1cm} (5.3)

In order to prove Theorem 5.4.2 we must be able to conclude that $L(V)$ is larger than or equal to the number of facets of $V$, i.e., we must show that there does not exist an MLP that can classify $V$ with a number of first-layer nodes that is less than the upper bound given by (5.3). Although intuitively clear, this conclusion must be postponed until the end of this section.

From the above results it follows that for a full-dimensional polyhedron $V$ its necessary halfspaces and its facet representing halfspaces are the same, and that these halfspaces are the only ones required for the description of $V$. If $V$ is not full-dimensional things are not at all clear. There are polyhedrons with facet-representing halfspaces that are not necessary for their description and necessary halfspaces that do not represent a facet; see Figure 5.2, which shows a polyhedron with different combinations of necessary and facet representing halfspaces. One can even easily construct polyhedrons that have no halfspaces that are necessary for their description, and no halfspaces that represent a facet. When considering these cases one should recall that whether or not a halfspace is necessary for the description of a given polyhedron depends on the specific description used, whereas whether or not a halfspace represents a facet of a given polyhedron does not depend on the description of the polyhedron. We do not consider non full-dimensional polyhedrons in detail, and end our discussion about arbitrary polyhedrons with some remarks.
If $V$ is a polyhedron of dimension $\dim(V) < N$, then one easily shows that $V$ lies in the intersection of $N - \dim(V)$ hyperplanes; see for instance Nemhauser & Wolsey [1988]. Since each of these hyperplanes can be described by the intersection of two halfspaces, we find the following result.

**Proposition 5.4.6.** Let $V \subseteq \mathbb{R}^N$ be a polyhedron. Then $L(V)$ is less than or equal to the number of facets of $V$ plus $2(N - \dim(V))$.

The upper bound for $L(V)$ given in Proposition 5.4.6 is in general not tight. We conjecture that a lower bound for $L(V)$ is given by the number of facets of $V$ plus the number of halfspaces that are necessary for the description of $V$ and that do not represent a facet of $V$. However, we have no idea about a proof of this result. In the following section we derive a general lower bound for subsets in $\bar{U}$, but this bound is the trivial zero bound for polyhedrons that are not full-dimensional.

### 5.4.2 Unions of pseudo polyhedrons

In this subsection we consider the general case that $V \in \bar{U}$, i.e., $V$ is a union of pseudo polyhedrons. Using similar ideas as in Section 5.4.1 we derive a lower bound for $L(V)$. We start by generalizing the notion of a facet to an interfacet. Interfacet is a combination of interior, interface and facet. An interfacet is any nonempty intersection of a hyperplane with the surface of a given subset $V \subseteq \mathbb{R}^N$, that locally looks like the facet of a full-dimensional polyhedron or like the complement of a facet of a full-dimensional polyhedron. However, we do not need an exact definition of an interfacet, we only need the definition of halfspaces representing an interfacet.

**Definition 5.4.11.** Let $V \subseteq \mathbb{R}^N$. The halfspace $W \in \bar{H}$ represents an interfacet of $V$, if there exists an open ball $B \subseteq \mathbb{R}^N$ such that

$$\emptyset \neq W \cap B \subseteq V \quad \land \quad \emptyset \neq W^* \cap B \subseteq V^*.$$
As the surface of \( V \) may or may not be part of \( V \), an interfacet may also be part of \( V \) or part of \( V^* \). This symmetry is fully expressed by the following trivial result.

**Proposition 5.4.7.** Let \( V \subseteq \mathbb{R}^N \). Then \( W \in \mathcal{H} \) represents an interfacet of \( V \) if and only if \( W^* \) represents an interfacet of \( V^* \).

In some cases it is irrelevant whether or not the interfacet is part of \( V \). In those cases we can use the following theorem.

**Theorem 5.4.4.** Let \( V \in \mathcal{U} \) and let \( W \in H \). Then there exists an open ball \( B \) such that

\[
\emptyset \neq W^* \cap B \subseteq V \land \emptyset \neq W^* \cap B \subseteq V^* ,
\]

(5.4)

if and only if \( W \) or \( W^* \) represents an interfacet of \( V \).

**Proof.** The 'if'-part is trivial, so it remains to prove the 'only-if'-part. To this end we use the following proposition, which is discussed below.

**Proposition 5.4.8.** Let \( W \in H \), let \( B \) be an open ball with \( W^* \cap B \neq \emptyset \), and let \( W_1, \ldots, W_K \in \mathcal{H} \) be a collection of halfspaces. Then there exists an \( z \in W^* \cap B \) such that \( z \notin W_i^* \), for all \( i \in \{1, \ldots, K\} \) with \( W_i^* \neq W^* \).

Now the proof of the 'only-if'-part of Theorem 5.4.4 is continued as follows.

Let \( V \in \mathcal{U} \), then \( V = \bigcup_{i=1}^K \bigcap_{j=1}^L W_{ij} \), for some \( W_{ij} \in \mathcal{H} \), \( K, L \in \mathbb{N} \). Let \( W \in H \) and let the open ball \( B \) satisfy (5.4). Then \( W^* \cap B \neq \emptyset \), and, using Proposition 5.4.8, we can choose an \( x_0 \in W^* \cap B \) such that \( x_0 \notin W_i^* \), for all \( i, j \) with \( W_i^* \neq W^* \). Consequently, we can take \( \delta > 0 \) such that \( B' = B(x_0, \delta) \) satisfies \( B' \subseteq B \) and \( B' \cap W_i^* = \emptyset \), for all \( i, j \) with \( W_i^* \neq W^* \).

This yields

\[
\emptyset \neq W^* \cap B' \subseteq V \land \emptyset \neq W^* \cap B' \subseteq V^* ,
\]

and we complete the proof by showing that either \( W^* \cap B' \subseteq V \) or \( W^* \cap B' \subseteq V^* \).

Assume not, i.e., assume that \( x_1 \in W^* \cap B' \cap V \) and \( x_2 \in W^* \cap B' \cap V^* \). Since \( x_1 \in V \) and \( x_2 \in V^* \), there is a pair \( i, j \) such that \( x_1 \in W_{ij} \) and \( x_2 \notin W_{ij} \). Since \( x_1, x_2 \in W^* \), we have that \( W_{ij}^* \neq W^* \), and, hence, \( B' \cap W_{ij}^* = \emptyset \), which contradicts \( x_1 \in B' \cap W_{ij} \) and \( x_2 \in B' \cap W_{ij}^* \).

\( \square \)

The above used Proposition 5.4.8 follows straightforwardly from Lemma 5.4.3 given below, by considering the subspace \( W^* \).

**Lemma 5.4.3.** Let \( B \subseteq \mathbb{R}^N \) be an open ball and \( W_1, \ldots, W_K \in \mathcal{H} \) a collection of halfspaces. Then there exists an \( x \in B \) such that \( x \notin W_i^* \), for all \( i \in \{1, \ldots, K\} \).

**Proof.** We construct a sequence of open balls \( B = B_0, B_1, \ldots, B_K \), such that

\[
B_i \subseteq B_{i-1} \cap (W_i^*)^* ,
\]

(5.5)

for all \( i = 1, \ldots, K \). Let \( B_0 = B \). Assume that \( B_0, B_1, \ldots, B_j \) have been constructed such that (5.5) holds for all \( i = 1, \ldots, j \). Since \( \dim(W_{ij}^*) = N < \dim(B_j) \), it follows that \( B_j \notin W_{ij}^* \). Let \( x_0 \in B_j \cap (W_{j+1}^*)^* \), then since both sets are open \( B_{j+1} = B(x_0, \delta) \subseteq B_j \cap (W_{j+1}^*)^* \), for some \( \delta > 0 \). This yields (5.5) for \( i = j + 1 \).

It is obvious that any \( x \in B_K \) satisfies \( x \notin W_i^* \), for all \( i = 1, \ldots, K \).
Our next result can be viewed as a generalization of Corollary 5.4.1. It says that only halfspaces used in the description of a given $V \in \bar{H}$ can represent any of its interfacets.

**Theorem 5.4.5.** Let $V = \bigcup_{i=1}^{K} \bigcap_{j=1}^{L_i} W_{ij}$, for some $W_{ij} \in \bar{H}$, $K, L_i \in \mathbb{N}$, and let $W \in \bar{H}$ represent an interfacet of $V$. Then $W = W_{ij}$, for some $i \in \{1, \ldots, K\}$ and $j \in \{1, \ldots, L_i\}$.

**Proof.** We consider the following two cases (i) $W \in H$ and (ii) $W^* \in H$.

(i) $W \in H$.

We start by showing that, without loss of generality, we may assume that $W_{ij} \neq W^*$, for all $i, j$. If $W_{ij} = W^*$, for some $i, j$, then we consider the set $V' \in \bar{H}$, given by $V' = \bigcup_{i=1}^{K} \bigcap_{j=1}^{L_i} W_{ij} \neq W_{ij}$. One easily verifies that $V \subseteq V' \subseteq V \cup W \leq (W^*)^* \subseteq V \cup W$, which implies that

$$\emptyset \neq W \cap B \subseteq V' \land \emptyset \neq W^* \cap B \subseteq (V')^*.$$

Therefore, we may assume that $W_{ij} \neq W^*$, for all $i, j$. Next, we also assume that $W \neq W_{ij}$, for all $i, j$, and show that this leads to a contradiction.

Since $W \cap B \neq \emptyset$ and $W^* \cap B \neq \emptyset$ we have that $W = \emptyset$ and $W^* = \emptyset$. Since $W = \emptyset$, we can choose an $x_0 \in W \cap B$ such that $x_0 \notin W^* \cap B$, for all $i, j$ with $W_{ij} \neq W^*$.

Using $W \in H$ we find $W = \emptyset$, and, hence, $x_0 \in W \cap B \subseteq V$. Consequently, $x_0 \in \bigcap_{i=1}^{L_i} W_{ij}$, for some $i \in \{1, \ldots, K\}$, say $i = 1$. Since $x_0 \notin W^* \cap B$, it follows that $W_{1j} \neq W^* \cap B$, and, hence, $x_0 \in W_{1j} \cap B$, for all $j = 1, \ldots, L_1$. This implies $x_0 \in V^*$, which contradicts $x_0 \notin W^* \cap B \subseteq (V^*)^*$.

(ii) $W^* \in H$.

From Proposition 5.4.7 it follows that $W^* \in H$ is an interfacet of $V^*$. By using (5.1) of Lemma 5.2.1 and applying (i) to $W^*$ and $V^*$, we obtain that $W^* = W_{ij}$, for some $i \in \{1, \ldots, K\}$ and $j \in \{1, \ldots, L_i\}$, and the result follows.

Next, we use Theorem 5.4.5 to conclude that any halfspace that represents an interfacet of a given subset, must be represented by a first-layer node of every MLP that classifies the given subset. We formulate this result separately for $1$LPs and $m$LPs, with $m \geq 1$. First, we consider $1$LPs. If $W$ represents an interfacet of a subset that is classifiable with an $1$LP, then, using Definition 5.4.11, it follows that neither the subset itself, nor its complement are empty. Hence, combining Theorem 5.3.1 with Theorem 5.4.5 for the case that $K = 1$ and $L_1 = 1$, we obtain the following result.

**Corollary 5.4.4.** Let $V = \mathcal{J}(f)$, for some $f \in R_{1,N,1}$, and let $W \in \bar{H}$ represent an interfacet of $V$. Then $W \in H$ and $W = \mathcal{J}(f)$.

This trivial result serves as an introduction for the general result for $m$LPs, with $m \geq 2$, given below. Using the representation of any $\mathcal{J}(f)$, for some $f \in R_m$, $m \geq 2$, given by Lemma 5.3.5, and combining it with Theorem 5.4.5, one can similarly prove the following result.

**Corollary 5.4.5.** Let $V = \mathcal{J}(f)$, for some $f \in R_m$, with $m \geq 2$, let $f = g \circ h$, with $g \in R_{m-1,1,1}$, $h \in R_{1,N,L}$, $L \in \mathbb{N}$, and let $W \in \bar{H}$ represent an interfacet of $V$. Then
5.4 Minimal number of first-layer nodes

\[ W = \mathcal{F}(h_i), \text{ if } W \in H, \text{ and } W = \mathcal{F}^*(h_i), \text{ if } W^* \in H, \text{ for some } i \in \{1, \ldots, L\}. \]

Note that Corollary 5.4.5 also holds, with "represent an interfacet of \( V \)" replaced by "represent an interfacet of \( V \) or \( V^* \)", which follows from Proposition 5.4.7.

Using Definition 5.4.1 and Corollary 5.4.5, we obtain the main result of this section.

**Corollary 5.4.6.** Let \( V \in \hat{U} \). Then

\[
L(V) \geq \left| \{ W \in H \mid W \text{ represents an interfacet of } V \} \right| \\
+ \left| \{ W \in H \mid W^* \text{ represents an interfacet of } V \} \right| \\
- \left| \{ W \in H \mid W \text{ and } W^* \text{ represent an interfacet of } V \} \right|.
\]

The above result gives the desired lower bound for the number of first-layer nodes that is required by any MLP that classifies a given subset. In Figure 5.3 two examples are given that demonstrate the use of Corollary 5.4.6. In Figure 5.3a we have depicted a subset \( V'_1 \) with \( L(V'_1) \geq 15 \). This set can be classified with a 2LP with this number of first-layer nodes, which proves that \( L(V'_1) = 15 \); see Proposition 2.3.2 in Chapter 2. As we already noted in Chapter 2, it is possible to alter \( V'_1 \) by a slight transformation of two of the bounding hyperplanes, such that 15 first-layer nodes are no longer sufficient to classify the resulting subset \( V''_1 \) with a 2LP. However, it still holds that \( L(V''_1) = 15 \), since it is trivial to construct a 3LP with 15 first-layer nodes that classifies \( V''_1 \).

(a) ![Hexagon Illustration](image1.png)  
(b) ![Hexagon Illustration](image2.png)

**Figure 5.3:** Two classifiable subsets with 15 interfacet representing halfspaces (a), and 6 interfacet representing halfspaces (b), respectively, which are indicated by the circles. As always, thick boundary lines belong to the subsets and thin boundary lines do not belong to the subsets.

The bow-tie shaped set \( V_2 \) in Figure 5.3b satisfies \( L(V_2) \geq 4 \). In Section 5.5 we show that this set cannot be classified with a 2LP, whatever the number of first-layer...
nodes. However, one can easily verify that the set shown in Figure 5.3b can be
classified with a 3LP with 4 nodes in the first layer and 2 nodes in the second
layer, which proves that \( L(\mathcal{V}_2) = 4 \). Note that \( L(\mathcal{V}_2^*) = 6 \), where \( \mathcal{V}_2^* \) corresponds to
the interior of the subset given in Figure 5.3b.

One can also construct examples for which the lower bound given in Corollary 5.4.6
is not tight. However, we prove that the lower bound is tight for the special case
of full-dimensional polyhedrons, by showing that in that case halfspaces representing
interfacs and halfspaces representing facets are the same thing. Except proving
this equivalence, Theorem 5.4.6 gives a third equivalent formulation which is useful
for characterizing (inter)facets of full-dimensional polyhedrons.

**Theorem 5.4.6.** Let the polyhedron \( \mathcal{V} \subseteq \bigcap_{i=1}^K W_i \), for some \( W_i \in \mathcal{H}, K \subseteq \mathbb{N} \), and
let \( W \in \Pi \). Then the following three statements are equivalent.

(i) \( V \) is full-dimensional and \( W \) represents a facet of \( V \).

(ii) \( W = W_i \), for some \( i = \{1, \ldots, K\} \), and there exists an \( x_0 \in W^n \) such that
\( x_0 \in W_j^c \), for all \( j \) with \( W_j \neq W \).

(iii) \( W \) represents an interfacet of \( V \).

**Proof.** We complete the proof in three steps.

(i) \( \Rightarrow \) (ii)

If \( W \) represents a facet of \( V \), then \( V \subseteq W \) and this facet is the face \( F = V \cap W^n \),
with \( \dim(F) = \dim(V) - 1 \). Since \( V \) is full-dimensional, \( \dim(V) = N \) and, hence,
\( \dim(F) = N - 1 \). Using Proposition 5.4.3 it follows that \( W_i \) represents \( F \), for some
\( i \). Since \( F \subseteq W^n \cap W_i^n \) and \( \dim(F) = N - 1 = \dim(W^n) = \dim(W_i^n) \), we have that
\( W^n = W_i^n \) and \( W = W_i \). It also implies that \( F \) is full-dimensional with respect to
the subspace \( W^n \). Hence, there exists an \( x_0 \in W^n \) such that \( x_0 \in W_j^c \), for all \( j \) with \( W_j \neq W \).
Since \( V \) is full-dimensional, \( W_j^c = W^n \) only if \( W_j = W \).

(ii) \( \Rightarrow \) (iii)

Choose \( x_0 \in W_i^n \) such that \( x_0 \in W_j^c \), for all \( j \) with \( W_j \neq W_i \). Then \( B = B(x_0, \delta) \subseteq \bigcap_{i=1}^K W_i \) for \( \delta > 0 \) sufficiently small. Hence, \( x_0 \in W_i \cap B \subseteq W_i \cap \bigcap_{i=1}^K W_i \), \( W_j = V \) and \( \emptyset \neq W_i \cap B \subseteq V^n \cap B \subseteq V^n \), which proves that \( W \)
represents an interfacet of \( V \).

(iii) \( \Rightarrow \) (i)

Let the open ball \( B \) be such that
\[ \emptyset \neq W \cap B \subseteq V \land \emptyset \neq W^n \cap B \subseteq V^n. \]
Since \( B \) is open, we have that \( \emptyset \neq W^n \cap B \subseteq V \), which proves that \( V \) has an interior
point and thus is full-dimensional. By Theorem 5.4.5 it follows that \( W = W_i \), for
some \( i \). Furthermore, since \( W \cap B \neq \emptyset \) and \( W^n \cap B \neq \emptyset \), we have that \( \emptyset \neq W^n \cap B \subseteq V \cap W^n = V \cap W_i^n \). Hence, the face \( V \cap W_i^n \) has dimension \( N - 1 = \dim(V) - 1 \), which implies that \( W_i = W \) represents a facet.

Using Theorem 5.4.5, it follows that there do not exist halfspaces \( W \in \mathcal{H} \), such that
\( W^n \) represents an interfacet of a polyhedron. Hence, combining Corollary 5.4.6 with
Corollary 5.4.3 and Theorem 5.4.6, we obtain the following result.
Corollary 5.4.7. Let $V \subseteq \mathbb{R}^n$ be a full-dimensional polyhedron. Then $L(V)$ is equal to the number of facets of $V$, which equals the number of facets of $V$, i.e.,

$$L(V) = |\{W \in H \mid W \text{ represents an facet of } V\}|.$$

From this the proof of Theorem 5.4.2 is trivial.

Proof of Theorem 5.4.2. Follows directly from Corollary 5.4.7, Theorem 5.4.3, Proposition 5.4.2, and Theorem 5.4.1. □

Note that this yields three expressions for the minimal number of first-layer nodes that is necessary for the classification of a full-dimensional polyhedron.

Many of the above results yield a non-trivial bound for full-dimensional polyhedrons and their complements only. However, if $V \subseteq W^*$, then we can apply the results to the affine subspace $W^*$, using the results for affine subspaces given in Section 5.2 and Section 5.3.

We end this section with a conjecture concerning a special class of subsets in $\tilde{U}$, which consist of a union of full-dimensional polyhedrons. It states that any halfspace necessary for the description of such a subset is an intersect representing halfspace. If this conjecture can be proven to be correct, it can act as the first step in finding the minimal number of first-layer nodes necessary for the classification of this kind of subsets.

Conjecture 5.4.1. Let $V \in U$, $V = \bigcup_{i=1}^{K'} W_i$, for some $V_i \in P$, $V_i$ full-dimensional, and $K \in \mathbb{N}$. Then

$$V = \bigcap_{i=1}^{K'} W_i$$

for some $G_i \subseteq \{W \in H \mid W \text{ represents an facet of } V\}$ and $K' \in \mathbb{N}$.

5.5 Characterizations of $C_2$

In this section we study the classification capabilities of 2LPs. The outline of this section is as follows. First, we use Farkas' lemma and its implications for the 1LP loading problem to derive a number of necessary conditions for a subset to be classifiable with a 2LP. The necessary conditions prove that $C_2 \neq \tilde{U}$. From a geometrical point of view the necessary conditions can be said to be of a local nature. Secondly, we study a special type of two-dimensional subsets for which we show that one of the necessary conditions is also a sufficient condition for these subsets to be classifiable with a 2LP. Thirdly, we derive a set of sufficient conditions, which prove that $\tilde{P}$ is a strict subclass of $C_2$. Geometrically viewed the derived sufficient conditions have a global character, and are based on a decomposition of the considered subset into a number of so called basic subsets. These basic subsets are examined in detail in the fourth subsection. Finally, the fifth subsection considers a decomposition algorithm for the special case where the basic subsets correspond to closed affine halfspaces.
5.5.1 Necessary conditions

In Lemma 5.3.2 of Section 5.3 we proved that $V \in \hat{U}$ is a necessary condition for the subset $V \subseteq \mathbb{R}^N$ to be classifiable with an MLP, for any $m \geq 1$. Hence, it is also a necessary condition for a subset to be classifiable with a 2LP. Furthermore, it was shown in Section 5.3 that $V \in \hat{U}$ is a sufficient condition for the existence of a 3LP that classifies $V$. In this section we demonstrate that $V \in \hat{U}$ is not sufficient for the existence of a 2LP that classifies $V$, if $N \geq 2$. Thus there exist subsets of $\mathbb{R}^N$, for $N \geq 2$, that can be classified with a 3LP but not with a 2LP, which implies that $C_2$ is a true subset of $C_3$. We show this by proving the necessity of some additional non-trivial conditions for a given subset to be classifiable with a 2LP. The necessary conditions we derive concern some local geometrical behavior of the considered subsets, and are based on the fact that the Exclusive-OR problem cannot be solved by a 1LP.

The problem of finding a 2LP that classifies a given subset might be solved using the following trial and error approach. First, one chooses a certain number of first-layer nodes and selects a set of weights and biases for these nodes. This corresponds to the positioning of a number of halfspaces, one for every first-layer node. From the previous section we know that the number of first-layer nodes must be at least as large as the number of halfspaces that represent an interface of the considered subset; see Section 5.4.2 for more details. In the second part one searches for a 1LP, such that the 2LP that is obtained by combining this 1LP with the first-layer nodes selected in the first part, classifies the given subset. This 1LP must have a number of inputs that equals the number of first-layer nodes of the first part. The problem of finding this 1LP is a loading problem, since one has to map a finite number of 0 1-inputs to a 0 1-output. Hence, one can use the conditions for solvability of 1LPPs that were discussed in Chapter 2, to find out whether a 1LP exists. If it exists, one has finished, if it does not exist, one goes back to the first part, and changes the number of first-layer nodes and/or their weights and biases.

The solution approach given above tells us the following. To prove that a given subset is not classifiable with a 2LP, we must show that, whatever choice of the number of first-layer nodes and whatever choice of their corresponding halfspaces determined by their weights and biases, there does not exist a solution of the resulting 1LP. The idea is to take an unsolvable 1LPP, and use this problem as a master problem. In Chapter 2 we discussed an example of a 1LPP that has no solution, namely the Exclusive-OR problem. This problem will act as the master problem. All subsets that we prove to be unclassifiable with a 2LP, are proved to be unclassifiable with a 2LP because every choice of first-layer nodes leads to a 1LP that is either the Exclusive-OR problem, or has this problem as a subproblem. We therefore start with a discussion of the Exclusive-OR problem and especially some 2LP solutions of its corresponding classification problem.

Consider the incomplete classification problem $(\Omega_1 \cup \Omega_0, B, \{\Omega_1, \Omega_0\})$, given by $\Omega_1 = B_1 \cup B_4$ and $\Omega_0 = B_1 \cup B_3$, where $B_1, B_2, B_3, B_4$ are four balls in $\mathbb{R}^2$ defined by

$$B_1 = \{x \in \mathbb{R}^2 | (x_1 - 1)^2 + (x_2 - 1)^2 < \frac{1}{4} \},$$
\[ B_2 = \{ x \in \mathbb{R}^2 \mid (x_1 + 1)^2 + (x_2 - 1)^2 < \frac{1}{2} \} \]

\[ B_3 = -B_1, \text{ and } B_4 = -B_2, \] respectively. This classification problem contains as a subproblem a 1LP loading problem that can be formulated as follows. Find the weights of an MLP represented by \( f : \{-1, +1\}^2 \to \mathbb{B} \) that satisfies \( f(1, 1) = f(-1, -1) = 0 \) and \( f(1, -1) = f(-1, 1) = 1 \). This is simply an alternative formulation of the Exclusive-OR problem. The 2LP depicted in Figure 2.6a of Chapter 2 that solves the Exclusive-OR problem, can be easily transformed into a solution of the incomplete classification problem given above. The decision region of the resulting 2LP is shown in Figure 5.4a. It consists of the intersection of two parallel halfspaces. Similarly, one can construct a 2LP that solves this classification problem and has a decision region corresponding to Figure 5.4b; see also Section 5.5.3. Note that in both examples there is at least one halfspace that separates one ball from the three others. We show below that this is typical for any 2LP that solves the considered incomplete classification problem, i.e., we show that if the halfspaces are positioned like in Figure 5.4c, it cannot lead to a 2LP solution of that problem.

![Figure 5.4](image)

Figure 5.4: The shaded regions in (a) and (b) correspond to the decision regions of two 2LPs that solve the Exclusive-OR problem. Without halfspaces that separate precisely one ball, this is not possible (c); see the text for further information.

We give two versions of the same result. First, we present the version given in Lemma 5.5.1, because it links up with Proposition 5.2.1. This enables us to give a short proof of Lemma 5.5.1. On the other hand, the version given by Lemma 5.5.2 more closely corresponds to the discussion based on Figure 5.4c above. Lemma 5.5.1 and Lemma 5.5.2 will be used below in the proofs of Theorem 5.5.2 and Theorem 5.5.1, respectively.

**Lemma 5.5.1.** Let \( f \in R_2 \), \( f = g \circ h \), with \( g \in R_{K,1} \) and \( h \in R_{1,N,K} \), for some \( K \in \mathbb{N} \). Then there do not exist \( x_1, x_2, x_3, x_4 \in \mathbb{R}^K \) such that \( f(x_1) = f(x_2) = 1 = 1 - f(x_3) = 1 = f(x_4) \), and satisfy

\[ \forall i = 1, \ldots, K : h_i(x_1) + h_i(x_3) = h_i(x_2) + h_i(x_4). \]  

(5.6)

**Proof.** Assume that \( x_1, x_2, x_3, x_4 \in \mathbb{R}^K \) exist as above. Then \( g \) separates the finite subset \( \{ h(x_1), h(x_2) \} \subseteq \mathbb{B}^K \) from the finite subset \( \{ h(x_3), h(x_4) \} \subseteq \mathbb{B}^K \). However, using Proposition 5.2.1 and (5.6) it follows that these sets are not separable, which yields a contradiction.

\[ \square \]

**Lemma 5.5.2.** Let \( f \in R_2 \), \( f = g \circ h \), with \( g \in R_{K,1} \) and \( h \in R_{1,N,K} \), for some
$K \in \mathbb{N}$. Then there do not exist $x_1, x_2, x_3, x_4 \in \mathbb{R}^N$ such that $f(x_1) = f(x_3) = 1 = 1 - f(x_2) = 1 - f(x_4)$, and satisfy
\[\forall i = 1, \ldots, K : [h_i(x_1) = h_i(x_2) \land h_i(x_3) = h_i(x_4)] \lor [h_i(x_1) = h_i(x_4) \land h_i(x_2) = h_i(x_3)]. \quad (5.7)\]

**Proof.** Follows directly from Lemma 5.5.1 and the equivalence of (5.6) and (5.7), which is trivial. \qed

Note that although we numbered $x_1, x_2, x_3, x_4$ such that they correspond with the balls $B_1, B_2, B_3, B_4$ given in Figure 5.4, Lemma 5.5.1 and Lemma 5.5.2 do not give an explicit specification of the position of these four points in $\mathbb{R}^N$. It is for instance not necessary that all four points lie in a two-dimensional affine subspace.

Next, we use Lemma 5.5.1 and Lemma 5.5.2 to derive two necessary and geometrically local conditions for a subset to be classifiable with a 2LP. These conditions are such that if they are not satisfied by the considered subset, then, whatever the positions of an arbitrary number of selected halfspaces, one can choose four points that satisfy the conditions of Lemma 5.5.1 or Lemma 5.5.2.

![Figure 5.5](image.png)

Figure 5.5: Three sets in $\mathbb{R}^2$ that cannot be classified with a 2LP. For (a) and (b) this can be shown using Theorem 5.5.1, and for (b) and (c) this can be shown using Theorem 5.5.2. Note that solid boundary lines do and thin boundary lines do not belong to the presented sets.

The first condition considers the local behavior of the examined subset around a hyperplane. It can be used to prove that the subsets of $\mathbb{R}^2$ presented in Figure 5.5a and Figure 5.5b cannot be classified with a 2LP. The theorem that uses this condition is named after the shape of the subset in Figure 5.5b, which resembles a bow-tie.

The second condition considers the local behavior of the examined subset around a point. It can be used to prove that the subsets of $\mathbb{R}^2$ presented in Figure 5.5c and Figure 5.5b cannot be classified with a 2LP. The theorem that uses this condition is named after the shape of the subset in Figure 5.5c, which resembles a twisted bow-tie. As will become clear from the formulation of the two conditions and their proofs below, there is a strong duality-like similarity between the two conditions. It might be possible to prove that the two conditions are indeed each others dual, by adapting the formulation.

**Theorem 5.5.1 (The bow-tie condition).** Let $V \in \mathcal{C}_2$ and $W \in \mathcal{H}$. Then there
cannot exist two open balls $B_1, B_2 \subset \mathbb{R}^N$ such that
\[ \emptyset \neq W^a \cap B_j \subseteq V \land \emptyset \neq W^s \cap B_j \subseteq V^s, \]
\[ \emptyset \neq W^* \cap B_j \subseteq V^* \land \emptyset \neq W^* \cap B_j \subseteq V. \] (5.8)

Proof. Let $V \in C_2$, i.e., $V = J(f)$, for some $f \in R_2$, $f = g \circ h$, $g \in R_{t, N, K}$, $h \in R_{t, N, K}$, and $K \in N$. Let $W \in H$ and assume that the two open balls $B_1, B_2 \subset \mathbb{R}^N$ satisfy (5.8). We prove that this leads to a contradiction.

First, we show that without loss of generality we may assume that $J^w(h_i) \cap B_j = \emptyset$, for all $i = 1, \ldots, K$ with $J^w(h_i) \neq W^w$ and $j = 1, 2$.

Suppose this does not hold, i.e., $J^w(h_i) \cap B_j \neq \emptyset$, for some $i \in \{1, \ldots, K\}$ and some $j \in \{1, 2\}$, say for $j = 1$. It is clear that $W^w \cap B_1 \neq \emptyset$, hence, using Proposition 5.4.8 it follows that there exists an $x_0 \in W^w \cap B_1$ such that $x_0 \notin J^w(h_i)$, for all $i = 1, \ldots, K$ with $J^w(h_i) \neq W^w$. Hence, we can choose $\delta > 0$ such that $B'_j = B(x_0, \delta) \subseteq B_1$ and $B'_j \cap J^w(h_i) = \emptyset$, for all $i = 1, \ldots, K$ with $J^w(h_i) \neq W^w$. It follows straightforwardly that $\emptyset \neq W^w \cap B'_j \subseteq V$ and $\emptyset \neq W^s \cap B'_j \subseteq V^s$.

Next, we take $x_1 \in W^w \cap B_1$, $x_2 \in W^s \cap B_1$, $x_3 \in W^w \cap B_2$, and $x_4 \in W^s \cap B_2$. Then (5.8) implies that $f(x_1) = f(x_2) = 1 = 1 - f(x_3) = 1 - f(x_4)$. Furthermore, the above implies that $h_i(x_1) = h_i(x_2)$ and $h_i(x_3) = h_i(x_4)$, for all $i = 1, \ldots, K$ with $J^w(h_i) \neq W^w$. Finally, it is obvious that $h_i(x_1) = h_i(x_4)$ and $h_i(x_2) = h_i(x_3)$, for all $i = 1, \ldots, K$ with $J^w(h_i) = W^w$. The contradiction now follows from Lemma 5.5.2.

Comparing the condition given by (5.8) in Theorem 5.5.1 with (5.4) in Theorem 5.4.4, yields that Theorem 5.3.1 can also be formulated as follows.

Corollary 5.5.1. Let $V \in C_2$. Then there cannot exist a halfspace $W \in H$ such that (i) $W$ or $W^w$ represents an interface of $V$, and (ii) $W$ or $W^s$ represents an interface of $V^s$.

However, the formulation of Theorem 5.5.1 is preferred because it gives more freedom in selecting the two balls.

Before we consider the twisted bow-tie condition we introduce a new notion for subsets of $\mathbb{R}^N$, called radial around a point. Informally we say that the subset $V \subseteq \mathbb{R}^N$ is radial around a certain center point, when whether or not a point belongs to $V$, does not depend on its distance to that center point. Furthermore, $V$ is locally radial around a certain center point, when this property holds within some ball around the center point. This is formalized in the following definition.

Definition 5.5.1. Let $V \subseteq \mathbb{R}^N$ and let $x_0 \in \mathbb{R}^N$. $V$ is said to be radial around $x_0$, if for all $\lambda > 0$ and $x \in \mathbb{R}^N$, it holds that $x_0 + \lambda x \in V$, if and only if $x_0 + x \in V$. $V$ is said to be locally radial around $x_0$, if there exist a $\delta > 0$, such that for all $0 < \lambda < 1$ and $x \in B(x_0, \delta)$, it holds that $\lambda x + (1 - \lambda)x_0 \in V$, if and only if $x \in V$.

In the formulation of the twisted bow-tie condition given in Theorem 5.5.2 below, we assume that the considered subset $V$ is locally radial around the origin. This is not a restriction, since a subset $V \in U$ is locally radial around any point in $\mathbb{R}^N$; see
Proposition 5.5.1 below. The proof of this result is straightforward and therefore omitted.

**Proposition 5.5.1.** Let \( V \in \hat{U} \) and \( x_0 \in \mathbb{R}^N \). Then \( V \) is locally radial around \( x_0 \).

The radial property of \( V \) expressed in Proposition 5.5.1 is another way of stating that \( V \) has piece-wise linear bounds. The reverse of Proposition 5.5.1 does not hold because there are piece-wise linear subsets that are not in \( \hat{U} \). These subsets have an infinite number of bounding hyperplanes, for instance the subset that represents the black fields of an infinite checker board.

In the following theorem, we fix the center point to 0 for notational convenience.

**Theorem 5.5.2 (The twisted bow-tie condition).** Let \( V \in \mathcal{C}_2 \). Let \( \delta > 0 \) be such that \( \lambda x \in V \), if and only if \( x \in V \), for all \( x \in B = B(0, \delta) \) and \( 0 < \lambda < 1 \).

Then there cannot exist two open balls \( B_1, B_2 \subset B \) such that

\[
\begin{align*}
B_1 & \subseteq V \land -B_1 \subseteq V, \\
B_2 & \subseteq V^\ast \land -B_2 \subseteq V^\ast.
\end{align*}
\] (5.9)

**Proof.** Let \( V \in \mathcal{C}_2 \), i.e., \( V = \mathcal{J}(f) \), for some \( f \in \mathcal{R}_2 \), \( f = g \circ h \), \( g \in \mathcal{R}_{1,K,1} \), \( h \in \mathcal{R}_{1,N,K} \), \( K \in \mathbb{N} \), and let \( \tilde{h}_i(x) = \theta(\tilde{h}_i(x)) \), for \( i = 1, \ldots, K \). Let \( B \) be as above and assume that the two open balls \( B_1, B_2 \subset B \) satisfy (5.9). We prove that this leads to a contradiction.

First, we show that without loss of generality we may assume that \( \mathcal{J}(\tilde{h}_i) \cap B = \emptyset \), for all \( i = 1, \ldots, K \) with \( \tilde{h}_i(0) \neq 0 \).

Suppose this does not hold, i.e., \( \mathcal{J}(\tilde{h}_i) \cap B \neq \emptyset \), for some \( i \in \{1, \ldots, K\} \) with \( \tilde{h}_i(0) \neq 0 \). Consider all \( x \in B \) for which an \( i \in \{1, \ldots, K\} \) exists with \( \tilde{h}_i(x) = 0 \) and \( \tilde{h}_i(0) \neq 0 \). Among these \( x \in B \), choose the one, say \( x_0 \), with minimal length \( ||x_0|| \).

Let \( \lambda = ||x_0||/\delta \), then \( 0 < \lambda < 1 \), and we consider the balls \( B' = \lambda B = \{\lambda x \mid x \in B\} = B(0, ||x_0||) \), \( B'_1 = \lambda B_1 \), and \( B'_2 = \lambda B_2 \). Then by construction it follows that \( \mathcal{J}(\tilde{h}_i) \cap B' = \emptyset \), for all \( i = 1, \ldots, K \) with \( \tilde{h}_i(0) \neq 0 \). Furthermore, using that \( x \in V \), if and only if \( \lambda x \in V \), for all \( x \in B \), we obtain that

\[
\begin{align*}
B'_1 & \subseteq V \land -B'_1 \subseteq V, \\
B'_2 & \subseteq V^\ast \land -B'_2 \subseteq V^\ast.
\end{align*}
\]

Hence, for all \( i = 1, \ldots, K \), either \( \tilde{h}_i(x) = h_i(y) \), for all \( x, y \in B \), or \( \tilde{h}_i(-x) = -\tilde{h}_i(x) \), for all \( x \in B \). The latter follows in case \( \tilde{h}_i(0) = 0 \), using that \( \tilde{h}_i \) is an affine function.

Finally, we choose \( x_1 \in B_1 \) and \( x_2 \in B_2 \) such that \( \tilde{h}_i(x_1) \neq 0 \), for all \( i = 1, \ldots, K \) and \( j = 1, 2 \), which is possible because of Lemma 5.4.3. Let further \( x_3 = -x_1 \) and \( x_4 = -x_2 \), then (5.9) implies that \( f(x_1) = f(x_2) = 1 - f(x_3) = 1 - f(x_4) \). From the above it can be deduced easily that \( \tilde{h}_i(x_1) + \tilde{h}_i(x_3) = \tilde{h}_i(x_2) + \tilde{h}_i(x_4) \), for all \( i = 1, \ldots, K \). The contradiction now follows from Lemma 5.5.1. \( \square \)

The obvious question is whether the combination of the given two conditions is sufficient for classifiability with a 2LP, or, alternatively, whether we require any other necessary conditions. The answer is that the conditions are not sufficient, at least not for an arbitrary subset in \( \mathbb{R}^N \). This can be shown as follows.
5.5 Characterizations of $C_2$

Take a subset $V \subseteq \mathbb{R}^2$ that is not classifiable with a 2LP, for instance any of the subsets in Figure 5.5, and take $N \geq 3$. Place $V$ in $\mathbb{R}^N$ to form a subset of a two-dimensional subspace $T$, denoted by $S \subseteq \mathbb{R}^N$. Then $S$ cannot be classifiable with a 2LP, because otherwise $V = (S \cap T)_{\mathbb{R}^2}$ would also be classifiable; see Corollary 5.3.3. However, $S$ satisfies both the bow-tie condition and the twisted bow-tie condition, because $S$ does not contain any open ball. Hence, these conditions are not sufficient. We therefore extend the conditions of Theorem 5.5.1 and Theorem 5.5.2 to conditions in which one considers intersections of $V$ with an arbitrary affine subspace.

**Corollary 5.5.2.** Let $V \in C_2$, $W \in H$, and $T$ an affine subspace of $\mathbb{R}^N$. Then there cannot exist two open balls $B_1, B_2 \subset \mathbb{R}^N$ such that

\[
0 \neq W^* \cap B_1 \cap T \subseteq V \cap T \land 0 \neq W^* \cap B_1 \cap T \subseteq V^* \cap T,
\]

\[
0 \neq W^* \cap B_2 \cap T \subseteq V^* \cap T \land 0 \neq W^* \cap B_2 \cap T \subseteq V \cap T.
\]

(5.10)

Like Theorem 5.5.2, the extended version of Theorem 5.5.2 also considers $x_0 = 0$ as center point only. This is because of some notational difficulties in the general case.

**Corollary 5.5.3.** Let $V \in C_2$ and $T$ a subspace of $\mathbb{R}^N$. Let $\delta > 0$ be such that $\lambda x \in V \cap T$, if and only if $x \in V \cap T$, for all $x \in B = B(0, \delta)$ and $0 < \lambda < 1$. Then there cannot exist two open balls $B_1, B_2 \subset B$ such that

\[
B_1 \cap T \subseteq V \cap T \land -B_1 \cap T \subseteq V^* \cap T,
\]

\[
B_2 \cap T \subseteq V^* \cap T \land -B_2 \cap T \subseteq V \cap T.
\]

(5.11)

The answer given above to the question whether the combination of the bow-tie condition and the twisted bow-tie condition is sufficient for classifiability, raises two new questions. The first is the question whether the combination of the bow-tie condition and the twisted bow-tie condition is sufficient for classifiability with a 2LP of an arbitrary subset in $\mathbb{R}^2$. The second question is whether the combination of the extended bow-tie condition and the extended twisted bow-tie condition is sufficient for classifiability with a 2LP of an arbitrary subset in $\mathbb{R}^N$, for all $N \in \mathbb{N}$. Below we show that the answer to both questions is negative. To this end we consider the approximative capabilities of 2LPs.

We have shown that the subsets presented in Figure 5.5 cannot be classified with a 2LP. However, as we know from Chapter 2, every subset can be approximately classified with a 2LP, i.e., for every subset there exists a classifiable subset that approximates the original subset. We describe an approach for obtaining approximative classifiable subsets that can be used for subsets that do not satisfy the bow-tie condition or the twisted bow-tie condition, although we do not know whether this approach works for all such subsets.

The approach consists of taking the intersection of the considered subset with a subset that is a union of two parallel, disjoint, affine halfspaces. In two dimensions this corresponds to the deletion of a strip from the set. The part of the set that is deleted by the strip must be small, in order for the resulting subset to be an approximation of the original subset. In case that this is not feasible, one can alternatively consider the same approach applied to complement of the subset. Furthermore, the strip must be such that whatever causes the subset to dissatisfy the bow-tie condition or
the twisted bow-tie condition, is removed. Thus in case of a subset that does not satisfy the bow-tie condition, one of the two hyperplanes that bound the strip must equal the hyperplane that bounds the halfspace considered in the bow-tie condition.

![Diagram](image.png)

Figure 5.6: Subsets that can be classified with a 2LP and approximate the subsets given in Figure 5.5b and Figure 5.5c.

If this approach is applied to the subset of Figure 5.5a, it corresponds to lifting one of the two small boxes with respect to the other one. The resulting subset is classifiable with a 2LP, which can be shown similarly as for the approximative subset for the subset of Figure 5.5b, which is discussed next. In order to apply the approach to the subset of Figure 5.5b, we have to delete two strips with four bounding hyperplanes, because there are two halfspaces that do not satisfy the bow-tie condition. Comparing the resulting subset to the original subset, it is like the two triangles are pulled away from each other; see Figure 5.6a. In Section 5.5.2, Section 5.5.3, and Section 5.5.4, we show in three steps that the subset of Figure 5.6a is classifiable with a 2LP.

In case the subset is not classifiable with a 2LP because it does not satisfy the twisted bow-tie condition, like the subset of Figure 5.5c, it can be approximated by a classifiable subset that is obtained from the given set by using the strip to cut off one of the cones that has as its top the center point that appears in the twisted bow-tie condition. For instance, Figure 5.6b and Figure 5.6c present two subsets that are obtained from the subset of Figure 5.5c using the described approach. Obviously, the smaller the cut, the better the approximation of the original subset.

In Section 5.5.2, Section 5.5.3, and Section 5.5.4, we show in three steps that the subsets of Figures 5.6b and c are classifiable with a 2LP. Furthermore, for all three subsets of Figure 5.6, it will follow that the better the approximation of the original subsets, the more hidden nodes are required. This corresponds well with the general approximation results of 2LPs that are discussed in Chapter 2; see Chapter 2 for a definition of the goodness of an approximation of a subset and for a discussion of the approximative capabilities of 2LPs.

Using Figure 5.6, given above, and Figure 5.7, presented below, we can explain why the first two necessary conditions discussed in this section are not sufficient for a
subset in $\mathbb{R}^2$ to be classifiable with a 2LP. As mentioned earlier, the proof that the subsets of Figure 5.6 are classifiable with a 2LP is given in three steps in Section 5.5.2, Section 5.5.3, and Section 5.5.4, respectively. However, the third step fails if one of the triangles in either of the three subsets of Figure 5.6 is unbounded. We can show that if both the triangles in either one of the three subsets are unbounded, the subset is no longer classifiable with a 2LP. Thus, for instance the subsets of Figures 5.7a and b are unclassifiable with a 2LP. The proof that these subsets are unclassifiable with a 2LP is straightforward using Lemma 5.5.2. This result indicates that in some cases boundedness of the considered subset is an additional necessary condition for this subset to be classifiable with a 2LP.

Figure 5.7: Unbounded subsets of $\mathbb{R}^2$. The subsets in (a) and (b) cannot be classified with a 2LP, while the subset in (c) can be classified with a 2LP.

We have not been able to generalize the unclassifiability of the subsets in Figures 5.7a and b to a necessary condition for classifiability of arbitrary subsets of $\mathbb{R}^2$, although we think that this might lead to some kind of twisted bow-tie condition at infinity. Connected with this is the fact that it can be shown that the half unbounded subset of Figure 5.7c is classifiable with a 2LP. The proof of this result is slightly more elaborated than the proof for the subset of Figure 5.6a, and somewhat resembles the proof of the classifiability of the subset of Figure 5.6c; see Section 5.5.4. We also think that the half unbounded versions of the subsets of Figures 5.6b and c can be classified with a 2LP, although we have not yet found a proof for all four possible types of subsets.

The above shows that the combination of the bow-tie condition and the twisted bow-tie condition is not sufficient for an arbitrary subset in $\mathbb{R}^2$ to be classifiable with a 2LP. However, it can be sufficient for some special types of subsets. In the following section we show that the twisted bow-tie condition is a necessary and sufficient condition for radial subsets of $\mathbb{R}^2$, i.e., subsets that are radial around an arbitrary point, to be classifiable with a 2LP, if their border points are left without consideration.

The above results on the unbounded subsets of $\mathbb{R}^2$ are straightforwardly used to show that the combination of the extended bow-tie condition and the extended twisted bow-tie condition is not sufficient for classifiability with a 2LP of an arbitrary subset.
in $\mathbb{R}^N$, for all $N \in \mathbb{N}$, for instance by considering the unbounded subset shown in Figure 5.7a. In fact, we can use the above approximation results to show that this also holds for bounded subsets. To this end we consider the subset $V \subseteq \mathbb{R}^2$ defined by

$$V = \{ x \in \mathbb{R}^2 \mid 0 \leq x_3 \leq \max[\min(x_1, x_2), \min(-x_1, -x_2)] \},$$

(5.12)

which is radial around 0, and is displayed in Figure 5.8. Clearly, $V$ does not satisfy either of the two extended conditions, because its cut with the affine subspace $x_3 = 0$, given by

$$\{ x \in \mathbb{R}^2 \mid (x_1 \geq 0 \land x_2 \geq 0) \lor (x_1 \leq 0 \land x_2 \leq 0) \},$$

(5.13)

does not satisfy the bow-tie condition, nor the twisted bow-tie condition. Hence, $V \not\subseteq C_2$. Next, we consider $V^*$, which is given by

$$V^* = \{ x \in \mathbb{R}^2 \mid 0 < x_3 < \max[\min(x_1, x_2), \min(-x_1, -x_2)] \}.$$

(5.14)

One can easily verify that $V^*$ satisfies both the extended bow-tie condition as well as the extended twisted bow-tie condition, given by Theorem 5.5.2 and Theorem 5.5.3, respectively. Now assume that $V^*$ is classifiable with a 2LP, say with $L \in \mathbb{N}$ hidden nodes. Then cuts of $V^*$ with the affine subspace $x_3 = \varepsilon$, are also classifiable with a 2LP with $L$ hidden nodes, for any $\varepsilon > 0$. Consequently, an arbitrary good approximation of the subset defined by (5.13) can be obtained with a 2LP that requires only $L$ hidden nodes. Again, one can straightforwardly use Lemma 5.5.2 to prove that this is not possible.

We conclude this section with some thoughts about the possibility of generalizing our approach for finding other necessary conditions. This could be done at two places. Firstly, one could use Lemma 5.5.2 or Lemma 5.5.1 to derive other conditions similar to the bow-tie condition and the twisted bow-tie condition. The above discussion about the unbounded subsets, as well as the discussion about the subset defined by (5.12), may provide starting-points for obtaining such conditions. Secondly, one could use an unsolvable 1LP loading problem other than the Exclusive-OR problem as the basis of a lemma similar to Lemma 5.5.1. Obviously, this makes sense only if this problem does not contain the Exclusive-OR problem as a subproblem. However, we think that no such problems exist, which is based on the following conjecture.

**Conjecture 5.5.1.** Let $N \in \mathbb{N}$, $\Omega_1, \Omega_0 \subseteq \mathbb{B}^N$, and $\Omega_1 \cap \Omega_0 = \emptyset$. Then there exist
5.5 Characterizations of $C_2$

![Diagram](image)

Figure 5.9: Two examples of two-dimensional radial subsets. The subset in (a) is classifiable with a 2LP, the subset in (b) is not classifiable with a 2LP; see Theorem 5.5.3.

$$\lambda_x \geq 0, \ z \in B^N, \text{ not all zero, that satisfy}$$

$$\sum_{x \in \Omega_1} \lambda_x z = \sum_{x \in \Omega_0} \lambda_x z_1,$$

$$\sum_{x \in \Omega_1} \lambda_x = \sum_{x \in \Omega_0} \lambda_x,$$

if and only if there exist $y_1, y_2 \in \Omega_1$ and $y_3, y_4 \in \Omega_0$ with

$$y_1 + y_2 = y_3 + y_4.$$

To put it differently, the instance of ILPLP given by $\Omega_1$ and $\Omega_0$ is a "yes"-instance, if and only if for every two-dimensional affine subspace $T$, the instance of ILPLP given by $\Omega_1 \cap T$ and $\Omega_0 \cap T$ is a "yes"-instance. One should note the connection with Lemma 5.5.1 and Proposition 5.2.1. It can be shown that Conjecture 5.5.1 is true for the case that $\Omega_1 \cup \Omega_0 = B^N$ and $N = 3$, by considering all 256 possible choices for $\Omega_1$ and $\Omega_0$. Since the number of possibilities increases rapidly with $N$, such a case checking proof becomes infeasible for larger $N$, but we do not know of a general proof.

5.5.2 A necessary and sufficient condition for a special class of subsets

In this section we consider two-dimensional subsets that are radial around an arbitrary center point; see Figure 5.9 for two examples of such subsets. Without loss of generality we assume that the center point is the origin, in which case we simply speak of radial subsets. From the previous section we know that the twisted bow-tie condition is a necessary condition for radial subsets to be classifiable with a 2LP. In this section we prove that the twisted bow-tie condition is also a sufficient condition for radial subsets to be classifiable with a 2LP, provided that their border points are left without consideration. For instance, this approach will show that the subset shown in Figure 5.9a is classifiable with a 2LP, and will show that the subset of Figure 5.9b is not classifiable with a 2LP.
As mentioned in the previous section, the necessary conditions derived in that section are in general not sufficient for a subset to be classifiable with a 2LP. The result proved in this section shows that the necessary conditions are sufficient for the classifiability of the special kind of two-dimensional radial subsets discussed here. Probably one of the reasons for the twisted bow-tie condition to be sufficient here, is that the subsets have a local geometry that equals their global geometry. Note that the necessary conditions derived in the previous section concern a geometrically local property of the considered subsets. In the following section we derive a set of sufficient conditions, which concern geometrically global properties of the considered subsets.

Below we present the main result of this section; see Theorem 5.5.3. Because we exclude the border points of the considered subsets, it is more convenient to formulate the result of the theorem in terms of functions. However, the result can be easily translated in terms of subsets by substituting the characteristic function of a subset. The proof of Theorem 5.5.3 given below is an existential proof based on Farkas’ lemma, that follows more or less the same lines as used in the previous section. Consequently, it does not give the weights of the 2LP that classifies the considered radial subset, if such a 2LP exists. In a separate theorem we therefore present an explicit expression for the weights of a classifying 2LP, if it exists; see Theorem 5.5.4.

**Theorem 5.5.3.** Let \( p \in \mathbb{N} \), let \( \alpha_1, \ldots, \alpha_{2p+1} \in \mathbb{R} \), with \( 0 \leq \alpha_1 < \alpha_2 < \cdots < \alpha_p < \pi \) and \( \alpha_{i+2p} = \alpha_i + \pi \), \( i = 1, \ldots, p+1 \), and let \( \beta_i \in \mathbb{B} \), \( i = 1, \ldots, 2p \). Then there exists a 2LP represented by the function \( f \in R_{2,2} \), that satisfies
\[
\forall \varphi > 0 \ \forall i = 1, \ldots, 2p \ \forall \alpha_i < \varphi < \alpha_{i+1} : f(r \cos \varphi, r \sin \varphi) = \beta_i,
\]
if and only if
\[
[\forall i = 1, \ldots, p : \beta_i = 1 \lor \beta_{i+p} = 1] \lor [\forall i = 1, \ldots, p : \beta_i = 0 \lor \beta_{i+p} = 0].
\]

**Proof.** That (5.16) is a necessary condition can be shown straightforwardly using Theorem 5.5.2 as follows. Assume that a 2LP \( f \) exists that satisfies (5.15), for \( \beta_1, \ldots, \beta_{2p} \in \mathbb{B} \) that do not satisfy (5.16). Then there exist \( i_1, i_2 \in \{1, \ldots, 2p\} \) such that \( \beta_{i_1} = 1 \land \beta_{i_2+p} = 1 \) and \( \beta_{i_1} = 0 \land \beta_{i_2+p} = 0 \). Let \( \varphi_j \in (\alpha_{i_j}, \alpha_{i_j+1}) \), \( x^{(j)} = (r \cos(\varphi_j), r \sin(\varphi_j)) \), and \( H_j = H(x^{(j)}, \delta_j) \), for \( j = 1, 2 \). Then using (5.13) one easily verifies that \( V = J(f) \), \( B = B(0, 2r) \), \( B_1 \), and \( B_2 \), satisfy (5.9) and the other conditions of Theorem 5.5.2, for \( r > 0 \) and \( \delta > 0 \) sufficiently small. This yields a contradiction.

Sufficiency of (5.16) follows directly from the following lemma, which is proved below.

**Lemma 5.5.3.** Let \( p, \alpha_1, \ldots, \alpha_{2p+1} \), and \( \beta_1, \ldots, \beta_{2p} \), be as in Theorem 5.5.3. Then there exists a set of numbers \( a_0, a_1, \ldots, a_p \in \mathbb{R} \) such that \( f : \mathbb{R}^2 \to \mathbb{B} \) defined by
\[
f(x) = f(x_1, x_2) = \theta[a_0 + \sum_{j=1}^{p} a_j \theta(-\sin(\alpha_j)x_1 + \cos(\alpha_j)x_2)],
\]
satisfies (5.15), if and only if (5.16) holds.

This completes the proof of Theorem 5.5.3. □
5.3 Characterisations of $C_2$

We used Lemma 5.5.3 to prove that (5.16) is a sufficient condition and Theorem 5.5.2 to prove that (5.16) is a necessary condition in the result of Theorem 5.5.3. However, the latter can also be proved using Lemma 5.5.3, essentially by showing that one does not have to consider other 2LPs than those given by Lemma 5.5.3. This can be done using a similar argument as used in the proof of Theorem 5.5.2.

Theorem 5.5.3 does not discuss the border points of the considered subsets, which are given by $\varphi = \alpha_i$, $i = 1, \ldots, 2p$. This is because we have not found a necessary and sufficient condition that includes these border points. In Section 5.5.4 we give an alternative approach for proving that condition (5.16) is sufficient, by demonstrating that it is a special case of one of the general sufficient conditions for classifiability with 2LPs that are derived in that section; see Theorem 5.5.14. The alternative proof is of a geometrical nature and is more constructive than the proofs given in this section. Furthermore, it might be useful for obtaining a necessary and sufficient condition that includes the border points, and it reveals some special properties of the two-dimensional classifiable radial subsets.

Next, we use Farkas' lemma to prove Lemma 5.5.3.

**Proof of Lemma 5.5.3.** Let $\varphi_i \in (\alpha_i, \alpha_{i+1})$ and $x^{(i)} = (\cos(\varphi_i), \sin(\varphi_i))$, for all $i = 1, \ldots, 2p$. Then $f$ defined by (5.17) satisfies (5.15), if and only if

$$f(x^{(i)}) = \theta[a_0 + \sum_{j=1}^{p} a_j \theta(\sin(\varphi_i - \alpha_j))] = \beta_i,$$

for all $i = 1, \ldots, 2p$. Define the $(2p) \times (p + 1)$ matrix $A$ by

$$A_{ij} = \begin{cases} (2\beta_i - 1) & \text{if } i = 1, \ldots, 2p, j = 0, \\ (2\beta_i - 1)\theta(\sin(\varphi_i - \alpha_j)) & \text{if } i = 1, \ldots, 2p, j = 1, \ldots, p. \end{cases}$$

Then it follows that the existence of $a_0, a_1, \ldots, a_p \in \mathbb{R}$ for which $f$ satisfies (5.15), is equivalent with the existence of an $a \in \mathbb{R}^{p+1}$ such that $Aa \geq 1$, where $1 \in \mathbb{R}^{2p}$ denotes a vector of only ones.

By the variant of Farkas' Lemma given by Proposition 2.2.2, this is equivalent with the non-existence of a $v \in \mathbb{R}^{2p}$ such that $vA = 0$ and $v \cdot 1 = \sum_{i=1}^{p} v_i > 0$. Before we examine these equations we note that $\alpha_i < \alpha_{i+1}$ and $\alpha_{i+p} = \alpha_i + \pi$, for all $i = 1, \ldots, p$, implies that

$$A_{ij} = \begin{cases} 2\beta_i - 1 & \text{if } j = 0 \text{ or } i - p < j \leq i, \\ 0 & \text{otherwise}. \end{cases}$$

Hence, $vA = 0$ is equivalent with

$$0 = (vA)_0 = \sum_{i=1}^{2p} v_i(2\beta_i - 1),$$

and

$$0 = (vA)_j = \sum_{i=j}^{j+p-1} v_i(2\beta_i - 1),$$

for all $j = 1, \ldots, p$. Combining these equalities we find that $vA = 0$ is equivalent with

$$\sum_{i=j}^{j+p-1} v_i(2\beta_i - 1) = 0,$$

for all $j = 1, \ldots, p+1$. Next, we subtract (5.18) with $j = l$ from (5.18) with $j = l+1,$
for $i = 1, \ldots, p$. This yields that the existence of a $u \in \mathbb{R}^p_+$ such that $uA = 0$ and
\[ \sum_{i=1}^{2p} u_i > 0, \] is equivalent with the existence of $v_i \geq 0$, $i = 1, \ldots, 2p$, not all zero, such that $v_i(2\beta_i - 1) = v_{i+p}(2\beta_{i+p} - 1)$, for all $i = 1, \ldots, p$, and $\sum_{i=1}^{2p} v_i(2\beta_i - 1) = 0$. One easily shows that such $u_i$'s exist, if and only if (5.16) does not hold, which completes the proof of the lemma. □

Below, we give an explicit expression for the weights of the 2LP that classifies a given classifiable subset of the form discussed in this section. Note that Theorem 5.5.4 alone does not prove that some of the radial subsets are unclassifiable, it only proves that for such subsets there does not exist a solution of the proposed form. If Theorem 5.5.4 is used in combination with Theorem 5.5.3, the result becomes much stronger, since this yields that the proposed solutions are the only ones that have to be considered.

Theorem 5.5.4. Let $p \in \mathbb{N}$, let $a_0, \ldots, a_{p+1} \in \mathbb{R}$, with $0 \leq a_0 < a_1 < \ldots < a_p < p$ and $a_{i+p} = a_i + p$, $i = 1, \ldots, p+1$, and let $\beta_i \in \mathbb{B}$, $i = 1, \ldots, 2p$. Let $a_1, \ldots, a_p \in \mathbb{Z}$ and $b_0, b_1, \ldots, b_p \in \mathbb{Z}$ be defined as
\[ a_i = 2(\beta_i - \beta_{i-1} - \beta_{i+p} + \beta_{i+p-1}), \quad i = 1, \ldots, p, \]
\[ b_i = 2(\beta_i - \beta_0 - \beta_{i+p} + \beta_p), \quad i = 0, 1, \ldots, 2p, \]
where we define $\beta_0 = \beta_{2p}$ and $\beta_{i+p} = \beta_i$, for all $i = 1, \ldots, p$. Then we have

(i) for all $a_0 \in \mathbb{R}$ that satisfy
\[ \max\{1 - b_i \mid \beta_i = 1, 1 \leq i \leq 2p\} \leq a_0 \leq \min\{1 - b_i \mid \beta_i = 0, 1 \leq i \leq 2p\}, \tag{5.19} \]
the function $f : \mathbb{R}^2 \to \mathbb{B}$ defined by (5.17) satisfies (5.15), and

(ii) there exists an $a_0 \in \mathbb{R}$ that satisfies (5.19), if and only if (5.16) holds.

Proof. The proof consists of two parts.

(i) Using the first part of the proof of Lemma 5.5.3 it follows that we must show that $Aa \geq 1$, where $A$ is a $(2p) \times (p+1)$ matrix that satisfies
\[ A_{ij} = \begin{cases} 2\beta_i - 1 & \text{if } j = 0 \text{ or } i - p < j \leq i, \\ 0 & \text{otherwise}, \end{cases} \]
for all $i = 1, \ldots, 2p$ and $j = 0, \ldots, p$. Using the definition of $a_i$ and $b_i$ above, one easily shows that $\sum_{j=1}^{p} a_j = b_k$ and $\sum_{j=k+1}^{2p} a_j = b_{k+p}$, for all $k = 0, \ldots, p$. Hence, it follows that
\[ (Aa)_i = (2\beta_i - 1)(a_0 + \sum_{j=\max(1,i+p+1)}^{\min(i,p)} a_j) \]
\[ = (2\beta_i - 1)(a_0 + b_i), \tag{5.20} \]
for all $i = 1, \ldots, 2p$. From (5.19) we obtain that $a_0 + b_i \geq 1$, if $\beta_i = 1$, and $a_0 + b_i \leq -1$, if $\beta_i = 0$. Together with (5.20) this proves $Aa \geq 1$.

(ii) There exists a solution $a_0 \in \mathbb{R}$ of (5.19), if and only if $b_i - b_{i'} \geq 2$, or equivalently,
\[ \beta_i - \beta_{i+p} = \beta_i - \beta_{i+p} \geq 1, \]
for all $i, i' \in \{1, \ldots, 2p\}$ with $\beta_i = 1$ and $\beta_{i'} = 0$. One easily verifies that this is the case, if and only if (5.16) holds. □
Figure 5.10: An example demonstrating the use of Theorem 5.5.4. It shows the subset of Figure 5.9a, together with its \( \beta_i \)-values, \( i = 0, 1, \ldots, 14 \), the angles given by \( \alpha_1 \) and \( \alpha_2 \), and the values of \( a_i \) and \( b_i \), \( i = 0, 1, \ldots, 14 \), corresponding to the solution given by Theorem 5.5.4.

The use of Theorem 5.5.4 is demonstrated in the following example. Consider the subset \( V \subseteq \mathbb{R}^2 \) presented in Figure 5.10, which is a copy of the subset of Figure 5.9a. The values \( \beta_i = r \cos \varphi, r \sin \varphi = \beta_i, \alpha_i < \varphi < \alpha_{i+1} \), are shown for \( i = 1, \ldots, 14 \), and \( \beta_0 = \beta_{14} \). The values of \( a_i \), with \( a_1 = 0 \), can easily be obtained from Figure 5.10, if wanted. Using Theorem 5.5.4, one calculates \( a_1, a_2, \ldots, a_7 \) and \( b_0, b_1, \ldots, b_4 \); see the table in Figure 5.10. Then (5.19) yields \( 3 \leq a_0 \leq 3 \), which proves that there exists a unique solution to the problem of finding a 2LP that classifies \( V \). This solution is given by (5.17), with \( a_0, a_1, \ldots, a_7 \) as given in Figure 5.10. Note that there exist subsets for which there is more than one solution of the proposed form.

Using the results of this section, we can make the first step in proving that the subsets of Figure 5.6 are classifiable with a 2LP. Consider the subsets presented in Figure 5.11. From Theorem 5.5.3 it is clear that the subsets of Figures 5.11a and b are classifiable with a 2LP. To turn them into the corresponding subsets of Figure 5.6, we have to remove the augmentation, which is done in two steps in Section 5.5.3 and Section 5.5.4, respectively. The proof that the subset in Figure 5.11c is classifiable with a 2LP is somewhat more complicated and is therefore treated in detail in Section 5.5.4.

In the following section we study general sufficient conditions for the classifiability of subsets with a 2LP.

5.5.3 Sufficient conditions

In Section 5.3 we showed that \( \hat{P} \subseteq C_2 \subseteq \hat{U} \). In Section 5.5.1 we derived a set of necessary conditions which proves that \( C_2 \subseteq \hat{U} \). In this section and the following sec-
tion we derive a set of sufficient conditions which shows that $\hat{P} \subseteq C_2$. The sufficient conditions are of a totally different form than the necessary conditions. Roughly speaking, the necessary conditions concern local geometrical properties, while the sufficient conditions concern global geometrical properties of the examined subsets. This difference of form alone makes it difficult to compare the sufficient and the necessary conditions. However, we demonstrated in Section 5.5.1 that the presented necessary conditions are not sufficient, which implies that they cannot match the sufficient conditions that are derived in the forthcoming sections. Thus in general there is still a gap between the necessary and sufficient conditions. On the other hand we have shown in Section 5.5.2 that the necessary conditions can be sufficient for certain classes of subsets to be classifiable with a 2LP; see Theorem 5.5.3. Although we cannot show in general that the sufficient conditions are necessary, we show that the necessary and sufficient condition given in Theorem 5.5.3 is a special case of one of the general sufficient conditions derived; see Section 5.5.4. Furthermore, we show that the sufficient conditions can be used to prove that the approximating subsets, discussed at the end of Section 5.5.1, are indeed classifiable with a 2LP.

The sections that are concerned with the sufficient conditions are organized as follows. In this section we start with a general sufficient condition for the classifiability of an intersection of two subsets that are both classifiable with a 2LP. As a result we identify a special class of subsets, subsets which can be intersected with any classifiable subset to form a new classifiable subset. Furthermore, this class contains a subclass of subsets, which we name the basic subsets, that can also be unified with any classifiable subset to form a new classifiable subset. We show that the basic subsets can be used as building blocks to construct arbitrary complex classifiable subsets. Although most of the sufficient conditions are of an existential nature, we also give versions that present explicit values of the weights of the classifying 2LPs. Section 5.5.4 considers the basic subsets in great detail and yields a full characterization of the one-, two-, and three-dimensional basic subsets. Finally, Section 5.6 discusses the question of decomposing a given subset into a number of basic subsets, in order to determine whether the given subset is classifiable with a 2LP, and presents a decomposition algorithm for a certain restricted collection of subsets.
Our approach for finding sufficient conditions is of a constructive nature. Given two sets that can be classified with 2LPs, we search for conditions that guarantee that their intersection or their union is also classifiable with a 2LP. The motivation for this approach is that since $C_2 \subset \bar{U}$, we only have to consider subsets in $\bar{U}$, which are unions of subsets in $\bar{P} \subset C_2$. Note that since $V \in C_2$, if and only if $V^* \in C_2$, it does not matter whether one studies intersections or unions, since results obtained for unions are easily translated into results for intersections and vice versa.

As in Section 5.5.1, we start with a remark essentially about 1LPs. By definition the hyperplane $ax + b = 0$ separates the two disjoint sets $\Omega_1, \Omega_0 \subset \mathbb{B}^N$, if $ax + b \geq 0$, for all $x \in \Omega_1$, and $ax + b < 0$, for all $x \in \Omega_0$. We noted in Chapter 2 that $\Omega_1$ and $\Omega_0$ can be separated, if and only if there exist $a \in \mathbb{R}^N$ and $b \in \mathbb{R}$ such that $ax + b \geq 1$, for all $x \in \Omega_1$, and $ax + b \leq -1$, for all $x \in \Omega_0$; see for instance the proof of Theorem 2.2.1. This has the following consequence for classification with 2LPs.

**Proposition 5.5.2.** Let $V \subseteq \mathbb{R}^N$. Then $V \in C_{2,N}$, if and only if there exists an $f \in R_{2,2,1}$ such that $V = \mathcal{J}(f)$ and $f = \theta \circ f$ with $f(x) \notin (-1,+1)$, for all $x \in \mathbb{R}^N$.

In the major part of this section, we assume that $f(x) \notin (-1,+1)$, for all considered 2LPs represented by $f = \theta \circ f$. Consequently, if $V = \mathcal{J}(\theta \circ f)$, then $V^* = \mathcal{J}(\theta \circ (-f))$, which is used frequently in the remainder of this section. We apply Proposition 5.5.2 in the following theorem, which gives a sufficient condition for the intersection of two classifiable subsets to be classifiable with a 2LP. As usual, the maximum and minimum over an empty set are defined to be $-\infty$ and $+\infty$, respectively.

**Theorem 5.5.5.** Let $V_1, V_2 \in C_2$, $V_i = \mathcal{J}(f_i)$, with $f_i \in R_2$, $f_i = \theta \circ \tilde{f}_i$, $\tilde{f}_i(x) \notin (-1,+1)$, $i = 1, 2$. Let $P \leq Q$, where $P$ and $Q$ are given by

$$P = \max \left\{ \frac{1 + \tilde{f}_i(x)}{1 - \tilde{f}_i(x)} \right\} \text{ if } \tilde{f}_i(x) \leq -1, \tilde{f}_i(x) \geq +1, x \in \mathbb{R}^N, \quad (5.21)$$

$$Q = \min \left\{ \frac{1 + \tilde{f}_i(x)}{1 - \tilde{f}_i(x)} \right\} \text{ if } \tilde{f}_i(x) > +1, \tilde{f}_i(x) \leq -1, x \in \mathbb{R}^N, \quad (5.22)$$

respectively. Then $V_1 \cap V_2 \in C_2$.

**Proof.** If $P = -\infty$, then $V_1^* \cap V_2 = \{x \in \mathbb{R}^N | \tilde{f}_i(x) \leq -1, \tilde{f}_i(x) \geq +1\} = \emptyset$, and hence, $V_1 \cap V_2 = V_2 \in C_2$.

Otherwise, there exists an $\alpha \in \mathbb{R}$ such that $0 < P \leq \alpha \leq Q$. Let $f(x) = \alpha(\tilde{f}_i(x) - 1) + \tilde{f}_i(x)$, for all $x \in \mathbb{R}^N$. We show that $V_1 \cap V_2 = \mathcal{J}(\theta \circ f)$, by considering the following cases.

Let $x \in V_1 \cap V_2$, then $\tilde{f}_i(x) \geq 1$ and $\tilde{f}_i(x) \geq 1$. Hence, since $\alpha > 0$, we obtain $f(x) \geq 1$.

Let $x \in (V_1 \cap V_2)^*$, then either $\tilde{f}_i(x) \leq -1$ or $\tilde{f}_i(x) \leq -1$. If $\tilde{f}_i(x) \leq -1$, then since $\alpha > 0$, it follows that $\tilde{f}_i(x) \leq -1$. If $\tilde{f}_i(x) \geq +1$, then since $\alpha \leq Q$, it follows that $\tilde{f}_i(x) \leq +1$, if $\tilde{f}_i(x) \geq +1$ and $\tilde{f}_i(x) \leq -1$, then since $\alpha \geq P$, it follows that $f(x) \leq -1$. □

In the following example we demonstrate the use of Theorem 5.5.5. In Figures 5.12a and b, two subsets of $\mathbb{R}^2$ are shown. In Figures 5.13a and b, two functions $f_1 =$
θ ⊘ f₁ and f₂ = θ ⊘ f₂, are defined that correspond to two 2LPs that classify these subsets. In Figures 5.12d and e, the values of the two functions f₁ and f₂ are given. Theorem 5.5.5 can now be used to show that the intersection of the two subsets is classifiable with a 2LP. To this end one calculates P and Q, as defined by (5.21) and (5.22), with the help of the Figures 5.12d and e. Note that to be exact these figures must be superimposed. This yields that P = 2 and Q = 2, respectively, which proves that the intersection of the two subsets is indeed classifiable with a 2LP. From the proof of Theorem 5.5.5 we know that the 2LP corresponding to the function f = θ ⊘ f₁ with f defined by f = 2(f₁ − 1) + f₂, classifies the intersection of the two subsets; see Figure 5.13c for an explicit expression for f. The classification of the intersection by f can be verified with the help of Figure 5.12c, which shows the intersection of the two subsets, and Figure 5.12f, which gives the values of the function f.

![Diagrams showing the classification of subsets](image)

Figure 5.12: Construction of a classifiable subset using Theorem 5.5.5; see the text for the details.

It is clear from this example that the verification of the condition given in Theorem 5.5.5 can in general be time consuming. Furthermore, one should note that Theorem 5.5.5 is not symmetric, i.e., in some cases the order of the two subsets V₁, V₂ in which they appear in the intersection V₁ ∩ V₂, influences whether or not the theorem decides that the intersection is classifiable with a 2LP. Therefore, we consider a non-trivial special case in which the verification is much easier. There are two easy cases for P ≤ Q to be satisfied. The first case is P = −∞, corresponding to the trivial case where V₁^c ∩ V₂ = ∅, which we already examined in the proof of Theorem 5.5.5. The second case is Q = +∞, and corresponds to the situation that \( \{ x \in \mathbb{R}^N \mid f₁(x) > +1, f₂(x) \leq -1 \} = \emptyset \). Obviously, this is satisfied if f₁(x) ≤ +1,
5.5 Characterizations of $C_2$

<table>
<thead>
<tr>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1(x) = \theta(f_1(x))$</td>
<td>$f_2(x) = \theta(f_2(x))$</td>
<td>$f(x) = \theta(f(x))$</td>
</tr>
<tr>
<td>$f_1(x) = 2h_1(x) + 2h_3(x) - 1$</td>
<td>$f_2(x) = 4h_3(x) + 4h_4(x) + 2h_5(x) + 2h_6(x) - 9$</td>
<td>$f(x) = 2(f_1(x) - 1) + f_2(x)$</td>
</tr>
<tr>
<td>$h_1(x) = \theta(h_1(x))$ (i = 1, 2)</td>
<td>$h_2(x) = \theta(h_2(x))$ (i = 3, ..., 6)</td>
<td>$h_1(x) = \theta(h_1(x))$ (i = 1, ...)</td>
</tr>
<tr>
<td>$h_3(x) = -2x_1 + 3x_2 - 3$</td>
<td>$h_4(x) = 4x_1 - 4x_2 - 1$</td>
<td>$h_1(x) = -2x_1 + 3x_2 - 3$</td>
</tr>
<tr>
<td>$h_5(x) = -3x_1 + x_2 + 9$</td>
<td>$h_6(x) = 4x_1 + 6x_2 - 21$</td>
<td>$h_5(x) = -2x_3 + 3$</td>
</tr>
<tr>
<td>$h_7(x) = -6x_1 - 2x_2 + 27$</td>
<td>$h_8(x) = -6x_2 - 2x_2 + 27$</td>
<td>$h_8(x) = 4x_1 + 6x_2 - 21$</td>
</tr>
</tbody>
</table>

Figure 5.13: Representations of the 2LPS that classify the corresponding subsets presented in Figure 5.12.

for all $x \in \mathbb{R}^N$. The subclass of functions that can be computed by a 2LP and have this property, is given a special name below, as well as two related subclasses.

**Definition 5.5.2.** The subclasses $R_{2,N,1}^+$, $R_{2,N,1}^-$, and $R_{2,N,1}^B$ of $R_{2,N,1}$ are defined by

- $R_{2,N,1}^+ = \{ f \in R_{2,N,1} | f = \theta \circ \hat{f}, \hat{f}(x) \in (-\infty, -1] \cup \{+1\}, x \in \mathbb{R}^N \}$,
- $R_{2,N,1}^- = \{ f \in R_{2,N,1} | f = \theta \circ \hat{f}, \hat{f}(x) \in [-1, +\infty), x \in \mathbb{R}^N \}$,
- $R_{2,N,1}^B = \{ f \in R_{2,N,1} | f = \theta \circ \hat{f}, \hat{f}(x) \in [-1, +1], x \in \mathbb{R}^N \}$,

respectively.

Thus, $R_{2,N,1}^+$ is the set of functions representing 2LPs of the form $\theta \circ \hat{f}$, for which $\hat{f}$ attains at most one positive value. The set $R_{2,N,1}^B$ corresponds to the set of basic functions. The reason for these functions to be called basic becomes clear below. As with $R_{2,N,1}$, we usually abbreviate $R_{2,N,1}^+$, $R_{2,N,1}^-$, and $R_{2,N,1}^B$ to $R_2^+$, $R_2^-$, and $R_2^B$, respectively. The corresponding subclasses $C_{2,N}^+$, $C_{2,N}^-$, and $C_{2,N}^B$, are defined as follows.

**Definition 5.5.3.** The subclasses $C_{2,N}^+$, $C_{2,N}^-$, and $C_{2,N}^B$ of $C_{2,N}$ are defined by

- $C_{2,N}^+ = \{ f(x) | f \in R_{2,N,1}^+ \}$,
- $C_{2,N}^- = \{ f(x) | f \in R_{2,N,1}^- \}$,
- $C_{2,N}^B = \{ f(x) | f \in R_{2,N,1}^B \}$,

respectively.

Again, we usually abbreviate $C_{2,N}^+$, $C_{2,N}^-$, and $C_{2,N}^B$ to $C_2^+$, $C_2^-$, and $C_2^B$, respectively. It is obvious that $R_{2}^B \subseteq R_2^+ \cap R_2^-$ and $C_{2,N}^B \subseteq C_{2,N}^+ \cap C_{2,N}^-$. We think that equality holds here, but we do not know of any proof. The collection $C_{2,N}^B$ corresponds to the collection of basic subsets. Below we first give some properties of the introduced subsets, and secondly, characterize the subsets that are contained in $C_2^+$ and $C_2^B$. 
The first result is a direct consequence of Theorem 5.5.1. It is this result that is the reason for introducing the above collections of subsets.

**Corollary 5.5.4.** Let \( V_1 \in C^+_2 \) and \( V_2 \in C_2 \). Then \( V_1 \cap V_2 \in C_2 \).

Next, we show that if we intersect two subsets of \( C^+_2 \) we obtain another subset of \( C^+_2 \), which proves that \( C^+_2 \) is closed under taking the intersection.

**Theorem 5.5.6.** Let \( V_1 \in C^+_2 \) and \( V_2 \in C^+_2 \). Then \( V_1 \cap V_2 \in C^+_2 \).

**Proof.** Let \( V_i = F(f_i) \), \( f_i \in R^+_2 \), \( f_i = \theta \circ \tilde{f}_i \), and \( \tilde{f}_i(x) \in (-\infty, -1] \cup \{+1\} \) for all \( x \in R^N \), \( i = 1, 2 \). Define \( f : R^N \to R \) by

\[
\tilde{f}(x) = \tilde{f}_1(x) + \tilde{f}_2(x) - 1,
\]

for all \( x \in R^N \). Then one easily verifies that \( f = \theta \circ \tilde{f} \in R_2 \), \( \tilde{f}(x) \in (-\infty, -1] \cup \{+1\} \) for all \( x \in R^N \), and \( V_1 \cap V_2 = F(f) \). \( \square \)

Repeated application of Theorem 5.5.6 and Corollary 5.5.4 yields the following two results.

**Corollary 5.5.5.** Let \( V_1, \ldots, V_L \in C^+_2 \), for some \( L \in \mathbb{N} \). Then \( \bigcap_{i=1}^L V_i \in C^+_2 \).

**Corollary 5.5.6.** Let \( V_1, \ldots, V_L \in C^+_2 \), for some \( L \in \mathbb{N} \). Then \( V_1 \setminus V_2 \setminus \cdots \setminus V_L \in C_2 \).

Note that we use \( V_1 \setminus V_2 \setminus \cdots \setminus V_L \) as a shorthand notation for \( V_1 \setminus (V_2 \setminus (\cdots (V_{L-1} \setminus V_L) \cdots)) \).

To prove Corollary 5.5.6, one combines Corollary 5.5.4 with the fact that \( V \in C_2 \), if and only if \( V^* \in C_2 \), by Lemma 5.3.3, and using that by definition

\[
V_1 \setminus V_2 \setminus \cdots \setminus V_L = V_1 \cap (V_2 \setminus V_3 \setminus \cdots \setminus V_L)^*.
\]

The following trivial result enables the translation of the results for \( C^+_2 \) given above into results for \( C^*_2 \).

**Proposition 5.5.3.** Let \( V \subseteq R^N \). Then \( V \in C^+_2 \), if and only if \( V^* \in C^*_2 \).

We mention two results that can be obtained using Proposition 5.5.3. For the first, it is combined with Corollary 5.5.4.

**Corollary 5.5.7.** Let \( V_1 \in C^+_2 \) and \( V_2 \in C_2 \). Then \( V_1 \cup V_2 \in C_2 \).

Combining Proposition 5.5.3 with Corollary 5.5.5, we obtain the following result.

**Corollary 5.5.8.** Let \( V_1, \ldots, V_L \in C^*_2 \), for some \( L \in \mathbb{N} \). Then \( \bigcup_{i=1}^L V_i \in C^*_2 \).

Using that \( C^*_2 \) is a subclass of both \( C^+_2 \) and \( C^*_2 \), and combining this with the Corollaries 5.5.5 and 5.5.8, gives the following result for basic subsets.

**Corollary 5.5.9.** Let \( V_1, \ldots, V_L \in C^*_2 \), for some \( L \in \mathbb{N} \). Then \( \bigcap_{i=1}^L V_i \in C^*_2 \) and \( \bigcup_{i=1}^L V_i \in C^*_2 \).

Again we do not know if there are subsets in \( C^*_2 \) that cannot be written as the intersection of a finite collection of basic subsets. A similar remark can be made about the subsets in \( C^*_2 \). Some applications of the above results are shown in Figure 5.14. It presents three examples of subsets that can be shown to be classifiable with a 2LP, using some of derived sufficient conditions.
5.5 Characterizations of $C_2$

Figure 5.14: Three subsets that can be proved to be in $C_2$. For (a) this can be done using Theorem 5.5.5 and Lemma 5.5.3, for (b) using Corollary 5.5.6 and Corollary 5.5.10, i.e., using Corollary 5.5.11, and for (c) using Corollary 5.5.9 and Corollary 5.5.12, which is presented in Section 5.5.4.

The results of the Corollaries 5.5.4, 5.5.7 and 5.5.9 motivated us to name the basic functions and basic subsets, because they act as the basic building blocks in our proofs of the classifiability of a given subset with a 2LP. Obviously, before we can give such proofs we have to characterize the basic functions and basic subsets. This is the main subject of Section 5.5.4. Here we shortly anticipate on the results of that section and study the most elementary basic functions and basic subsets, which are based on 1LPs.

Consider a 1LP with one output, represented by a function $h \in R_{1,N,1}$. Since $h$ is a function from $R^N$ to $B$, it follows that $+2h(x) - 1 \in \{-1, +1\}$, and $-2h(x) + 1 \in \{-1, +1\}$, for all $x \in R^N$. This implies that the functions $f_1, f_2 \in R_{2,N,1}$ defined by $f_1(x) = \theta(2h(x) - 1)$, and $f_2(x) = \theta(-2h(x) + 1)$, $x \in R^N$, are basic functions. They correspond to 2LPs, with one node in the hidden layer, one output, and their second layer weights and biases given by $(+2, -1)$ and $(-2, +1)$, respectively. Using that $J(f_1) = J(h)$, $J(f_2) = J^*(h)$, and using Theorem 5.3.1, we have proved the following result.

**Theorem 5.5.7.** $\hat{H} \cup \{0, R^N\} \subseteq C_{2,N}^b$.

**Proof.** See above. □

The open and closed affine halfspaces, $\emptyset$, and $R^N$, form our first collection of basic subsets. In Section 5.5.4 we show that the halfspaces, $\emptyset$, and $R^N$, are not the only basic subsets, which implies that the inclusion of Theorem 5.5.7 is strict. In fact in Section 5.5.4 we rigorously derive a characterization of $C_{2,N}^b$, which we prove to be complete for $N = 1, 2, 3$. The elementary basic subsets identified in Theorem 5.5.7, can already be used in combination with the above results for the construction of subsets that can be classified with a 2LP. For instance, we directly obtain the following result using Corollary 5.5.9.

**Corollary 5.5.10.** $\hat{P} \subseteq C_{2}^+$. 

This result states that the pseudo polyhedrons are not only classifiable with a 2LP, as we already saw in Section 5.3, but they have a special property that enables them
to be combined with other classifiable subsets to form more complex classifiable subsets. To be more specific, Corollary 5.5.10 and Corollary 5.5.4 imply that the intersection of a pseudo polyhedron and a classifiable subset is again classifiable with a 2LP. As an example, we can use this result to transform the classifiable subsets of Figure 5.11 into the classifiable subsets of Figure 5.15, by intersecting them with the appropriate pseudo polyhedrons. This is the first step in the deletion of the augmentation of Figure 5.11. The second step of the deletion of the augmentation of Figure 5.11 uses Corollary 5.5.4 also, but it exploits other basic subsets that are discussed in Section 5.5.4, and is therefore treated in that section.

Figure 5.15: The first step in deleting the augmentation of the subsets of Figure 5.11 in order to obtain the subsets of Figure 5.6.

A second application of Corollary 5.5.10 is obtained by combining it with Corollary 5.5.6. This yields the following result.

**Corollary 5.5.11.** Let \( V_1, \ldots, V_L \in \tilde{P} \), for some \( L \in \mathbb{N} \). Then \( V_1 \setminus V_2 \setminus \cdots \setminus V_L \in C_2 \).

Note that this result implies that \( \tilde{P} \subset C_2 \). The result of Corollary 5.5.11 is extensively discussed in Section 5.6, which studies the decomposition of a given subset \( V \subseteq \mathbb{R}^N \) in the form \( V_1 \setminus V_2 \setminus \cdots \setminus V_L \), for some \( L \in \mathbb{N} \) and \( V_1, \ldots, V_L \in \tilde{P} \).

The sufficient conditions discussed in this section are of an existential nature, i.e., these conditions characterize a certain class of subsets that can be classified with a 2LP, without giving an explicit description of the classifying 2LP. For some of the discussed results, such an explicit description can be obtained straightforward from their proofs, see for instance Theorem 5.5.6, which is easily extended to Corollary 5.5.5 and Corollary 5.5.8. An explicit version of Corollary 5.5.4 can be obtained indirectly via the proof of Theorem 5.5.5. We end this section with an explicit version of Corollary 5.5.6; see Theorem 5.5.8. Before we give the explicit version of Corollary 5.5.6, we present an explicit version of an alternative of Corollary 5.5.4, which is used in the proof of Theorem 5.5.8; see Lemma 5.5.4. Both formulations make use of the fact that for subsets in \( C_{\ast,n}^* \), there exist \( K \in \mathbb{N}, \gamma \in \mathbb{N}, g = \theta \circ \bar{g} \in R_{t,K,1} \), and \( h \in R_{t,n,K} \), such that \( V = f(g \circ h) \) and \( \bar{g}(h(x)) \in (-\gamma, -\gamma + 1, \ldots, 0) \), for all \( x \in \mathbb{R}^N \), which follows straightforwardly from the definition of \( C_{\ast,n}^* \).

**Lemma 5.5.4.** Let \( V_1 \in C_{\ast}^* \) and \( V_2 \in C_2 \). Let \( K_1, K_2 \in \mathbb{N}, \gamma \in \mathbb{N}, \alpha \in \mathbb{N} \),
\[ \beta \in \mathbb{N}_0, \quad g_i = \theta \circ g_i \in R_{i,K_i \uparrow}, \quad i = 1, 2, \quad \text{and} \quad h_i \in R_{i,N,K_i}, \quad i = 1, 2, \quad \text{be such that} \]
\[ V_i = \mathcal{J}(g_i \circ h_i), \quad i = 1, 2, \quad \gamma_i = (h_i(x)) \in \{-\gamma_i - \alpha_i + 1, \ldots, 0\}, \quad \text{for all} \ x \in \mathbb{R}^N, \]
\[ g_i(h_i(x)) \in \{-\gamma_i - \alpha_i + 1, \ldots, \beta_i\}, \quad \text{for all} \ x \in \mathbb{R}^N. \]

Then \[ V_1 \setminus V_2 = \mathcal{J}(g \circ h), \quad \text{with} \]
\[ h \in R_{i,N,K_i \uparrow}, \quad \text{and} \quad g = \theta \circ \tilde{g} \in R_{i,K_i \uparrow}, \quad \text{given by} \quad h(x) = (h_1(x), h_2(x)) \]
\[ \tilde{g}(w_1, w_2) = \alpha \tilde{g}_1(w_1) - \tilde{g}_2(w_2) - 1, \quad (5.23) \]
respectively. Furthermore, we have that \[ \tilde{g}(h(x)) \in \{-\gamma - \beta - 1, \ldots, \alpha - 1\}, \quad \text{for all} \ x \in \mathbb{R}^N. \]

\[ \text{Proof.} \quad \text{If} \ x \notin V_1, \text{then} \quad \tilde{g}_1(h_1(x)) \leq -1 \quad \text{and hence}, \quad \tilde{g}(h(x)) \leq -\alpha - \tilde{g}_2(h_2(x)) - 1 \leq -1. \quad \text{If} \ x \in V_1, \text{then} \quad \tilde{g}_1(h_1(x)) = 0 \quad \text{and hence}, \quad \tilde{g}(h(x)) = -\tilde{g}_2(h_2(x)) - 1 \geq 0, \quad \text{if and only if} \ x \notin V_2. \]

It remains to show that \[ \tilde{g}(h(x)) \in \{-\gamma - \beta - 1, \ldots, \alpha - 1\}, \quad \text{for all} \ x \in \mathbb{R}^N, \]
which follows directly from (5.23). \[ \square \]

Using Lemma 5.5.4 we can prove an explicit version of Corollary 5.5.6.

**Theorem 5.5.8.** Let \( L \in \mathbb{N} \) and \( V_i \in C_f^+, \quad i = 1, \ldots, L \). For all \( i = 1, \ldots, L \), let \( K_i \in \mathbb{N}, \gamma_i \in \mathbb{N}, \quad g_i = \theta \circ g_i \in R_{i,K_i \uparrow}, \) and \( h_i \in R_{i,N,K_i}, \) be such that \( V_i = \mathcal{J}(g_i \circ h_i), \) and \( \tilde{g}_i(h_i(x)) \in \{-\gamma_i - \alpha_i + 1, \ldots, 0\}, \) for all \( x \in \mathbb{R}^N \). Let \( \alpha_0, \ldots, \alpha_L \in \mathbb{N} \) be recursively defined by \( \alpha_0 = 1, \alpha_i = \gamma_i, \) and \( \alpha_{L+1} = \gamma_L - \alpha_i + \alpha_i - 1, \) for \( i = 1, \ldots, L-1 \). Then \( V_1 \setminus V_2 \setminus \cdots \setminus V_L = \mathcal{J}(g \circ h), \) where \( h \in R_{i,N,K_i \uparrow}, \) and \( g = \theta \circ \tilde{g} \in R_{i,K_i \uparrow}, \) are given by \( h(x) = (h_1(x), \ldots, h_L(x)) \) and
\[ \tilde{g}(w_1, \ldots, w_L) = \sum_{i=1}^L (-1)^{i-1} \alpha_{L-i} \tilde{g}_i(w_i) - \frac{1}{2}((-1)^{L+i} + 1), \quad (5.24) \]
respectively. Furthermore, we have that \( \tilde{g}(h(x)) \in \{-\alpha_L, \ldots, \alpha_{L-1} - 1\}, \quad \text{for all} \ x \in \mathbb{R}^N. \)

**Proof.** The proof is by induction to \( L \in \mathbb{N}. \) The result obviously holds for \( L = 1. \)
Next, we assume that the result holds for some \( L \in \mathbb{N}, \) and show that it also holds for \( L + 1. \) The proof of the induction step is based upon Lemma 5.5.4.

Assume that the result holds for some \( L \in \mathbb{N}. \) Then, after (re-)numbering the variables from 2 to \( L + 1 \) instead of 1 to \( L, \) we have that \( V_{L+1} \setminus V_L \setminus \cdots \setminus V_2 = \mathcal{J}(g \circ h), \) where \( h \in R_{i,N,K_i \uparrow}, \) and \( g = \theta \circ \tilde{g} \in R_{i,K_i \uparrow}, \) are given by \( h(x) = (h_2(x), \ldots, h_{L+1}(x)) \) and
\[ \tilde{g}(w_2, \ldots, w_{L+1}) = \sum_{i=2}^{L+1} (-1)^{i-1} \alpha_{L+1-i} \tilde{g}_i(w_i) - \frac{1}{2}((-1)^{L+1} + 1), \quad (5.25) \]
respectively, with \( \tilde{g}(h(x)) \in \{-\alpha_L, \ldots, \alpha_{L-1} - 1\}, \quad \text{for all} \ x \in \mathbb{R}^N. \) The coefficients \( \alpha_i, \ i = 0, 1, \ldots, L, \) are recursively defined by \( \alpha_0 = 1, \alpha_i = \gamma_i, \) and \( \alpha_{i+1} = \gamma_i - \alpha_i + \alpha_i - 1, \) for \( i = 1, \ldots, L-1. \)
Next, we apply Lemma 5.5.4 to \( V_1 \setminus V_2 \setminus \cdots \setminus V_{L+1} = V_1 \setminus (V_2 \setminus \cdots \setminus V_{L+1}). \) Having \( \gamma = \gamma_1, \alpha = \alpha_L, \) and \( \beta = \alpha_{L-1} - 1, \) this yields that \( V_1 \setminus V_2 \setminus \cdots \setminus V_L = \mathcal{J}(e \circ f), \) where \( f \in R_{i,N,K_i \uparrow}, \) and \( e = \theta \circ \xi \in R_{i,K_i \uparrow}, \) are given by \( f(x) = (h_1(x), h(x)) = (h_1(x), h_2(x), \ldots, h_{L+1}(x)) \) and
\[ \xi(w_1, w_2, \ldots, w_{L+1}) = \alpha \xi_1(w_1) - \xi_2(w_2, \ldots, w_{L+1}) - 1 \]
\[ \alpha_{L+1} \hat{g}(w_i) - \sum_{i=2}^{L+1} (-1)^i \alpha_{L+1-i} \hat{g}(w_i) + \frac{1}{2}((-1)^L + 1) - 1 \]

respectively. Furthermore, we have that

\[ \hat{e}(f(x)) \in \{ -\gamma_0 - \beta - 1, \ldots, \alpha - 1 \} \]

for all \( x \in \mathbb{R}^N \). By defining \( \alpha_{L+1} = \gamma_1 \alpha_L + \alpha_{L-1} \), we have proved the result for \( L+1 \), which completes the proof of the theorem. \( \square \)

Both Theorem 5.5.8 and Lemma 5.5.4 are based on the knowledge of a bound for \( \hat{g}(h(x)) \), \( x \in \mathbb{R}^N \), for those \( g = \theta \circ \hat{g} \in \mathcal{R}_{1, K, 1} \), and \( h \in \mathcal{R}_{1, N, K} \), such that \( \mathcal{J}(g \circ h) = V \), where \( K \in \mathbb{N} \) and \( V \) is an arbitrary subset in \( \mathcal{C}_+^K \). We do not know of a general approach for calculating this bound. However, if \( V \in \mathcal{C}_+^K \) is constructed using Corollary 5.5.9, then a bound can be obtained straightforwardly, as follows from the following result.

Proposition 5.5.4. Let \( L, V \in \mathcal{N} \) and \( V_i \in \mathcal{C}_+^K \), \( i = 1, \ldots, L \). Then there exist \( K \in \mathbb{N} \), \( g = \theta \circ \hat{g} \in \mathcal{R}_{1, K, 1} \), and \( h \in \mathcal{R}_{1, N, K} \), such that \( \cap_{i=1}^L V_i = \mathcal{J}(g \circ h) \) and \( \hat{g}(h(x)) \in \{ \ldots, -L, -L+1, \ldots, 0 \} \), for all \( x \in \mathbb{R}^N \).

Proposition 5.5.4 applies in particular to subsets \( V \in \mathcal{P} \). Hence, using Theorem 5.5.8, one easily obtains an explicit version of Corollary 5.5.11.

This completes our study of sufficient conditions for subsets to be classifiable with a 2LP. In the following section we study the basic subsets in detail. As a result we obtain a complete characterization of the basic subsets in \( \mathbb{R}^1 \), \( \mathbb{R}^2 \) and \( \mathbb{R}^3 \).

5.5.4 Basic functions and basic subsets

In this section we study the basic functions and basic subsets. This is done as follows. First, we consider the one-dimensional basic subsets. We show that all one-dimensional subsets in \( \mathcal{U} \) are basic. Secondly, we derive a set of conditions for the second layer weights of a 2LP with one input, such that this 2LP represents a basic function. This result is then used to obtain a similar result for a 2LP with two inputs, which we use to identify the two-dimensional basic subsets. The results are also used to give an alternative and extended proof of the sufficiency part of Theorem 5.5.3. Furthermore, we show that all three-dimensional basic functions are essentially two-dimensional, and finally, we conjecture the same for higher-dimensional basic functions, i.e., we conjecture that \( N \)-dimensional basic functions are essentially two-dimensional. We conclude this section with the last step of the proof that the subsets of Figure 5.6 are classifiable with a 2LP.

The following theorem shows that every one-dimensional subset that is classifiable with a 3LP is a basic subset, and thus certainly classifiable with a 2LP.

Theorem 5.5.9. Let \( V \subseteq \mathcal{R} \). Then \( V \in \mathcal{C}_{+1}^K \), if and only if \( V \subseteq \mathcal{U} \).

Proof. Since \( \mathcal{C}_+^K \subseteq \mathcal{C}_{+1}^K \subseteq \mathcal{U} \), it remains to prove that \( \mathcal{U} \subseteq \mathcal{C}_+^K \).
Let \( V \subseteq \mathbb{R}, V \subseteq \hat{U}, \) then the proof is completed if we show that there exists an \( f \in R_{2,1,1}^g \) such that \( V = \mathcal{F}(f) \), or equivalently, \( f = \chi_V \), where \( \chi : \mathbb{R} \to \mathcal{B} \) is the characteristic function of \( V \) defined by
\[
\chi_V(x) = \begin{cases} 
1 & \text{if } x \in V, \\
0 & \text{if } x \not\in V.
\end{cases}
\]

Since \( V \subseteq \hat{U} \), there exist \( K \in \mathbb{N}_0 \) and \( d_1, \ldots, d_K \in \mathbb{R} \), with \( d_1 < d_2 < \cdots < d_K \), such that \( \chi_V \) is constant on every interval \((d_j, d_{j+1})\), \( j = 0, \ldots, K \), with \( d_0 = -\infty \) and \( d_{K+1} = +\infty \).

Choose \( x_i \in (d_j, d_{j+1}) \), \( i = 0, \ldots, K \), let \( g(x) = 2\chi_V(x) - 1, x \in \mathbb{R} \), and let \( f = \theta \circ \hat{f} \), where \( \hat{f} \) is defined by
\[
\hat{f}(x) = a_0 + \sum_{i=1}^{K} \left\{ a_{2i-1}(x - d_i) + a_{2i}(1 - \theta(-x + d_i)) \right\},
\]
for all \( i = 1, \ldots, K \).

Let \( x \in (d_j, d_{j+1}) \) for some \( j \in \{0, \ldots, K\} \). Then it follows from (5.26) that
\[
\hat{f}(x) = a_0 + \sum_{i=1}^{j} a_{2i-1} + \sum_{i=1}^{j} a_{2i} = a_0 + \sum_{i=1}^{2j} a_i.
\]

Similarly, it follows for all \( j = 1, \ldots, K \), that
\[
\hat{f}(d_j) = a_0 + \sum_{i=1}^{j} a_{2i-1} + \sum_{i=1}^{j} a_{2i} = a_0 + \sum_{i=1}^{2j-1} a_i.
\]

Next, using induction on \( j \in \{0, \ldots, K\} \), one obtains from (5.27), (5.28), and (5.29), that \( \hat{f}(x_j) = \hat{f}(x) = g(x_j) = g(x) \), for all \( x \in (d_j, d_{j+1}) \) and \( j = 0, \ldots, K \). Consequently, using (5.30), combined with (5.29) and (5.28), we also obtain
\[
\hat{f}(d_j) = a_0 + \sum_{i=1}^{j} a_i - a_{2j}
\]
\[
= \hat{f}(x_j) - a_{2j}
\]
\[
= g(x_j) - (g(x_j) - g(d_j))
\]
\[
= g(d_j),
\]
for all \( j = 1, \ldots, K \). Thus \( \hat{f}(x) = g(x) \in \{-1, +1\} \), for all \( x \in \mathbb{R} \), which proves that \( f \in R_{2,1,1}^g \), and proves that \( f(x) = \theta(\hat{f}(x)) = \theta(g(x)) = \theta(2\chi_V(x) - 1) = \chi_V(x) \), for all \( x \in \mathbb{R} \). 

The sufficiency part of Theorem 5.5.9 can be viewed as a special case of Corollary 5.5.12, which is given below. This result characterizes a collection of basic subsets in arbitrary dimension that are based on a number of parallel hyperplanes and can be proved by making small adjustments to the proof of the sufficiency part of Theorem 5.5.9. Note that Theorem 5.5.7, given at the end of Section 5.5.3, is also a special case of Corollary 5.5.12.

**Corollary 5.5.12.** Let \( V \subseteq \mathbb{R}^N, V = \bigcup_{i \in \mathcal{H}} \cap_{i \in \mathcal{L}} W_{ij}, \) for some \( W_{ij} \in \mathcal{H}, K, L_i \in \mathbb{N}, \) such that \( W_{ij} \cap W_{ij'} = \emptyset \) or \( W_{ij} = W_{ij'}, \) for all \( i, i' \in \{1, \ldots, K\}, j \in \{1, \ldots, L_i\}, \) and \( j' \in \{1, \ldots, L_{ij'}\}. \) Then \( V \in C_{2,N}. \)
Figure 5.16: Typical examples of basic subsets in $\mathbb{R}^2$. The subset shown in (a) corresponds to a two-dimensional example of Corollary 5.5.12, and the subset shown in (b) corresponds to an example of Theorem 5.5.12. As always, thick boundary lines belong to the subsets and thin boundary lines do not belong to the subsets.

A two-dimensional example of a basic subset of the kind of Corollary 5.5.12 is given in Figure 5.16a.

Another way of expressing the relation between Theorem 5.5.9 and Corollary 5.5.12 is based on the following observation. If $f \in R_{a,b}^N$ is a one-dimensional basic function and $c \in \mathbb{R}^N$ is an arbitrary $N$-dimensional vector, then $g(u) = f(c \cdot u)$ is an $N$-dimensional basic function. Obviously, this approach applied to the basic functions appearing in the proof of Theorem 5.5.9 yields the basic functions corresponding to the basic subsets of Corollary 5.5.12. In general we have the following result, which says that a basic function preceded by an affine transformation is again a basic function. Obviously, this follows directly from Proposition 5.3.1 and Definition 5.5.2, the definition of basic functions.

**Proposition 5.5.5.** Let $f \in R_{a,b}^N$, let $K \in \mathbb{N}$, let $B$ be a $K \times N$ matrix, let $x_0 \in \mathbb{R}^N$, and let the function $g : \mathbb{R}^N \rightarrow B$ be defined by $g(u) = f(x_0 + uB)$, for all $u \in \mathbb{R}^K$. Then $g \in R_{a,b}^{1,1}$.

This result can be used in two ways. Firstly, it can be used upward, i.e., with $K > N$, to construct higher dimensional basic functions from lower dimensional basic functions, similarly as we did above. Secondly, it can be used downward, i.e., with $K < N$, as follows. If we have a set of conditions for lower dimensional basic functions, we can use it to obtain a set of conditions for higher dimensional basic functions, by constructing a lower dimensional basic function from the considered higher dimensional basic function, and applying the conditions for these functions. In the remainder of this section we use both the approaches for finding and characterizing the two- and three-dimensional basic functions, respectively. To be able to derive a set of conditions for the two-dimensional basic functions, we first need to derive a set of conditions for the one-dimensional basic functions, which is done in the following theorem.
Theorem 5.5.10. Let $K \in \mathbb{N}_0$, let $a_0, a_1, \ldots, a_{2K} \in \mathbb{R}$, let $d_1 < d_2 < \cdots < d_K$, and let $f = \theta \circ f$, where $\theta$ is defined by

$$
\hat{f}(x) = a_0 + \sum_{i=1}^{K} \{a_{2i-1}\theta(x - d_i) + a_{2i}(1 - \theta(-x + d_i))\}.
$$

(5.31)

Then $f \in R_{2,1,1}^R$ if and only if the following conditions are satisfied.

(i) $a_0 \in \{-1, +1\}$.

(ii) There exist $t \in \{0, 1, \ldots, 2K\}$ and $t_0, t_1, \ldots, t_{t+1} \in \mathbb{N}_0$ with $0 = t_0 < t_1 < \cdots < t_{t+1} = 2K + 1$, such that

$$
a_i = \begin{cases} 
0, & \text{if } t_i < i < t_{i+1} \text{ and } s = 0, \ldots, t, \\
2(-1)^s a_0, & \text{if } i = t_s \text{ and } s = 1, \ldots, t.
\end{cases}
$$

Proof. Let $d_0 = -\infty, d_{K+1} = +\infty$, and $x_j \in (d_j, d_{j+1})$, for all $j = 0, \ldots, K$. Then, similarly as in the proof of Theorem 5.5.9, it follows that

$$
\hat{f}(x_j) = a_0 + \sum_{i=1}^{j} a_i,
$$

$$
\hat{f}(d_j) = a_0 + \sum_{i=1}^{j-1} a_i,
$$

for all $j = 0, \ldots, K$ and all $j = 1, \ldots, K$, respectively. Since it is obvious that $f \in R_{2,1,1}^R$, we find that $f \in R_{2,1,1}^R$ if and only if $a_0 + \sum_{i=1}^{t+1} a_i \in \{-1, +1\}$, for all $j = 0, \ldots, 2K$.

The 'if'-part now directly follows from $a_i = 0$, for $t_i < i < t_{i+1}$, and, hence,

$$
a_0 + \sum_{i=0}^{t_i} a_i = a_0 + \sum_{s=1}^{t} \{2(-1)^s a_0\}
$$

$$
= a_0 + (\{-1\} - \{-1\}^{s+1})a_0
$$

$$
= (-1)^s a_0
$$

$$
\in \{-1, +1\},
$$

for all $s = 0, \ldots, t$.

To prove the 'only-if'-part, one applies induction to $j \in \{0, \ldots, 2K\}$, together with the fact that $a \in \{-1, +1\}$, $a + b \in \{-1, +1\}$, and $b \neq 0$, implies that $b = -2a$. □

Next, we turn our attention towards two-dimensional basic functions and basic subsets. As already noted, one can obtain a two-dimensional basic function as follows. Take a one-dimensional basic function $f \in R_{2,1,1}^R$ and a two-dimensional vector $c \in \mathbb{R}^2$. Then the function defined by $f(c \cdot u)$, for all $u \in \mathbb{R}^2$, is a two-dimensional basic function. The corresponding basic subsets are of the kind of Corollary 5.5.12 and consist of parallel lines that form the borders of a number of strips. The borders can be arbitrarily chosen to be included or not; see Figure 5.16a for an example.

In the following three theorems we consider two-dimensional basic functions that correspond to basic subsets that are radial around a certain center point, which implies that all the border lines pass through that point; see Figure 5.16b for an example. For notational convenience this center point is chosen to be $(0, 0)$, but it follows from Proposition 5.5.5 that this is not a restriction. Firstly, Theorem 5.5.11
gives a set of necessary and sufficient conditions for the second layer of weights of a
2LP, such that the corresponding function is a two-dimensional basic function. In the
proof of this theorem we make use of Theorem 5.5.10. Secondly, in Theorem 5.5.12,
we use the results of Theorem 5.5.11 to obtain a detailed characterization of the
corresponding basic subsets. Thirdly, a relaxed characterization of those kind of
basic subsets is given in Theorem 5.5.13. This result is used to give an alternative
proof of the sufficiency part of Theorem 5.5.3, which extends the previous proof in
that it gives more insight; see Theorem 5.5.14. Finally, we show that there are no
other two-dimensional basic subsets than the ones discussed. Thus either all border
lines are parallel or all border lines pass through one point, with the alternating
shading as indicated by Figure 5.16.

In the following theorem we pose some additional conditions for the set of considered
functions. These restrictions ensure that there are at least two distinct border lines
that pass through (0, 0), which simplifies the formulation of the theorem. We can
make these restrictions because the case that there is only one border line is already
reduced by Corollary 5.5.12.

**Theorem 5.5.11.** Let \( K \geq 2 \), let \( 0 \leq \alpha_1 < \alpha_2 < \cdots < \alpha_K < \pi \), let \( a_0, a_1, \ldots, a_{2K} \in \mathbb{R} \)
such that \( a_{2i-1} \neq 0 \) or \( a_{2i} \neq 0 \), \( i = 1, \ldots, K \), let \( h_i(x_1, x_2) = \theta(-\sin(\alpha_i) x_1 + \cos(\alpha_i) x_2) \), \( i = 1, \ldots, K \), and let \( \tilde{f} = \theta \circ f \), where \( f \) is given by

\[
\tilde{f}(x_1, x_2) = a_0 + \sum_{i=1}^{K} (a_{2i-1} h_i(x_1, x_2) + a_{2i}(1 - h_i(-x_1, -x_2))).
\]

Then \( f \in R_{2,2,1}^\theta \), if and only if the following conditions are satisfied.

(i) \( a_0 \in \{-1, +1\} \).

(ii) \( \forall i = 1, \ldots, K : a_{2i-1} = 0 \lor a_{2i} = 0 \).

(iii) \( \forall i = 1, \ldots, K : a_{2i-1} + a_{2i} = 2(-1)^i a_0 \).

(iv) \( K \) is odd.

(v) \( \sum_{i=1}^{K} a_{2i-1} = 0 \lor \sum_{i=1}^{K} a_{2i} = 0 \).

**Proof.** It is obvious that \( f \in R_{2,2,1} \), and thus it remains to prove that \( \tilde{f}(x_1, x_2) \in \{-1, +1\} \), for all \( (x_1, x_2) \in \mathbb{R}^2 \), if and only if the above conditions hold.

Because it holds that

\[
f(r \cos \varphi, r \sin \varphi) = \tilde{f}_{1, - \cdots, K}(r \cos \varphi, r \sin \varphi) = \tilde{f}_{1, a_1, \ldots, K}(r \cos(\varphi + \epsilon), r \sin(\varphi + \epsilon)),
\]

for all \( r, \varphi, \epsilon \in \mathbb{R} \), we may assume without loss of generality that \( a_1 > 0 \).

Let \( \tilde{f}_1, \tilde{f}_2 : \mathbb{R} \rightarrow \mathbb{R} \) be defined by \( \tilde{f}_1(x) = \tilde{f}(-x, 1) \) and \( \tilde{f}_2(x) = \tilde{f}(-x, -1) \), respectively. Then, because of the special form of \( \tilde{f} \), and because \( a_1 > 0 \), it follows that

(a) \( \tilde{f}(0, 0) \in \{-1, +1\} \), and

(b) \( \tilde{f}(z) \in (-1, +1) \), for all \( z \in \mathbb{R} \).

In order to be able to use Theorem 5.5.10, we define \( d_i = -\cos(\alpha_i)/\sin(\alpha_i) = \)
5.5 Characterizations of $C_2$

$- \cot(\alpha_i)$, for all $i = 1, \ldots, K$. Then one easily verifies that

$$\tilde{f}_1(x) = a_0 + \sum_{i=1}^{K} \{a_{2i-1} \theta(x - d_i) + a_{2i}(1 - \theta(-x + d_i))\}, \quad (53.3)$$

and

$$\tilde{f}_2(x) = a_0 + \sum_{i=1}^{K} \{a_{2i-1} \theta(x + d_i) + a_{2i}(1 - \theta(-x - d_i))\}$$

$$= a_0 + \sum_{i=1}^{K} \{a_{2K-2i+1} \theta(x + d_{K+i-1}) + a_{2K-2i+2}(1 - \theta(-x - d_{K+i-1}))\}, \quad (53.4)$$

respectively. Using that $0 < \alpha_1 < \alpha_2 < \cdots < \alpha_K < \pi$, it follows that $d_1 < d_2 < \cdots < d_K$, and, equivalently, $-d_K < -d_{K-1} < \cdots < -d_1$.

Next, we use Theorem 5.5.10 to prove that the conditions (i) through (v) are necessary and sufficient for (a) and (b) to hold.

We start with the 'sufficiency'-part, and assume that the conditions (i) through (v) are satisfied. To prove (a), we must show that $f(0, 0) = a_0 + \sum_{i=1}^{K} a_{2i-1} \in \{-1, +1\}$. If $\sum_{i=1}^{K} a_{2i-1} = 0$, this follows from (i). By (v) the only remaining possibility is that $\sum_{i=1}^{K} a_{2i} = 0$. This combined with (iii) gives

$$a_0 + \sum_{i=1}^{K} a_{2i-1} = a_0 + \sum_{i=1}^{K} \{a_{2i-1} + a_{2i}\}$$

$$= a_0 + \sum_{i=1}^{K} \{2(-1)^{i+1}a_0\}$$

$$= (-1)^{K+1}a_0$$

$$\in \{-1, +1\}.$$

To prove (b), we note that (iii) and (iv) imply that

$$a_{2K-2i+1} + a_{2K-2i+2} = 2(-1)^{K+i-1}a_0 = 2(-1)^ia_0, \quad (53.5)$$

for all $i = 1, \ldots, K$. Using (53.3), (i), (ii), and (iii), and using (53.4), (i), (ii), and (iii), one straightforwardly shows that $\theta \circ \tilde{f}_1$ and $\theta \circ \tilde{f}_2$, respectively, satisfy the conditions of Theorem 5.5.10. Thus $\theta \circ \tilde{f}_1 \in R^\theta_{ii,i,i}$ and $\theta \circ \tilde{f}_2 \in R^\theta_{ii,i,i}$, which yields (b).

Finally, we prove the 'necessary'-part. Using (a), it follows that

$$a_0 + \sum_{i=1}^{K} a_{2i-1} \in \{-1, +1\}. \quad (53.6)$$

Furthermore, from (b) it follows that $\theta \circ \tilde{f}_1, \theta \circ \tilde{f}_2 \in R^\theta_{1,i,i}$, which implies that we can use Theorem 5.5.10. Firstly, this yields $a_0 \in \{-1, +1\}$, which proves the necessity of (i). Secondly, we find that there exist $i \in \{0, 1, \ldots, 2K\}$ and $l_0, l_1, \ldots, l_{2K} \in \mathbb{N}_0$ with $0 = l_0 < l_1 < \cdots < l_{2K+1} = 2K + 1$, such that

$$a_i' = \begin{cases} 0, & \text{if } l_s < i < l_s + 1 \text{ and } s = 0, \ldots, t, \\ 2(-1)^{s+1}a_0, & \text{if } i = l_s \text{ and } s = 1, \ldots, t, \end{cases} \quad (53.7)$$

and, thirdly, we find that there exist $t' \in \{0, 1, \ldots, 2K\}$ and $l'_0, l'_1, \ldots, l'_{2K} \in \mathbb{N}_0$ with $0 = l'_0 < l'_1 < \cdots < l'_{2K+1} = 2K + 1$, such that

$$a'_i = \begin{cases} 0, & \text{if } l'_s < i < l'_{s+1} \text{ and } s = 0, \ldots, t', \\ 2(-1)^{s+1}a_0, & \text{if } i = l'_s \text{ and } s = 1, \ldots, t', \end{cases} \quad (53.8)$$
where \( a'_{2i-1} = a_{2K-2i+1} \) and \( a'_i = a_{2K-2i+2}, \ i = 1, \ldots, K \).

We proceed by showing the necessity of (ii). We therefore assume that (ii) does not hold and show that this leads to a contradiction. First, assume that \( a_{2i-1} \neq 0 \) and \( a_i \neq 0 \), for all \( i = 1, \ldots, K \). Then, by (5.37) it follows that \( a_{2i-1} = -2a_0 \), for all \( i = 1, \ldots, K \). Hence, using \( K > 2 \) and \( a_0 \in \{-1, +1\} \), we find that \( a_0 + \sum_{i=1}^{K} a_{2i-1} = (1 - 2K)a_0 \not\in \{-1, +1\} \), which contradicts (5.36).

Consequently, we may assume that either

- there exists an \( i \in \{1, \ldots, K - 1\} \) such that \( a_{2i-1} \neq 0, a_i \neq 0, \) and \( a_{2i+1} = 0 \) or \( a_{2i+2} = 0 \), or
- there exists an \( i \in \{2, \ldots, K\} \) such that \( a_{2i-1} \neq 0, a_i \neq 0, \) and \( a_{2i-3} = 0 \) or \( a_{2i-2} = 0 \).

We show that the first assumption leads to a contradiction. The same then holds for the second assumption by symmetry.

Assume that \( a_{2i-1} \neq 0, a_i \neq 0, \) and \( a_{2i+1} = 0 \) or \( a_{2i+2} = 0 \). Consider the case that \( a_{2i+1} = 0, \) the other case is treated similarly. By the assumptions of the theorem \( a_{2i+1} \neq 0 \). Hence, using (5.37), we obtain that \( a_{2i-1} = a_{2i+1} = -a_0 \in \{-2, +2\} \).

Furthermore, if we write these coefficients as \( a_{2K-2iK-1} \neq 0, a_{2K-2i+1} = 0, a_{2K-2iK+1} = 0, \) and \( a_{2K-2i+2} \neq 0 \), then using (5.38) we find that \( a_{2i+1} = a_i = -a_0, i \in \{-2, +2\} \). Obviously these two conclusions conflict.

Thus (ii) holds, which implies that for all \( i = 1, \ldots, K \), either \( a_{2i-1} = 0, \) or, exclusively, \( a_i = 0 \). Using (5.37) and (5.38) once more, we obtain

\[ \forall i = 1, \ldots, K : a_{2i-1} + a_i = 2(-1)^i a_0, \tag{5.39} \]

and

\[ \forall i = 1, \ldots, K : a_{2K-2i+1} + a_{2K-2i+2} = 2(-1)^i a_0. \tag{5.40} \]

It is obvious that (5.39) and (5.40) hold, if and only if (iii) and (iv).

Finally, we prove the necessity of (v). Assume \( \sum_{i=1}^{K} a_{2i-1} \neq 0, \). Then using (i) and (5.36) we obtain that \( \sum_{i=1}^{K} a_{2i-1} = -2a_0 \). From (iii) and (iv) it follows that

\[ \sum_{i=1}^{K} (a_{2i-1} + a_{2i}) = (-1)^K a_0 - a_0 = -2a_0, \]

which implies that \( \sum_{i=1}^{K} a_{2i} = 0 \).

In the following theorem the results of Theorem 5.5.11 are translated into a set of conditions for two-dimensional radial subsets to be basic. We may exclude the subsets corresponding to the whole \( \mathbb{R}^2 \), the emptyset, and the affine subspaces, because these subsets are already treated by Corollary 5.5.12. The lengthy formulation used in Theorem 5.5.12 for the characterization of the considered subsets is necessary, because the representation given by (i) through (iv) is not unique.

**Theorem 5.5.12.** Let \( V \subseteq \mathbb{R}^2 \) such that the following two conditions are satisfied.

(a) \( V \) is radial around \((0, 0)\), i.e., \( \lambda x \in V \), if and only if \( x \in V \), for all \( x \in \mathbb{R}^2 \) and \( \lambda > 0 \).

(b) There exists an \( x \in V \) for which \( -x \not\in V \).
Then \( V \) is a two-dimensional basic subset, i.e., \( V \subseteq C^2_{\xi} \), if and only if there exist \( K \in \mathbb{N},\ a_0, a_1, \ldots, a_{2K+1} \in \mathbb{R},\ \beta_0, \beta_1, \ldots, \beta_{2K} \in \mathbb{B},\ \gamma_1, \ldots, \gamma_{2K} \in \mathbb{B}, \text{ and } \xi \in \mathbb{B}, \) that satisfy the following conditions.

(i) \( 0 \leq \alpha_0 < \alpha_1 < \cdots < \alpha_K < \pi \text{ and } \alpha_{i+K} = \alpha_i + \pi, \ i = 0, \ldots, K + 1. \)

(ii) \( \forall r > 0 \ \forall i = 0, \ldots, 2K \forall \alpha_i < \varphi < \alpha_{i+1} : \chi_{V}(r \cos \varphi, r \sin \varphi) = \beta_i. \)

(iii) \( \forall r > 0 \ \forall i = 1, \ldots, 2K : \chi_{V}(r \cos(\alpha_i), r \sin(\alpha_i)) = \gamma_i. \)

(iv) \( \chi_{V}(0, 0) = \xi. \)

(v) \( \forall i = 1, \ldots, 2K : \beta_{i-1} + \beta_i = 1. \)

(vi) \( \forall i = 1, \ldots, K : \gamma_i = \gamma_{i+K}. \)

(vii) \( K \) is odd.

(viii) \( \sum_{i=1}^{K} \gamma_i = \xi + (K - 1)/2. \)

**Proof.** We want to use the basic functions introduced in Theorem 5.5.11. To this end we first note the following property of these functions. Let \( h_i(x_1, x_2) = \theta(-\sin(\alpha_i)x_1 + \cos(\alpha_i)x_2), \ i = 1, \ldots, K, \) let \( \tilde{f} : \mathbb{R}^2 \to \mathbb{R} \) be defined by

\[
\tilde{f}(x_1, x_2) = a_0 + \sum_{i=1}^{K} \{a_{2i-1}h_i(x_1, x_2) + a_{2i}(1 - h_i(-x_1, -x_2))\},
\]

for some \( a_0, a_1, \ldots, a_{2K} \in \mathbb{R}, \) and let \( \mu(\varphi) = \tilde{f}(\cos \varphi, \sin \varphi), \) for all \( \varphi \in \mathbb{R}. \) Then we have

\[
\mu(\varphi) = a_0 + \sum_{i=1}^{K} \{a_{2i-1}\theta(\sin(\varphi - \alpha_i)) + a_{2i}(1 - \theta(-\sin(\varphi - \alpha_i))))\},
\]

for all \( \varphi \in \mathbb{R}, \) and, consequently

\[
\forall j = 0, \ldots, K \forall \varphi \in (\alpha_j, \alpha_{j+1}) : \mu(\varphi) = a_0 + \sum_{i=1}^{j} a_{2i-1} + \sum_{i=1}^{j} a_{2i},
\]

\[
\forall j = K, \ldots, 2K \forall \varphi \in (\alpha_j, \alpha_{j+1}) : \mu(\varphi) = a_0 + \sum_{i=1}^{K} a_{2i-1} + \sum_{i=1}^{K} a_{2i},
\]

\[
\forall j = 1, \ldots, K : \mu(\alpha_j) = a_0 + \sum_{i=1}^{j} a_{2i-1} + \sum_{i=1}^{j} a_{2i},
\]

\[
\forall j = K + 1, \ldots, 2K : \mu(\alpha_j) = a_0 + \sum_{i=1}^{K} a_{2i-1} + \sum_{i=1}^{K} a_{2i}.
\]

Next, we prove the 'if'-part.

Let \( a_0 = 2\beta_0 - 1,\ a_{2i-1} = 2\gamma_i - 2\beta_{i-1}, \) and \( a_{2i} = 2\beta_i - 2\gamma_i, \ i = 1, \ldots, K. \) Then using (ii), (iii), (v), (vi), (vii), and (viii) through (5.45), one straightforwardly shows that \( \mu(\varphi) = 2\chi_{V}(\cos \varphi, \sin \varphi) - 1, \) for all \( \varphi \in (a_0, a_{2K+1}) \subset [0, 2\pi]. \) Furthermore, using (v), (vii), and (viii), we find that

\[
\tilde{f}(0, 0) = a_0 + \sum_{i=1}^{K} a_{2i-1}
\]

\[
= 2\beta_0 - 1 + \sum_{i=1}^{K} (2\gamma_i - 2\beta_{i-1})
\]

\[
= 2\beta_0 - 1 + 2 \sum_{i=1}^{K} \gamma_i - 2 \sum_{i=0}^{K-1} \beta_i
\]
\[ f = 2 \beta_0 - 1 + 2(\xi + (K - 1)/2) - 2(\beta_0 + (K - 1)/2) \]
\[ = 2\xi - 1. \]

Hence, \( \tilde{f}(r \cos \varphi, r \sin \varphi) = 2X_{\nu}(r \cos \varphi, r \sin \varphi) - 1 \), for all \( r \geq 0 \) and \( \varphi \in [0, 2\pi] \).

This yields \( f = \theta \circ \tilde{f} \in R_{2,1}^{(l)}, \tilde{f} = \theta \circ \tilde{f} = X_{\nu}, \) and, hence, \( V \in C_{2,1}^0 \).

Finally, we prove the ‘only-if’ part.

Let \( V = \mathcal{F}(f) \), for some \( f \in R_{2,1}^{(l)} \). Then we may assume that \( f = \theta \circ \tilde{f} \), with
\[ \tilde{f}(x_1, x_2) = c_0 + \sum_{i=1}^{K} c_i h_i(x_1, x_2) \]
for some \( h_i \in R_{1,2,1} \), \( h_i(x_1, x_2) = \theta(h_i(x_1, x_2)) \), \( i = 1, \ldots, K \).

Assume that \( L = \{1, \ldots, K\} \) is such that \( h_i(0, 0) = 0 \), for \( i = 1, \ldots, L \), and \( h_i(0, 0) \neq 0 \), for \( i = L + 1, \ldots, K \). We prove that \( \tilde{f}(x_1, x_2) = c_0 + \sum_{i=1}^{L} c_i h_i(x_1, x_2) \), for all \( (x_1, x_2) \in \mathbb{R}^2 \).

Firstly, we note that the function defined by \( \tilde{g}(x_1, x_2) = \sum_{i=1}^{K} c_i h_i(x_1, x_2) \), for all \( (x_1, x_2) \in \mathbb{R}^2 \), is constant in a sufficiently small disc with center \((0, 0)\). Secondly, using (a) it follows that \( f \) is constant on every halfline \((\lambda x_1, \lambda x_2)\), \( \lambda > 0 \). Since \( f \) is basic, this implies that also \( \tilde{f} \) is constant on every halfline \((\lambda x_1, \lambda x_2)\), \( \lambda > 0 \). Since further \( c_0 + \sum_{i=1}^{L} c_i h_i(x_1, x_2) \) is constant on every halfline, we find that \( \tilde{g}(x_1, x_2) \) is constant on every halfline \((\lambda x_1, \lambda x_2)\), \( \lambda > 0 \). Hence, \( \tilde{g}(x_1, x_2) = \tilde{g}(0, 0) \), for all \( (x_1, x_2) \in \mathbb{R}^2 \), which is possible only if \( \tilde{g}(x_1, x_2) = 0 \).

Consequently, we may assume that \( f \) is of the form given by (5.41), for some \( K \geq 0 \), \( h_i(x_1, x_2) = \theta(-\sin(\alpha_i)x_1 + \cos(\alpha_i)x_2) \), \( i = 1, \ldots, K \), with \( 0 \leq \alpha_1 < \cdots < \alpha_K < \pi \), and \( a_0, a_1, \ldots, a_K \in \mathbb{R} \) such that \( a_{2i-1} \neq 0 \) or \( a_{2i} \neq 0 \), \( i = 1, \ldots, K \).

Using (b) it follows that \( K \neq 0 \). Define \( a_0 = a_K - \pi, a_{i+K} = a_i + \pi, i = 1, \ldots, K-1, \beta_i = [1 + \mu(\alpha_i)]/2, \) for some \( \psi \in (\alpha_i, \alpha_{i+1}) \), \( i = 0, \ldots, 2K \), \( \gamma_i = [1 + \mu(\alpha_i)]/2, \) \( i = 1, \ldots, 2K \), \( \xi = [1 + f(0, 0)]/2 \). Then, since \( f \) is basic, we obtain (i) through (iv), and it remains to prove (v) through (viii). We distinguish between the cases \( K = 1 \) and \( K \geq 2 \).

\( K = 1. \)

Then automatically (vii) holds. Furthermore, we directly find from (5.42), (5.43), and (5.41) that \( 2\gamma_1 - 1 = 2\gamma_2 - 1 = 2\xi - 1 = a_0 + a_1 \), which proves (vi) and (viii).

Finally, (v) follows from (b).

\( K \geq 2. \)

Then we may apply Theorem 5.5.11. This yields (vii). Using that \( a_{2i-1} + a_{2i} = 2(-1)^i a_0, i = 1, \ldots, 2K, K \) is odd, (5.42) and (5.43), it is easily shown that \( 2\beta_i - 1 = (-1)^i a_0, i = 0, \ldots, 2K \), which proves (v). Next, we use that \( a_{2i-1} = 0 \) or \( a_{2i} = 0 \).

Combined with (5.44) and (5.45) this implies that either
\[ \gamma_i = \beta_i = \beta_{i+K-1} = \gamma_{i+K}, \]
or
\[ \gamma_i = \beta_{i-1} = \beta_{i+K} = \gamma_{i+K}, \]
which proves (vi). Finally, by (5.44) we have that \( 2\gamma_i - 1 = 2\beta_i - 1 + a_{2i-1}, i = 1, \ldots, K, \) which is easily used to prove (viii).
around an arbitrary center point, and have an odd number of border lines that pass through the center point. A border line entirely belongs to the considered set or to the complement of the set, possibly except for the center point. Half the number of border lines belongs to the set and the other half belongs to the complement of the set. Finally, if a sector of $\mathbb{R}^2$, bounded by two half border lines, belongs to the set, the corresponding sector across the center point does not belong to the set, and vice versa.

The last of the above mentioned properties can be stated as $\beta_i + \beta_{i+K} = 1$, for all $i = 1, \ldots, K$, which follows directly from the conditions (v) and (vii) of Theorem 5.5.12. It is this property that is the most characteristic property of radial basic subsets.

In the following theorem we show that this property alone guarantees that a radial subset can be classified with a 2LP corresponding to a basic function, provided that the border points of the subset are left without consideration. Moreover, and here lies a great deal of the strength of the theorem, the number of border lines does not have to be restricted to odd values in advance. Instead we implicitly prove that at least one of the border lines is redundant, if there are an even number of border lines. Note that in Theorem 5.5.13 we use $p$ instead of $K$ for denoting the number of border lines, to stress upon the difference with Theorem 5.5.12 and to correspond to the formulation used in Theorem 5.5.3.

**Theorem 5.5.13.** Let $p \in \mathbb{N}$, let $\alpha_1, \ldots, \alpha_{2p} \in \mathbb{R}$, with $0 \leq \alpha_1 < \alpha_2 < \cdots < \alpha_p < \pi$ and $\alpha_{2p} = \alpha_1 + \pi$, $i = 1, \ldots, p + 1$, and let $\beta_i \in \mathbb{B}$, $i = 1, \ldots, 2p$. Then there exists a basic function represented by $f \in R_{2,1}^\theta$ that satisfies

$$\forall r > 0 \forall i = 1, \ldots, 2p \forall \varphi < \alpha_{i+1} : f(r \cos \varphi, r \sin \varphi) = \beta_i,$$  \hspace{1cm} (5.46)

if

$$\forall i = 1, \ldots, p : \beta_i + \beta_{i+p} = 1.$$

(5.47)

**Proof.** Let $\beta_0 = \beta_{2p}$, then (5.47) holds for $i = 0$ also. Using (5.47) twice, once for $i$ and once with $i$ replaced by $i - 1$, and adding these equations, we obtain that

$$\forall i = 1, \ldots, p : \beta_{i-1} + \beta_i = 1 \Leftrightarrow \beta_{i+p-1} + \beta_{i+p} = 1.$$  \hspace{1cm} (5.48)

Define $L \in \mathbb{N}$ and $i_1, \ldots, i_L \in \{1, \ldots, 2p\}$ such that, firstly, $i_1 < i_2 < \cdots < i_L$, and, secondly, $\beta_{i-1} + \beta_i = 1$, if and only if $i \in \{i_1, \ldots, i_L\}$. From (5.47) it follows that $L \geq 1$. Let $1 \leq q \leq L$ be such that $i_1, \ldots, i_q \in \{1, \ldots, p\}$ and $i_{q+1}, \ldots, i_L \in \{p+1, \ldots, 2p\}$. Then, using (5.48), we find that $L = 2q$ and $i_{j+q} = i_j + p$, for all $j = 1, \ldots, q$.

From the definition of $i_j$ we know that $2\beta_{i_j} - 1 = (-1)^{j-1}(2\beta_{i_j} - 1)$, for all $j = 1, \ldots, 2q$.

Combining (5.47) once more with the above results yields

$$2\beta_{i_j} - 1 = -(-2\beta_{i_j+p} - 1) = -(-2\beta_{i_j+1} - 1) = (-1)^{q+1}(2\beta_{i_j} - 1),$$

which proves that $q$ is odd.

Using Theorem 5.5.12, one can now easily show that there exists an $f \in R_{2,1}^\theta$ that satisfies

$$\forall r > 0 \forall j = 1, \ldots, 2q \forall \alpha_{i_j} < \varphi < \alpha_{i_{j+1}} : f(r \cos \varphi, r \sin \varphi) = \beta_{i_j}.$$
Since $\beta_i = \beta_{i+1}$, for all $i_1 \leq i < i_{j+1}$, $j = 1, \ldots, 2q$, it follows that this $f \in R_{2,2,1}^q$ also satisfies (5.46).

We use Theorem 5.5.13 to give an alternative proof of the ‘sufficiency’ part of Theorem 5.5.3; see Theorem 5.5.14 below. The idea is to prove that the considered radial subset can either be written as the union or as the intersection of two basic subsets, and then to apply Corollary 5.5.9. The construction of the two basic subsets is in both cases the same and can be described informally as follows. Choose a line through the center point of the considered radial subset $V \subseteq \mathbb{R}^2$. This line may be any line, but it is most convenient to take one of the border lines of $V$. Copy everything of $V$ at one side of the line to the first candidate basic subset, and everything of $V$ at the other side of the line to the second candidate basic subset. Complete both the candidate basic subsets by simultaneously inverting and mirroring the sets in the center point, such that they satisfy the conditions of Theorem 5.5.13.

The above construction is illustrated by Figures 5.17a, b, and c. The subset of Figure 5.17a is the same as in Figure 5.9a. We choose the horizontal axis as the line where we cut the subset in two halves. The subsets of Figures 5.17b and c are obtained by taking the upper and lower half of Figure 5.17a, respectively, and completing them by inverting and mirroring this half to the other half. Then by Theorem 5.5.13 there exist two basic subsets that equal the subsets of Figures 5.17b and c, except for their border points. One verifies that the union of the subsets of Figures 5.17b and c yields the subset of Figures 5.17a, which therefore also holds for the union of the two basic subsets, possibly except for their border points. In the proof of Theorem 5.5.14 below, the above construction is formalized.

**Theorem 5.5.14.** Let $p \in \mathbb{N}$, let $\alpha_1, \ldots, \alpha_{2p+1} \in \mathbb{R}$, with $0 \leq \alpha_1 < \alpha_2 < \ldots < \alpha_p < \pi$ and $\alpha_{2p} = \alpha_1 + \pi$, $i = 1, \ldots, p+1$, and let $\beta_i \in \mathbb{B}$, $i = 1, \ldots, 2p$. Then there exists a 2LP represented by the function $f \in R_{2,2,1}$ that satisfies

$$
\forall r > 0 \forall i = 1, \ldots, 2p \forall \alpha_i < \varphi < \alpha_{i+1} \colon f(r \cos \varphi, r \sin \varphi) = \beta_i,
$$

(5.49)
5.5 Characterizations of $C_2$

If
\[ [V_i = 1, \ldots, p : \beta_i = 1 \vee \beta_{i+p} = 1] \cup [V_i = 1, \ldots, p : \beta_i = 0 \vee \beta_{i+p} = 0] \]

Proof. Define $\beta_i^{(1)}, \beta_i^{(2)} \in B, i = 1, \ldots, 2p,$ by
\[
\beta_i^{(1)} = \begin{cases} 
\beta_i, & i = 1, \ldots, p, \\
1 - \beta_{i+p}, & i = p + 1, \ldots, 2p,
\end{cases}
\]
and
\[
\beta_i^{(2)} = \begin{cases} 
1 - \beta_{i+p}, & i = 1, \ldots, p, \\
\beta_i, & i = p + 1, \ldots, 2p,
\end{cases}
\]
respectively. Using Theorem 5.5.13, it follows that there exist basic functions $f^{(1)}, f^{(2)} \in R_{2,2,1}^2$ such that $f^{(1)}(r \cos \varphi, r \sin \varphi) = \beta_i^{(1)}$, for all $r > 0$, $\alpha_i < \varphi < \alpha_{i+1}$, $i = 1, \ldots, 2p$, and $j = 1, 2$.

Using Corollary 5.5.9, we find that $\mathcal{J}(f^{(1)}) \cap \mathcal{J}(f^{(2)}) \subseteq C_2$ and $\mathcal{J}(f^{(1)}) \cup \mathcal{J}(f^{(2)}) \subseteq C_2$, respectively. Hence, there exist two 2LPs represented by $f^1, f^2 \in R_{2,2,1}^2$, such that $\mathcal{J}(f^1) = \mathcal{J}(f^{(1)}) \cup \mathcal{J}(f^{(2)})$ and $\mathcal{J}(f^2) = \mathcal{J}(f^{(1)}) \cap \mathcal{J}(f^{(2)})$. The proof is completed by verifying that $f = f^1$ satisfies (5.49) if $\beta_i = 1 \vee \beta_{i+p} = 1$, for all $i = 1, \ldots, p$, and $f = f^2$ satisfies (5.49) if $\beta_i = 0 \vee \beta_{i+p} = 0$, for all $i = 1, \ldots, p$.

Note that the proof of Theorem 5.5.14 implies that the classifiable two-dimensional radial subsets are either in $C_2^+ \setminus C_2$ or in $C_2^+ \setminus C_2$, depending on whether $\beta_i = 1 \vee \beta_{i+p} = 1$, for all $i = 1, \ldots, p$, or $\beta_i = 0 \vee \beta_{i+p} = 0$, for all $i = 1, \ldots, p$, respectively.

Next, we return to the basic subsets. In the following theorem we show that there are no other types of two-dimensional basic subsets than the ones identified by Corollary 5.5.12 and Theorem 5.5.12, corresponding to the examples given in Figures 5.16a and b, respectively. Thus either all border lines are parallel, or all border lines pass through one point.

Theorem 5.5.15. Let $V \in C_2^\circ$ be a two-dimensional basic subset. Then $V$ is either of the form given by Corollary 5.5.12, for $N = 2$, or of the form given by Theorem 5.5.12, possibly after a translation.

Proof. Let $V = \mathcal{J}(f)$, for some $f \in R_{2,2,1}^2$. Then we may assume that $f = \theta \circ \tilde{f}$, with $\tilde{f}(x_1, x_2) = c_0 + \sum_{i=1}^K c_i h_i(x_1, x_2)$, $K \in \mathbb{N}_0$, $c_0 \in \mathbb{R}$, $c_i \in \mathbb{R}, h_i \in R_{1,2,1}(0,1)$, $h_i(x_1, x_2) = \theta(h_i(x_1, x_2))$, $i = 1, \ldots, K$, and $h_i \neq h_j$, $i \neq j$. We distinguish the following three cases for the configuration of the lines $\mathcal{J}^=(h_i)$, $i = 1, \ldots, K$, which cover all possible configurations.

- The lines $\mathcal{J}^=(h_i)$ are all parallel, i.e., $\mathcal{J}^=(h_i) \cap \mathcal{J}^=(h_j) = \emptyset$ or $\mathcal{J}^=(h_i) = \mathcal{J}^=(h_j)$, for all $i, j = 1, \ldots, K$.

- All the lines $\mathcal{J}^=(h_i)$ intersect in one point, and there are at least two different lines, i.e., $\mathcal{J}^=(h_i) \neq \mathcal{J}^=(h_j)$, for some $i, j \in \{1, \ldots, K\}$.

- There are at least two distinct points where two or more different lines $\mathcal{J}^=(h_i)$ intersect.

In the first case $V$ is of the form given by Corollary 5.5.12. In the second case, we translate $V$ such that the intersection point is $(0,0)$. Then it follows that $f$ can be written in the form given by Theorem 5.5.11, which implies that $V$ is of the form
given by Theorem 5.5.12. The proof is completed by showing that the third case cannot occur.

The first step is to prove that through every intersection point of two different lines \( f^L(h_i) \) there is a third line distinct from the other two. Consider a point in \( \mathbb{R}^2 \) where two or more different lines intersect. Without loss of generality we take this point to be \((0,0)\). Then, possibly after renumbering the \( h_i \)'s, it follows straightforwardly that there exist \( L_1, L_2 \in \mathbb{N} \) with \( 2 \leq L_1 \leq L_2 \leq K \), \( a_0, a_1, \ldots, a_{2L} \in \mathbb{R} \) with \( a_{2i-1} \neq 0 \) or \( a_{2i} \neq 0 \), for all \( i = 1, \ldots, L_1 \), and \( b_{L_1+1}, \ldots, b_{L_2} \in \mathbb{R} \setminus \{0\} \), such that \( f \) can be written as

\[
\hat{f}(x_1, x_2) = a_0 + \sum_{i=1}^{L_1} (a_{2i-1} h_i(x_1, x_2) + a_{2i}(1 - h_i(-x_1, -x_2))),
\]

with \( h_i(x_1, x_2) = \theta(-\sin(\alpha_i) x_1 + \cos(\alpha_i) x_2), \) \( i = 1, \ldots, L_1 \), for some \( 0 \leq \alpha_1 < \alpha_2 < \cdots < \alpha_{L_1} < \pi \), and \( h_i(0,0) \neq 0 \), \( i = L_1 + 1, \ldots, L_2 \). Define \( g \in R_{L_2}^B \) by \( g = \theta \circ \hat{g} \),

where

\[
\hat{g}(x_1, x_2) = a_0 + \sum_{i=1}^{L_1} (a_{2i-1} h_i(x_1, x_2) + a_{2i}(1 - h_i(-x_1, -x_2))),
\]

for all \((x_1, x_2) \in \mathbb{R}^2\). Then \( \hat{g}(x_1, x_2) = \hat{f}(x_1, x_2) \in \{-1, +1\} \), for all \((x_1, x_2) \) in a sufficiently small disk with center \((0,0)\), which is easily shown to imply that \( g \in R_{L_2}^B \). Consequently, we may apply Theorem 5.5.11 to obtain that \( L_1 \) is odd, and, hence, \( L_1 \geq 3 \).

The second and final step is given by the following proposition.

Proposition 5.5.6. There does not exist a solution to the dual of Sylvester's problem, which is stated as follows: Find a finite number of lines not all parallel or going through the same point such that through every point lying on two distinct lines there passes a third distinct line.

This completes the proof of the theorem. \( \square \)

Proposition 5.5.6 can be proved similarly as the well-known result that Sylvester's problem of collinear points has no solution; see Coxeter [1989] for a description of this problem and a proof due to Kelly.

The last class of basic subsets studied in detail is the class of the three-dimensional basic subsets. We prove that the three-dimensional basic subsets are essentially two-dimensional and thus do not introduce new building blocks. As we know, we can construct a three-dimensional basic function from a two-dimensional basic function using Proposition 5.5.5, similar as we constructed two-dimensional basic functions from one-dimensional basic functions. Let \( f \in R_{L_2}^B \) be a two-dimensional basic function, and let \( B \) be a \( 3 \times 2 \) matrix. Then the function \( g : \mathbb{R}^3 \rightarrow B \) defined by \( g(x) = f(xB) \), \( x \in \mathbb{R}^2 \), represents a three-dimensional basic function. In case that the basic function \( f \) corresponds to a basic subset in which all the border lines are parallel, the border planes of the three-dimensional basic subset corresponding to \( g \) are all parallel also; see Figure 5.18a for a possible configuration of the border planes. In case that the border lines of the subset corresponding to \( f \) all intersect in one point, the border planes of the subset corresponding to \( g \) all intersect in one line; see Figure 5.18b for a possible configuration in this case.
Figure 5.18: The only possible configurations of the hyperplanes $J^a(h_i)$ for a three-dimensional basic function or subset. They correspond with the two different types of two-dimensional basic subsets depicted in Figures 5.16a and b.

Below we prove that there are no other types of three-dimensional basic subsets. This result could be stated similar to Theorem 5.5.15, if we had a three-dimensional version of Theorem 5.5.12. Because of the above, the following more convenient formulation suffices.

**Theorem 5.5.16.** Let $f \in R_{2,3,1}$. Then $f \in R_{2,3,1}^B$ if and only if there exist a two-dimensional basic function $g \in R_{2,2,1}^B$ and a matrix $B \in R^{2 \times 2}$, such that $f(x) = g(xB)$, for all $x \in R^3$.

**Proof.** The 'if'-part follows directly from Proposition 5.5.5.

It remains to prove the 'only-if'-part. Let $f \in R_{2,3,1}^B$, say $f = \theta \circ \tilde{f}$, with $\tilde{f}(x_1, x_2) = c_0 + \sum_{i=1}^{K} c_i h_i(x_1, x_2)$, $K \in N_0$, $c_0 \in R$, $c_i \in R \setminus \{0\}$, $h_i \in R_{1,2,1} \setminus \{0, 1\}$, $h_i(x_1, x_2) = \theta(h_i(x_1, x_2))$, $i = 1, \ldots, K$, and $h_i \neq h_j$, $i \neq j$.

We start by proving that the planes $J^a(h_i)$ all share at least one direction. To this end we construct a two-dimensional basic function form $f$, by taking a cross-section along a plane $\{x \in R^2 | x = u(0) + \lambda_1 u(1) + \lambda_2 u(2), \lambda_1, \lambda_2 \in R\}$, for some $u(0), u(1), u(2) \in R^3$. We choose $u(0), u(1), u(2)$ as follows.

(a) Firstly, choose the vectors $u(1), u(2) \in R^3$ such that the plane $\{x \in R^2 | x = \lambda_1 u(1) + \lambda_2 u(2), \lambda_1, \lambda_2 \in R\}$ is not parallel to any of the sets $J^a(h_i) \cap J^a(h_j)$, for those $i, j \in \{1, \ldots, K\}$ for which this intersection is not empty.

(b) Secondly, choose the vector $u(0) \in R^3$ such that the plane $P = \{x \in R^2 | x = u(0) + \lambda_1 u(1) + \lambda_2 u(2), \lambda_1, \lambda_2 \in R\}$ does not contain any of the sets $J^a(h_i) \cap J^a(h_j)$, for those $i, j, l \in \{1, \ldots, K\}$ for which this intersection is not empty.

Define $g : R^2 \to B$ by $g(x_1, x_2) = f(u(0) + x_1 u(1) + x_2 u(2))$, then by Proposition 5.5.5 it follows that $g \in R_{2,2,1}^B$. Let $\tilde{g}(x_1, x_2) = \tilde{h}(u(0) + x_1 u(1) + x_2 u(2))$, let $e(x_1, x_2) = \theta(\tilde{h}(x_1, x_2))$, and let $\tilde{g}(x_1, x_2) = c_0 + \sum_{i=1}^{K} c_i e_i(x_1, x_2)$, for all $(x_1, x_2) \in R^2$. Then it follows that $g = \theta \circ \tilde{g}$, $K \in N_0$, $c_0 \in R$, $c_i \in R \setminus \{0\}$, and $e_i \in R_{1,2,1}$, for all $i = 1, \ldots, K$. 
Furthermore, using (a) and (b), one easily verifies that $e_i \notin \{0, 1\}$, $i = 1, \ldots, K$, and $e_i \neq e_j$, $i \neq j$, respectively. Hence, using the proof of Theorem 5.5.15 applied to $g \in R_{2,1}$, we find that the lines $J^\tau(e_i)$, $i = 1, \ldots, K$, are either all parallel or all intersect in one point. We use this to prove that the planes $J^\tau(h_i)$, $i = 1, \ldots, K$, all share at least one direction. In fact, we prove that the planes $J^\tau(h_i)$ are either all parallel or all contain the same line, and use this to conclude that they all share at least one direction.

If all the planes $J^\tau(h_i)$, $i = 1, \ldots, K$, are parallel, they share at least one direction, so assume that they are not all parallel. Then there exist two distinct planes $J^\tau(h_i)$ and $J^\tau(h_j)$, for some $i \neq j$, whose intersection is a line, say, $L = J^\tau(h_i) \cap J^\tau(h_j)$. Using (a), we find that the line $L$ must intersect the plane $P$ in a point $x = v(0) + x_1v(1) + x_2v(2)$, for some $x_1, x_2 \in \mathbb{R}$. Consequently, the lines $J^\tau(e_i)$ and $J^\tau(e_j)$ intersect in $(x_1, x_2)$. Furthermore, using that the planes $J^\tau(h_i)$ and $J^\tau(h_j)$ are distinct, combined with (b), we derive that the lines $J^\tau(e_i)$ and $J^\tau(e_j)$ are not parallel. This implies that all the lines $J^\tau(e_i)$, $i = 1, \ldots, K$, must intersect in $(x_1, x_2)$, and, hence, all the planes $J^\tau(h_i)$, $i = 1, \ldots, K$, contain $L$. Assume that a plane $J^\tau(h_i)$ does not contain the line $L$. Then $J^\tau(h_i) \cap J^\tau(h_j) = L \cap P$, which contradicts (b). Thus the planes $J^\tau(h_i)$, $i = 1, \ldots, K$, are either all parallel or all contain the same line, which implies that they all share at least one direction.

Let the vector $v(3) \in \mathbb{R}^2$, $\|v(3)\| = 1$, represent the direction shared by all the planes $J^\tau(h_i)$, $i = 1, \ldots, K$. Then there exist $x(0), x(1) \in \mathbb{R}^2$, such that $J^\tau(h_i) = \{x \in \mathbb{R}^2 | x = x(0) + \lambda_1 v(0) + \lambda_2 v(1), \lambda_1, \lambda_2 \in \mathbb{R}\}$, for all $i = 1, \ldots, K$. Note that this implies that $h_i(x + \mu v(3)) = h_i(x)$, for all $i = 1, \ldots, K$, and, hence, $f(x + \mu v(3)) = f(x)$, for all $x \in \mathbb{R}^2$ and $\mu \in \mathbb{R}$. Let $v(1), v(2) \in \mathbb{R}^3$ be such that the matrix $[v(1) v(2) v(3)]$ is orthogonal. Then we define the basic function $g \in R_{2,1}$ by $g(u_1, u_2) = f(u_1v(1) + u_2v(2))$, for all $(u_1, u_2) \in \mathbb{R}^2$ and the matrix $B \in \mathbb{R}^{2 \times 2}$ by $B = [v(1) v(2)]$. The final result now follows from

$$f(x) = f((v(1) \cdot x)v(1) + (v(2) \cdot x)v(2) + (v(3) \cdot x)v(3)) = f((v(1) \cdot x)v(1)) = g((v(1) \cdot x), v(2)) = g(xB),$$

for all $x \in \mathbb{R}^3$. \hfill \qed

Similarly as we did above for $N = 3$, we can use Proposition 5.5.5 to construct $N$-dimensional basic functions for arbitrary $N \geq 3$. We think that this yields all possible $N$-dimensional basic functions, but we do not know of a proof for $N > 3$.

Conjecture 5.5.2. Let $N \in \mathbb{N}$, $N \geq 3$, and $f \in R_{2,N,1}$, then $f \in R_{2,N,1}$ if and only if there exist a two-dimensional basic function $g \in R_{2,1}$ and a matrix $B \in \mathbb{R}^{N \times 2}$, such that $f(x) = g(xB)$, for all $x \in \mathbb{R}^N$.

Note that Conjecture 5.5.2 corresponds well with Conjecture 5.5.1. In essence both conjectures state that the question whether or not a subset is classifiable with a 2LP is rather a two-dimensional business, i.e., in three or more dimensions there do not occur new phenomena. Furthermore, both conjectures have been proven for $N = 3$ and are open for $N > 3$. 

138

The MLP Complexity of CCPs
Figure 5.19: A decomposition of the subset presented in Figure 5.6c, which shows that it can be classified with a 2LP; see the text for an explanation of the different steps.

In the remainder of this section we complete the proof that the subsets of Figure 5.6 are classifiable with a 2LP. To this end we take the classifiable subsets of Figure 5.15 and delete the remaining augmentation, by subsequently intersecting the subsets with the appropriate basic subsets and using Corollary 5.5.4. The idea is to delete small strips, using subsets that consist of the union of two parallel and disjunct affine halfspaces, which are basic due to Corollary 5.5.12. The reader should have no difficulty to apply this technique to the subsets of Figures 5.15a and b. Obviously, the smaller the strips must be or the more space that has to be removed, the more strips, and thus, the more hidden nodes of the classifying 2LP are required.
We give the proof of the classifiability of the subset of Figure 5.6c in detail; see Figure 5.19. First, we construct the subset of Figure 5.19c, which is the intersection of the basic subsets in Figure 5.19a and b. It can be verified that this is essentially the construction approach used in the proof of Theorem 5.5.14. Secondly, we unify the subset in Figure 5.19c with the basic subset shown in Figure 5.19d, which yields the classifiable subset in Figure 5.19e; see Corollary 5.5.7 for a justification of this construction. Next, we intersect the subset in Figure 5.19e with the polyhedron given in Figure 5.19f, to obtain the classifiable subset in Figure 5.19g; see also Figure 5.11 in Section 5.5.3. In Figure 5.19h we have shown an intersection of a number of basic subsets corresponding to inverted strips, which by Corollary 5.5.9 forms a subset in $C^+_U$. Hence, using Corollary 5.5.4, we find that the subset of Figure 5.19h, which corresponds to the intersection of the subset of Figure 5.19h and the subset of Figure 5.19g, is classifiable with a 2LP. The above construction yields a 2LP that classifies the subset of Figure 5.6c and requires a total of 25 hidden nodes.

As the above example shows, the decomposition of a subset into a number of basic subsets can be far from trivial. Obviously, one would like to have a methodical approach for this decomposition, instead of doing it by hand on trial and error basis. The following section discusses a decomposition method, together with a sketch for a possible implementation, for a restricted collection of subsets.

5.6 Decomposition

In Section 5.5.3 and Section 5.5.4 we discussed sufficient conditions for subsets of $R^N$ to be classifiable with a 2LP. All of the discussed results are of the following form: if the subset is of a certain type, it can be classified with a 2LP. For instance, if the subset is a pseudo polyhedron, it can be classified with a 2LP; see Corollary 5.5.10. And also, if a subset is radial and satisfies certain additional properties, it can be classified with a 2LP; see Theorem 5.5.14. The obvious question is how can we determine whether a given subset is of one of those types. Thus, how can we detect that a subset is a pseudo polyhedron, or that a subset is radial. We call methods that answer such questions *decomposition* methods. For instance, a decomposition method might be developed to find affine halvespaces $W_1, \ldots, W_L$ that satisfy $V = \bigcap_{i=1}^L W_i$, for a given subset $V \in U$, in case that such a decomposition exists, which would answer the question whether or not a subset is a pseudo polyhedron. Similarly, a decomposition method might be developed to find basic subsets $V_1, \ldots, V_L$ that satisfy $V = \bigcap_{i=1}^L V_i$, for a given subset $V \in C^+_U$, in case that such a decomposition exists; see Corollary 5.5.9.

In this section we discuss a decomposition method for a restricted version of the sufficient condition presented in Corollary 5.5.11, i.e., we consider the problem of finding $V_1, \ldots, V_L \in P$ that satisfy $V = V_1 \setminus V_2 \setminus \cdots \setminus V_L$, for a given subset $V \in U$, in case that such a decomposition exists. Thus we consider Corollary 5.5.11 with polyhedrons instead of pseudo polyhedrons. A second restriction we have to make is that the considered subset is $V \in U$ is rational, i.e., we assume that there exist rational, affine halvespaces $W_{ij} \in H$ such that $V = \bigcup_{i=1}^K \bigcap_{j=1}^L W_{ij}$, for some $K, L \in N_0$. With a *rational* affine halfspace we denote an affine halfspace of the form $\{x \in R^N | a \cdot x + b \geq 0\}$ or $\{x \in R^N | a \cdot x + b > 0\}$, with $a \in Q^N$ and $b \in Q$. 


5.6 Decomposition

First, we examine the case that \( L = 1 \), i.e., we answer the question of deciding whether a given rational subset \( V \in \hat{U} \) is a polyhedron. This corresponds to finding a set of rational, closed affine halfspaces \( W_1, \ldots, W_K \) that satisfy \( V = \bigcap_{i=1}^{K} W_i \) in case that such a decomposition exists. This problem already contains most of the technical difficulties of the general problem, which is treated afterwards. Note that our discussion aims at proving the existence of a decomposition method. The computational efficiency of the obtained decomposition method is left without consideration.

Given a subset \( V \in \hat{U} \), we consider the question whether or not \( V \in P \). Note that the assumption that \( V \in \hat{U} \), is already a restriction, but we cannot do without it. Below we prove the property that in case that \( V \in \hat{U} \), it holds that \( V \) is a polyhedron, if and only if \( V \) is convex and closed. This implies that verifying whether \( V \in \hat{U} \), is equivalent with verifying whether \( V \) is convex and closed. Hence, a possible verification method is to compute the closure of the convex-hull of \( V \). If this yields \( V \), then \( V \) is a polyhedron, otherwise it is not a polyhedron. The above ideas are formalized in the following theorem, which uses \( \text{conv-hull}(V) \) to denote the closure of the convex-hull of \( V \); see Section 5.2 for a definition of the closure and the convex-hull of a subset in \( R^N \).

**Theorem 5.6.1.** Let \( V \in \hat{U} \). Then \( V \in P \), if and only if \( \text{conv-hull}(V) = V \).

**Proof.** The 'only-if'-part follows directly from \( \text{conv-hull}(V) = V \) and \( \overline{V} = V \), if \( V \in P \). The 'if'-part is directly obtained from the following lemma.

**Lemma 5.6.1.** Let \( V \in \hat{U} \). Then \( \text{conv-hull}(V) \in P \). This completes the proof of the theorem.

Lemma 5.6.1 is proved below, after we complete the description of our decomposition method for answering the question whether or not a given subset is a polyhedron. Note that Theorem 5.6.1 does not hold with \( \text{conv-hull}(V) \) replaced with \( \text{conv-hull}(V) \), since there exist subsets not even in \( \hat{P} \), for which \( \text{conv-hull}(V) = V \); see the subset of Figure 5.21d presented below. This implies that we have to restrict ourselves to subsets in \( P \). If we had a mapping \( q : \hat{U} \rightarrow \hat{U} \), such that \( q(V) = V \), if and only if \( V \in P \), the decomposition method could be extended to subsets in \( \hat{P} \).

To convert Theorem 5.6.1 into a procedure for testing whether a subset is a polyhedron, we design four algorithmic functions. To this end, we use the description of \( V \in \hat{U} \), which is given by \( K \in \mathbb{N}_0 \), \( L_i \in \mathbb{N}_0 \), \( i = 1, \ldots, K \), and \( W_{ij} \in \bar{H} \), \( i = 1, \ldots, K \), \( j = 1, \ldots, L_i \), such that \( V = \bigcup_{i=1}^{K} \bigcap_{j=1}^{L_i} W_{ij} \). If \( V \in P \), then \( K = 1 \) and \( W_{ij} \in H \). The four functions take the following form.

**function INTERSECT(V1, V2);**
**input:** description of \( V_1 \in P \) and \( V_2 \in \hat{U} \)
**output:** description of \( V_1 \cap V_2 \in \hat{U} \)

**function INVERT(V);**
**input:** description of \( V \in \hat{U} \)
**output:** description of \( V^* \in \hat{U} \)**
function EMPTY(V);
input: description of $V \in \hat{U}$
output: true if $V = \emptyset$, false otherwise

function CLOS_CONV_HULL(V);
input: description of $V \in \hat{U}$
output: description of $\text{conv}(\text{hull})(V) \in \hat{P}$

Before we sketch an implementation of these functions, we show how they can be used to construct a procedure for verifying whether a subset is a polyhedron. Using Theorem 5.6.1, and using that $\text{conv}(\text{hull})(V) \supseteq V$, for all $V \subseteq \mathbb{R}^N$, we obtain the following result.

---

procedure DEC1(input: $V \in \hat{U}$; output: $\text{poly} \in \{\text{true}, \text{false}\}$, $V' \in \hat{P}$);
begin
$V' := \text{CLOS_CONV_HULL}(V)$;
$\text{poly} := \text{EMPTY}(\text{INTERSECT}(V', \text{INVERT}(V)))$
end;

---

Figure 5.20: The procedure DEC1.

Corollary 5.6.1. Let $V \in \hat{U}$ and DEC1 the procedure presented in Figure 5.20. Then DEC1 correctly verifies whether or not $V \in \hat{P}$, and the result is indicated by the Boolean output variable $\text{poly}$. If $\text{poly} = \text{true}$, then a decomposition of $V$ as an intersection of closed affine halfspaces is given by the description of $V' \in \hat{P}$.

Next, we sketch an implementation of the four functions introduced above.

INTERSECT($V_1$, $V_2$). Since $V_1 \in \hat{P}$, it is trivial to determine the description of $V_1 \cap V_2$, given the description of $V_1$ and $V_2$.

INVERT(V). See the proof of Lemma 5.2.1.

EMPTY(V). We present a method for verifying whether the subset $V \in \hat{U}$ is empty, that is based on the following three results. All three results can be proven straightforwardly.

1. Let $V \in \hat{U}$, say $V = \bigcup_{i=1}^{K} V_i$, for some $K \in \mathbb{N}_0$, and $V_i \in \hat{P}$, $i = 1, \ldots, K$. Then $V = \emptyset$, if and only if $V_i = \emptyset$, for all $i = 1, \ldots, K$.
2. Let $V \in \hat{P}$, say $V = \{x \in \mathbb{R}^N | a^i \cdot x \leq b_i, i = 1, \ldots, K, a^i \cdot x < b_i, i = K + 1, \ldots, K + L\}$, for some $K, L \in \mathbb{N}_0$, $a^i \in \mathbb{Q}^N$, and $b_i \in \mathbb{Q}$. Then $V = \emptyset$, if and only if $V_j = \emptyset$, for all $j = 1, \ldots, L$, where $V_j$ is given by

$V = \{x \in \mathbb{R}^N | a^i \cdot x \leq b_i, i = 1, \ldots, K + L, i \neq j, a^i \cdot x < b_j\}$.

3. Let $V = \{x \in \mathbb{R}^N | a^i \cdot x \leq b_i, i = 1, \ldots, K, c \cdot x < d\}$, for some $K \in \mathbb{N}_0$, $a^i, c \in \mathbb{Q}^N$, and $b, d \in \mathbb{Q}$. Then $V = \emptyset$, if and only if $\{x \in \mathbb{R}^N | a^i \cdot x \leq b_i, i = 1, \ldots, K\} = \emptyset$, or $\max\{c \cdot x | x \in \mathbb{R}^N, a^i \cdot x \leq b_i, i = 1, \ldots, K\} \geq d$.
5.6 Decomposition

Verifying whether \( \{ x \in \mathbb{R}^N | a_i \cdot x \leq b_i, i = 1, \ldots, K \} = \emptyset \), or \( \max \{ c \cdot x \}, x \in \mathbb{R}^N, a_i \cdot x \leq b_i, i = 1, \ldots, K \} \geq d \), with \( a_i, c \in \mathbb{Q}^N \) and \( b, d \in \mathbb{Q} \), can be done using standard techniques from linear programming; see for instance Papadimitriou & Steiglitz [1982]. Consequently, combining the three results gives an approach for verifying whether a subset \( V \subseteq \tilde{U} \) is empty.

CLOS\_CONV\_HULL\( (V) \). In Theorem 5.6.2 below we present a method to find the description of \( \text{conv.hull}(V) \), for a given subset \( V \subseteq \tilde{U} \). The method consists of three steps. In the first step we use Corollary 5.2.1, which shows that the closure of a subset in \( \tilde{U} \), is in \( U \), and hence, can be written as a union of a finite number of polyhedrons. The second and the third step apply Lemma 5.2.2 in two ways. For implementations of these steps one requires an implementation of Lemma 5.2.2, which is well-discussed in literature; see also Schrijver [1986] and Nemhauser & Wolsey [1988].

**Theorem 5.6.2.** Let \( V \subseteq \tilde{U} \). Then \( V' = \text{conv.hull}(V) \) can be calculated using the following three steps.

1. Determine \( K \in \mathbb{N}_0 \) and \( V_i \subseteq P \setminus \{ \emptyset \}, i = 1, \ldots, K \), such that \( V = \bigcup_{i=1}^{K} V_i \).

2. Determine \( p_i \in \mathbb{N}_0, q_i \in \mathbb{N}_0, i = 1, \ldots, K, x_{ij} \in \mathbb{R}^{m_j}, i = 1, \ldots, K, j = 1, \ldots, p_i \), and \( y_{ij} \in \mathbb{R}^{m_j}, i = 1, \ldots, K, j = 1, \ldots, q_i \), such that

   \[
   V_i = \text{conv.hull}(\{x_{ij}\}_{j=1}^{p_i}) + \text{cone}(\{y_{ij}\}_{j=1}^{q_i}),
   \]

   for all \( i = 1, \ldots, K \).

3. Determine \( V' \subseteq P \) such that \( V' = \text{conv.hull}(\{x_{ij}\}_{i=1}^{K}) = \text{cone}(\{y_{ij}\}_{i=1}^{K}) \).

**Proof.** Let \( V \subseteq \tilde{U} \), then using Corollary 5.2.1, it follows that \( \overline{V} \subseteq U \). Hence, \( V = \bigcup_{i=1}^{K} V_i \), for some \( K \in \mathbb{N}_0 \) and \( V_i \subseteq P \setminus \{ \emptyset \}, i = 1, \ldots, K \).

For the second step we apply Lemma 5.2.2, which yields that

\[
V_i = \text{conv.hull}(\{x_{ij}\}_{j=1}^{p_i}) + \text{cone}(\{y_{ij}\}_{j=1}^{q_i}),
\]  

for some \( p_i, q_i \in \mathbb{N}_0, i = 1, \ldots, K, x_{ij} \in \mathbb{R}^{m_j}, i = 1, \ldots, K, j = 1, \ldots, p_i \), and \( y_{ij} \in \mathbb{R}^{m_j}, i = 1, \ldots, K, j = 1, \ldots, q_i \). Since \( V_i \neq \emptyset, i = 1, \ldots, K \), it follows that \( p_i > 0, i = 1, \ldots, K \).

Let \( V' \subseteq \mathbb{R}^N \) be defined by

\[
V' = \text{conv.hull}(\{x_{ij}\}_{i=1}^{K}) + \text{cone}(\{y_{ij}\}_{i=1}^{K}).
\]  

Then, by Lemma 5.2.2, it follows that \( V' \subseteq P \), which completes the third step.

In order to complete the proof we have to show that \( \overline{\text{conv.hull}(V)} = \overline{V'} \). If \( K = 0 \) or \( K = 1 \) this is trivial. Therefore, we assume in the remainder of the proof that \( K \geq 2 \).

Let \( W = \text{conv.hull}(\bigcup_{i=1}^{K} V_i) \) and let \( W' \subseteq \mathbb{R}^N \) be defined by

\[
W' = \{ x \in \mathbb{R}^N | x = \sum_{i=1}^{K} \lambda_i v_i, v_i \in V_i, \lambda_i \geq 0, \sum_{i=1}^{K} \lambda_i = 1 \}.
\]  

Then, since \( \overline{\text{conv.hull}(V)} = \overline{\text{conv.hull}(V')} = \overline{W} \), the proof is completed when we show that \( W \subseteq W' \subseteq \overline{V} \subseteq W \), because \( \overline{V'} = V' \).


$W \subseteq W'$. Follows straightforwardly, using that $V_i$ is convex, for all $i = 1, \ldots, K$.

$W' \subseteq V'$. Follows easily from (5.52), (5.50), and (5.51).

$V' \subseteq W$. Let $x \in V'$, then from (5.51) it follows that

$$x = \sum_{i=1}^{K} \sum_{j=1}^{p_i} \mu_{ij} \bar{x}_{ij} + \sum_{i=1}^{K} \sum_{j=1}^{q_i} \tau_{ij} \bar{y}_{ij},$$

for some $\mu_{ij} \geq 0$, $\tau_{ij} \geq 0$ with $\sum_{i=1}^{K} \sum_{j=1}^{p_i} \mu_{ij} = 1$. We assume without loss of generality that $\mu_{11} > 0$.

Take $\varepsilon > 0$. Let $\delta = \min(\frac{\varepsilon}{2} \mu_{11}, \varepsilon (\sum_{i=1}^{K} \|x_{1i}\| + 1)^{-1}) > 0$, and define $y = \sum_{i=1}^{K} \lambda_i v_i$, where

$$v_i = \sum_{j=1}^{p_i} \tilde{\mu}_{ij} \bar{x}_{ij} + \sum_{j=1}^{q_i} \tilde{\tau}_{ij} \bar{y}_{ij},$$

$i = 1, \ldots, K$, $\lambda_1 = \sum_{j=1}^{p_1} \tilde{\mu}_{1j} - \delta > 0$, $\lambda_i = \sum_{j=1}^{p_i} \tilde{\mu}_{ij} + \delta/(K-1) > 0$, $i = 2, \ldots, K$, $\tilde{\mu}_{11} = (\mu_{11} - \delta)/\lambda_1$, $\tilde{\mu}_{ij} = (\mu_{ij} + \delta/(K-1))/\lambda_i$, $i = 2, \ldots, K$, $\tilde{\mu}_{ij} = \mu_{ij}/\lambda_i$, $i = 1, \ldots, K$, $j = 2, \ldots, p_i$, and $\tilde{\tau}_{ij} = \tau_{ij}/\lambda_i$, $i = 1, \ldots, K$, $j = 1, \ldots, q_i$.

Then it follows that $\sum_{j=1}^{p_i} \tilde{\mu}_{ij} = 1$, $i = 1, \ldots, K$, which implies that $v_i \in V_i$, $i = 1, \ldots, K$. Hence, using that $\lambda_i > 0$, $i = 1, \ldots, K$, and $\sum_{i=1}^{K} \lambda_i = 1$, we find that $y \in W$. In fact it proves that $y \in W'$, but this we do not need.

Finally, we have that $\|x - y\| = \|\varepsilon x_{11} - \sum_{i=2}^{K} \sum_{j=1}^{p_i} \delta/(K-1)\| \leq \delta (\sum_{i=1}^{K} \|x_{1i}\|) < \varepsilon$.

Since $\varepsilon > 0$ was arbitrarily chosen, this yields $x \in W$. \Box

Without proof we state that INTERSECT, INVERT, and CLOS.Conv.Hull do not change the rationality of the description of the considered subsets, i.e., if their inputs can be described with rational halfspaces, then so can their output. This completes our description of a decomposition procedure for determining whether or not a subset is a polyhedron. It remains to prove Lemma 5.6.1.

**Proof of Lemma 5.6.1.** Follows directly from Theorem 5.6.2. \Box

Next, we consider the general case, i.e., we present a decomposition method that for a given subset $V \in \hat{U}$ finds a number of polyhedrons $V_{i_1}, \ldots, V_{i_l} \in P$, $l \in \mathbb{N}_0$, that satisfy $V = V_{i_1} \setminus V_{i_2} \setminus \cdots \setminus V_{i_l}$, in those cases that such a decomposition of $V$ exists. This decomposition method is based on ideas that arise when we consider the more general problem of finding $V_{i_1}, V_{i_2} \in \hat{P}$, that satisfy $V = V_{i_1} \setminus V_{i_2}$, for some $V \in \hat{U}$.

Assume that $V = V_{i_1} \setminus V_{i_2}$, for some unknown subsets $V_{i_1}, V_{i_2} \in \hat{P}$. In the search for the unknown sets $V_{i_1}$ and $V_{i_2}$, we note that $V = V', (V' \setminus V')$, if and only if $V' \supseteq V$. Hence, it suffices to find a $V' \supseteq V$ for which we have $V' \in \hat{P}$ and $V' \setminus V \in \hat{P}$. To find such a $V'$ we exploit the property that all subsets in $\hat{P}$ are convex, which implies that $V'$ and $V' \setminus V$ must be convex. Using that $V' \setminus V = V' \setminus (V' \setminus V') = (V' \setminus V') \cup (V' \setminus V')$, we conclude that $V' \setminus V$ is convex if $V'$ is convex and $V_{i_1} \supseteq V'$. Since $V_{i_1}$ can be any convex set satisfying $V_{i_1} \supseteq V$, it follows that $V'$ has to be the smallest convex set with $V' \supseteq V$, which is conv.hull($V'$). However, the convexity of $V'$ and $V' \setminus V$ does not necessarily guarantee that $V' \in \hat{P}$ and $V' \setminus V \in \hat{P}$. In Figure 5.21 below we give an example where the use of the convex-hull indeed does not suffice.
Figure 5.21: Example demonstrating the necessity of using the closure of the convex-hull. Note that thin boundary lines and circles are not part of the given subsets.

In Figures 5.21a and 5.21b, two subsets $V_1, V_2 \in P$ are given. Suppose that we want to decompose $V = V_1 \setminus V_2$, which is shown in Figure 5.21c. To this end we construct $V' = \operatorname{conv.hull}(V)$; see Figure 5.21d. Note that $V' \notin \hat{P}$. The only way to ensure that $V' \in \hat{P}$ is to take $V' = \operatorname{conv.hull}(V)$, in which case we have $V' \in P$ by Lemma 5.6.1. Again this implies that we have to restrict ourselves to subsets in $\hat{U}$ that have a decomposition consisting of subsets in $P$, since one can easily construct a $V = V_1 \setminus V_2$, with $V_1, V_2 \in \hat{P}$, for which there do not exist $W_1, W_2 \in P$ such that $V = W_1 \setminus W_2$. In the following theorem the above ideas are used to prove the correctness of the decomposition procedure presented in Figure 5.22, that calculates the decomposition of a given subset, and which is based on the same functions as used for DEC1.

\begin{verbatim}
procedure DEC2(input: $V \in \hat{U}$; output: $l \in \mathbb{N}_0$, $V_1, \ldots, V_l \in P$);
begin
  $l := 0$;
  $T := V$;
  while not EMPTY($T$) do
    $l := l + 1$;
    $V_l := \operatorname{CLOS.CONV.HULL}(T)$;
    $T := \operatorname{INVERT}(T)$;
    $T := \operatorname{INTERSECT}(V_l, T)$
  end;
end;
\end{verbatim}

Figure 5.22: The procedure DEC2.

Theorem 5.6.3. Let $V \in \hat{U}$ and DEC2 the procedure presented in Figure 5.22. Then we have the following.

(i) $l \in \mathbb{N}_0$, $V = V_1 \setminus V_2 \setminus \cdots \setminus V_l \setminus T$, and $V_i \in P$, for all $i = 1, \ldots, l$, are invariants of the while-loop.

(ii) The procedure terminates, if and only if $V = W_1 \setminus W_2 \setminus \cdots \setminus W_k$, for some $k \in \mathbb{N}_0$ and $W_1, \ldots, W_k \in P$. 


The MLP Complexity of CCPs

Proof. (i) It is obvious that \( l \in W_0 \) is an invariant of the while-loop. Using
induction to \( l = 0, 1, \ldots \), combined with the equality \( T = \text{conv.hull}(T') \setminus \text{conv.hull}(T) \),
\( T \) follows straightforwardly that \( V = \mathcal{V}_1 \setminus \mathcal{V}_2 \setminus \cdots \setminus \mathcal{V}_l \setminus T \) is an invariant. Similarly,
using induction to \( l = 0, 1, \ldots \), combined with Lemma 5.6.1, one shows that \( T \in T' \),
and \( V_i \subseteq P \), for all \( i = 1, \ldots, l \), are also invariants.

(ii) The ‘only-if’-part follows directly from (i), using that \( T = \emptyset \) at termination. It
remains to prove the ‘if’-part.

Let \( V = W_1 \setminus W_2 \setminus \cdots \setminus W_k \), for some \( k \in \mathbb{N}_0 \) and \( W_1, \ldots, W_k \in P \). Define the sets \( T_i \),
\( \mathcal{V}_i \), and \( Z_i \), \( i = 0, \ldots, k \), iteratively by
\[
T_0 = \emptyset,
\]
\[
Z_i = W_{i+1} \setminus Z_{i+1},
\]
\[
\mathcal{V}_0 = \mathbb{R}^n,
\]
\[
\mathcal{V}_i = \text{conv.hull}(\mathcal{V}_{i-1} \cap Z_{i-1}),
\]
\[
T_i = \mathcal{V}_i \setminus T_{i+1}.
\]

Using (5.53), one can easily shows that \( \mathcal{V}_{i-1} \cap Z_{i-1} = (\mathcal{V}_{i-1} \cap W_i) \setminus W_{i+1} \setminus \cdots \setminus W_k \), for
\( i = 1, \ldots, k \). Hence, using (5.54), \( \mathcal{V}_0 \subseteq P \), Lemma 5.6.1, and induction applied to \( i \),
we find that \( \mathcal{V}_i \subseteq P \), for \( i = 0, \ldots, k \). This implies that we also have \( \mathcal{V}_i \cap \mathcal{V}_{i-1} \cap W_i \subseteq P \), for all \( i = 1, \ldots, k \), which we use to derive
\[
\mathcal{V}_{i-1} \cap Z_{i-1} = \mathcal{V}_i \setminus Z_i,
\]
for all \( i = 1, \ldots, k \). The derivation proceeds as follows. Using (5.54) and (5.53) we
find that
\[
\mathcal{V}_{i-1} \cap Z_{i-1} = \mathcal{V}_i \cap \mathcal{V}_{i-1} \cap Z_{i-1}
\]
\[
= (\mathcal{V}_i \cap \mathcal{V}_{i-1} \cap W_i) \setminus Z_i
\]
\[
\subseteq \mathcal{V}_i \cap \mathcal{V}_{i-1} \cap W_i
\]
\[
\subseteq \mathcal{V}_i = \text{conv.hull}(\mathcal{V}_{i-1} \cap Z_{i-1}),
\]
and hence, since \( \mathcal{V}_i \cap \mathcal{V}_{i-1} \cap W_i \subseteq P \) implies that \( \mathcal{V}_i \cap \mathcal{V}_{i-1} \cap W_i \) is closed and convex,
we have that \( \mathcal{V}_i \cap \mathcal{V}_{i-1} \cap W_i = \text{conv.hull}(\mathcal{V}_{i-1} \cap Z_{i-1}) = \mathcal{V}_i \). Substituting this back
into (5.57) we get (5.56).

Next, we use (5.56) to show that
\[
T_i = \mathcal{V}_i \cap Z_i,
\]
for all \( i = 0, 1, \ldots, k \). We start with \( T_0 = V = \mathbb{R}^n \cap V = \mathcal{V}_0 \setminus Z_0 \). Next, assume (5.58)
holds for some \( i \in \{0, \ldots, k-1\} \), then from (5.56) it follows that \( T_i = \mathcal{V}_{i+1} \setminus Z_{i+1} \),
and hence
\[
T_{i+1} = \mathcal{V}_{i+1} \setminus T_i
\]
\[
= (\mathcal{V}_{i+1} \setminus (\mathcal{V}_{i+1} \setminus Z_{i+1}))
\]
\[
= \mathcal{V}_{i+1} \setminus Z_{i+1},
\]
hereby completing the proof of (5.58) by induction to \( i \). From (5.58), (5.54) and
(5.53) we conclude that \( \mathcal{V}_i \) and \( T_i \) satisfy
\[
\mathcal{V}_{i+1} = \text{conv.hull}(T_i),
\]
\[
T_{i+1} = \mathcal{V}_{i+1} \setminus T_i,
\]
for all $i = 0, \ldots, k - 1$. Since $T_0 = V$, this proves that $T = T_k$ and $V_i = V_i$ are invariants of DEC2. The result now follows from $T_k = V_k \cap Z_k = \emptyset$.

\[\begin{array}{c}
\text{(a)} \\
\text{(b)} \\
\text{(c)}
\end{array}\]

Figure 5.23: In (b) the results of the decomposition algorithm are given, when it is applied to the subset shown in (a). In (c) a subset is presented that can be classified with a 2LP, but for which the algorithm does not terminate because it has no decomposition of the described form.

Figure 5.23b presents the intermediate steps of DEC2 when it is applied to the subset $V$ of Figure 5.23a. We obtain $V = V_1 \setminus V_2 \setminus \cdots \setminus V_0$, where $V_1, V_2, \ldots, V_5$ are rectangles obviously belonging to $P$. Note that DEC2 can be used to prove that most of the example subsets presented in [Huang & Lippmann, 1987], have a decomposition of the form $V_1 \setminus V_2 \setminus \cdots \setminus V_i$, for some $i \in N_0$ and $V_1, \ldots, V_i \in P$.

Although we do not consider the computational complexity of DEC2, we mention one simplification that speeds up the procedure, because it gives more insight in the functioning of DEC2. The following theorem shows that the calculation of $T^*$ on every iteration of the loop, can be replaced by a single calculation of $V^*$ at the beginning of the procedure.

**Theorem 5.6.4.** The procedure DEC3 presented in Figure 5.24 is equivalent to the procedure DEC2 introduced in Theorem 5.6.3.

**Proof.** We prove the theorem by showing that $T = V_i \cap V^{(-i)^*}$, $i = 0, 1, \ldots$, is an invariant of DEC2, where $V_0 = R^N$, $V^{(0)^*} = V$, and $V^{(-1)^*} = V^*$.

First, we show that $V_{i+1} \subseteq V_i$, for all $i \in N_0$, where we define $V_0 = R^N$. One easily verifies that $T \subseteq V_i$ is an invariant of DEC2. Using Theorem 5.6.3, we conclude that $V_i \in P$ is also an invariant. Combining these two invariants it follows that $V_{i+1} = \text{conv hull}(T) \subseteq \text{conv hull}(V_i) = V_i$, for all $i \in N_0$.

Next, we prove with induction to $i$ that $T = V_i \cap V^{(-i)^*}$, for all $i = 0, 1, \ldots$. Obviously, this holds for $i = 0$. Assume that it holds for some $i \geq 0$, then we have

\[
\begin{align*}
V_{i+1} \setminus T &= V_{i+1} \cap T^* \\
&= V_{i+1} \cap (V_i \cap V^{(-i)^*})^* \\
&= V_{i+1} \cap (V_i^* \cup V^{(-i)^*}) \\
&= (V_{i+1} \cap V_i^*) \cup (V_{i+1} \cap V^{(-i)^*}) \\
&= V_{i+1} \cap V^{(-i)^*},
\end{align*}
\]

for all $i = 0, \ldots, k - 1$. Since $T_0 = V$, this proves that $T = T_k$ and $V_i = V_i$ are invariants of DEC2. The result now follows from $T_k = V_k \cap Z_k = \emptyset$.
procedure DEC3(input: \( V \in \hat{U} \); output: \( l \in \mathbb{N}_0, V_1, \ldots, V_l \in P \));
begin
\( l := 0; \)
\( T := V; \)
\( W := \text{INVERT}(V); \)
while not EMPTY(\( T \))
do
\( l := l + 1; \)
\( V_l := \text{CLOS.\,CONV.\,HULL}(T); \)
if \( l \mod 2 = 0 \)
then \( T := \text{INTERSECT}(V_l, V) \)
else \( T := \text{INTERSECT}(V_l, W) \)
fi
od
end;

Figure 5.34: The procedure DEC3.

which proves the induction step. Hence, \( T = V_l \cap V^{(-1)}_l \) is an invariant of DEC2, and, consequently, DEC3 is equivalent to DEC2. \( \square \)

In some sense, the use of Theorem 5.6.3 seems a bit paradoxical, because one can only compute the decomposition of a subset if it is known to exist, and the latter can only be guaranteed by giving this decomposition. However, DEC2 can also be used to find out whether a given subset \( V \in \hat{U} \) can be decomposed. If the procedure terminates when applied to \( V \), then it follows from Theorem 5.6.3 that a decomposition is found. If on the other hand the procedure does not terminate, then Theorem 5.6.3 implies that there does not exist a decomposition of \( V \).

The remaining problem is to determine whether DEC2 or, equivalently, DEC3 terminates. When the procedure is applied to \( V \in \hat{U} \), then one can easily show that \( V_{l+1} \subseteq V_l \), for all \( l \in \mathbb{N} \); see the proofs of Theorem 5.6.3 and Theorem 5.6.4. We claim that the procedure does not terminate, if and only if \( V_{l+1} = V_l \), for some \( l \in \mathbb{N} \), or in other words, the situation where \( V_{l+1} \subseteq V_l \), for all \( l \in \mathbb{N} \) cannot occur. The first part of this claim is expressed in the following theorem.

**Theorem 5.6.5.** Let DEC2 be applied to a subset \( V \in \hat{U} \). Then the procedure does not terminate, if at some stage \( k \) in the execution of the procedure \( V_{k+1} \) equals \( V_k \).

**Proof.** Let \( T_l \) denote the set \( T \) at stage \( l \) of the procedure, then \( V_{k+1} = \text{conv} \text{hull}(T_k) \) and \( T_{l+1} = V_{l+1} \setminus T_l \). Assume that \( V_{k+1} = V_k \) at stage \( k \) of the procedure, then we have that

\[
T_{k+1} = V_{k+1} \setminus T_k = V_k \setminus (V_k \setminus T_{k-1})
\]
5.7 General CCPs

\[ V_k \cap T_{k-1} = T_{k-1} \]

using that \( V_k = \text{conv.hull}(T_{k-1}) \supseteq T_{k-1} \). Since the procedure reaches stage \( k \), it follows that \( T_i \neq \emptyset \), for all \( i \leq k \). Hence, \( T_{k+1} \neq \emptyset \) and the procedure reaches stage \( k+1 \), where it calculates \( V_{k+2} = \text{conv.hull}(T_{k+1}) = \text{conv.hull}(T_{k-1}) = V_k = V_{k+1} \). Using the same argument in an iterative way, we find that \( T_{l+1} = T_{l-1} \neq \emptyset \) and \( V_{l+1} = V_l \), for all \( l \geq k \), which proves that the procedure does not terminate. □

Although we believe that the opposite of Theorem 5.6.5 is also true, we have not been able to prove this. We therefore present it as a conjecture with a sketch of a possible proof.

**Conjecture 5.6.1.** Let DEC2 be applied to a subset \( V \in \hat{U} \). Then the procedure does not terminate, only if at some stage \( k \) in the execution of the procedure \( V_{k+1} \) equals \( V_k \).

A proof of the above conjecture might be constructed along the following lines. Since \( V \in \hat{U}, V \) is defined using a number, say \( n \), of affine halfspaces. Assume that the procedure when applied to \( V \) does not terminate, then at stage \( n + 2 \) we have that \( V = V_1 \setminus V_2 \setminus \cdots \setminus V_{n+2} \) \( \supseteq V_1 \supseteq V_2 \supseteq \cdots \supseteq V_{n+2} \supseteq T \). To complete the proof we must show that \( V_1 \supseteq V_2 \supseteq \cdots \supseteq V_{n+2} \supseteq T \). The first step is to observe that if \( V_i, V_{i+1} \in F \) and \( V_i \supseteq V_{i+1} \), then one of the halfspaces that defines \( V_i \) is not needed for defining \( V_{i+1} \). The essential part of the proof is to show that this implies that \( n + 1 \) distinct halfspaces can be chosen, one for every pair \( V_i, V_{i+1} \), that are all needed for the definition of \( V \). If correct, the above argument would imply that if DEC2 does not terminate by itself, then after at most \( n + 1 \) iterations of the loop the procedure can be stopped since one may conclude that it will never terminate.

As an example of a subset for which the procedure does not terminate we present the subset of Figure 5.23c. One can easily verify that for this subset we obtain \( V_2 = V_1\), which implies that \( V_2 = V_3 = V_4 = \ldots \), and proves that this subset has no decomposition of the proposed form. However, this subset can be classified with a 2LP, which proves that the sufficient condition of Corollary 5.5.11 is not a necessary condition.

In the following section we discuss the possibility of translating the results for BCPs, that were obtained in this section and the previous sections, to results for general CCPs.

5.7 General CCPs

In this section we briefly discuss the MLP-complexity of combinatorial classification problems in general. The objective of this section is to show how the results obtained for complete disjoint binary classification problems (BCPs) can be used to obtain results for arbitrary combinatorial classification problems (CCPs); see for a definition of these notions Section 5.1. This is done by demonstrating that an arbitrary CCP can be reduced to a number of incomplete disjoint BCPs. Therefore, we start by considering how to solve an arbitrary incomplete disjoint BCP with an MLP.
Consider a BCP represented by \((\Omega, L, \Gamma)\), with \(\Omega \subseteq \mathbb{R}^N\), for some \(N \in \mathbb{N}\), \(L = B\), \(\Gamma = \{\Omega_0, \Omega_1\}\), \(\Omega_0, \Omega_1 \subseteq \Omega\), and \(\Omega_0 = \Omega \setminus \Omega_1\). It is obvious that any MLP that solves the BCP \((\mathbb{R}^N, L, \Gamma)\), also solves \((\Omega, L, \Gamma)\). Furthermore, \((\mathbb{R}^N, L, \Gamma)\) is a classification problem of the kind that is treated in the first part of this chapter. It can be solved by finding an MLP represented by \(f \in \mathcal{R}_{m,N,1}\), for some \(m \in \mathbb{N}\), such that \(\mathcal{J}(f) = \Omega_1\). As we know from Section 5.3, this is possible if and only if \(\Omega_1 \in \mathcal{U}\).

Although it is sufficient, \(\Omega_1 \in \mathcal{U}\) is not a necessary condition for finding an MLP that solves \((\Omega, L, \Gamma)\). This follows from a more precise observation, and the result is formulated in Proposition 5.7.1. The proof of this result is trivial and therefore omitted.

**Proposition 5.7.1.** Let \((\Omega, \Gamma, \Gamma)\) be a BCP with \(\Omega \subseteq \mathbb{R}^N\), for some \(N \in \mathbb{N}\), \(L = B\), \(\Gamma = \{\Omega_0, \Omega_1\}\), \(\Omega_0, \Omega_1 \subseteq \Omega\), and \(\Omega_0 = \Omega \setminus \Omega_1\). Let \(m \in \mathbb{N}\). Then an \(m\)LP with \(N\) inputs and one output node, represented by \(f \in \mathcal{R}_{m,N,1}\), solves \((\Omega, L, \Gamma)\), if and only if \(\Omega_1 \in \mathcal{J}(f) \cap \Omega\).

Proposition 5.7.1 implies that the problem becomes to find an \(m \in \mathbb{N}\) and a \(V \subseteq C_{m,N}\) such that \(\Omega_1 = V \cap \Omega\) or, equivalently, to find an \(m \in \mathbb{N}\) and a \(V \subseteq C_{m,N}\) such that \(\Omega_1 \subseteq V \subseteq \mathbb{R}^N \setminus \Omega_0 = \Omega_0^c\). We do not know of a general approach for solving this extended problem. Therefore, we advise to consider this aspect to be handled separately for each CCP.

Next, we show that every CCP can be reduced to a number of BCPs of the kind described by Proposition 5.7.1. To this end we first discuss Lemma 5.7.1 below, which considers the special case that all the classifying subsets are disjunct.

**Lemma 5.7.1.** Let \((\Omega, L, \Gamma)\) be a BCP with \(\Omega \subseteq \mathbb{R}^N\), for some \(N \in \mathbb{N}\), \(L \subseteq B^K\), for some \(K \in \mathbb{N}\), \(\Gamma = \{\Omega_1 \subseteq \Omega\} \cap L\), \(\cup_{l \in L} \Omega_l = \Omega_1\), and \(\Omega_l \cap \Omega_k = \emptyset\), for all \(l \neq k\). Let, for all \(i = 1, \ldots, K\) and \(q = 0, 1\), the set \(V_q^{(i)}\) be defined by

\[
V_q^{(i)} = \bigcup_{l \in L, l \in L_q} \Omega_l.
\]

(5.59)

Then the following results hold.

(i) \(V_0^{(i)} \cup V_1^{(i)} = \Omega\), for all \(i = 1, \ldots, K\).

(ii) \(V_0^{(i)} \cap V_1^{(i)} = \emptyset\), for all \(i = 1, \ldots, K\).

(iii) For all \(l \in B^K\) and \(x \in \mathbb{R}^N\), \(l \in L\) and \(x \in \Omega_l\), if and only if \(l \in B\) and \(x \in V_1^{(i)}\), for all \(i = 1, \ldots, K\).

**Proof.** (i) Follows from (5.59) by

\[
V_0^{(i)} \cup V_1^{(i)} = \left(\bigcup_{l \in L, l \in L_q} \Omega_l\right) \cup \left(\bigcup_{l \in L, l \in L_q} \Omega_l\right) = \bigcup_{l \in L, l \in \{0, 1\}} \Omega_l = \Omega.
\]

(ii) Follows from (5.59) by

\[
V_0^{(i)} \cap V_1^{(i)} = \left(\bigcup_{l \in L, l \in L_q} \Omega_l\right) \cap \left(\bigcup_{l \in L, l \in L_q} \Omega_l\right) \subseteq U(\Omega) \cap U(\Omega) = \emptyset.
\]

(iii) First, we prove the 'only-if'-part. Assume \(x \in \Omega_l\) for some \(l \in L\), then obviously
5.7 General CCPs

$l_i \in B$, for all $i = 1, \ldots, K$. Furthermore, using (5.59), it follows directly that $x \in V_i^{(0)}$, for all $i = 1, \ldots, K$.

It remains to prove the 'if'-part. Let, for all $i = 1, \ldots, K$, $l_i \in B$ be such that $x \in V_i^{(0)}$. Then, using (i), it follows that $x \in \Omega_k$, for some $k \in L$, which implies $x \in V_k^{(0)}$, for all $i = 1, \ldots, K$. The case that $k_i \neq l_i$, for some $i \in \{1, \ldots, K\}$, cannot occur because of (ii). Hence, $k = l$, and the result follows.

Theorem 5.7.1 gives the main result of this section. It states that a CCP can be solved by an MLP, if and only if it can be replaced by an equivalent CCP that is reducible to a number of BCPS, of kind described by Proposition 5.7.1.

**Theorem 5.7.1.** Let $(\Omega, L, \Gamma)$ be a CCP with $\Omega \subseteq \mathbb{R}^N$, for some $N \in \mathbb{N}$, $L \subseteq B^K$, for some $K \in \mathbb{N}$, and $\Gamma = \{\Omega_i \subseteq \Omega \mid i \in L\}$. Let $m \in \mathbb{N}$. Then an mLP with $N$ inputs and $K$ outputs, represented by $f \in R_{m,N,K}$, solves $(\Omega, L, \Gamma)$, if and only if there exists a CCP represented by $(\tilde{\Omega}, \tilde{L}, \tilde{\Gamma})$, with $\tilde{\Gamma} = \{\tilde{\Omega}_l \subseteq \tilde{\Omega} \mid l \in \tilde{L}\}$, that satisfies the following conditions.

1. $\bigcup_{l \in \tilde{L}} \tilde{\Omega}_l = \tilde{\Omega} = \Omega$.
2. $\tilde{\tilde{L}} \subseteq \tilde{L}$.
3. $\tilde{\Omega}_l \subseteq \tilde{\Omega}_k$, for all $l \in \tilde{L}$.
4. $\tilde{\Omega}_l \cap \tilde{\Omega}_k = \emptyset$, for all $l, k \in \tilde{L}$, $l \neq k$.
5. $\tilde{V}_i^{(0)} = \mathcal{J}(f_i) \cap \tilde{\Omega}_i$, for all $i = 1, \ldots, K$, where for all $i = 1, \ldots, K$ and $q = 0, 1$, the set $V_q^{(0)}$ is defined by

$$V_q^{(0)} = \bigcup_{i \in \tilde{L}, l \in \tilde{L}} \tilde{\Omega}_l.$$ (5.60)

**Proof.** The 'if'-part. It follows from (i), (ii), and (iv), that $(\tilde{\Omega}, \tilde{L}, \tilde{\Gamma})$ satisfies the conditions of Lemma 5.7.1. This, combined with (v), yields that $V_0^{(0)} \cup \tilde{V}_1^{(0)} = \tilde{\Omega}$ and $V_0^{(0)} \cap \tilde{V}_1^{(0)} = \emptyset$, for all $i = 1, \ldots, K$. Hence, using (v) once more, it follows that

$$\tilde{V}_i^{(0)} = \tilde{\Omega} \setminus \tilde{V}_1^{(0)} = \tilde{\Omega} \setminus (\mathcal{J}(f_1) \cap \tilde{\Omega}) = \tilde{\Omega} \setminus \{x \in \tilde{\Omega} \mid f_1(x) = 1\} = \{x \in \tilde{\Omega} \mid f_1(x) = 0\},$$

for all $i = 1, \ldots, K$. This implies that $x \in \tilde{V}_i^{(0)}$, for all $x \in \tilde{\Omega}$ and $i = 1, \ldots, K$.

Using the third result of Lemma 5.7.1, we find that $f(x) \in \tilde{L}$ and $x \in \tilde{\Omega}_f(x)$, for all $x \in \tilde{\Omega}$. Combining this with (i), (ii), and (iii), it follows that $f$ solves $(\Omega, L, \Gamma)$.

The 'only-if'-part. If $f$ solves $(\Omega, L, \Gamma)$ then $f(x) \in \tilde{L}$ and $x \in \tilde{\Omega}_f(x)$, for all $x \in \tilde{\Omega}$. Define $(\tilde{\Omega}, \tilde{L}, \tilde{\Gamma})$ with $\tilde{\Gamma} = \{\tilde{\Omega}_l \mid l \in \tilde{L}\}$, by $\tilde{\Omega} = \Omega$, $\tilde{L} = \{f(x) \mid x \in \Omega\}$, and $\tilde{\Omega}_l = \{x \in \Omega \mid f_1(x) = 1\}$, for all $l \in \tilde{L}$. Then (i), (ii), (iii), and (iv) follow directly. Since by definition $f(x) \in \tilde{L}$ and $x \in \tilde{\Omega}_f(x)$, for all $x \in \tilde{\Omega}$, (v) follows from Lemma 5.7.1.

Using Theorem 5.7.1, Corollary 5.3.1, and the above, we obtain a necessary and sufficient condition for a CCP to be solvable by an MLP.
Corollary 5.7.1. Let \((\Omega, L, \Gamma)\) be a CCP with \(\Omega \subseteq \mathbb{R}^N\), for some \(N \in \mathbb{N}\), \(L \subseteq \mathcal{B}^K\), for some \(K \in \mathbb{N}\), and \(\Gamma = \{\Omega_l \subseteq \Omega | l \in L\}\). Then there exists an MLP that solves \((\Omega, L, \Gamma)\), if and only if there exists a CCP represented by \((\bar{\Omega}, \bar{L}, \bar{\Gamma})\), with \(\bar{\Gamma} = \{\bar{\Omega}_l \subseteq \bar{\Omega} | l \in \bar{L}\}\), that satisfies the Conditions (i), (ii), (iii), and (iv), given in Theorem 5.7.1, and such that for all \(i = 1, \ldots, K\), there exists a subset \(V \subseteq \bar{U}\) with \(V_i \subseteq V \subseteq (V_i)^\circ\), where for all \(i = 1, \ldots, K\) and \(q = 0, 1\), the set \(V_i^{(q)}\) is defined by (5.60).

Although of theoretical interest, the result of Corollary 5.7.1 seems practically of little use, due to the fact that the conditions posed on \((\bar{\Omega}, \bar{L}, \bar{\Gamma})\) are not easily verified. Below we discuss a sufficient condition that is much easier to verify; see Corollary 5.7.2. The underlying ideas are explained using the following argument.

Theorem 5.7.1 lodges a two-step approach for finding an MLP that solves a given CCP. First, one must find a CCP represented by \((\bar{\Omega}, \bar{L}, \bar{\Gamma})\) that satisfies Conditions (i) through (iv) of Theorem 5.7.1. Secondly, one must find \(K\) one-output mLPS, for a certain \(m \in \mathbb{N}\), that solve the BCPs represented by \((\bar{\Omega}, \bar{B}, \{\bar{V}_i^{(0)}, \bar{V}_i^{(0)}\})\), for \(i = 1, \ldots, K\), respectively. Combining these \(K\) one-output mLPS into one \(K\)-output MLP, yields the desired MLP that solves \((\Omega, L, \Gamma)\).

The second step of this approach has already been discussed below Proposition 5.7.1. It remains to consider the first step, which is done in Theorem 5.7.2. This theorem describes a general method for finding a CCP \((\tilde{\Omega}, \tilde{L}, \tilde{\Gamma})\) with disjunct subsets, that is equivalent to \((\Omega, L, \Gamma)\). The method is based on an arbitrary injection function on \(\mathcal{B}^K\), which quantifies a certain preference on the collection of subsets \(\Gamma\). If in \(\tilde{\Gamma}\), an object \(x \in \tilde{\Omega}\) is a member of a number of subsets \(\tilde{\Omega}_l\), the construction of \(\tilde{\Gamma}\) guarantees that in \(\tilde{\Gamma}\), \(x\) is a member of one \(\tilde{\Omega}_l\) only, corresponding to the \(\Omega_l\) for which the label has the highest preference value.

Theorem 5.7.2. Let \((\Omega, L, \Gamma)\) be a CCP with \(L \subseteq \mathcal{B}^K\), for some \(K \in \mathbb{N}\). Let \(p : \mathcal{B}^K \to \mathbb{R}\) be a function with \(p(l) \neq p(k)\), for all \(l, k \in \mathcal{B}^K\), \(l \neq k\). Let \((\bar{\Omega}, \bar{L}, \bar{\Gamma})\) represent a CCP defined by \(\bar{\Omega} = \Omega\), \(\bar{L} = L\), and \(\bar{\Gamma} = \{\bar{\Omega}_l | l \in L\}\), where

\[
\bar{\Omega}_l = \Omega_l \cap \bigcap_{k \in L \setminus \{l\}} \Omega_k^* \tag{5.61}
\]

for all \(l \in L\). Then \((\bar{\Omega}, \bar{L}, \bar{\Gamma})\) satisfies (i), (ii), (iii), and (iv) of Theorem 5.7.1.

Proof. That the CCP \((\bar{\Omega}, \bar{L}, \bar{\Gamma})\) defined above satisfies (ii) and (iii), follows directly from its definition. Let \(x \in \bar{\Omega} = \bar{\Omega}_l\), then \(x \in \Omega_l\), for some \(l \in L\). Let \(l\) be the highest, with respect to the value of the preference function \(p(l)\), element of \(L\), such that \(x \in \Omega_l\) and \(x \notin \Omega_k\), for all \(k \in L\) with \(p(k) > p(l)\), which by (5.61) implies that \(x \in \Omega_l\). Hence, using also that \(\bar{\Omega} = \Omega\) and \(\bar{L} = L\), we find (i). Finally, we must show that (iv) holds. This follows from the fact that \(\tilde{\Omega}_l \cap \tilde{\Omega}_k \subseteq \Omega_l \cap \Omega_k^* = \emptyset\), in case that \(p(l) > p(k)\), \(\tilde{\Omega}_l \cap \tilde{\Omega}_k \subseteq \Omega_l^* \cap \Omega_k = \emptyset\), in case that \(p(k) > p(l)\), and \(p(l) > p(k)\) or \(p(k) > p(l)\), if \(k \neq l\).

Obviously, there may well exist other methods for constructing a CCP \((\bar{\Omega}, \bar{L}, \bar{\Gamma})\) that satisfies the conditions of Theorem 5.7.1. It is important to note that the choice of \((\bar{\Omega}, \bar{L}, \bar{\Gamma})\) may affect the complexity, with respect to the number of layers and the
number of nodes, of the MLPs that solve \((\tilde{\Omega}, B, \{\tilde{V}^{(0)}, \tilde{V}^{(1)}\})\), if they exist. Here, we do not address these issues in general, except for the trivial result noted in Corollary 5.7.2 below, which discusses the existence of a 3LP that solves the considered CCP. In the following chapter we discuss some complexity questions for the class of classification problems that are obtained from combinatorial optimization problems. Combining the complexity results for complete disjoint BCPs found in the first part of this chapter, with the solution approach described by Theorem 5.7.1 and Theorem 5.7.2, we obtain a number of explicit results on the complexity of MLPs that solve combinatorial optimization problems.

**Corollary 5.7.2.** Let \((\Omega, L, \Gamma)\) be a CCP with \(\Omega \subseteq \mathbb{R}^{N}\), for some \(N \in \mathbb{N}\), \(L \subseteq \mathbb{B}^{K}\), for some \(K \in \mathbb{N}\), \(\Gamma = \{\Omega_{l} \subseteq \Omega \mid l \in L\}\), and \(\Omega_{l} \subseteq \tilde{\Omega}\), for all \(l \in L\). Then there exists a 3LP represented by an \(f \in \mathbb{R}^{3,N,K}\) that solves \((\Omega, L, \Gamma)\).

### 5.8 Concluding remarks

In this chapter we gave a characterization of the classification capabilities of multilayered perceptrons (MLPs) with real-valued inputs and hard-limiting response functions. The literature does not contain many previous results in this area. Most results found in the sixties and the seventies concern results on feedforward networks with binary-valued inputs; see Amari [1990], Minsky & Papert [1969], and Murgoca [1971] for overviews of the most important results of that time period. After the revival of the interest in neural networks, which is mainly due to the discovery of the backpropagation learning algorithm by Rumelhart, Hinton & Williams [1986], the need was felt to know more about the capabilities of MLPs that use real-valued inputs. One of the first to study the capabilities of this type of neural networks was Lippmann. In his introductory paper, Lippmann [1987] briefly considers the classification capabilities of 1LPs, 2LPs, and 3LPs, with real-valued inputs and hard-limiting response functions. He summarizes his conclusions in a well-known and often quoted figure, which shows examples of decision regions that are classifiable with a 1LP, a 2LP, and a 3LP, respectively. The figure demonstrates that 1LPs can classify affine halfspaces, that 2LPs can classify piece-wise linearly bounded and convex subsets, and that 3LPs can classify arbitrary complex piece-wise linearly bounded subsets. However, the interpretation of this figure is usually that 2LPs can classify convex subsets only, and that 3LPs can classify arbitrary complex subsets. Later, several people, including Lippmann, presented hand-crafted examples of non-convex decision regions that can be classified with a 2LP; see for instance Huang & Lippmann [1987], Makhouk, Schwartz & El-Jaroudi [1989], Wieland & Leighton [1987], Amari [1990], and Li [1991]. However, neither of the examples presented in these papers resulted in some kind of general result on the classification capabilities of 2LPs.

This is probably one of the reasons why many researchers considered the approximate classification capabilities of MLPs. One of the obtained results shows that 2LPs, that use the hard-limiting response function, can approximately classify any subset; see Cybenko [1989] and also Chapter 2. Although of theoretical importance, these results are often not very useful in practice, since they usually do not say anything about the complexity of the classification problem at hand, with respect to the
required number of hidden nodes. In some cases only a very rough upper bound on
the required number of nodes can be given; see for instance Cybenko [1990], Barron
[1991], and Chapter 2.

Our approach to determine the complexity of classification problems is based on a
precise characterization of the classification capabilities of MLPs, with and without
a bounded number of first-layer nodes. In the unbounded case, this resulted in
a complete and fairly trivial characterization of the capabilities of MLPs, except
for 2LPs; see Section 5.3, and for some earlier results [Zwietering, Aarts & Wessels,
1991a]. The capabilities of 2LPs are described by presenting necessary and sufficient
conditions for a subset to be classifiable with a 2LP; see the Sections 5.5.1, 5.5.2,
and 5.5.3. One of the necessary conditions, the bow-tie condition, appears without
a proof in a paper by Gibson & Cowan [1990], and with an intricate proof in one
of our previous papers [Zwietering, Aarts & Wessels, 1992a]. The other conditions
are new in literature. Our description of the capabilities of 2LPs is not complete,
since the presented necessary conditions and sufficient conditions match for one-
dimensional subsets only. However, we give some arguments that indicate that the
gap in general not very large, at least not for two- and three-dimensional subsets;
see the Sections 5.5.1 and 5.5.4. Obviously, one would like to have a methodical
approach for verifying whether a subset is classifiable with a 2LP, and which yields
the description of that 2LP in case that it exists. We have presented a methodical
approach for verifying whether a subset satisfies one of our sufficient conditions for
a subset to be classifiable with a 2LP; see Section 5.6. A preliminary version of this
section appears in a previous paper [Zwietering, Aarts & Wessels, 1992a]. It turns
out that, although the considered condition is rather simple, finding a methodical
approach for verifying this condition is already quite complicated.

The results on the capabilities of MLPs with a bounded number of first-layer nodes,
yield a lower bound on the number of first-layer nodes required to classify a given
subset with any MLP; see Section 5.4. These results are used in Chapter 6 to
find the minimal number of first-layer nodes required to solve a given combinatorial
optimization problem (COP). Furthermore, the sufficient condition for the existence
of a 3LP, and the necessary condition for the existence of a 2LP, that classify a given
subset, are translated into the corresponding conditions for the existence of a 3LP,
respectively a 2LP, that solves a given COP.
Chapter 6

The MLP Complexity of COPs

6.1 Introduction

In this chapter we study the complexity of combinatorial optimization problems, with respect to the number of layers and the number of nodes required by multi-layered perceptrons that solve these problems. For an introduction to multi-layered perceptrons (MLPs), combinatorial optimization problems (COPs), and the complexity of problems in general, we refer to Chapter 2, Chapter 3, and Chapter 4, respectively. The main idea applied in this chapter is to translate a COP into an equivalent combinatorial classification problem (CCP) and, then, to use the results of Chapter 5 on the complexity of MLPs for solving CCPs. Therefore, similarly as in Chapter 5, we only consider MLPs that use the hard-limiting response function \( \theta \), for solving COPs; see also Chapter 2, where this kind of MLPs is denoted \( \theta \)-MLPs. Consequently, we assume that the considered COPs are given using a 0-1 formulation; see Chapter 3 for details.

Using the results of Chapter 5, we formulate in Section 6.2 a set of conditions that guarantee that a given COP can be solved by an MLP. As we know from Chapter 5, the existence of an MLP implies the existence of a 3LP that solves the considered COP. We present an explicit construction of a 3LP for some special classes of COPs. One of these special classes concerns COPs for which the set of feasible solutions is parameter independent, and for which the cost function is an affine function of the parameter. This is the class of COPs that is further studied in the Sections 6.3 and 6.4. In Section 6.3 we present the main results of this chapter, which concern an expression for the minimal number of first-layer nodes of any MLP that solves a given COP of the class mentioned above. Again, we present constructive results for a 3LP with the minimal number of first-layer nodes that solves the COP. We give a description how to calculate the minimal number of first-layer nodes for a specific COP, and we present upper and lower bounds on the required number of first-layer nodes, that can be used if the calculation becomes too complex. A large part of this section deals with issues that originate from local search, a well-known technique for handling COPs. In fact, this section present results that links the construction of MLPs with a minimal sized first layer, to the concepts related to the minimal exact
neighborhood. We note that most of the results of Section 6.3 are based on the results about the minimal number of first-layer nodes required for solving a given CCP, that are derived in Section 5.4.

Finally, Section 6.4 examines the question whether a COP can be solved by a 2LP. The results on the classification capabilities of 2LPs, that have been presented in Chapter 5, are used to obtain two necessary conditions for COPs to be solvable by a 2LP. The first of these conditions is based on the bow-tie condition and the second is based on the twisted bow-tie condition; see also Section 5.5.1. We have not been able to formulate general sufficient conditions for a COP to be solvable by a 2LP. Consequently, in order to show that a given COP can be solved by a 2LP, we have to give a tailored proof for this problem.

In Chapter 7, the results of the present chapter will be applied to the five COPs that have been introduced in Chapter 3.

6.2 Sufficient conditions for the existence of a 3LP

In this section we derive a set of sufficient conditions for a given 0-1 formulated COP to be solvable by an MLP with hard-limiting response functions. In fact these conditions guarantee that the considered COP can be solved by a 3LP. For some special cases, an explicit construction of a 3LP that solves a given COP is given, which is based on a complete enumeration of the set of feasible solutions of the considered COP.

The results presented in this section are obtained by translating COPs to CCPs, and subsequently using the results of Section 5.7, which presents a sufficient condition for a CCP to be classifiable with an MLP. In order to obtain an explicit construction of a 3LP that solves a COP, the corresponding CCP has to be decomposed into a number of binary classification problems (BCPs), which also follows the lines of Section 5.7. The final step is then to construct 3LPs that solve the obtained BCPs, for which we use some parts of the general construction of 3LPs that solve BCPs, that was presented in Section 5.3.

Before we introduce a CCP that is equivalent to a considered COP, we repeat the definition of a COP given in Chapter 3, tailored to 0-1 formulated COPs, and give a definition of solving a COP with an MLP.

Definition 6.2.1. A 0-1 formulated combinatorial optimization problem is given by a 3-tuple \((I, F, c)\), where (i) for some \(N \in \mathbb{N}\), \(I \subseteq \mathbb{R}^N\) denotes a set of instance defining parameters, (ii) for some \(K \in \mathbb{N}\) and for all \(x \in I\), \(F(x) \subseteq \mathbb{B}^K\) denotes a set of feasible solutions for the instance defined by \(x\), and (iii) for all \(x \in I\), \(c(\cdot; x) : F(x) \to \mathbb{R}\) denotes a cost function on the set of feasible solutions for the instance defined by \(x\). For a given instance \((F(x), c(\cdot; x))\) defined by some \(x \in I\), the problem is to find a feasible solution with minimal cost, i.e., we must find an \(y \in F(x)\), such that \(c(y; x) \leq c(z; x)\), for all \(z \in F(x)\). It is assumed that \(F(x) \neq \emptyset\), for all \(x \in I\), which guarantees that each instance of this problem can be solved.

Note that although \(c(y; z)\) is meaningful for \(x \in I\) and \(y \in F(x)\) only, we technically
6.2 Sufficient conditions for the existence of a 3LP

assume that \( c(y, z) \) represents a real number for all \( z \in \mathbb{R}^N \) and \( y \in \mathbb{B}^K \); see also the examples discussed in Chapter 3. The following definition formalizes what we mean by solving a COP by a MLP.

**Definition 6.2.2.** A COP \((I, F, c)\), with \( I \subseteq \mathbb{R}^N \) and \( F(\cdot) \subseteq \mathbb{B}^K \), for some \( N, K \in \mathbb{N} \), is solved by an MLP with \( N \) inputs and \( K \) outputs, represented by the function \( f : \mathbb{R}^N \rightarrow \mathbb{B}^K \), if \( f(x) \in F(z) \) and \( c(f(x); z) \leq c(z; x) \), for all \( x \in I \), and for all \( z \in F(x) \).

Definitions 6.2.1 and 6.2.2 are analogous to Definitions 5.1.1 and 5.1.2, defining a CCP and its MLP solution, respectively. For each COP of the form given by Definition 6.2.1, we define a corresponding CCP in the following way.

**Definition 6.2.3.** Let \((I, F, c)\) be a COP. Then the CCP corresponding to this COP is given by \((\Omega, L, \Gamma)\), with \( \Omega \), \( L \), and \( \Gamma \) defined by

(i) \( \Omega = I \), i.e., the set of objects corresponds to the set of instance parameters,

(ii) \( L = \bigcup_{x \in I} F(x) \), i.e., the set of labels corresponds to the set of all feasible solutions, and

(iii) \( \Gamma = \{ \Omega_y \mid y \in L \} \), where for each \( y \in L \), the subset \( \Omega_y \subseteq I \) denotes the set of parameters for which \( y \) is a feasible, optimal solution, and is given by

\[
\Omega_y = \{ x \in I \mid y \in F(x) \land [\forall z \in F(x) : c(y; z) \leq c(z; x)] \}. \quad (6.1)
\]

One can verify that if the COP \((I, F, c)\) satisfies the requirements of Definition 6.2.1, its corresponding CCP \((\Omega, L, \Gamma)\) satisfies the requirements of Definition 5.1.1. Furthermore, the COP and its corresponding CCP are equivalent, in the sense that for each instance defining parameter \( x \in I \), the problem of finding a feasible solution \( y \in F(x) \), such that \( c(y; x) \leq c(z; x) \), for all \( z \in F(x) \), is equivalent to finding a label \( y \in L \), such that \( x \in \Omega_y \). Consequently, both the considered COP and its corresponding CCP have the same MLP-complexity, which is formalized by the following proposition.

**Proposition 6.2.1.** Let \((I, F, c)\) be a COP with \( I \subseteq \mathbb{R}^N \) and \( F(\cdot) \subseteq \mathbb{B}^K \), for some \( N, K \in \mathbb{N} \), and let \((\Omega, L, \Gamma)\) be its corresponding CCP. Then an MLP with \( N \) inputs and \( K \) outputs solves \((I, F, c)\), if and only if it solves \((\Omega, L, \Gamma)\).

Proposition 6.2.1 implies that it suffices to study the complexity of the CCP corresponding to a given COP, which can be done using the results of Chapter 5. The first result is a sufficient condition for the existence of an MLP that solves the COP at hand, which is obtained from Corollary 5.7.2. Recall that \( P \), \( \bar{P} \), and \( \bar{U} \), denote the set of all polyhedrons, the set of all pseudo polyhedrons, and the set of all unions of pseudo polyhedrons, respectively, as defined in Definition 5.2.3 and Definition 5.2.6.

**Theorem 6.2.1.** Let \((I, F, c)\) be a COP with \( I \subseteq \mathbb{R}^N \) and \( F(\cdot) \subseteq \mathbb{B}^K \), for some \( N, K \in \mathbb{N} \), and let \((\Omega, L, \Gamma)\) be its corresponding CCP. Let the subsets \( \Phi_y \subseteq I \) and \( \Psi_{y,z} \subseteq I \) be defined by

\[
\Phi_y = \{ x \in I \mid y \in F(x) \}, \quad (6.2)
\]

for all \( y \in L \), and

\[
\Psi_{y,z} = \{ x \in I \mid c(y; z) \leq c(z; x) \}, \quad (6.3)
\]
for all \( y, z \in L \), respectively, and let \( \Phi_y \in \hat{U} \) and \( \Psi_{y,z} \in \hat{U} \), for all \( y, z \in L \). Then there exists a 3LP that solves \((I, F, c)\).

**Proof.** One easily verifies that

\[
\Omega_y = \{ x \in I \mid y \in F(x) \land \forall z \in F(x) : c(y; z) \leq c(z; x) \} = \{ x \in I \mid y \in F(x) \land \forall z \in L : z \notin F(x) \lor c(y; z) \leq c(z; x) \} = \{ x \in I \mid y \in F(x) \} \cap \bigcap_{z \in L} (\Phi_z^* \cup \Phi_{y,z}).
\]

Consequently, if \( \Phi_y \in \hat{U} \) and \( \Psi_{y,z} \in \hat{U} \), for all \( y, z \in L \), then, using the results of Section 5.2, it follows that \( \Omega_y \in \hat{U} \), for all \( y \in L \). Combining Corollary 5.7.2 with Proposition 6.2.1, then yields the desired result. 

The condition given in Theorem 6.2.1 is not a necessary condition. One can easily find artificially constructed COPs that do not satisfy the condition of Theorem 6.2.1, and that are still solvable by an MLP. A necessary and sufficient condition can be derived from Corollary 5.7.1, but this leads to a condition that is not easily verifiable; see also Section 5.7. The reason for giving the formulation used in Theorem 6.2.1 is that we do not expect that there exist real-world COPs that do not satisfy the condition of Theorem 6.2.1 and that are still solvable by an MLP.

Using the proof of Theorem 6.2.1 and the construction of a 3LP that classifies an arbitrary subset in \( \hat{U} \), which is given in Section 5.3, one can in principle construct a 3LP that solves a COP that satisfies the conditions of Theorem 6.2.1. However, since the conditions posed in Theorem 6.2.1 do not present any restrictions on the complexity of the functions \( F \) and \( c \), nothing can be said about the complexity of the resulting 3LP with respect to its number of hidden nodes. In other words, to be able to present statements about the complexity of MLPs, we have to introduce restrictions on the set of COPs defined by \( I, F, \) and \( c \). In the remainder of this chapter we therefore make the following restrictions.

**Condition 1a.** The set of feasible solutions is parameter independent, i.e., \( F(x) = F(x') \), for all \( x, x' \in I \). In the remainder we use \( F \) to denote the set of feasible solutions.

**Condition 1b.** For all \( y \in F, c(y; \cdot) : \mathbb{R}^n \to \mathbb{R} \) is an affine function, i.e., \( c(y; z) = a(y)^T z + b(y) \), for some \( a(y) \in \mathbb{R}^n, b(y) \in \mathbb{R} \), and all \( x \in \mathbb{R}^n \).

Below, we elaborate on these conditions in more detail.

One can verify that a COP that satisfies Conditions 1a and 1b, satisfies the conditions of Theorem 6.2.1 in case that \( I \in \hat{U} \). Below we show that the condition \( I \in \hat{U} \) is not necessary for the existence of a 3LP that solves the considered COP. However, Condition 1a is essential for the construction of the 3LP given below. Without Condition 1a, the straightforward approach exploited in Theorem 6.2.2 leads to a 4LP, which can be reduced to a 3LP at the cost of a large number of hidden nodes only; see Zwietering, Aarts & Wessels [1991a] for a description of a possible construction in that case. Note that Condition 1a may exclude some interesting COPs,
for instance the knapsack problem discussed in Zwietering, Aarts & Wessels [1991a]. Condition 1a is also essential for the results that we derive in Section 6.3. We have not been able to generalize the results of that section to the case that Condition 1a does not hold. In a strict sense, Condition 1b is not essential, since most results can be extended to the case that the subsets \( \{ x \in \mathbb{R}^N | c(y; x) \leq c(z; x) \} \) correspond to closed affine halfspaces, but this does not seem to be a serious extension, in the sense that the set of admissible COPs is enlarged significantly. We require Condition 1b because all the results of Chapter 5 are based on affine halfspaces. However, we are aware of the fact that it is a restrictive condition, and for instance excludes COPs in which the cost function contains the maximum function or the absolute value function.

The class of COPs described by Conditions 1a and 1b, contains the class of discrete linear optimization problems distinguished by Savage and others; see also Savage [1973], Papadimitriou & Steiglitz [1982], and Section 6.3. This class of problems can be defined as follows.

**Definition 6.2.4 (Savage [1973]).** Let \((I, F, c)\) be a COP. This COP is called a discrete linear optimization problem if \((I, F, c)\) satisfies (i) \( I = \mathbb{R}^N \), for some \( N \in \mathbb{N} \), (ii) \( F(x) = F \subseteq B^N \), for all \( x \in I \), and some \( K \in \mathbb{N} \), and (iii) \( c(y; x) = a(y)^T x \), for all \( y \in F \) and \( x \in I \).

One can easily verify that traveling salesman, sorting, and minimum cost spanning tree, which were introduced in Chapter 3, are examples of discrete linear optimization problems. However, the other three introduced problems, shortest network path, shortest network route, and discrete dynamic lotsizing, fall outside this class. Two of the mentioned problems, traveling salesman and minimum cost spanning tree, are also considered by Savage; see Savage [1973]. Note that Savage uses \( a(y) \) to denote a solution, instead of \( y \) itself. Generalized versions of the results that Savage obtained for his class of discrete linear optimization problems, play an important part in Section 6.3.

Next, we describe the construction of a 3LP that solves a COP, if this COP satisfies Conditions 1a and 1b. We first give a sketch of this construction by explaining the basic ideas. Using Condition 1a and using (6.1), it follows that the subsets \( \Omega_y \) of the corresponding CCP satisfy

\[
\Omega_y = \{ x \in I | \forall z \in F : c(y; x) \leq c(z; x) \},
\]

for all \( y \in F \). Hence, using Condition 1b, it follows that \( \Omega_y = I \cap A_y \), where for all \( y \in F \) the subset \( A_y \subseteq R^N \) is defined by

\[
A_y = \{ x \in \mathbb{R}^N | \forall z \in F : [a(y) - a(z)]^T x \leq [b(z) - b(y)] \}. \tag{6.5}
\]

Obviously, \( A_y \subseteq F \), which implies that there exists a 2LP that classifies \( A_y \); see Section 5.3 and Section 5.5.3. Hence, using Corollary 5.3.2, it follows that there exist \( K \) 3LPs, represented by \( f_i \in C_{x,Y,K} \), \( i = 1, \ldots, K \), such that \( f_i \cap I = V^{(i)}_q \), \( i = 1, \ldots, K \), where for all \( i = 1, \ldots, K \) and \( q = 0,1 \), the set \( V^{(i)}_q \) is defined by

\[
V^{(i)}_q = \bigcup_{y \in F \cup \{ y \}} \Omega_y.
\]

If we could show that the subsets \( \Omega_y \), \( y \in F \), were disjoint, then Theorem 5.7.1 would prove that these \( K \) one-output 3LPs, combined into one \( K \)-output 3LP, solve
the considered COP. However, in general the subsets \( \Omega_y, y \in F \), are not disjoint. Therefore, we introduce a preference function, similar like we did in Theorem 5.7.2. Let \( p : \mathbb{B}^K \rightarrow \mathbb{R} \) be an arbitrary function with \( p(y) \neq p(z) \), for all \( y, z \in \mathbb{B}^K, y \neq z \). Instead of designing a 3LP that solves \((I, F, c)\), we will in fact design a 3LP that solves \((I, F, c)\), and, in case of a tie, selects the optimal solution \( y \in F \) with the highest preference value \( p(y) \). The corresponding CCP is given by \((\Omega, L, \Gamma)\), with \( \Omega = \Omega = I, L = L = F, \) and \( \Gamma = \{\Omega_y | y \in F\} \), where for all \( y \in F, \Omega_y \) is given by

\[
\Omega_y = \{x \in I | \forall z \in F \setminus \{y\} : c(y; z) < c(z; x) \lor [c(y; z) = c(z; x) \land p(y) > p(z)]\}. \tag{6.6}
\]

One easily verifies that \((\hat{\Omega}, \hat{L}, \hat{\Gamma})\) satisfies the conditions (i), (ii), (iii), and (iv) of Theorem 5.7.1. Moreover, this CCP equals the CCP that is constructed in Theorem 5.7.2, in case that one starts with the corresponding CCP \((\Omega, L, \Gamma)\) defined in Definition 6.2.3.

The above shows that we must find 3LPs that classify the subsets \( \bigcup_{y \in F : y \neq 0} \hat{\Omega}_y \), for \( i = 1, \ldots, K \). In the following theorem we present in detail the construction of a 3LP, of which we prove that it solves the considered COP, by showing that it solves \((\hat{\Omega}, \hat{L}, \hat{\Gamma})\).

**Theorem 6.2.2.** Let \((I, F, c)\) be a COP that satisfies Conditions 1a and 1b, with \( I \subseteq \mathbb{R}^N \) and \( F \subseteq \mathbb{B}^K \), for some \( N, K \in \mathbb{N} \). Let \( p : \mathbb{B}^K \rightarrow \mathbb{R} \) be a function with \( p(y) \neq p(z) \), for all \( y, z \in \mathbb{B}^K, y \neq z \). Let \( f : \mathbb{R}^B \rightarrow \mathbb{B}^K \) be the function defined by \( f = g^{(2)} \circ g^{(1)} \circ g^{(0)} \), with \( g^{(0)} \), \( i = 1, 2, 3 \), given by

\[
\forall y, z \in F, p(y) > p(z) \forall x \in \mathbb{R}^N \quad : \quad g^{(0)}_y(x) = \theta[c(y; x) - c(y; x)], \tag{6.7}
\]

\[
\forall y \in F \quad \forall w \in \mathbb{R}^{\frac{1}{2}|F|(|F| - 1)} \quad : \quad g^{(1)}_y(w) = \theta[\sum_{y \in F} w_{y, y} - 1 - \sum_{y \in F} w_{y, y}], \tag{6.8}
\]

\[
\forall i = 1, \ldots, K \quad \forall y \in \mathbb{R}^{|F|} \quad : \quad g^{(2)}_y(s) = \theta[-1 + \sum_{y \in F} s_i], \tag{6.9}
\]

respectively. Then \( f \) represents a 3LP, with \( N \) inputs, \( \frac{1}{2}|F|(|F| - 1) \) nodes in the first hidden layer, \( |F| \) nodes in the second hidden layer, and \( K \) outputs, that solves \((I, F, c)\).

**Proof.** The above functions are properly defined, because Condition 1a holds. Furthermore, using Condition 1b, it follows that \( g^{(1)} \) represents a 1LP. Since \( g^{(2)} \) and \( g^{(3)} \) also represent 1LPs, and \( g^{(1)}, g^{(2)}, \) and \( g^{(3)} \) have matching dimensions, \( f \) represents a 3LP with the given sizes.

Let \((\hat{\Omega}, \hat{L}, \hat{\Gamma})\) be the CCP corresponding to the COP \((I, F, c)\), and let \((\hat{\Omega}, \hat{L}, \hat{\Gamma})\) be defined as above, with \( \hat{\Omega}_y \), given by (6.6). Using Condition 1a, one easily verifies that \( \hat{\Omega}_y \subseteq \Omega_y \), for all \( y \in F \), \( \bigcup_{y \in F} \hat{\Omega}_y = \hat{\Omega} = \Omega \), and \( \hat{\Omega}_y \cap \hat{\Omega}_z = \emptyset \), for all \( y \neq z \). Hence, using Theorem 5.7.1 and Proposition 6.2.1, it follows that we have proven that \( f \) solves \((I, F, c)\), if we show that \( \mathcal{J}(f) \cap I = \hat{V}^{(0)}_i \), for all \( i = 1, \ldots, K \), where for all \( i = 1, \ldots, K \) and \( q = 0, 1 \), the set \( \hat{V}^{(0)}_i \) is defined by

\[
\hat{V}^{(0)}_i = \bigcup_{y \in F : y \neq 0} \hat{\Omega}_y. \tag{6.10}
\]

We show this in two steps. First, we prove that for all \( x \in I \) and \( y \in F \), we have that \((g^{(2)}_y \circ g^{(1)})(x) = 1\), if and only if \( x \in \hat{\Omega}_y \). Let \( x \in I \) and \( y \in F \), then by (6.8)
6.3 The minimal number of first-layer nodes

and (6.7), we find
\[
g^{(3)}_{i,j}(g^{(1)}(x)) = 1 \iff \forall z \in F, p(y) > p(x) : g^{(1)}_{i,j}(x) = 1
\wedge \forall z \in F, p(z) > p(y) : g^{(1)}_{i,j}(x) = 0
\iff \forall z \in F, p(y) > p(x) : c(y; x) \leq c(z; x)
\wedge \forall z \in F, p(z) > p(y) : c(y; x) < c(z; x)
\implies \exists z \in \Omega_y.
\] (6.11)

Next, we describe the second step. Using (6.9), it follows that for all \( x \in \mathbb{R}^N \) and \( i = 1, \ldots, K \), we have that \( g_i^{(3)}(g^{(1)}(x)) = 1 \), if and only if there exists a \( z \in F \) with \( z_i = 1 \) and \( g^{(3)}_i(g^{(1)}(y)) = 1 \). This combined with (6.11) and (6.10), yields that for all \( x \in I \) and \( i = 1, \ldots, K \), \( f_i(x) = 1 \), if and only if \( x \in \tilde{V}^{(i)} \). This completes the proof of the theorem.

We have shown that for COPs that satisfy Conditions 1a and 1b, the minimal number of layers required by an MLP to solve this problem is three. In Section 6.4 we discuss the possibility of solving such a COP with a 2LP.

Theorem 6.2.2 describes a 3LP with \( \frac{1}{2} |F|(|F| - 1) \) nodes in the first hidden layer and \( |F| \) nodes in the second hidden layer, which corresponds to a complete enumeration of the feasible solutions. In the following section we prove that the size of the first hidden layer can be reduced, without increasing the number of layers and without increasing the number of hidden nodes in the second hidden layer, by deriving the minimal number of first layer nodes that is required to solve the considered COP, in case that the considered COP satisfies some additional constraints.

Our final remark concerns the number of non-zero weights of the 3LP presented in Theorem 6.2.2. The construction of the 3LP presented in Theorem 6.2.2. is based on \( K \) one-output 3LPs that classify the \( K \) subsets \( \tilde{V}^{(i)} = \bigcup_{x \in F, y_i = 1} \tilde{\Omega}_y, i = 1, \ldots, K \), with \( \tilde{\Omega}_y \) given by (6.6). One can straightforwardly show that \( \tilde{V}^{(i)} = \bigcup_{x \in F, y_i = 1} \tilde{\Omega}_y^{(i)} \), with \( \tilde{\Omega}_y^{(i)} \) defined by
\[
\tilde{\Omega}_y^{(i)} = \{ x \in I \mid \forall z \in F, z_i \neq y_i : c(y; z) < c(z; x) \lor [c(y; z) = c(z; x) \wedge p(y) > p(z)] \}
\] (6.12)
for all \( i = 1, \ldots, K \) and \( y \in F \). If one uses this result to construct a 3LP similar to the 3LP presented in Theorem 6.2.2, the obtained 3LP has the same dimensions, except that the number of non-zero weights in the second hidden layer is half as much as required for the 3LP of Theorem 6.2.2.

6.3 The minimal number of first-layer nodes

In this section we derive an expression for the minimal number of first-layer nodes, that is required by any MLP that solves a COP satisfying some additional constraints of the kind discussed in the previous section. This is done by translating the results of Section 5.4 on the minimal number of first-layer nodes required for classifying polyhedrons, into results for COPs. As in Section 6.2, we use the CCP \((\Omega, L, \Gamma)\), corresponding to a given COP, and \((\tilde{\Omega}, \tilde{L}, \tilde{\Gamma})\), that has disjoint subsets due to the addition of a preference function, as intermediums.

The main issue is to find minimal representations of the subsets \( \Omega_y \) and \( \tilde{\Omega}_y \). To this
and we present an additional condition that ensures that $\Omega_\nu$ and $\Omega_\nu$ correspond to polyhedrons and pseudo polyhedrons, respectively. Consequently, the determination of the minimal representations of $\Omega_\nu$ and $\Omega_\nu$ involves the identification of facets and intercepts of $\Omega_\nu$ and $\Omega_\nu$, respectively. Once these are known, it remains to show that (i), every MLP that solves the considered COP indeed uses first layer nodes that correspond to the halfspaces present in the minimal representations of $\Omega_\nu$ and $\Omega_\nu$, and (ii), there exists a construction of an MLP based on the minimal representations of $\Omega_\nu$ and $\Omega_\nu$, that solves the COP. The first construction we present to prove the second part, concerns a 4LP. Because this is somewhat unsatisfactory, we show that it can be reduced to a 3LP, with minimal sized first layer and $|\mathcal{F}|$ nodes in the second layer, that solves the considered COP. Hence, we show that the 3LP given in Section 6.2 can be reduced with respect to the size of its first hidden layer, without increasing the number of hidden layers and without increasing the number of nodes in the other hidden layers.

In the present section an important part is played by local search, a well-known approach for solving COPs; see for instance Aarts, Korst & Zwiering (1992). The basic idea of local search is that one searches for a feasible solution that is locally optimal with respect to a certain neighborhood. One then hopes that the solution found is also globally optimal, which can be guaranteed only if the neighborhood is exact. The reason for considering local search is that we can show that finding minimal representations of $\Omega_\nu$, is equivalent to finding the minimal exact neighborhood of the considered COP. The concept of minimal exact neighborhoods is introduced by Savage, who described a method for determining this neighborhood for his class of discrete linear optimization problems; see Definition 6.2.4 in Section 6.2, and Savage (1973). We extend on his results by proving that his expression for the minimal exact neighborhood is valid also for the larger class of COPs defined by Conditions 1a and 1b, given in the previous section, plus two additional conditions. These additional conditions ensure that the subsets $\Omega_\nu$ are full-dimensional polyhedrons, and generalize a similar additional condition required by Savage.

An important distinction between the approach of Savage and our approach is that while Savage works mainly in the solution space given by $\mathbb{R}^{N+1} \supseteq \{(a(y), b(y)) | y \in \mathcal{F}\}$, we consider the parameter space $\mathbb{R}^N \supseteq I$ instead. In fact Savage considers COPs with $b(y) = 0$ only, which implies that his solution space is $\mathbb{R}^N \supseteq \{a(y) | y \in \mathcal{F}\}$; see Savage (1973) and Papadimitriou & Steiglitz (1982), for further discussions of the solution space approach. Although there is a clear duality-based relation between the two approaches, it appears to be very convenient to work in parameter space.

We show that the minimal number of first-layer nodes required to solve a COP, can be expressed using the minimal exact neighborhood of the problem. This implies that in order to find the minimal first layer, one has to determine the minimal exact neighborhood of the considered COP first. Two approaches for determining the minimal exact neighborhood are discussed in this section. Furthermore, it is shown that the size of the minimal exact neighborhood is a lower bound for the number of first-layer nodes that is required to solve the COP by an MLP. Hence, in case that the minimal exact neighborhood can be shown to be exponential, this implies that the considered COP cannot be solved by an MLP with a polynomial number
The remainder of this section is partitioned in four subsections. Firstly, we discuss the basic notions of local search, including the minimal exact neighborhood introduced by Savage. Secondly, we study the relation between the minimal exact neighborhood and the minimal representations of $\Omega_u$. Thirdly, we derive an expression for the minimal number of first-layer nodes, that is based on the minimal representation of $\Omega_u$. Fourthly, we present two approaches for determining the minimal exact neighborhood, when we discuss the determination of the minimal size of the first layer for the COP at hand.

### 6.3.1 Local search and the minimal exact neighborhood

In this section we briefly introduce the basic concepts of local search. Subsequently, we consider a local search procedure, and show that this leads to a search for the minimal exact neighborhood, a concept introduced by Savage [1973]. We discuss some of the results of Savage, that are relevant in connection with our search for MLPs with minimal sized first layer.

Local search is a general approach for solving COPs, based on the use of neighborhood structures; see Yannakakis [1990] for an overview of local search. The notions neighborhood, neighborhood structure, local optimality, exact neighborhood, and minimal exact neighborhood, all relate to the concept of local search, and are discussed briefly in this section. For a more elaborate discussion of these subjects we refer to Aarts & Korst [1989], Aarts, Korst & Zwietering [1992], Papadimitriou & Steiglitz [1982], and Yannakakis [1990].

Consider a COP $(I, F, c)$, as defined in Section 6.2. Throughout this section we assume that the considered COP satisfies Conditions 1a and 1b of Section 6.2. This implies that the set of feasible solutions is parameter independent; see Condition 1a. Similarly, we only consider neighborhood structures that are parameter independent, which are defined below. The results of this section do not apply to neighborhoods that depend on the instance parameter, which are discussed occasionally; see for instance Papadimitriou & Steiglitz [1977]. However, most papers on local search consider parameter independent neighborhood structures; see Aarts, Korst & Zwietering [1992], Papadimitriou & Steiglitz [1982], Yannakakis [1990], and the references therein.

**Definition 6.3.1.** Let $(I, F, c)$ be a COP that satisfies Condition 1a, with $I \subseteq \mathbb{R}^N$ and $F(x) = F \subseteq \mathbb{B}^N$, for all $x \in I$ and some $N, K \in \mathbb{N}$. Then a neighborhood structure is a mapping $N : F \rightarrow 2^F$, which defines for each feasible solution $y \in F$ a set $N_y \subseteq F \setminus \{y\}$ of feasible solutions. The set $N_y$ is called the neighborhood of $y$, and each $z \in N_y$ is called a neighbor of $y$. For an instance defined by $x \in I$, the feasible solution $y \in F$ is called locally optimal with respect to the neighborhood structure $N$, if $c(y; x) \leq c(z; x)$, for all $z \in N_y$. A neighborhood structure is called exact, if for each $x \in I$, a feasible solution $y \in F$ that is locally optimal with respect to $N$, is also globally optimal, i.e., $c(y; x) \leq c(z; x)$, for all $z \in F$.

A local search algorithm is an approximation algorithm based on the exploration
The \textbf{MLP Complexity of COPs}

procedure LOCAL-SEARCH(\textbf{input}: \( z \in I \); \textbf{output}: \( y \in F \));
\{all solutions \( z \in F \) are unmarked\}
\begin{align*}
\text{begin} \\
\text{INITIALIZE}(y \in F); \\
\text{MARK}(y); \\
\{\text{invariant of loop: } c(y; z) \leq c(z; x), \text{ for all marked } z \in F\} \\
\text{while} \ [\text{there exists an unmarked solution in } \mathcal{N}_y] \\
\text{do} \\
\text{GENERATE}(\text{unmarked } z \in \mathcal{N}_y); \\
\text{if } c(z; x) < c(y; x) \text{ then } y := z; \text{fi}; \\
\text{MARK}(z) \\
\text{end}; \\
\{c(y; z) \leq c(z; x), \text{ for all } z \in \mathcal{N}_y\}
\end{align*}

Figure 6.1: A basic local search procedure.

of a given neighborhood; see Aarts, Korst & Zwietering [1992] for a discussion of the difference between optimization and approximation algorithms. In Figure 6.1 an
annotated version of a basic local search procedure is given, that uses a certain neighbor-
hood structure \( \mathcal{N} \). For a given instance of the COP defined by \( z \in I \), it starts
with a certain initial feasible solution \( y \in F \), that is usually generated randomly.
Then the procedure applies iterative improvement by searching the neighborhood of
the current solution for a feasible solution with lower cost. If such a solution is
found, the current solution is replaced by this solution. Otherwise, the procedure
continues with the current solution. It terminates when no strict improvements can
be obtained anymore, which implies that the procedure terminates in a local min-
imum. In order to make the procedure provably correct, we have added a MARK
procedure, that marks inspected feasible solutions. This enables us to prove that the
local search procedure is an optimization algorithm, in case that the neighborhood
structure \( \mathcal{N} \) is exact.

\textbf{Theorem 6.3.1.} Let \((I, F, c)\) be a COP that satisfies Condition 1a, and let \( \mathcal{N} \) be
an exact neighborhood structure for \((I, F, c)\). Then for each \( x \in I \), the local search
procedure of Figure 6.1 applied to the instance defined by \( x \), yields an optimal
solution.

\textbf{Proof.} Let \( x \in I \) and consider the local search procedure applied to the instance
defined by \( x \). The procedure halts since the number of unmarked solutions is finite,
and decreases every iteration of the inner loop. Furthermore, one easily verifies that
\[ c(y; z) \leq c(z; x), \text{ for all marked solutions } z \in F^m, \]
\text{is an invariant of the loop. When the procedure halts, all solutions of } \mathcal{N}_y \text{ are marked. Consequently, at termination we have } c(y; z) \leq c(z; x), \text{ for all } z \in \mathcal{N}_y. \text{ Since } \mathcal{N} \text{ is an exact neighborhood structure, it follows that } c(y; z) \leq c(z; x), \text{ for all } z \in F. \]
Next, we briefly discuss the running time of the local search procedure of Figure 6.1, and show how it leads to the search for the minimal exact neighborhood. Let the running time of the local search procedure of Figure 6.1 be defined as the number of times that the cost of a solution is determined. For most implementations, this number equals the number of times that a solution is marked. Hence, the worst-case running time \( T \) satisfies
\[
\max_{\gamma \in F} |\mathcal{N}_\gamma| \leq T \leq |F|.
\]  
Consequently, if one is interested in finding an optimization algorithm based on a local search procedure that uses an exact neighborhood structure, and that has a small worst-case running time, one is tempted to search for the exact neighborhood structure for which \( \max_{\gamma \in F} |\mathcal{N}_\gamma| \) is minimal. Below, we define the minimal exact neighborhood, which solves this problem.

**Definition 6.3.2.** Let \((I,F,c)\) be a COP that satisfies Condition 1a. A neighborhood structure \(\mathcal{M}\) is called the minimal exact neighborhood for \((I,F,c)\), if \(\mathcal{M}\) is exact and satisfies the following condition. For all exact neighborhood structures \(\mathcal{N}\) of \((I,F,c)\), it holds that \(\mathcal{M}_y \subseteq \mathcal{N}_y\), for all \(y \in F\).

Note that a COP might not have a minimal exact neighborhood, because we search for a neighborhood \(\mathcal{M}\) for which \(|\mathcal{M}_y|\) is minimal for all \(y \in F\) simultaneously. The obvious question is how to find the minimal exact neighborhood of a problem, if it exists. We briefly discuss the results of Savage for the class of discrete linear optimization problems; see also [Savage, 1973; Savage, 1976; Savage, Weiner & Bagchi, 1976; Savage, Weiner & Krone, 1972; Weiner, Savage & Bagchi, 1973]. Furthermore, we show that these results can be extended, using the results of Section 5.4 on the minimal representation of polyhedrons. Savage introduces the following neighborhood structure, which we denote \(\mathcal{M}^S\).

**Definition 6.3.3 (Savage, Weiner & Bagchi [1976], Savage [1973]).**

Let \((I,F,c)\) be a COP that satisfies Condition 1a. We define the neighborhood structure \(\mathcal{M}^S\) of \((I,F,c)\) as follows. For each \(y \in F\), the neighborhood \(\mathcal{M}^S_y\) contains all feasible solutions \(z \in F\) that can be uniquely optimal when \(y\) is second to optimal, i.e., for each \(y \in F\)
\[
\mathcal{M}^S_y = \{ z \in F | \exists x \in I \forall z' \in F \setminus \{x\} : c(z;x) < c(y;x) \leq c(z';x) \}.
\]  

Note that \(\mathcal{M}^S\) depends on \(I\), in the sense that if \(I\) is changed, \(\mathcal{M}^S\) can change also. In order to prove that \(\mathcal{M}^S\) is the minimal exact neighborhood of a given COP, we must prove that (i) \(\mathcal{M}^S\) is exact, and (ii) \(\mathcal{M}^S_y \subseteq \mathcal{N}_y\), for all \(y \in F\), and for all exact neighborhood structures \(\mathcal{N}\) of the considered COP. The second requirement can be proved straightforwardly, as is shown by Savage. We copy his argument since it illustrates the definition of \(\mathcal{M}^S\).

**Lemma 6.3.1 (Savage, Weiner & Bagchi [1976]).** Let \((I,F,c)\) be a COP that satisfies Condition 1a, and let \(\mathcal{N}\) be an arbitrary exact neighborhood structure for this COP. Then \(\mathcal{M}^S_y \subseteq \mathcal{N}_y\), for all \(y \in F\).

**Proof.** Let \(y \in F\) and \(z \in \mathcal{M}^S_y\). Then by (6.14), there exists a parameter \(x \in I\), such that for the instance defined by \(z\), \(z\) is uniquely optimal and \(y\) is second to optimal. Assume that \(z \notin \mathcal{N}_y\). Then \(c(y;z) \leq c(z';z)\), for all \(z' \in \mathcal{N}_y\), and hence,
since $\mathcal{N}$ is exact, $y$ must be declared optimal. Since $y$ is not optimal, it follows that $z \in \mathcal{N}_y$. □

It remains to be shown that $\mathcal{M}^S$ is exact. In [Savage, Weiner & Bagchi, 1976], Savage, Weiner & Bagchi give a tailored proof of this result for TRAVELING SALESMAN. A more general result is given in [Savage, 1973], where Savage proves that $\mathcal{M}^S$ is an exact neighborhood structure for all problems that are member of his class of discrete linear optimization problems. A recent proof of the same result can be found in [Papadimitriou & Steiglitz, 1982]. In order to prove that $\mathcal{M}^S$ is exact, Savage requires one additional restriction, which we generalize to the following two conditions.

**Condition 2a.** The set of instance defining parameters forms a polyhedron in $\mathbb{R}^N$, i.e., $I \in P$.

**Condition 2b.** For each feasible solution, there exists a ball of parameters that all define instances for which this solution is strictly optimal, i.e.,

$$\forall y \in F \exists z \in I^o \forall z \in F \setminus \{y\} : c(y; z) < c(z; x).$$

Because Savage considers problems with $I = \mathbb{R}^N$ only, he does not require Condition 2a, and his version of Condition 2b contains $\mathbb{R}^N$ instead of $I^o$; see Savage [1973]. Below it becomes clear why Savage's version has to be generalized to Condition 2b, with $I^o$ instead of $\mathbb{R}^N$ and not $I$, when we discuss the interpretations of Condition 2b in solution space and parameter space.

In case that Condition 2a does not hold, because $I \not\in P$, one might consider solving $(I', F', c)$, with $I' \supset I$ such that $I' \in P$. For instance, if $I \in U$, then $I' = \text{conv}(U)$ indeed satisfies $I' \supset I$ and $I' \in P$; see Lemma 5.6.1 presented in Section 5.6. Obviously, any MLP that solves $(I', F', c)$, solves $(I, F, c)$ also. However, enlarging the set of instance parameters may affect the size of the minimal representation of $\Omega_\mu$; see Section 6.3.2. As we will see in Section 6.3.3 and Section 6.4, this may also affect the complexity of the problem with respect to the required number of hidden nodes and the required number of layers, respectively.

In case that Condition 2b does not hold, because $I^o = \emptyset$, one determines the smallest, in dimension, affine subspace that contains $I$, and considers the problem with respect to this subspace, i.e., one makes the substitution indicated by Corollary 5.2.2. Obviously, $I^o \neq \emptyset$ with respect to this determined subspace. In case that Condition 2b does not hold, although $I^o \neq \emptyset$, one might consider solving $(I, F', c)$, with $F' = \{y \in F \mid \exists x \in I^o \forall z \in F' \setminus \{y\} : c(y; z) < c(z; x)\}$. One can easily show that any MLP that solves $(I, F', c)$, solves $(I, F, c)$ also.

Sometimes we can pose weaker conditions on the set of considered COPs. For later use we present the following conditions.

**Condition 3a.** The set of instance parameters has a non-empty interior, i.e., $I^o \neq \emptyset$.

**Condition 3b.** For each pair of feasible solutions there exists a parameter for which these solutions have different cost, i.e.,

$$\forall y, z \in F, y \neq z \exists x \in I : c(y; x) \neq c(z; x).$$
6.3 The minimal number of first-layer nodes

**Condition 3c.** For each feasible solution there exists a parameter for which this solution is optimal, i.e.,

$$\forall y \in F \exists x \in I \forall z \in F : c(y; x) \leq c(z; x).$$

The three Conditions 3a, 3b, and 3c, all form a part of Condition 2b. Hence, Condition 2b implies the three Conditions 3a, 3b, and 3c. However, if all three Conditions 3a, 3b, and 3c hold, then Condition 2b does not necessarily have to hold.

In order to prove that $M^S$ is exact, both Savage and Papadimitriou & Steiglitz apply what might be called a solution space approach. This is because they mainly consider the space $R^{n+1}$, in which the points $(a(y), b(y))$, $y \in F$, corresponding to the 'solutions' of Savage, form a polytope, i.e., a bounded polyhedron. Condition 2b has the following geometrical interpretation in solution space. Recall that by Condition 1b, $c(y; x) = a(y)^T x + b(y)$, for all $x \in R^n$ and $y \in F$. Then Condition 2b implies that for each feasible solution $y \in F$, there exists a hyperplane in $R^{n+1}$ with $(a(y), b(y))$ on one side and all remaining 'solutions' on the other side. Therefore, no 'solution' $(a(y), b(y))$ can be expressed as a convex combination of other 'solutions', which implies that the points $(a(y), b(y))$ form the extreme points of the polytope in $R^{n+1}$. Using this approach, Savage shows that $M^S$ is exact in case that $I = R^0$ and $b(y) = 0$, by proving that in this case the neighbors of $y$ given by $M^S$ correspond to the polytope neighbors; see also Savage [1973] and Papadimitriou & Steiglitz [1982].

It should be possible to extend the analysis of Savage to the general case given by $b(y) \neq 0$ and $I \in P$. However, we show below that the argument is very short if one considers parameter space instead of solution space.

The interpretation of Condition 2a in parameter space is that, combined with Conditions 1a and 1b, it implies that $\Omega_y$ is a polyhedron, for all $y \in F$. If Condition 2a holds, then the interpretation of Condition 2b in parameter space is that for each $y \in F$, $\Omega_y$ is a full-dimensional polyhedron, if and only if Condition 2b holds. Before we study the implications of this observation, we discuss the main result of Savage for discrete linear optimization problems; see Savage [1973] for a proof of this result.

**Proposition 6.3.1 (Savage [1973]).** Let $(I, F, c)$ be a discrete linear optimization problem that satisfies Condition 2b. Then the neighborhood structure $M^S$ defined by (6.14) is exact.

Lemma 6.3.1 and Proposition 6.3.1 together prove that $M^S$ is the minimal exact neighborhood for discrete linear optimization problems that satisfy Condition 2b. Below we extend this result of Savage to the class of COPs that satisfy Conditions 1a, 1b, 2a, and 2b. This is done by a parameter space approach, as opposed to the solution space approach of Savage and Papadimitriou & Steiglitz. In fact, it follows as a by-product of our search for the minimal representations of $\Omega_y$ and $\Omega_x$, that leads to the construction of MLPs with minimal sized first layers.
6.3.2 Minimal representations of $\Omega_y$

In this section we examine the problem of finding minimal representations of $\Omega_y$, and discuss the relation with the problem of finding the minimal exact neighborhood, as discussed in the previous subsection. We start with a definition that links $\Omega_y$ to neighborhoods.

**Definition 6.3.4.** Let $(I, F, c)$ be a COP that satisfies Condition 1a, and let $\mathcal{N}$ be a neighborhood structure for this COP. Then the subset $\Omega_y(\mathcal{N}) \subseteq I$, is for all $y \in F$ defined by

$$
\Omega_y(\mathcal{N}) = \{ x \in I | \forall z \in \mathcal{N}_y : c(y; x) \leq c(z; x) \}. 
$$

(6.15)

This definition enables us to give an alternative formulation of the exactness of a neighborhood structure. Since the proof of this result is trivial it is omitted.

**Proposition 6.3.2.** Let $(I, F, c)$ be a COP that satisfies Condition 1a, and let $\mathcal{N}$ be a neighborhood structure for this COP. Then $\mathcal{N}$ is exact, if and only if $\Omega_y(\mathcal{N}) = \Omega_y$, for all $y \in F$, where $\Omega_y(\mathcal{N})$ is given by (6.15) and $\Omega_y$ is given by (6.4).

The above result implies that searching for the minimal exact neighborhood is the same as searching for the minimal representation of $\Omega_y$ as an intersection of closed affine halfspaces. Because we assume that the considered COPs satisfy Conditions 1a, 1b, and 2a, $\Omega_y$ is a polyhedron. Hence, we can use the results of Section 5.4 that studies the minimal representations of polyhedrons. Our first observation is that in general $\Omega_y$ has two types of faces, faces of the form $c(y; x) \leq c(z; x)$, for some $y, z \in F$, and faces that are defined by the faces of $I$. In our search for minimal representations of $\Omega_y$, the faces defined by $I$ can be left without consideration, because these are fixed. Furthermore, faces defined by $I$ will neither be represented in a considered neighborhood structure, nor will they be represented in the first layer of an MLP that solves the considered problem. One should note however, that the presence of a face defined by $I$ might cause that a face of the first type is redundant for the description of $\Omega_y$. Consequently, altering $I$ may change the minimal representation of $\Omega_y$.

Next, we explicitly describe the relation between the minimal exact neighborhood and the minimal representation of $\Omega_y$, using the results of Section 5.4. Comparing Savage's definition of the neighborhood structure $\mathcal{M}^5$, given by (6.14), with the general definition of necessary halfspaces given by Definition 5.4.2, we directly obtain the following characterization of $\mathcal{M}^5$.

**Proposition 6.3.3.** For all $y, z \in F$, we have that $x \in \mathcal{M}^5_y$, if and only if the subset $\{ x \in \mathbb{R}^N | c(y; x) \leq c(z; x) \}$ represents a halfspace that is necessary for the description of $\Omega_y$.

In other words, the description of the polyhedron $\Omega_y(\mathcal{M}^5)$ is obtained from the description of the polyhedron $\Omega_y$, by removing all redundant halfspaces of the form $\{ x \in \mathbb{R}^N | c(y; x) \leq c(z; x) \}$, for some $z \in F$.

In general, the necessary halfspaces are not sufficient for the characterization of a polyhedron. However, there is one case for which we have shown that the necessary halfspaces are sufficient for the characterization of a polyhedron. This is the case
where the polyhedron is full-dimensional, because in that case, every necessary half-space represents a facet; see Theorem 5.4.1 and Corollary 5.4.2. Thus $\Omega_y(\mathcal{M}^y) = \Omega_y$, for all $y \in F$, and consequently $\mathcal{M}^y$ is exact, if we know that for each $y \in F$ the polyhedron $\Omega_y$ is full-dimensional. As already noted, this is the case if and only Condition 2b holds, which yields the following result.

**Theorem 6.3.2.** Let $(I, F, c)$ be a COP that satisfies Conditions 1a, 1b, 2a, and 2b. Then the neighborhood structure $\mathcal{M}^y$ defined by (6.14) is exact.

**Proof.** Let $y \in F$. Conditions 1a, 1b, and 2a, guarantee that $\Omega_y$ is a polyhedron, and Condition 2b guarantees that $\Omega_y^c \neq \emptyset$, i.e., $\Omega_y$ is full-dimensional; see also Proposition 5.4.2. Consequently, by combining Proposition 6.3.3 and Theorem 5.4.1, we find that $\Omega_y(\mathcal{M}^y) = \Omega_y$. Since $y \in F$ was arbitrarily chosen, the result follows from Proposition 6.3.2. \( \square \)

Theorem 6.3.2 extends the result of Savage given in Proposition 6.3.1, using a short and transparent argument that is based on some basic but powerful results on polyhedrons. However, this result was not our goal in this chapter. Our main objective is to find the minimal number of first-layer nodes that is required by any MLP that solves a given COP. The link between these two subjects is formed by the interfaces, defined in Definition 5.4.11, and, more specifically, by the result on the lower bound for the number of first-layer nodes that is required for the classification of a given subset. This result is presented in Corollary 5.4.6, and specialized to full-dimensional polyhedrons in Corollary 5.4.7. The following definition introduces a neighborhood structure based on interface representing halfspaces, which can be compared to the result of Proposition 6.3.3.

**Definition 6.3.5.** Let $(I, F, c)$ be a COP that satisfies Condition 1a. Then for each $y \in F$, the neighborhood $\mathcal{M}^y_{IF}$ consists of those feasible solutions $z \in F \setminus \{y\}$, for which the subset $\{x \in \mathbb{R}^N \mid c(y; x) \leq c(z; x)\}$ corresponds to a halfspace that represents an interface at $\Omega_y$.

Using Theorem 5.4.6, we obtain the following useful characterization of the neighborhood structure $\mathcal{M}^y_{IF}$. Note that we assume that Conditions 1b and 3b hold, because we want the subsets $\{x \in \mathbb{R}^N \mid c(y; x) \leq c(z; x)\}$, $y, z \in F$, to correspond to halfspaces for all $y \neq z$.

**Proposition 6.3.4.** Let $(I, F, c)$ be a COP that satisfies Conditions 1a, 1b, and 3b. Then, for each $y \in F$, the neighborhood $\mathcal{M}^y_{IF}$ of $y$ is given by

$$
\mathcal{M}^y_{IF} = \{z \in F \setminus \{y\} \mid \exists z \in F \setminus \{y\} \in F \setminus \{y\} : c(y; z) = c(y; z) < c(z; x)\}. \quad (6.16)
$$

The characterization (6.16) of $\mathcal{M}^y_{IF}$ can be compared with the definition (6.14) of $\mathcal{M}^y$. If Condition 2b does not hold, then it is possible that $\mathcal{M}^y_{IF}$ is empty, for some $y \in F$, while $\mathcal{M}^y$ is not empty. In this case we would have that $\mathcal{M}^y_{IF} \neq \mathcal{M}^y$. On the other hand, if both the Conditions 2a and 2b hold, then $\Omega_y$ is a full-dimensional polyhedron, which causes necessary halfspaces, halfspaces that represent a facet, and halfspaces that represent an interface, to be all the same thing; see Corollary 5.4.1, Corollary 5.4.2, and Theorem 5.4.6. Hence, we have the following result.

**Corollary 6.3.1.** Let $(I, F, c)$ be a COP that satisfies Conditions 1a, 1b, 2a, and 2b. Then $\mathcal{M}^y = \mathcal{M}^y_{IF}$, for all $y \in F$. 
This result has also been noted by Savage for the class of discrete linear optimization problems, where he uses it to show that $\mathcal{M}_S$ is symmetric, i.e., $z \in \mathcal{M}_S$, if and only if $y \in \mathcal{M}_S$; see Savage [1973]. Obviously, using Theorem 6.3.2, this result also shows that in case that the considered COP satisfies Conditions 1a, 1b, 2a, and 2b, $\mathcal{M}_{IF}$ is an exact neighborhood, which implies that $\mathcal{M}_{IF}$ is the minimal exact neighborhood for the considered COP.

**Corollary 6.3.2.** Let $(I, F, c)$ be a COP that satisfies Conditions 1a, 1b, 2a, and 2b. Then the neighborhood structure $\mathcal{M}_{IF}$ given by (6.16) is exact.

In the following subsection we show that there exists a relation between the neighborhood structure $\mathcal{M}_{IF}$ and the minimal size of the first hidden layer of any MLP that solves a given COP.

### 6.3.3 The minimal sized first layer

In the previous subsections, we discussed some results on the relation between minimal representations of the polyhedrons $\Omega_y$, $y \in F$, and the exact neighborhood structures originating from local search. One of the results, namely Corollary 6.3.1, identifies the facet representing halfspaces of $\Omega_y$, and showed their relation with the minimal exact neighborhood. In this subsection, we exploit these results and derive an expression for the minimal number of first-layer nodes that is required to solve a COP, which is based on the neighborhood structure $\mathcal{M}_{IF}$. In fact we show that the outputs of all the nodes of the minimal sized first layer are of the form $\theta(c(y; z) - c(z; z))$, for some $y \in F$ and $z \in \mathcal{M}_{IF}$, in which all the doubles have been eliminated. This can be formalized with the help of the following notions.

**Definition 6.3.6.** Let $(I, F, c)$ be a COP that satisfies Conditions 1a and 1b, and let $\mathcal{N}$ be a neighborhood structure. The two neighboring pairs $(y, z)$ and $(u, v)$, with $y \in F$, $z \in \mathcal{N}_y$, $u \in F$, and $v \in \mathcal{N}_u$, are equivalent, if there exists a $\tau \in \mathbb{R}$ such that

$$c(y; z) - c(z; z) = \tau(c(u; z) - c(v; z)),$$

for all $x \in \mathbb{R}^N$. Furthermore, $L(I, F, c, \mathcal{N})$ denotes the total number of non-equivalent neighboring pairs $(y, z)$, with $y \in F$ and $z \in \mathcal{N}_y$.

Thus if $L(I, F, c, \mathcal{N}) = L$, then there exists a set of $L$ non-equivalent neighboring pairs $(y, z)$, with $z \in \mathcal{N}_y$, and this is the largest set with that property. Savage introduced a similar notion, called parallel vectors; see Savage [1973].

Using the notion $L(I, F, c, \mathcal{N})$, we can formulate the main result of this section.

**Theorem 6.3.3.** Let $(I, F, c)$ be a COP that satisfies Conditions 1a, 1b, 2a, and 2b. Then the minimal number of first-layer nodes of any MLP that solves $(I, F, c)$ is $L(I, F, c, \mathcal{M}_{IF})$.

We prove this result in two steps. First, we show that $L(I, F, c, \mathcal{M}_{IF})$ first-layer nodes are necessary for any MLP that solves $(I, F, c)$; see Corollary 6.3.3. Secondly, we show that there exists an MLP with $L(I, F, c, \mathcal{M}_{IF})$ nodes in the first hidden layer, that solves $(I, F, c)$; see Corollary 6.3.4.
6.3 The minimal number of first-layer nodes

In the first step, we prove that each halfspace that represents an interfaced of \( \Omega_y \), for some \( y \in F \), must be represented by a first-layer node in any MLP that solves the considered COP. To this end, we use Corollary 5.4.4 and Corollary 5.4.5, by proving that an interfaced representing halfspace of any \( \Omega_y \), is also an interfaced representing halfspace of \( J(f_i) \), for some \( i = 1, \ldots, K \), if \( f \) represents an MLP that solves the considered COP. In the proof of Theorem 6.3.4 we use a direct argument, instead of using an argument based on Theorem 5.7.1 and Proposition 6.2.1.

**Theorem 6.3.4.** Let \((I, F, c)\) be a COP that satisfies Conditions 1a, 1b, and 3b, with \( I \subseteq \mathbb{R}^N \) and \( F \subseteq \mathcal{B}_K^N \), for some \( N, K \in \mathbb{N} \). Let \( f \in R_{m,N,K} \) represent an MLP that solves \((I, F, c)\), for some \( m \in \mathbb{N} \). Let \( y \in F \), and \( z \in M_{y}^\mathcal{B}^N \). Then there exists a first-layer node of the MLP represented by \( f \), whose output equals \( \Theta(c(y; z) - c(z; z)) \) or \( \Theta(c(z; z) - c(y; z)) \).

**Proof.** Let \( y \in F \), and \( z \in M_{y}^\mathcal{B}^N \). Then using (6.16), it follows that there exists an \( x_0 \in I^y \), such that \( c(z; x_0) = c(y; x_0) < c(x'; x_0) \), for all \( x' \in F \setminus \{y, z\} \). Since \( x_0 \in I^y \), and, by Condition 1b, \( c(x'; z) \) is continuous in \( x \), there exists a \( \delta > 0 \), such that \( B(x_0, \delta) \subseteq I \), \( c(y; z) < c(x'; z) \), and \( c(z; z) < c(x'; z) \), for all \( x' \in F \setminus \{y, z\} \) and for all \( x \in B = B(x_0, \delta) \). Since \( f \) solves \((I, F, c)\), this implies that \( f(x) = \{y, z\} \), for all \( x \in B \).

Let \( W = \{ x \in R^N \mid c(y; x) \leq c(z; x) \} \). Because \( y \neq z \), Conditions 1b and 3b imply that \( W \in H \). Furthermore, from the above it follows that \( f(x) = y \), for all \( x \in W^* \cap B \), and \( f(x) = z \), for all \( x \in W^* \cap B \). Since \( x_0 \in W^* \cap B \), both \( W^* \cap B \) and \( W^* \cap B \) are not empty.

Since \( y \neq z \), there exists an \( i \in \{1, \ldots, K\} \), such that \( y_i \neq z_i \). Then, in case that \( y_i = 1 \), we have

\[
0 \neq W^* \cap B \subseteq J^*(f_i) \wedge 0 \neq W^* \cap B \subseteq J^*(f_i).
\]

(6.17)

and, in case that \( z_i = 1 \), we have

\[
0 \neq W^* \cap B \subseteq J^*(f_i) \wedge 0 \neq W^* \cap B \subseteq J^*(f_i).
\]

(6.18)

Using Theorem 5.4.4 twice, we obtain from (6.17) and (6.18) that either \( W \) or \( W^* \) represents an interfaced of either \( J^*(f_i) \) or \( J^*(f_i) \). By Proposition 5.4.7, this implies that either \( W \), \( W^* \), or \( W^- = (W^*)^* \), represents an interfaced of \( J^*(f_i) \). The result now follows from Corollary 5.4.4, in case that \( m = 1 \), and from Corollary 5.4.5, in case that \( m \geq 2 \).

\( \square \)

Combining Theorem 6.3.4 with Definition 6.3.6, directly yields the following result.

**Corollary 6.3.3.** Let \((I, F, c)\) be a COP that satisfies Conditions 1a, 1b, and 2b. Then every MLP that solves \((I, F, c)\) requires at least \( L(I, F, c; M_{y}^\mathcal{B}^N) \) first-layer nodes.

The above result shows that at least \( L(I, F, c; M_{y}^\mathcal{B}^N) \) first-layer nodes are required by any MLP that solves \((I, F, c)\). Next, we prove that \( L(I, F, c; M_{y}^\mathcal{B}^N) \) first-layer nodes are sufficient, in case that the considered COP satisfies the additional Conditions 2a and 2b. This result is based on the following more general approach. We prove for each exact neighborhood structure \( N \), there exists an MLP with \((I, F, c, N)\) first-layer nodes that solves \((I, F, c)\), in case that this COP satisfies Conditions 3a and 3b. This is done by constructing a 4LP with \((I, F, c, N)\) nodes in the first hidden layer,
\(|F|\) nodes in the second hidden layer, and \(|F|\) nodes in the third hidden layer, that solves \((I, F, c)\); see Theorem 6.3.5. This result proves that \(L(I, F, c; M^{(p)})\) first-layer nodes are sufficient in case that Conditions 2a and 2b are satisfied, since in that case Conditions 3a and 3b hold, and the neighborhood structure \(M^{(p)}\) is exact; see Corollary 6.3.2.

From a practical point of view, the result presented below in Theorem 6.3.5 is unsatisfactory, since although the total number of nodes required for the construction of the 4LP may be less than the number of nodes used in the construction of the 3LP given in Theorem 6.2.2, the number of layers used increases from three to four. Therefore, we present a second construction, which produces a 3LP with \(L(I, F, c; N)\) first-layer nodes and \(|F|\) second-layer nodes that solves the considered COP in case that \(N\) is exact; see Theorem 6.3.6. In fact the second construction is obtained by proving that the second hidden layer of the first construction is superfluous, for which we use a dual characterization of exact neighborhoods; see Lemma 6.3.1. There is a relatively small cost that has to be paid for the removal of one layer, since the conditions have to be strengthened through the addition of Condition 3c.

As in Section 6.2, we make use of a preference function in the forthcoming constructions. However, in contrast to the construction presented in Theorem 6.2.2, we now require a specially designed preference function that is based on the cost function itself. The following lemma introduces an essential ingredient for this preference function.

**Lemma 6.3.2.** Let \((I, F, c)\) be a COP satisfying Conditions 1a, 1b, 3a, and 3b. Then there exists an \(\hat{z} \in I\), such that \(c(y; \hat{z}) \neq c(z; \hat{z})\), for all \(y, z \in F, y \neq z\).

**Proof.** Condition 3a implies that there exists a ball \(B \subseteq I\). Furthermore, Conditions 1b and 3b imply that the subsets \(W_{y^*} = \{x \in \mathbb{R}^q | c(y; x) \leq c(z; x)\}\) correspond to halfspaces, for all \(y \neq z\). Using Lemma 5.4.3, it then follows that there exists an \(\hat{z} \in B\), such that \(\hat{z} \notin W_{y^*}\), for all \(y \neq z\), which completes the proof of the lemma. \(\square\)

Lemma 6.3.2 implies that the function \(p : F \rightarrow \mathbb{R}\) defined by \(p(y) = -c(y; \hat{z})\), for all \(y \in F\), is a preference function. It is this preference function that is used in the construction of the two MLPs presented in Theorem 6.3.5 and Theorem 6.3.6. To this end we define the subsets \(\hat{\Omega}_y\) and \(\hat{\Omega}_y(N)\), similarly as \(\hat{\Omega}_y\) and \(\Omega_y(N)\) given by (6.6) and (6.15), respectively.

**Definition 6.3.7.** Let \((I, F, c)\) be a COP that satisfies Conditions 1a, 1b, 3a, and 3b, let \(N\) be a neighborhood structure for this COP, and let \(\hat{z} \in I\) be as in Lemma 6.3.2. Then the subsets \(\hat{\Omega}_y \subseteq I\) and \(\hat{\Omega}_y(N) \subseteq I\), are for all \(y \in F\) defined by

\[
\hat{\Omega}_y = \{x \in I | \forall z \in F \setminus \{y\} : c(y; x) < c(z; x) \lor [c(y; x) = c(z; x) \land c(y; \hat{z}) < c(z; \hat{z})]\},
\]

and \(\hat{\Omega}_y(N) = \{x \in I | \forall z \in N_y : c(y; x) < c(z; x) \lor [c(y; x) = c(z; x) \land c(y; \hat{z}) < c(z; \hat{z})]\}\), respectively.
Next, we present the construction of a 4 LP with \( L(I, F, c, N) \) nodes in the first-layer, that solves \((I, F, c)\), in case that \( N \) is an exact neighborhood structure. The first-layer nodes correspond to the \( L(I, F, c, N) \) non-equivalent neighboring pairs. The second hidden layer consists of \(|F|\) nodes, that are proven to classify the subsets \( \hat{\Omega}_y(N) \), \( y \in F \). We prove that these subsets satisfy the requirements (i) and (iii), of Theorem 5.7.1, i.e., we show that \( \hat{\Omega}_y(N) \subseteq \Omega_y \), for all \( y \in F \), and \( \bigcup_{y \in F} \hat{\Omega}_y(N) = \bigcup_{y \in F} \Omega_y = I \). The third hidden layer serves to satisfy the requirement (iv), using a similar approach as used in Theorem 5.7.2. The fourth hidden layer completes the construction, similar as in Theorem 6.2.2. In Theorem 6.3.5 we use the function \( \text{sign} : \mathbb{R} \to \{-1, 0, +1\} \), which satisfies \( \text{sign}(\lambda) = +1 \), if \( \lambda > 0 \), \( \text{sign}(\lambda) = -1 \), if \( \lambda < 0 \), and \( \text{sign}(\lambda) = 0 \), if \( \lambda = 0 \).

**Theorem 6.3.5.** Let \((I, F, c)\) be a COP that satisfies Conditions 1a, 1b, 3a, and 3b, with \( I \subseteq \mathbb{R}^N \) and \( F \subseteq \mathbb{B}^K \), for some \( N, K \in \mathbb{N} \). Let \( N \) be an exact neighborhood structure for this COP. Let \( L = L(I, F, c, N) \), and let \( \{(u^{(1)}, v^{(1)}), \ldots, (u^{(l)}, v^{(l)})\} \) be a maximal set of non-equivalent neighboring pairs, with \( v^{(i)} \in N_{u^{(i)}} \), for all \( i = 1, \ldots, K \). Let for all \( y \in F \) and \( z \in N_y \), \( t_{yz} \in \mathbb{R} \) and \( t_{yz} \in \{1, \ldots, K\} \) be such that

\[
\text{c}(y; z) - \text{c}(z; z) = t_{yz}(c(u^{(t_{yz})}; z)) - c(v^{(t_{yz})}; z)),
\]

(6.21)

for all \( x \in \mathbb{R}^N \). Let \( \hat{x} \in I \) be as in Lemma 6.3.8, and define \( \Delta(y, z) \in \{-1, +1\} \), for all \( y \neq z \), by

\[
\Delta(y, z) = \text{sign}(\Delta(y; \hat{x}) - \Delta(z; \hat{x})).
\]

(6.22)

Let, finally, \( f : \mathbb{R}^N \to \mathbb{B}^K \) be the function defined by \( f = g^{(4)} \circ g^{(3)} \circ g^{(2)} \circ g^{(1)} \), with \( g^{(i)}, i = 1, 2, 3, 4 \), given by

\[
\forall i = 1, \ldots, L \forall x \in \mathbb{R}^N : \quad g^{(i)}(x) = \theta[\Delta(u^{(i)}, v^{(i)})(c(u^{(i)}; x)) - c(v^{(i)}; x))],
\]

(6.23)

\[
\forall y \in F \quad \forall w \in \mathbb{R}^K : \quad g^{(2)}(w) = \theta[\sum_{x \in N_y} \Delta(z, y)w_{zy} - \theta(\Delta(z; y))]
\]

(6.24)

\[
\forall y \in F \quad \forall s \in \mathbb{R}^{|F|} : \quad g^{(3)}(s) = \theta[s - 1 - \sum_{x \in P(y)} \Delta(y, x)s_x]
\]

(6.25)

\[
\forall i = 1, \ldots, K \forall t \in \mathbb{R}^{|F|} : \quad g^{(4)}(t) = \theta[-1 + \sum_{x \in F} x_t]
\]

(6.26)

respectively. Then \( f \) represents a 4 LP, with \( N \) inputs, \( L(I, F, c, N) \) nodes in the first hidden layer, \(|F|\) nodes in the second hidden layer, \(|F|\) nodes in the third hidden layer, and \( K \) outputs, that solves \((I, F, c)\).

**Proof.** As in the proof of Theorem 6.2.2, one verifies that \( g^{(1)}, g^{(2)}, g^{(3)}, \) and \( g^{(4)} \) represent 1 LPs with matching dimensions, and hence, \( f \) represents a 4 LP with the given sizes. We complete the proof by showing that the MLP represented by \( f \) solves \((I, F, c)\).

The main line is as in the proof of Theorem 6.2.2. Here, we prove that, firstly

\[
\{x \in I | g^{(4)}_y(g^{(3)}(x)) = 1\} = \hat{\Omega}_y(N),
\]

(6.27)

for all \( y \in F \), where \( \hat{\Omega}_y(N) \) is given by (6.20), and, secondly

\[
\{x \in I | g^{(4)}_y(g^{(3)}(g^{(2)}(x))) = 1\} = \hat{\Omega}_y,
\]

(6.28)

for all \( y \in F \), where \( \hat{\Omega}_y \) is given by (6.19). For the remainder of the proof we then refer to the proof of Theorem 6.2.2.
For the proof of (6.27), we first note that substituting $\hat{z}$ into (6.21), and using (6.22), yields
\[
\Delta(y, z) = \text{sign}(\tau_{y,z})\Delta(u^{(v_{x,y})}, v^{(v_{x,z})}),
\]
for all $y \in F$ and $z \in \mathcal{N}_y$. Secondly, using (6.22), (6.23), (6.29), and (6.21), it follows that for all $x \in \mathbb{R}^N$
\[
g^{(1)}_{0,x,y}(x) = 1 \land \Delta(z, y) = +1
\]
\[
\Leftrightarrow \Delta(u^{(v_{x,y})}, v^{(v_{x,z})})(c(u^{(v_{x,y})}; x) - c(v^{(v_{x,z})}; x)) \geq 0 \land
\Delta(y, z) = +1 \land c(y; \hat{z}) - c(z; \hat{z}) > 0
\]
\[
\Leftrightarrow \tau_{y,z} \Delta(u^{(v_{x,y})}, v^{(v_{x,z})})(c(y; z) - c(z; z)) \geq 0 \land
\tau_{y,z} \Delta(u^{(v_{x,y})}, v^{(v_{x,z})})< 0 \land c(y; \hat{z}) - c(z; \hat{z}) > 0
\]
\[
\Leftrightarrow c(y; z) \leq c(z; x) \land c(y; \hat{z}) < c(z; \hat{z}) .
\]
Similarly, one can show that $g^{(1)}_{0,x,y}(x) = 0$ and $\Delta(z, y) = -1$, if and only if $c(y; x) < c(z; x)$ and $c(y; \hat{z}) > c(z; \hat{z})$, for all $x \in \mathbb{R}^N$. Consequently, using that
\[
\Delta(z, y) \leq 0,
\]
for all $x \in I$
\[
g^{(2)}_{0,\xi}(g^{(1)}(x)) = 1 \Leftrightarrow \sum_{x \in \mathcal{N}_y} (\Delta(z, y) g^{(1)}_{0,x,y}(z) - \theta(\Delta(z, y))) \geq 0
\]
\[
\Leftrightarrow \forall z \in \mathcal{N}_y : \Delta(z, y) g^{(1)}_{0,x,y}(z) - \theta(\Delta(z, y)) = 0
\]
\[
\Leftrightarrow \forall z \in \mathcal{N}_y : (g^{(1)}_{0,x,y}(z) = 1 \land \Delta(z, y) = +1)
\]
\[
\lor (g^{(1)}_{0,x,y}(z) = 0 \land \Delta(z, y) = -1)
\]
\[
\Leftrightarrow \forall z \in \mathcal{N}_y : (c(y; x) \leq c(z; x) \land c(y; \hat{z}) < c(z; \hat{z}))
\]
\[
\lor (c(y; x) < c(z; x) \land c(y; \hat{z}) > c(z; \hat{z}))
\]
\[
\Leftrightarrow x \in \hat{N}_y(\mathcal{N})
\]
which completes the proof of (6.27). If we could prove that $\hat{N}_y(\mathcal{N}) \subseteq \Omega_y$, for all $y \in F$, $\cup_{y \in F} \hat{N}_y(\mathcal{N}) = \cup_{y \in F} \Omega_y$, and $\hat{N}_y(\mathcal{N}) \cap \hat{N}_x(\mathcal{N}) = \emptyset$, for all $y \neq x$, then as in the proof of Theorem 6.2.2, we could use Theorem 6.7.1 to complete the proof of the theorem. However, since we cannot prove the last requirement, we introduce an extra layer. In Theorem 6.3.6 below, we show that the last requirement holds, if we pose an additional restriction on the considered COP.

Next, we prove (6.28). Let $y \in F$, then by (6.25), we have that $g^{(2)}(y) = 1$, if and only if $s_y = 1$ and $s_z = 0$, for all $z \in F \setminus \{y\}$, with $\Delta(y, z) = 1$. Hence, using (6.27) and (6.22), we find that for all $x \in I$, $g^{(2)}(g^{(1)}(x))) = 1$, if and only if $x \in \hat{N}_y(\mathcal{N})$ and $x \notin \hat{N}_y(\mathcal{N})$, for all $z \in F \setminus \{y\}$, with $c(z; y) < c(y; \hat{z})$. Thus
\[
\{x \in I \mid g^{(2)}(g^{(1)}(x))) = 1\} = \hat{N}_y(\mathcal{N}) \setminus \bigcup_{y \in F \setminus \{y\}} \hat{N}_x(\mathcal{N}),
\]
for all $x \in F$. Let $V_y = \hat{N}_y(\mathcal{N}) \setminus \cup_{x \in \mathcal{N}_y \setminus \{x\}} \hat{N}_x(\mathcal{N})$, denote the right-hand side of (6.30). We complete the proof of (6.28), by showing that $V_y = \Omega_y$, for all $y \in F$. To this end, we make use of the fact that
\[
\hat{N}_y \subseteq \hat{N}_y(\mathcal{N}) \subseteq \hat{N}_y(\mathcal{N}) = \Omega_y,
\]
for all $y \in F$. The first two inclusions of (6.31) follow directly from the definitions of $\hat{N}_y$, $\hat{N}_y(\mathcal{N})$, and $\hat{N}_y(\mathcal{N})$, given by (6.19), (6.20), and (6.15), respectively. The equal-
6.3 The minimal number of first-layer nodes

Let \( \Omega_{y}(N) = \Omega_{y} \) follows from the fact that \( N \) is an exact neighborhood structure; see Proposition 6.3.2.

Using (6.19) and (6.31), we find

\[
\hat{\Omega}_{y} = \hat{\Omega}_{y} \setminus \bigcup_{x \in \mathcal{P}} \Omega_{x} \subseteq V_{y} \subseteq \hat{\Omega}_{y}(N) \subseteq \Omega_{y},
\]

(6.32)

for all \( y \in \mathcal{P} \). This implies

\[
\bigcup_{y \in \mathcal{P}} \Omega_{y} = \bigcup_{y \in \mathcal{P}} \hat{\Omega}_{y} \subseteq \bigcup_{y \in \mathcal{P}} V_{y} \subseteq \bigcup_{y \in \mathcal{P}} \Omega_{y}.
\]

(6.33)

It is easily verified using (6.19), and using the definition of \( V_{y} \), as given by the right-hand side of (5.30), that \( \hat{\Omega}_{y} \cap \hat{\Omega}_{z} = \emptyset \) and \( V_{y} \cap V_{z} = \emptyset \), for all \( y \neq z \). Combining this with (6.32) and (6.33), we conclude that \( V_{y} = \hat{\Omega}_{y} \), for all \( y \in \mathcal{P} \), which completes the proof of (6.28).

As in the proof of Theorem 6.2.2, one can finally show that

\[
\{ x \in I | (g^{(4)} \circ g^{(5)} \circ g^{(3)} \circ g^{(1)})(x) = 1 \} = \bigcup_{z \in \mathcal{P}, n=1} \hat{\Omega}_{z},
\]

for all \( i = 1, \ldots, K \). Hence, using Theorem 5.7.1 and Proposition 6.2.1, this proves that \( f = g^{(4)} \circ g^{(5)} \circ g^{(3)} \circ g^{(1)} \) solves \((I, F, c)\).

Combining Theorem 6.3.5 with Corollary 6.3.2, yields the other half of the proof of Theorem 6.3.3.

**Corollary 6.3.4.** Let \((I, F, c)\) be a COP that satisfies Conditions 1a, 1b, 2a, and 2b. Then there exists an MLP with \( L(I, F, c; M^{\mathcal{F}}) \) first-layer nodes that solves \((I, F, c)\).

For completeness, we present the proof of Theorem 6.3.3.

**Proof of Theorem 6.3.3.** Follows directly from the fact that Condition 2b implies Condition 3b, and using Corollary 6.3.3 and Corollary 6.3.4.

At first we tried to construct an MLP without using a preference function in the definition of the second hidden layer. Then, we considered an arbitrary preference function, similar like we used it for the second hidden layer of the MLP constructed in Theorem 6.2.2. These attempts failed, because of the following reason. In both cases, one obtains a second layer that classifies subsets similar to \( \hat{\Omega}_{y}(N) \), that assume the following form,

\[
\{ x \in I | \forall z \in \mathcal{P}_{y} : c(y; x) < c(z; x) \lor [c(y; x) = c(z; x) \land q(y, z)] \},
\]

(6.34)

where \( q(y, z) \) is a boolean function that depends on the preference function used and on the sign of \( \tau_{yk} \). However, in these cases there is no equivalent of (6.29). Hence, we cannot eliminate \( \tau_{yk} \), by combining (6.21) and (6.29), which is an essential step in the proof of (6.27). Furthermore, it is not possible to "globalize" \( \tau_{yk} \) to all \( y, z \in F \), because it is a "local" function that has a meaning for \( x \in \mathcal{P}_{y} \) only. Therefore, we cannot construct subsets similar to \( \hat{\Omega}_{y} \), that are contained in the subset of (6.34). Consequently, we cannot prove that the subsets of (6.34) cover \( I \), which is crucial for the proof that the constructed MLP solves the considered COP.

The additional benefit of the use of the preference function \( p(y) = -c(y, z) \), is that it makes it possible to relate the preference order of a neighboring pair \((y, z)\), with
\( z \in N_z \) to the preference order of the corresponding neighboring pair \((w^{(x)}, y^{(x)})\), as given by (6.29). Below, we prove that the third hidden layer of the MLP given by Theorem 6.3.3 is superfluous. The proof of this result is based on the fact that for this choice of the preference function, the preference order extends even beyond the borders of the neighborhood, i.e., we show that \( \Omega_z(N) = \Omega_y \), for all \( y \in F \), in case that Condition 3c holds; see Theorem 6.3.6. Before we can prove this, we give two lemmas.

For the first lemma we return to the basics of halfspaces and polyhedrons; see also Sections 5.2 and 5.4 of Chapter 5. We present this lemma as an alternative condition for a halfspace to correspond to a valid inequality of a polyhedron; see Definition 5.4.8. It can be viewed as an extended version of Farkas' lemma, since it contains Farkas' lemma as a special case; see Papadimitriou & Steiglitz [1982], and also Proposition 2.2.1 in Chapter 2. Furthermore, it is strongly related to the so-called 'affine' form of Farkas' lemma; see Schrijver [1986]. As usual, it is convenient to prove this Farkas-type lemma by exploiting the strong duality property of the primal and dual formulation of a linear program; see also Papadimitriou & Steiglitz [1982], Nemhauser & Wolsey [1988], and Schrijver [1986].

**Lemma 6.3.3.** Let \( W = \{ x \in \mathbb{R}^N | a \cdot x \leq b \} \) be a halfspace, for some \( a \in \mathbb{R}^N \) and \( b \in \mathbb{R} \). Let \( V = \bigcap_{i=1}^K W_i \) be a non-empty polyhedron for some \( K \in \mathbb{N} \), with \( W_i = \{ x \in \mathbb{R}^N | a_i^T \cdot x \leq b_i \} \), for some \( a_i^T \in \mathbb{R}^N \) and \( b_i \in \mathbb{R} \), \( i = 1, \ldots, K \). Then \( W \) corresponds to a valid inequality for \( V \), i.e., \( V \subseteq W \), if and only if there exist \( \lambda_i \geq 0 \), \( i = 1, \ldots, K \), such that

\[
a \cdot x - b \leq \sum_{i=1}^K \lambda_i (a_i \cdot x - b_i),
\]

for all \( x \in \mathbb{R}^N \). Furthermore, if \( b = 0 \) and \( b_i = 0 \), for all \( i = 1, \ldots, K \), then the inequality in (6.35) can be replaced by equality.

**Proof.** The 'if'-part is trivial, so it remains to prove the 'only-if'-part.

Assume that \( V \subseteq W \). Then \( a \cdot x \leq b \), for all \( x \in V \), which is equivalent to \( \max \{ a \cdot x | a \cdot x \leq b_i, i = 1, \ldots, K, x \in \mathbb{R}^N \} \leq b \). Hence, using that \( V \neq \emptyset \), it follows by the strong duality property of linear programming that the dual problem

\[
\min \{ \sum_{i=1}^K \lambda_i b_i | \sum_{i=1}^K \lambda_i a_i = a, \lambda_i \geq 0, i = 1, \ldots, K \},
\]

has a feasible solution that satisfies \( \sum_{i=1}^K \lambda_i b_i \leq b \). Hence, we obtain that there exist \( \lambda_1 \geq 0, \ldots, \lambda_K \geq 0 \), such that \( \sum_{i=1}^K \lambda_i a_i = a \) and \( \sum_{i=1}^K \lambda_i b_i \leq b \), from which the result follows. \( \square \)

Farkas' Lemma is contained in Lemma 6.3.3 as the special case that \( b = 0 \) and \( b_i = 0 \), for all \( i = 1, \ldots, K \); see also Proposition 2.2.1.

Applying Lemma 6.3.3 to the polyhedrons \( \Omega_z(N) \) yields a second alternative condition for a neighborhood structure \( N \) to be exact. The condition that the polyhedron is not empty, which is an essential condition in order to be able to apply Lemma 6.3.3, is translated into Condition 3c.
Lemma 6.3.4. Let \((I, F, c)\) be a COP that satisfies Conditions 1a, 1b, 2a, and 3c. Let \(\mathcal{N}\) be a neighborhood structure for this COP. Then \(\mathcal{N}\) is exact, if and only if for all \(y, z \in F\), there exist \(\lambda_{y} \geq 0, y' \in \mathcal{N}_{y}\), such that
\[
 c(y; z) - c(z; x) \leq \sum_{x' \in \mathcal{N}_{y}} \lambda_{x'}(c(y; x) - c(x'; x)),
\]  
(6.36)
for all \(x \in I\). Furthermore, if \(I = \mathbb{R}^{N}\) and \(c(y; 0) = c(z; 0)\), for all \(y, z \in F\), then the inequality in (6.36) can be replaced by equality.

Proof. The 'if'-part is trivial, so it remains to prove the 'only-if'-part.

Assume that \(\mathcal{N}\) is exact, then using Proposition 6.3.2 it follows that \(\Omega_{y}(\mathcal{N}) \subseteq \Omega_{y}\), for all \(y \in F\), which implies that
\[
\Omega_{y}(\mathcal{N}) \subseteq \{ x \in \mathbb{R}^{N} | c(y; x) \leq c(z; x) \},
\]  
(6.37)
for all \(y, z \in F\). We complete the proof using Lemma 6.3.3. Let \(y, z \in F\). Then, since \((I, F, c)\) satisfies Conditions 1a, 1b, 2a, and 3c, \(\Omega_{y}(\mathcal{N})\) corresponds to a non-empty polyhedron. Let \(I = \{ x \in \mathbb{R}^{N} | a_i \cdot x \leq b_i, i = 1, \ldots, K \}\), for some \(K \in \mathbb{N}\), \(a_i \in \mathbb{R}^{N}\), and \(b_i \in \mathbb{R}\), for \(i = 1, \ldots, K\). Then \(\Omega_{y}(\mathcal{N})\) can be written as
\[
\Omega_{y}(\mathcal{N}) = \bigcap_{x' \in \mathcal{N}_{y}} \{ x \in \mathbb{R}^{N} | c(y; x) \leq c(z; x) \} \cap \bigcap_{i=1}^{K} \{ x \in \mathbb{R}^{N} | a_i \cdot x \leq b_i \}.
\]
Applying Lemma 6.3.3 to \(\Omega_{y}(\mathcal{N})\), implies that since (6.37) holds, there exist \(\lambda_{y} \geq 0,\)
\(x' \in \mathcal{N}_{y}\), and \(\lambda_{i} \geq 0, i = 1, \ldots, K\), such that
\[
c(y; z) - c(z; x) \leq \sum_{x' \in \mathcal{N}_{y}} \lambda_{x'}(c(y; x) - c(x'; x)) + \sum_{i=1}^{K} \lambda_{i}(a_i \cdot x - b_i),
\]
for all \(x \in \mathbb{R}^{N}\), with equality if \(c(y; 0) = c(z; 0)\), \(c(y; 0) = c(z'; 0)\), for all \(z' \in \mathcal{N}_{y}\), and \(b_i = 0\), for all \(i = 1, \ldots, K\). From this the result easily follows. \(\square\)

Lemma 6.3.4 is used below to show that the third hidden layer of the 4LP constructed in Theorem 6.3.5 is superfluous. To this end we prove that the second layer has the same output as the third hidden layer. Because we use Lemma 6.3.4, the requirements posed upon the considered COPs must be slightly strengthened in comparison with the requirements of Theorem 6.3.5. The conditions posed in Theorem 6.3.6 are the combined conditions of Theorem 6.3.5 and Lemma 6.3.4.

Theorem 6.3.6. Let \((I, F, c)\) be a COP that satisfies Conditions 1a, 1b, 2a, 3b, and 3c, with \(f \subseteq \mathbb{R}^{N}\) and \(F \subseteq \mathbb{B}^{N}\), for some \(N, K \in \mathbb{N}\). Let \(\mathcal{N}\) be an exact neighborhood structure for this COP. Let \(g^{(1)}, l = 1, 2, 3, 4\), be the functions defined in Theorem 6.3.5. Then \(f = g^{(0)} \circ g^{(2)} \circ g^{(1)}\) represents a 3LP, with \(N\) inputs, \(|I|\) nodes in the first hidden layer, \(|F|\) nodes in the second hidden layer, and \(K\) outputs, that solves \((I, F, c)\).

Proof. Consider the 4LP represented by \(g^{(0)} \circ g^{(2)} \circ g^{(1)}\), that is constructed in Theorem 6.3.5. In the proof of Theorem 6.3.5 it is shown that the outputs of the second and third hidden layer satisfy \(\{ x \in I | g^{(2)}(g^{(1)}(x)) \} = \Omega_{y}(\mathcal{N})\), for all \(y \in F\), and \(\{ x \in I | g^{(2)}(g^{(1)}(x)) \} = \Omega_{y}\), for all \(y \in F\), respectively, with \(\Omega_{y}(\mathcal{N})\) and \(\Omega_{y}\) given by (6.20) and (6.19); see (6.27) and (6.28). Using Lemma 6.3.4, we prove below that \(\Omega_{y}(\mathcal{N}) = \Omega_{y}\), for all \(y \in F\). This implies that the output of the second hidden layer, given by \(g^{(2)}(g^{(1)}(x))\), is equal to the output of the third
hidden layer, given by \( g^{(3)}(g^{(2)}(g^{(1)}(x))) \), for all \( x \in I \), which makes the third hidden layer superfluous.

It is trivial that \( \hat{\Omega}_g \subseteq \hat{\Omega}_y(\mathcal{N}) \), for all \( y \in F \); see also (6.31). Hence, it remains to show that \( \hat{\Omega}_y(\mathcal{N}) \subseteq \hat{\Omega}_y \), for all \( y \in F \). Let \( y \in F \) and assume that \( x \in \hat{\Omega}_y(\mathcal{N}) \). Then \( x \in I \), and for all \( z' \in \mathcal{N}_y \), \( c(y; z) < c(z'; x) \), or \( c(y; z) \leq c(z'; x) \) and \( c(y; \hat{z}) < c(z'; \hat{z}) \), where \( \hat{z} \) is the parameter distinguished in Lemma 6.3.2, that is used in the definition of \( \hat{\Omega}_y(\mathcal{N}) \). In order to show that \( x \in \hat{\Omega}_y \), we must prove that for all \( z \in F \setminus \{y\} \), \( c(y; z) < c(z; x) \), or \( c(y; z) \leq c(z; x) \) and \( c(y; \hat{z}) < c(z; \hat{z}) \).

Since \( \mathcal{N} \) is exact, it follows that \( c(y; x) \leq c(z; x) \), for all \( z \in F \). Assume that \( c(y; x) = c(z; x) \), for some \( z \in F \setminus \{y\} \), then we must prove that \( c(y; \hat{z}) < c(z; \hat{z}) \). To this end we note that by Lemma 6.3.4, there exist \( \lambda_{y'} \geq 0, \alpha' \in \mathcal{N}_y \), such that

\[
c(y; z') - c(z; x') \leq \sum_{z' \in \mathcal{N}_y} \lambda_{y'}(c(y; z') - c(z'; x')), \tag{6.38}
\]

for all \( z' \in I \). First, we use (6.38) with \( z' = x \). Since \( c(y; x) = c(z; x) \), \( c(y; x) \leq c(z'; x) \), for all \( z' \in \mathcal{N}_y \), and \( \lambda_{y'} \geq 0 \), for all \( z' \in \mathcal{N}_y \), we obtain from (6.38) with \( z' = x \) that either \( \lambda_{y'} = 0 \) or \( c(y; x) = c(z'; x) \), for all \( z' \in \mathcal{N}_y \). Consequently, \( \lambda_{y'} = 0 \) or \( c(y; \hat{z}) < c(z'; \hat{z}) \), for all \( z' \in \mathcal{N}_y \). Hence, using (6.38) once more with \( z' = \hat{z} \), yields \( c(y; \hat{z}) \leq c(z; \hat{z}) \). Since \( y \neq z \), this implies \( c(y; \hat{z}) < c(z; \hat{z}) \), which completes the proof of the theorem.

Theorem 6.3.6 yields a 3LP with minimal sized first layer and a second layer of size |\( F \)|, in case that one uses the neighborhood structure \( \mathcal{M}^{IF} \), which is exact if Condition 2b holds; see Corollary 6.3.2. Since Condition 2b implies Conditions 3a, 3b, and 3c, we find the following result.

**Corollary 6.3.5.** Let \((I, F, c)\) be a COP that satisfies Conditions 1a, 1b, 2a, and 2b. Then there exists a 3LP with \( L(I, F, c; \mathcal{M}^{IF}) \) first-layer nodes and \(|F| \) second-layer nodes that solves \((I, F, c)\).

The determination of the size \( L(I, F, c; \mathcal{M}^{IF}) \) of the first layer, for a given COP \((I, F, c)\), is the subject of the following subsection.

### 6.3.4 The determination of the minimal size of the first-layer for a given COP

In this section we discuss some issues related to the determination of \( L(I, F, c; \mathcal{M}^{IF}) \), which corresponds to the minimal number of first-layer nodes of any MLP that solves the COP \((I, F, c)\). Throughout this section we assume that the considered COP satisfies Conditions 1a, 1b, 2a, and 2b.

The determination of \( L(I, F, c; \mathcal{M}^{IF}) \) obviously starts with the determination of the neighborhood structure \( \mathcal{M}^{IF} \) for the problem at hand. The usual approach is that one tries to prove that a certain neighborhood structure \( \mathcal{N} \) is equal to \( \mathcal{M}^{IF} \). This can be done by showing that (i) \( \mathcal{N} \subseteq \mathcal{M}^{IF} \), for all \( y \in F \), and (ii) \( \mathcal{M}^{IF} \subseteq \mathcal{N} \), for all \( y \in F \). Using (6.16), the first part boils down to proving that for all \( y \in F \) and \( z \in \mathcal{N}_y \), and there exists an \( x \in I^c \) such that \( c(z; x) = c(y; z) < c(z'; x) \), for all \( z' \in F \setminus \{y, z\} \). Using (6.16) for the second part is not that convenient. Therefore, we consider two alternative approaches for proving that \( \mathcal{M}^{IF} \subseteq \mathcal{N} \), for all \( y \in F \).
6.3 The minimal number of first-layer nodes

Approach I. Show that $\mathcal{N}$ is exact.

Approach II. Show that $\mathcal{M}_y^F \subseteq \mathcal{N}_y$, for all $y \in F$.

The first approach is based on Lemma 6.3.1 and Corollary 6.3.1, while the second approach is based on Corollary 6.3.1 only. We compare these two approaches, to find out if there are any fundamental differences. To this end we give two variants of both approaches. We start with the first approach.

**Approach Ia.** Use the definition of an exact neighborhood structure, i.e., show that for all $y \in F$, $z \notin \mathcal{N}_y$, and $x \in I$, if $c(z; x) < c(y; x)$, then there exists a $x' \in \mathcal{N}_y$, with $c(x'; x) < c(y; x)$.

**Approach Ib.** Use Lemma 6.3.4, i.e., show that for all $y \in F$ and $x \notin \mathcal{N}_y$, there exist $\lambda_x \geq 0$, $x' \in \mathcal{N}_y$, such that
\[
c(y; x) - c(z; x) \leq \sum_{x' \in \mathcal{N}_y} \lambda_x(\lambda_y c(y; x) - c(z'; x)), \tag{6.39}
\]
for all $x \in I$.

Similarly, we have the following two variants of the second approach.

**Approach IIa.** Using (6.14), one can show that $\mathcal{M}_y^F \subseteq \mathcal{N}_y$, for all $y \in F$, if and only if for all $y \in F$, $z \notin \mathcal{N}_y$, and $x \in I$, if $c(z; x) < c(y; x)$, then there exists a $x' \neq x$, with $c(x'; x) < c(y; x)$.

**Approach IIb.** Similarly as done in Lemma 6.3.4, one can show using Lemma 6.3.3, that $\mathcal{M}_y^F \subseteq \mathcal{N}_y$, for all $y \in F$, if and only if for all $y \in F$ and $z \notin \mathcal{N}_y$, there exist $\lambda_x \geq 0$, $x' \neq x$, such that
\[
c(y; z) - c(z; x) \leq \sum_{x' \neq x} \lambda_x(\lambda_y c(y; x) - c(z'; x)), \tag{6.40}
\]
for all $x \in I$.

Comparing the burden of proof of both variants of the two approaches, it appears that in theory the second approach is more attractive. However, in practice there seems to be little difference. For instance, if one can show that for some $z \notin \mathcal{N}_y$ with $c(z; x) < c(y; x)$, there exists a $x' \neq x$ with $c(x'; x) < c(y; x)$, one can usually apply induction with respect to a properly defined distance on $F$, that leads to the conclusion that there exists a $z'' \in \mathcal{N}_y$, with $c(z''; x) < c(y; x)$. In fact this is a usual approach to prove that the neighborhood $\mathcal{N}$ is exact; see for instance Savage, Weiner & Bagchi [1976]. Similarly, we do not know of a COP and a neighborhood structure $\mathcal{N}$ for which it is easy to prove (6.40) without proving (6.39) at the same time; see also Chapter 7, where we determine $\mathcal{M}_I^F$ for some example COPs.

Once an explicit formulation of $\mathcal{M}_I^F$ has been determined, it is usual straightforward to compute $L(I, F, c; \mathcal{M}_I^F)$; see also Chapter 7. However, it is not always possible to find such an explicit formulation. For instance, Savage has found only a subset of the minimal exact neighborhood for the traveling salesman problem; see Savage [1973] and Savage, Weiner & Bagchi [1976]. In these cases it might be impossible to calculate $L(I, F, c; \mathcal{M}_I^F)$. Below we prove that $L(I, F, c; \mathcal{M}_I^F) \geq \max_{y \in F} |\mathcal{M}_y^F|$, which gives a lower bound for the number of nodes that is required by any MLP that solves the considered COP, if one has a lower bound on $\max_{y \in F} |\mathcal{M}_y^F|$. For completeness the theorem presents a trivial upper bound on $L(I, F, c; \mathcal{M}_I^F)$ also.
Theorem 6.3.7. Let \((I, F, c)\) be a COP that satisfies Conditions 1a and 1b. Then we have
\[
\max_{y \in F} |\mathcal{M}_y^{IP}| \leq L(I, F, c; \mathcal{M}^{IP}) \leq \frac{1}{2} |F| \max_{y \in F} |\mathcal{M}_y^{IP}|.
\]

Proof. The upper bound follows trivially from the definition of \(L(I, F, c; \mathcal{M}^{IP})\), using that \(\mathcal{M}^{IP}\) is symmetric. Hence, it remains to prove the lower bound.

Let \(y \in F\), such that \(|\mathcal{M}_y^{IP}| = \max_{y \in F} |\mathcal{M}_y^{IP}|\). Let \(z, z' \in \mathcal{M}_y^{IP}\), then there exist \(x, x' \in I\) such that \(c(z; x) = c(y; x) < c(z'; x)\) and \(c(z'; x') = c(y; x') < c(z; x')\). Hence, there cannot exist a \(\tau \in \mathbb{R}\), such that
\[
c(y; x) - c(z; x) = \tau (c(y; x) - c(z'; x)),
\]
for all \(x \in I\). This implies that all the neighboring pairs \((y, z)\), with \(z \in \mathcal{M}_y^{IP}\), are non-equivalent. Hence, using Definition 6.3.6, we find that \(L(I, F, c; \mathcal{M}^{IP}) \geq |\mathcal{M}_y^{IP}| = \max_{y \in F} |\mathcal{M}_y^{IP}|\).

Savage considered the determination of the minimal exact neighborhood of the traveling salesman problem, and showed that there exist feasible solutions for which the size of the minimal exact neighborhood is at least exponential; see Savage [1973] and Savage, Weiner & Bagchi [1976]. Hence, combining Theorem 6.3.7 and Corollary 6.3.3, it follows that any MLP that solves the traveling salesman problem, has a first hidden layer with exponential size. In Chapter 7 we consider the determination of \(\mathcal{M}^{IF}\) and \(L(I, F, c; \mathcal{M}^{IF})\) for five other COPs.

The final result of this section generalizes Theorem 6.3.7 to arbitrary neighborhood structures, which might be useful in combination with Theorem 6.3.5. To this end, we require Theorem 6.3.8 presented below.

Theorem 6.3.8 (Savage [1973]). Let \((I, F, c)\) be a COP satisfying Conditions 1a, 1b, and 2b. Then for all \(y, z, z' \in F\), with \(y, z,\) and \(z'\) all different, there does not exist a \(\tau \in \mathbb{R}\) such that
\[
c(y; x) - c(z; x) = \tau (c(y; x) - c(z'; x)),
\]
for all \(x \in I\).

Proof. Let \(y, z, z' \in F\), with \(y, z,\) and \(z'\) all different, and assume that \(\tau \in \mathbb{R}\) is such that (6.41) holds for all \(x \in I\). Then one easily verifies that also
\[
(\tau - 1)(c(z; x) - c(y; x)) = \tau (c(z; x) - c(z'; x)),
\]
for all \(x \in I\), and
\[
(1 - \tau)(c(z'; x) - c(y; x)) = c(z'; x) - c(z; x),
\]
for all \(x \in I\).

Using Condition 2b, it follows that there exist \(x_1, x_2, x_3 \in I\), such that \(c(y; x_1) < c(z; x_1)\) and \(c(y; x_1) < c(z'; x_1)\), \(c(z; x_2) < c(y; x_2)\) and \(c(z; x_2) < c(z'; x_2)\), and \(c(z'; x_3) < c(y; x_3)\) and \(c(z'; x_3) < c(z; x_3)\). Substituting \(x_1, x_2,\) and \(x_3\) in (6.41), (6.42), and (6.43), respectively, one easily verifies that at least one of the equations cannot hold.

As in the proof of Theorem 6.3.7, it follows by Theorem 6.3.8 that we have the following general result.
Corollary 6.3.6. Let \((I, F, c)\) be a COP that satisfies Conditions 1a, 1b, and 1b. Let \(\mathcal{N}\) be a neighborhood structure for this COP. Then we have
\[
\max_{e \in E} |\mathcal{N}_e| \leq L(I, F, c; \mathcal{N}) \leq |F| \max_{e \in E} |\mathcal{N}_e|.
\]

6.4 The minimal number of layers

In Section 6.2 we showed that all COPs that satisfy Conditions 1a and 1b, can be solved by a 3LP. Section 6.3 improves upon this result by presenting a construction of a 3LP with minimal sized first layer that solves the considered COP, in case that it satisfies the two additional Conditions 2a and 2b. In this section we study the question which COPs can be solved by a 2LP. Answering this question would yield the minimal number of layers that is required by any MLP that solves a considered COP, except for some trivial COPs that can be solved by a 1LP.

As in the previous sections, the approach is to translate the results for CCPs that are presented in Chapter 5, into results for COPs. Here, this means that we translate the results on the classification capabilities of 2LPs, into conditions for being able to solve a COP with a 2LP. The results on the classification capabilities concern a number of necessary conditions and a number of sufficient conditions for a subset to be classifiable with a 2LP. The necessary conditions that we have derived, are the bow-tie condition, the twisted bow-tie condition, and some variants of these two. Below we present two necessary conditions for a COP to be solvable by a 2LP, that are based on the bow-tie condition and the twisted bow-tie condition, respectively; see Theorem 6.4.1 and Theorem 6.4.2.

Both results concern four instances and four feasible solutions that are strictly optimal for these instances. Furthermore, the four feasible solutions are pairwise opposite at one coordinate; see condition (b) below. The two results mainly differ with respect to the last condition; see condition (e) in the two theorem formulations. The necessary condition based on the bow-tie condition has as additional requirement a condition concerning two coinciding hyperplanes in parameter space, while the necessary condition based on the twisted bow-tie condition has as additional requirement a condition concerning two coinciding points in parameter space.

In the proof of the following first result, based on the bow-tie condition, we use a similar approach as used in the proof of Theorem 6.3.4.

Theorem 6.4.1 (The bow-tie condition for COPs). Let \((I, F, c)\) be a COP that satisfies Conditions 1a and 1b, with \(I \subseteq \mathbb{R}^d\) and \(F \subseteq \mathbb{B}^d\), for some \(N, K \in \mathbb{N}\). Let \(y^{(i)}, y^{(3)}; y^{(3)}, y^{(4)} \in F\) and \(x_1, x_2, x_3, x_4 \in I\) satisfy the following conditions.

\[
\begin{align*}
(a) & \quad c(y^{(j)}; x_j) \prec c(z; x_j), \text{ for all } z \in F \setminus \{y^{(j)}\} \text{ and } j = 1, 2, 3, 4. \\
(b) & \quad y^{(i)}_i = y^{(3)}_i = 1 \text{ and } y^{(3)}_i = y^{(4)}_i = 0, \text{ for some } i \in \{1, \ldots, K\}. \\
(c) & \quad x_{13} = (x_1 + x_3)/2 \in I^s \text{ and } x_{24} = (x_2 + x_4)/2 \in I^s. \\
(d) & \quad c(y^{(1)}; x_{13}) = c(y^{(3)}; x_{13}) \prec c(z; x_{13}), \text{ for all } z \in F \setminus \{y^{(1)}, y^{(3)}\}, \text{ and } c(y^{(2)}; x_{24}) = c(y^{(4)}; x_{24}) \prec c(z; x_{24}), \text{ for all } z \in F \setminus \{y^{(2)}, y^{(4)}\}. \\
(e) & \quad \text{sign}[c(y^{(j)}; z) - c(y^{(3)}; z)] = \text{sign}[c(y^{(j)}; z) - c(y^{(3)}; z)], \text{ for all } z \in I.
\end{align*}
\]
Then \((I, F, c)\) cannot be solved by a 2LP.

**Proof.** Assume that \(f \in R_{2,N,K}\) represents a 2LP that solves \((I, F, c)\). Let \(V = J(f)\), where \(i \in \{1, \ldots, K\}\) is given by \((b)\). Then \(V \in C_{2,1}\), i.e., \(V\) is a subset in \(R^{N}\) that can be classified with a 2LP. However, we show that the above conditions imply that \(V\) does not satisfy the bow-tie condition presented in Section 5.5.1, which yields a contradiction.

Using \((c), (d)\), and the fact that, by Condition 1b, \(c(z; x)\) is continuous in \(x\), it follows that there exist \(\delta_1 > 0\) and \(\delta_2 > 0\), such that

- \(B(x_{13}, \delta_1) \subseteq I \text{ and } B(x_{24}, \delta_2) \subseteq I\).
- \(c(y^{(1)}; x) < c(z; x)\) and \(c(y^{(3)}; x) < c(z; x)\), for all \(z \in F \setminus \{y^{(1)}, y^{(3)}\}\), and \(x \in B_1 = B(x_{13}, \delta_1)\).
- \(c(y^{(2)}; x) < c(z; x)\) and \(c(y^{(4)}; x) < c(z; x)\), for all \(z \in F \setminus \{y^{(2)}, y^{(4)}\}\), and \(x \in B_2 = B(x_{24}, \delta_2)\).

Since \(f\) solves \((I, F, c)\), these results imply that \(f(x) \in \{y^{(1)}, y^{(3)}\}\), for all \(x \in B_1\), and \(f(x) \in \{y^{(2)}, y^{(4)}\}\), for all \(x \in B_2\).

Let \(W_{13} = \{x \in R^N | c(y^{(1)}; x) \leq c(y^{(3)}; x)\}\). Using \((a)\) and Condition 1b, one easily shows that \(W_{13} \in H\). Furthermore, using \((b)\) and the above, it follows that \(f_1(x) = y^{(1)} = 1\), for all \(x \in W_{13} \cap B_1\), and \(f_1(x) = y^{(3)} = 0\), for all \(x \in W_{13} \cap B_1\). Since \(x_{13} \in W_{13} \cap B_1\), both \(W_{13} \cap B_1\) and \(W_{13} \cap B_2\) are not empty.

Let \(W_{42} = \{x \in R^N | c(y^{(4)}; x) \leq c(y^{(2)}; x)\}\). Using \((a)\) and Condition 1b, one easily shows that \(W_{42} \in H\). Furthermore, using \((b)\) and the above, it follows that \(f_1(x) = y^{(4)} = 0\), for all \(x \in W_{42} \cap B_2\), and \(f_1(x) = y^{(2)} = 1\), for all \(x \in W_{42} \cap B_2\). Since \(x_{24} \in W_{42} \cap B_2\), both \(W_{42} \cap B_2\) and \(W_{42} \cap B_2\) are not empty.

By \((c)\), \(\text{sign}[c(y^{(1)}; x) - c(y^{(3)}; x)] = \text{sign}[c(y^{(4)}; x) - c(y^{(2)}; x)]\), for all \(x \in B_1\). Since \(x_{13} \in W_{13} \cap B_1\), this is easily shown to imply that \(W_{13} = W_{42}\). Hence, letting \(W = W_{13} = W_{42}\), we obtain the following result.

\[ \emptyset \neq W^* \cap B_1 \subseteq V \land \emptyset \neq W^* \cap B_2 \subseteq V^* \land \emptyset \neq W^* \cap B_2 \subseteq V. \]

Obviously, this results in a contradiction with Theorem 5.5.1.

In Chapter 7 we use Theorem 6.4.1 to prove that \textsc{real sorting} cannot be solved by a 2LP.

The proof of the second result, based on the twisted bow-tie condition, has the same structure as the proof of Theorem 6.4.1.

**Theorem 6.4.2 (The twisted bow-tie condition for COPs).** Let \((I, F, c)\) be a COP that satisfies Conditions ia and 1b, with \(I \subseteq R^N\) and \(F \subseteq B^K\), for some \(N, K \in \mathbb{N}\). Let \(y^{(1)}, y^{(2)}, y^{(3)}, y^{(4)} \in F\) and \(x_1, x_2, x_3, x_4 \in I\) satisfy the following conditions.

\[
(a) \ c(y^{(j)}; x_j) < c(z; x_j), \ \text{for all } z \in F \setminus \{y^{(j)}\} \ \text{and } j = 1, 2, 3, 4.
\]
6.4 The minimal number of layers

(b) \( y^{(1)} = y^{(2)} = 1 \) and \( y^{(3)} = y^{(4)} = 0 \), for some \( i \in \{1, \ldots, K\} \).

(c) \( z_{12} = (z_1 + z_2)/2 \in I^* \) and \( z_{34} = (z_3 + z_4)/2 \in I^* \).

(d) \( c(y^{(1)}; z_{12}) \leq c(z; z_{12}) \), for all \( z \in F \setminus \{y^{(1)}, y^{(2)}\} \), and

\[ c(y^{(3)}; z_{34}) \leq c(z; z_{34}) \], for all \( z \in F \setminus \{y^{(3)}, y^{(4)}\} \).

(e) \( z_{12} = z_{34} \).

Then \((I, F, c)\) cannot be solved by a 2LP.

Proof. Assume that \( f \in R_{2,N,K} \) represents a 2LP that solves \((I, F, c)\). Let \( V = J(f_i) \), where \( i \in \{1, \ldots, K\} \) is given by (b). Then \( V \in C_3 \), i.e., \( V \) is a subset in \( \mathbb{R}^N \) that can be classified with a 2LP. However, we show that the above conditions imply that \( V \) does not satisfy the twisted bow-tie condition presented in Section 5.5.1, which yields a contradiction.

By (e) and (c), we can set \( z_0 = z_{12} = z_{34} \in I^* \). Furthermore, Since \( V \in C_3 \), we know that \( V \in U \); see Lemma 5.3.2. Hence, using Proposition 5.5.1, it follows that \( V \) is locally radial around \( z_0 \), i.e., there exist a \( \delta_0 > 0 \) such that for all \( 0 < \lambda < 1 \) and \( z \in B(z_0, \delta_0) \), it holds that \( \lambda z + (1 - \lambda)z_0 \in V \), if and only if \( z \in V \); see also Definition 5.5.1. Since \( z_0 \in I^* \), we may assume without loss of generality that \( B_0 = B(z_0, \delta_0) \subset I \).

Choose \( \lambda_1, \lambda_3 \in (0, 1) \), such that \( \lambda_1 z_1 + (1 - \lambda_1)z_0 \in B_0 \) and \( \lambda_3 z_3 + (1 - \lambda_3)z_0 \in B_0 \). Define \( \lambda_2 = \lambda_1 \), \( \lambda_4 = \lambda_3 \), and \( \tilde{z}_j = \lambda_j z_j + (1 - \lambda_j)z_0 \), for \( j = 1, 2, 3, 4 \). Then using (c), one easily verifies that \( \tilde{z}_j \in B_0 \), for all \( j = 1, 2, 3, 4 \), and \( z_0 = (\tilde{z}_1 + \tilde{z}_2)/2 = (\tilde{z}_3 + \tilde{z}_4)/2 \). Furthermore, using (a), (d), and Condition 1b, we obtain

\[ c(y^{(j)}; \tilde{z}_j) = c(y^{(j)}; \lambda_j z_j + (1 - \lambda_j)z_0) \]

\[ = \lambda_j c(y^{(j)}; z_j) + (1 - \lambda_j) c(y^{(j)}; z_0) \]

\[ < \lambda_j c(z; z_j) + (1 - \lambda_j) c(z; z_0) \]

\[ = c(z; z_j) \]

for all \( z \in F \setminus \{y^{(j)}\} \), and \( j = 1, 2, 3, 4 \). Hence, we can choose \( \delta_j > 0 \), \( j = 1, 2, 3, 4 \), such that \( B(\tilde{z}_j, \delta_j) \subset B_0 \) and \( c(y^{(j)}; x) < c(z; x) \), for all \( z \in F \setminus \{y^{(j)}\} \), \( z \in B(\tilde{z}_j, \delta_j) \), and \( j = 1, 2, 3, 4 \).

Let finally \( \mu_1 = \min\{\delta_1, \delta_2\} \), \( \mu_3 = \min\{\delta_3, \delta_4\} \), and

\[ B_1 = B(\tilde{z}_1, \mu_1) = B(z_0 + (\tilde{z}_1 - \tilde{z}_2)/2, \mu_1) \],

\[ B_2 = B(\tilde{z}_2, \mu_1) = B(z_0 - (\tilde{z}_1 - \tilde{z}_2)/2, \mu_1) \],

\[ B_3 = B(\tilde{z}_3, \mu_3) = B(z_0 + (\tilde{z}_3 - \tilde{z}_4)/2, \mu_3) \],

\[ B_4 = B(\tilde{z}_4, \mu_3) = B(z_0 - (\tilde{z}_3 - \tilde{z}_4)/2, \mu_3) \].

Then, since \( B_j \subset B(\tilde{z}_j, \delta_j) \subset B_0 \subset I \), for \( J = 1, 2, 3, 4 \), and \( f \) solves \((I, f, c)\), the above implies \( f(x) = y^{(j)} \), for all \( x \in B_j \) and \( j = 1, 2, 3, 4 \). Combined with (b), this yields

\[ B_1 \subset V \land B_2 \subset V, \]

\[ B_3 \subset V^* \land B_4 \subset V^* \]

Obviously, this results in a contradiction with Theorem 5.5.2. \( \square \)
In Chapter 7 we use Theorem 6.4.2 to prove that REAL DYNAMIC LOTSIZING cannot be solved by a 2LP.

We have chosen the formulations of Theorems 6.4.1 and 6.4.2 in such a way that they resemble each other as much as possible, and do not lose their minimality. With this we mean that in those cases where one can show that a COP is not solvable by a 2LP, using the bow-tie condition or the twisted bow-tie condition, the considered COP must satisfy the conditions of Theorem 6.4.1, or those of Theorem 6.4.2, respectively. This can be demonstrated using (6.12) and Corollary 5.4.5.

We have not been able to translate any of the sufficient conditions for a subset to be classifiable with a 2LP, into a sufficient condition for an arbitrary COP to be solvable by a 2LP. This is mainly due to the fact that our approach for solving COPs with MLPs is based on the classification of a number of unions of pseudo polyhedrons, as given by (6.10), for which we do not have a general sufficient condition that guarantees their classifiability with a 2LP. In Chapter 7 we present a formulation of REAL SORTING that can be solved by a 2LP. This also stresses the fact that whether or not a problem can be solved by a 2LP, may depend on its formulation.

6.5 Concluding remarks

In this chapter we have derived sufficient conditions for a COP to be solvable with a 3LP. Using Theorem 6.2.2, one can easily find the weights of a 3LP that solves a given COP. In a paper by Zwietering, Aarts & Wessels an example is given for REAL SORTING; see Zwietering, Aarts & Wessels [1991a], and also Zwietering, Aarts & Wessels [1993], Zwietering, Aarts & Wessels [1992b], and the discussion of REAL SORTING in the Chapters 3 and 7.

We have found an expression for the minimal number of first-layer nodes that is required by any MLP that solves a given COP, which is based on the minimal exact neighborhood for the considered problem. We have presented a construction of a 3LP with minimal sized first layer and \(|F|\) nodes in the second layer that solves the considered COP. As such, this result is a generalization of a similar construction of a 3LP with minimal sized first layer for the sorting problem; see [Zwietering, Aarts & Wessels, 1993]. For our general construction we require a second hidden layer with \(|F|\) nodes. Future research might yield a general approach to reduce the size of the second hidden layer, like we did for the sorting problem in [Zwietering, Aarts & Wessels, 1993]; see also Chapter 7. Furthermore, it might be interesting to consider the capabilities of MLPs for approximately solving COPs, using MLPs that are constructed similarly to the ones presented in this section, starting with neighborhood structures that are not exact; see Aarts, Korst & Zwietering [1992] for some well-known examples of not exact neighborhood structures.

Finally, we have derived two necessary conditions for COP to be solvable by a 2LP, that were based on the bow-tie condition and the twisted bow-tie condition for subsets to be classifiable with a 2LP.
Chapter 7

Case Studies

7.1 Introduction

In this chapter, we apply the theory developed in the previous chapter to the five well-known COPs introduced in Chapter 3; SORTING, MINIMUM COST SPANNING TREE, SHORTEST NETWORK PATH, SHORTEST NETWORK ROUTE, and DISCRETE DYNAMIC LOTSIZING. Note that SHORTEST NETWORK ROUTE is introduced as a variant of MANY-TO-ONE SHORTEST PATH; see Chapter 3. For these five problems we give an explicit characterization of the minimal exact neighborhood, and use it to find the minimal number of first-layer nodes that is required by any MLP that solves the considered problem. Furthermore, for two problems, SORTING and DISCRETE DYNAMIC LOTSIZING, we determine the minimal number of layers that is required by any MLP that solves them.

The problems discussed in this section are selected for the following reasons. The first reason is that their integer versions are all in \( \mathcal{TC} \); see Chapter 4. Now, SORTING is a well-known test problem. Moreover, its analysis covers all aspects of the general theory developed in Chapter 6. Furthermore, the MLP that solves SORTING and that is presented in Section 7.2, has a polynomial number of nodes, and is minimal, both with respect to its number of first-layer nodes and with respect to its number of layers. MINIMUM COST SPANNING TREE is selected because it shows an example of a non-trivial COP that can be solved by an MLP with a minimal number of first-layer nodes that is polynomial in the number of inputs. Thirdly, SHORTEST NETWORK PATH is an example of a problem of which the integer version is in \( \mathcal{TC} \), but that cannot be solved by an MLP with a polynomial number of nodes, whatever the number of layers used. SHORTEST NETWORK ROUTE shows that the result for SHORTEST NETWORK PATH is not due to the fact that its minimal exact neighborhood has exponential size. Finally, DISCRETE DYNAMIC LOTSIZING is selected because it is a non-trivial, practical problem in \( \mathcal{TC} \) that has a non-linear cost function. This complicates the analysis, both with respect to the minimal exact neighborhood, making it necessary to use an extended version of Farkas' lemma, and with respect to the minimal number of layer, making it necessary to use the twisted bow-tie condition.
In what follows, we show that the presented formulations of all five problems satisfy the conditions of Theorem 6.3.3, by showing that the Conditions 1a, 1b, 2a, and 2b are satisfied. This enables the determination of the minimal number of first-layer nodes that is required by an MLP that solves a given COP. Before we examine the five problems in greater detail, we make some general remarks about the minimal exact neighborhoods for these problems. In all five cases, the minimal exact neighborhood $M^S_y$ of a given feasible solution $y \in F$, consists of those solutions that are in some sense the closest to $y$. Usually, a $z \in M^S_y$ can be obtained by applying some kind of 2-change to $y$, which can be described as follows. Take an elementary small part of $y$ and replace it with another elementary part not in $y$, to obtain $z$. Furthermore, if $z \in M^S_y = M^S_z$, then $y$ and $z$ are mutually optimal for $x = (x_y + x_z)/2$, where $x_y$ ($x_z$) is a parameter for which $y$ ($z$) is uniquely optimal. It is usually straightforward to find a parameter vector $x_y$ for which a given solution $y$ is uniquely optimal. In fact, for all five problems except sorting and discrete dynamic lotsizing, a solution $y$ is shown to be uniquely optimal for $x_y = 1 - u(y)$, where 1 is a suitable vector of ones.

7.2 Sorting

In this section we study the MLP-complexity of two variants of the well-known sorting problem. In both variants we search for a non-decreasing ordering of a given array of $n$ real-valued numbers. The difference between the two variants lies in the representation of the solution of the problem, i.e., the representation of the output. In the first version, which we name absolute sorting, and which corresponds to the usual formulation of sorting, the solution is represented as a permutation of the indices of the numbers. The $i$-th value of this permutation corresponds to the index of the number that should be placed at the $i$-th position in order to obtain a sorted list. In the second version, which we name relative sorting, the solution is a pair of an index that indicates the smallest of the $n$ numbers, and a cyclic permutation of the indices, that yields the index of the immediate successor of each number, except the last, in the sorted list.

Below, we present two 0-1 formulated COPs that are shown to be equivalent to the two considered versions of sorting, and that satisfy the general conditions posed in Chapter 6. Hence, we can use the results of Chapter 6 to prove that both variants of sorting can be solved by an MLP. Moreover, we show that the minimal number of first-layer nodes required by any MLP that solves these problems is $\frac{1}{2}n(n - 1)$. Furthermore, we show that the minimal number of layers required for solving absolute sorting and relative sorting is three and two, respectively. Finally, we complete our discussion of the two variants of sorting, by giving the construction of two MLPs that solve them. The first construction concerns a 3LP with $\frac{1}{3}n(n - 1)$ nodes in the first hidden layer, and $n^2$ nodes in the second hidden layer, that solves absolute sorting. The second construction concerns a 2LP with $\frac{1}{2}n(n - 1)$ nodes in the first hidden layer, that solves relative sorting, and is based on the use of the so-called basic functions that are distinguished in Chapter 5.

**Absolute Sorting**
7.2 Sorting

First, we consider ABSOLUTE SORTING, which can be defined as follows.

**Definition 7.2.1. (Absolute Sorting)**

Given an array of \( n \) real numbers, for some \( n \in \mathbb{N} \), the problem is to find the absolute sequence in which the numbers have to be placed in order to obtain a sorted list. Let \( \pi(i) \) denote the index of the number at position \( i \) in the sorted list. Then the problem can be formulated mathematically as follows.

- Given \( n \in \mathbb{N} \) and \( x_1, \ldots, x_n \in \mathbb{R} \).
- Find a mapping \( \pi : \{1, \ldots, n\} \to \{1, \ldots, n\} \) that satisfies the following conditions.
  1. \( \pi \) is a permutation.
  2. \( x_{\pi(i)} \leq x_{\pi(i+1)} \), for all \( i = 1, \ldots, n-1 \).

In order to be able to use the results of Chapter 6, it is convenient to formulate ABSOLUTE SORTING as a COP. We propose the following formulation, which is proved to be equivalent to ABSOLUTE SORTING in Theorem 7.2.1 below.

**Definition 7.2.2. (COP Formulation of Absolute Sorting)**

The COP formulation of ABSOLUTE SORTING is defined as the tuple \((I, F, c)\), given by

\[
I = \mathbb{R}^n,
F = \{ \pi \mid \pi \text{ is a permutation of } \{1, \ldots, n\} \},
\]

\[
c(\pi; x) = \sum_{i=1}^{n} (n+1-i)x_{\pi(i)}.
\]  

(7.1)

where for a given \( x \in I \) the problem is to find a \( \pi \in F \) that minimizes \( c(\pi; x) \).

The term \( n+1 \), occurring in the above formulation is arbitrarily chosen, and may be replaced by any other constant term, without altering any of the forthcoming results.

Next, we prove that the given COP formulation of ABSOLUTE SORTING is equivalent to ABSOLUTE SORTING. This is in fact proved, by showing that ABSOLUTE SORTING is the problem of finding a local minimum with respect to a certain neighborhood, that turns out to be the exact minimal neighborhood of the COP formulated by (7.1). This neighborhood is essentially introduced in Definition 7.2.3 below. The neighbors of a given permutation are the permutations that can be obtained by swapping two subsequent entries. The exactness of this neighborhood is proved in Lemma 7.2.1 and Theorem 7.2.1, where we use it as a proof of the equivalence of ABSOLUTE SORTING and the COP formulation of ABSOLUTE SORTING. The minimality of this neighborhood is shown in Theorem 7.2.2, after we have introduced a 0-1 version of the COP formulation.

**Definition 7.2.3.** For all \( k = 1, \ldots, n-1 \), the permutation \( \pi^{(k)} \) is obtained from the permutation \( \pi \) by swapping the \( k \)th and \((k+1)\)th entry, i.e.,

\[
\pi^{(k)}(i) = \begin{cases} 
\pi(i), & i \neq k, k+1 \\
\pi(2k+1-i), & i = k, k+1
\end{cases}
\]

(7.2)

for all \( i = 1, \ldots, n \).
The permutations \( \pi^{(1)}, \ldots, \pi^{(n-1)} \), form a neighborhood of \( \pi \); see Theorem 7.2.2. In the following lemma we essentially prove that \( c(\pi; x) \) satisfies (6.35), which implies that this neighborhood is exact.

**Lemma 7.2.1.** For all \( \pi, \kappa \in F \) we have

\[
c(\pi; x) - c(\kappa; x) = \sum_{j=1}^{n-1} \lambda_j^{\pi, \kappa}(c(\pi; x) - c(\pi^{(j)}; x)),
\]

(7.3)

for some \( \lambda_j^{\pi, \kappa} \geq 0, j = 1, \ldots, n-1 \), that satisfy \( \lambda_j^{\pi, \kappa} = 0 \), for all \( j = 1, \ldots, n-1 \), if and only if \( \pi = \kappa \).

**Proof.** Let \( \pi, \kappa \in F \), and define the permutation \( \tau = \kappa^{-1} \circ \pi \), then we have

\[
c(\pi; x) - c(\kappa; x) =
\]

\[
= \sum_{i=1}^{n} (n+1-i)\pi(i) - \sum_{i=1}^{n} (n+1-i)\pi(i)
\]

\[
= \sum_{i=1}^{n} (n+1-i)\pi(i) - \sum_{i=1}^{n} (n+1-\tau(i))\pi(i)
\]

\[
= \sum_{i=1}^{n} (\tau(i) - i)\pi(i)
\]

\[
= \sum_{i=1}^{n} (\tau(i) - i) \left[ \sum_{j=1}^{n-1} (\pi(j) - \pi(j+1)) + \pi(n) \right]
\]

\[
= \sum_{j=1}^{n-1} \sum_{i=1}^{n} (\tau(i) - i)(\pi(j) - \pi(j+1)) + \sum_{i=1}^{n} (\tau(i) - i)\pi(n)
\]

\[
= \sum_{j=1}^{n-1} \sum_{i=1}^{n} (\tau(i) - i)(\pi(j) - \pi(j+1)) + \sum_{i=1}^{n} (\tau(i) - i)\pi(n)
\]

\[
= \sum_{j=1}^{n-1} \lambda_j^{\pi, \kappa}(c(\pi; x) - c(\pi^{(j)}; x)),
\]

where \( \lambda_j^{\pi, \kappa} = \sum_{i=1}^{n} (\tau(i) - i) = \sum_{i=1}^{n} (\kappa^{-1}(\pi(i)) - i) \), for all \( j = 1, \ldots, n-1 \). One easily verifies that, firstly, \( \lambda_j^{\pi, \kappa} \geq 0 \), for all \( j = 1, \ldots, n-1 \), and, secondly, \( \lambda_j^{\pi, \kappa} = 0 \), for all \( j = 1, \ldots, n-1 \), if and only if \( \pi = \kappa \).

We prove below that \textsc{absolute sorting} and its COP formulation are equivalent, essentially by showing that the above mentioned neighborhood is exact.

**Theorem 7.2.1.** The Definitions 7.2.1 and 7.2.2 are equivalent, in the sense that for all \( x \in I = \mathbb{R}^n \), any permutation \( \pi \) of \( \{1, \ldots, n\} \) that solves \textsc{absolute sorting}, minimizes \( c(\pi; x) \), and vice versa.

**Proof.** Let \( x \in I = \mathbb{R}^n \). The proof consists of two parts, that represent the two directions of the statement given in the theorem.

1. Assume that \( \pi \in F \) minimizes \( c(\pi; x) \). Then certainly \( x_{\pi(j)} - x_{\pi(j+1)} = c(\pi; x) - c(\pi^{(j)}; x) \leq 0 \), for all \( j = 1, \ldots, n-1 \). Hence, \( \pi \) solves \textsc{absolute sorting}.

2. Assume that \( \pi \in F \) solves \textsc{absolute sorting}. Then \( c(\pi; x) - c(\pi^{(j)}; x) = x_{\pi(j)} - x_{\pi(j+1)} \leq 0 \), for all \( j = 1, \ldots, n-1 \). Hence, using Lemma 7.2.1, we find that \( c(\pi; x) \leq c(\kappa; x) \), for all \( \kappa \in F \), which proves that \( \pi \) minimizes \( c(\pi; x) \).
7.2 Sorting

A 0-1 formulation

In order to be able to consider \( \theta \)-MLPs for solving ABSOLUTE SORTING, we need to have a 0-1 formulation of ABSOLUTE SORTING. This is obtained from the COP formulation by representing a permutation as a 0-1 matrix.

**Definition 7.2.4. (0-1 FORMULATION OF ABSOLUTE SORTING (ABSSORT))**

Given \( x_1, \ldots, x_n \in \mathbb{R} \), for some \( n \in \mathbb{N} \), the problem is to sort these numbers in non-decreasing order. Let \( y_{ij} \in \mathcal{B} \) be a 0-1 variable, where \( y_{ij} = 1 \), if the number \( x_j \) is to be put on the \( i^{th} \) position and \( y_{ij} = 0 \) otherwise. Then the problem can be formalized as a tuple \((I, F, c)\), with

\[
I = \mathbb{R}^n,
\]

\[
F = \{ y \in \mathcal{B}^{n \times n} \mid \sum_{i=1}^n y_{ij} = \sum_{j=1}^n y_{ij} = 1 \},
\]

\[
c(y;x) = \sum_{i=1}^n \sum_{j=1}^n (n+1-i) y_{ij} x_j,
\]

where for a given \( x \in I \) the problem is to find a \( y \in F \) that minimizes \( c(y;x) \).

The translation back and forth to the basic version of ABSOLUTE SORTING follows trivially from Theorem 7.2.1, and can be expressed as follows, where we use \( 1(i) \) to denote the true-false indicator defined by \( 1(true) = 1 \) and \( 1(false) = 0 \).

**Proposition 7.2.1.** The Definitions 7.2.1 and 7.2.4 are equivalent, in the sense that for all \( x \in I = \mathbb{R}^n \)

- if \( \pi \) is a permutation that solves ABSOLUTE SORTING, then \( y \in \mathcal{B}^{n \times n} \) defined by \( y_{ij} = 1(i \preceq j), \) for all \( i,j = 1, \ldots, n \), minimizes \( c(y;x) \), and

- if \( y \in \mathcal{B}^{n \times n} \) minimizes \( c(y;x) \), then the permutation \( \pi \) defined by \( \pi(i) = \sum_{j=1}^n j y_{ij} \), for all \( i = 1, \ldots, n \), solves ABSOLUTE SORTING.

Minimal Exact Neighborhood

Before we determine the minimal exact neighborhood for ABSOLUTE SORTING, we show that the above presented 0-1 formulation satisfies the Conditions 1a, 1b, 2a, and 2b of Chapter 6. One easily shows that the above formulation of ABSOLUTE SORTING satisfies the Conditions 1a, 1b, and 2a, with \( N = n, K = n^2, a_j(y) = \sum_{i=1}^n (n+1-i) y_{ij} \), for all \( y \in F \), and \( b(y) = 0 \), for all \( y \in F \). That the 0-1 formulation of ABSOLUTE SORTING satisfies Condition 2b can be shown as follows. Let \( y \in F \) and define the permutation \( \pi \) by \( \pi(i) = \sum_{j=1}^n j y_{ij} \), for all \( i = 1, \ldots, n \). Let \( x \in I^x \) be defined by \( x_i = \pi^{-1}(i) = \sum_{k=0}^{n-1} k y_{ki} \), then obviously \( x_{\pi(i)} = i \preceq i+1 = x_{\pi(i+1)} \), for all \( i = 1, \ldots, n \), which implies that \( \pi \) solves ABSOLUTE SORTING for this \( x \). Hence, using Proposition 7.2.1, we find that \( y \) minimizes \( c(y;x) \). Furthermore, using Lemma 7.2.1, one can easily show that \( c(y;x) < c(z;x) \), for all \( z \in F \setminus \{y\} \), which proves that Condition 2b is satisfied.

Since we have Lemma 7.2.1, we can use Approach 1b for proving that a certain neighborhood is the minimal exact neighborhood for ABSOLUTE SORTING. For notational convenience the arguments are based on the use of permutations. However, it is straightforward to translate the arguments in the 0-1 notation.
Theorem 7.2.2. The minimal exact neighborhood for absolute sorting is given by

\[ \mathcal{M}_n^{\text{ABSORT}} = \{ \pi(k) | k = 1, \ldots, n - 1 \}, \]

for all \( \pi \in F \).

Proof. That \( \mathcal{M}_n^{\text{ABSORT}} \) is exact follows directly from Lemma 7.2.1; see also Theorem 7.2.1 and Lemma 6.3.4.

Next, we show that \( \mathcal{M}_n^{\text{ABSORT}} \subseteq \mathcal{M}_n^{\text{IF}} \), for all \( \pi \in F \). Let \( \pi \in F \) and \( \kappa \in \mathcal{M}_n^{\text{ABSORT}} \). Then \( \kappa = \pi(k) \), for some \( k \in \{1, \ldots, n - 1\} \). Define \( x \in \mathcal{I}^n \) by

\[ x_{\pi(i)} = \begin{cases} i, & i \neq k, k + 1 \\ k + \frac{1}{2}, & i = k, k + 1. \end{cases} \]

Then we have

\[ c(\pi; x) - c(\pi^{(i)}; x) = x_{\pi(i)} - x_{\pi(i+1)} = \begin{cases} -1, & j \neq k - 1, k, k + 1 \\ -\frac{1}{2}, & j = k - 1, k + 1 \\ 0, & j = k. \end{cases} \] \hspace{1cm} (7.5)

for all \( j = 1, \ldots, n - 1 \), which yields \( c(\kappa; x) = c(\pi; x) \). Let \( \kappa' \in F \setminus \{ \pi, \kappa \} \) and \( r = \kappa'^{-1} \circ \pi \), then it follows that \( \lambda_0^{\kappa'} = \sum_{i=1}^{n}(\pi(i) - 1) \geq 1 \), for some \( j \neq k \). Hence, combining (7.3) and (7.5), we obtain that \( c(\pi; x) - c(\kappa'; x) \leq -1 < 0 \), which proves that \( \kappa \in \mathcal{M}_n^{\text{IF}} \).

\[ \square \]

The size of the minimal exact neighborhood of a given solution \( y \in F \), follows directly from the definition of this neighborhood as given in Theorem 7.2.2.

Proposition 7.2.2. For all \( \pi \in F \), we have \( |\mathcal{M}_n^{\text{ABSORT}}| = n - 1 \).

Minimal size first layer of MLP

Finally, we can compute the minimal number of first-layer nodes that any MLP requires for solving the sorting problem. We start with the observation that for all \( \pi \in F \) and \( \kappa \in \mathcal{M}_n^{\text{ABSORT}} \), we have \( c(\pi; x) - c(\kappa; x) = x_i - x_k \), for some \( i, k \in \{1, \ldots, n\} \) with \( i \neq k \). Furthermore, all of these differences are non-equivalent and occur. Hence, \( L(I, F; c; \mathcal{M}_n^{\text{ABSORT}}) \) is equal to the number of ways one can select two different indices from \( \{1, \ldots, n\} \), which yields the following result.

Proposition 7.2.3. The minimal number of first-layer nodes of an MLP that solves absolute sorting is

\[ L_{\text{ABSORT}} = \binom{n}{2} = \frac{1}{2} n(n - 1). \]

Minimal number of layers of MLP

One can easily verify that absolute sorting can be solved by a 2LP, in case that \( n \leq 2 \). Using the bow-tie condition for COPs derived in Chapter 6, we prove that absolute sorting cannot be solved by a 2LP, in case that \( n \geq 3 \). First, we consider the case that \( n = 3 \).

Let \( x^{(1)} = (1, 3, 4) \), \( x^{(2)} = (7, 5, 4) \), \( x^{(3)} = (3, 1, 4) \), \( x^{(4)} = (5, 7, 4) \), and let \( y^{(j)} \in \mathbb{B}^{2 \times 3} \), \( j = 1, 2, 3, 4 \), such that \( c(y^{(j)}; x^{(j)}) < c(z; x^{(j)}) \), for all \( z \in F \setminus \{y^{(j)}\} \), and \( j = 1, 2, 3, 4 \). Then one easily verifies that the following conditions are satisfied.

- \( y^{(1)}_{22} = y^{(2)}_{22} = 1 \) and \( y^{(3)}_{22} = y^{(4)}_{22} = 0 \).
7.2 Sorting

- \( x_{13} = (x^{(1)} + x^{(3)})/2 = (2, 2, 4) \in I^3 \) and \( x_{24} = (x^{(2)} + x^{(4)})/2 = (6, 6, 4) \in I^3 \).
- \( c(y^{(1)}; x_{13}) = c(y^{(2)}; x_{13}) < c(z; x_{13}) \), for all \( z \in F' \setminus \{y^{(1)}, y^{(2)}\} \), and
  \( c(y^{(3)}; x_{24}) = c(y^{(4)}; x_{24}) < c(z; x_{24}) \), for all \( z \in F' \setminus \{y^{(3)}, y^{(4)}\} \).
- \( \text{sign}(c(y^{(1)}; x) - c(y^{(3)}; x)) = \text{sign}(x_1 - x_3) = \text{sign}(c(y^{(2)}; x) - c(y^{(4)}; x)) \), for all \( x \in I \).

Hence, using Theorem 6.4.1, it follows that **absolute sorting** cannot be solved by a 2LP, in case that \( n = 3 \). Note that the above values of \( x^{(i)} \) and \( y^{(j)} \), \( j = 1, 2, 3, 4 \), also satisfy the conditions of Theorem 6.4.2.

---

**Figure 7.1:** The set presented in (a) is \( \{ (x_1, x_2) \in \mathbb{R}^2 \mid x_1 \leq x_2 \leq x_3 < x_3 < x_1 \} \),
for a given value of \( x_3 \in \mathbb{R} \). The points (1), (2), (3), and (4), correspond to the points
\( x^{(1)}, x^{(3)}, x^{(3)}, \) and \( x^{(4)}, \) that are introduced in the text, in case that \( x_3 = 4 \). The set
presented in (b) is \( \{ (x_1, x_2) \in \mathbb{R}^2 \mid x_1 \leq x_2 \leq x_3 \vee x_3 < x_1 \leq x_2 \vee x_2 \leq x_3 < x_1 \} \),
for a given value of \( x_3 \in \mathbb{R} \).

The idea behind the above argument can be illustrated using Figure 7.1a, in which
we have shown a two-dimensional cut of the subset \( V \subseteq \mathbb{R}^3 \), defined by
\[
V = \{ x \in \mathbb{R}^3 \mid x_1 \leq x_2 \leq x_3 \vee x_3 < x_2 < x_1 \}. \tag{7.6}
\]
This set corresponds to all values \( x_1, x_2, x_3 \) for which \( y_{22} = 1 \). Since this set does
not satisfy the bow-tie condition, it cannot be classified with a 2LP, and, hence, **absolute sorting**
cannot be solved by a 2LP, in case that \( n = 3 \).

That **absolute sorting** cannot be solved by a 2LP in case that \( n > 3 \), follows directly from the fact that it cannot be solved by a 2LP in case that \( n = 3 \), and
the fact that if \( f : \mathbb{R}^n \rightarrow \mathbb{B}^{\leq n} \) represents a 2LP that solves **absolute sorting** in
case \( n > 3 \), then \( f : \mathbb{R}^3 \rightarrow \mathbb{B}^3 \), defined by \( \hat{f}(x_1, x_2, x_3) = f_i(x_1, x_2, x_3, 0, 0, \ldots, 0), \)
\( i = 1, 2, 3 \), represents a 2LP that solves **absolute sorting** in case that \( n = 3 \).

An MLP with minimal sized first layer and minimal number of layers.
In a previous paper we have shown that there exists a 3LP with minimal sized first
layer that solves absolute sorting; see Zwietering, Aarts & Wessels [1992a]. This result can be stated as follows.

Proposition 7.2.4 (Zwietering, Aarts & Wessels [1992a]).
Let \( f : \mathbb{R}^n \to \mathbb{R}^{n \times n} \) be the function defined by \( f = g^{(3)} o g^{(2)} o g^{(1)} \), with \( g^{(i)} \), \( i = 1, 2, 3 \), given by
\[
\forall i, j = 1, ..., n, \quad g^{(1)}_{ij}(x) = \theta[x_j - x_i],
\]
\[
\forall i, j = 1, ..., n, \quad g^{(2)}_{ij}(w) = \theta[n + 1 - i - j + \sum_{k=1}^{n} w_{kj} - \sum_{k=j+1}^{n} w_{ij}],
\]
\[
\forall i, j = 1, ..., n, \quad g^{(3)}_{ij}(s) = \theta[s_{ij} - s_{i+1,j} - 1],
\]
respectively, where one should note that \( s_{n+1,i} \equiv 0 \), for all \( j = 1, ..., n \). Then \( f \) represents a 3LP, with \( n \) inputs, \( \frac{1}{2} n(n-1) \) nodes in the first hidden layer, \( n^2 \) nodes in the second hidden layer, and \( n^2 \) outputs, that solves \( (I, F, c) \).

Relative Sorting

In our search for a formulation of sorting that could be solved with a 2LP, we came across relative sorting. This is the second considered variant of sorting, which can be defined as follows.

Definition 7.2.5. (Relative Sorting)
Given an array of \( n \) real numbers, for some \( n \in \mathbb{N} \), the problem is to find the relative sequence in which the numbers have to be placed in order to obtain a sorted list, i.e., give the index of the smallest number and for each number, except the largest, give the index of the number that follows this number in the sorted list. Let \( s \) denote the index of the smallest number and \( \alpha(1) \) denote the index of the number that is the successor of number \( 1 \) in the sorted list. Then the problem can be formulated mathematically as follows.

- Given \( n \in \mathbb{N} \) and \( x_1, ..., x_n \in \mathbb{R} \).
- Find \( s \in \{1, ..., n\} \) such that \( x_s \leq x_i \), for all \( i = 1, ..., n \), and a mapping \( \alpha : \{1, ..., n\} \to \{1, ..., n\} \) that satisfies the following conditions.
  (i) \( \alpha \) is a cyclic permutation.
  (ii) For all \( i = 1, ..., n \), either \( x_i \leq x_{\alpha(i)} \), or \( x_j \leq x_i \), for all \( j = 1, ..., n \).

Note that a permutation \( \alpha \) is cyclic, if \( \{\alpha^0(s), \alpha^1(s), ..., \alpha^{n-1}(s)\} = \{1, ..., n\} \).

To be able to use the results obtained for absolute sorting, we present the following relation between relative sorting and absolute sorting.

Proposition 7.2.5. Relative Sorting and Absolute Sorting are related as follows. For all \( n \in \mathbb{N} \) and \( x_1, ..., x_n \in \mathbb{R} \), we have

- If \((s, \alpha)\) is a solution of relative sorting, then the permutation \( \pi \) defined by \( \pi(i) = \alpha^{-1}(s) \), for all \( i = 1, ..., n \), solves absolute sorting, and
- If \( \pi \) is a solution of absolute sorting, then \((s, \alpha)\) defined by \( s = \pi(1) \), \( \alpha(\pi(i)) = \pi(i + 1) \), for all \( i = 1, ..., n - 1 \), and \( \alpha(\pi(n)) = \pi(1) \), solves relative sorting.
7.2 Sorting

A 0-1 formulation
Using Proposition 7.2.5 and the 0-1 formulation of ABSOLUTE SORTING, we come to the following 0-1 formulation of RELATIVE SORTING.

Definition 7.2.6. (0-1 FORMULATION OF RELATIVE SORTING (RELSORT))
Given \(x_1, \ldots, x_n \in \mathbb{R}\), for some \(n \in \mathbb{N}\), the problem is to sort these numbers in non-decreasing order. Let \(v \in \mathbb{B}^n\) be a 0-1 vector and \(w \in \mathbb{B}^{nxn}\) be a 0-1 matrix, where \(v_i = 1\) if the number \(x_i\) is to be put on the first position, and \(w_{ij} = 1\), if \(x_j\) is the immediate successor of \(x_i\) in the sorted list. Then the problem can be formalized as a tuple \((I, F, c)\), with

\[
I = \mathbb{R}^n,
\]

\[
F = \{(v, w) \in \mathbb{B}^n \times \mathbb{B}^{nxn} | \sum_{i=1}^{n} v_i = \sum_{i=1}^{n} w_{ij} = 1, \forall i = 1, \ldots, n, 1 \leq j \leq n : (w^i)_j \neq 1\},
\]

\[
c(v, w; x) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} (n+1-i)w_{ijk}x_j,
\]

where for a given \(x \in I\) the problem is to find a pair \((v, w) \in F\) that minimizes \(c(v, w; x)\).

Using Proposition 7.2.5 and Proposition 7.2.1, one can straightforwardly prove the following result.

Proposition 7.2.6. The Definitions 7.2.5 and 7.2.6 are equivalent, in the sense that for all \(x \in I = \mathbb{R}^n\)

- if \((s, \alpha)\) solves RELATIVE SORTING, then \((v, w) \in \mathbb{B}^n \times \mathbb{B}^{nxn}\) defined by \(v_i = 1_{s(i) = 1}, w_{ij} = 1_{s(i) = j}, \forall i, j = 1, \ldots, n,\) minimizes \(c(v, w; x)\), and

- if \((v, w) \in \mathbb{B}^n \times \mathbb{B}^{nxn}\) minimizes \(c(v, w; x)\), then the pair \((s, \alpha)\) defined by \(s = \sum_{i=1}^{n} v_i, \alpha = \sum_{i=1}^{n} jw_{ij}\), for all \(i = 1, \ldots, n,\) solves RELATIVE SORTING.

Next, we discuss the MLP-complexity of RELATIVE SORTING. Similarly as for ABSOLUTE SORTING, one can show that the 0-1 formulation of RELATIVE SORTING satisfies the Conditions 1a, 1b, 2a, and 2b. Hence, RELATIVE SORTING can be solved with an MLP. Furthermore, one can prove that the minimal exact neighborhood for RELATIVE SORTING is a subproblem of ABSOLUTE SORTING, given by Theorem 7.2.2. This yields that the minimal number of first-layer nodes of any MLP that solves RELATIVE SORTING is \(\binom{n}{2}\). However, this can also be demonstrated using a somewhat more sophisticated argument.

Assume that RELATIVE SORTING is solved by an MLP, for some \(m \in \mathbb{N}\), with \(L\) nodes in the first layer. Then one straightforwardly constructs an \((m + 2)\)LP with \(L\) nodes in the first hidden layer that solves ABSOLUTE SORTING. Hence, we have

\[
L_{\text{absort}} \geq L_{\text{realsort}} \quad (7.8)
\]

Similarly, if ABSOLUTE SORTING is solved by an MLP, for some \(m \in \mathbb{N}\), with \(L\) nodes in the first layer, then one straightforwardly constructs an \((m + 2)\)LP with \(L\) nodes in the first hidden layer that solves RELATIVE SORTING. This yields

\[
L_{\text{realsort}} \geq L_{\text{absort}} \quad (7.9)
\]
Combining (7.8) and (7.9) we find that $L_{\text{relsort}} = L_{\text{amsort}} = \binom{n}{2}$, as we already noted.

An MLP with minimal sized first layer and minimal number of layers. In the following theorem we prove that RELATIVE SORTING can be solved with a 2LP that has minimal first layer.

**Theorem 7.2.3.** Let $h : \mathbb{R}^n \to \{-1,+1\}^{n \times n}$ be defined by

$$h_{ij}(x) = \begin{cases} 2\theta[x_j - x_i] - 1, & \text{if } i \leq j, \\ 1 - 2\theta[x_i - x_j], & \text{if } i > j, \end{cases} \quad (7.10)$$

let $f : \mathbb{R}^n \to \mathbb{R}^n$ be defined by

$$f_i(x) = \theta[\sum_{k=1}^{n} h_{ik}(x) - 1]. \quad (7.11)$$

and let $g : \mathbb{R}^n \to \mathbb{R}^n$ be defined by $g_{ij}(x) = 1$, in case that $n = 1$, $g_{ij}(x) = 0$, for all $i = 1, \ldots, n$, in case that $n > 1$, and

$$g_i(x) = \theta[\sum_{k=1}^{n} (h_{ki}(x) + h_{ij}(x) + h_{jk}(x) - 1)]. \quad (7.12)$$

Then $(f, g)$ represents a 2LP with $n$ inputs, $\frac{1}{2}n(n-1)$ nodes in the hidden layer, and $n(n+1)$ outputs, that solves $(I, F, c)$ given by (7.7).

**Proof.** Inserting (7.10) in (7.11) and (7.12), yields two 2LPs represented by $f$ and $g$ respectively. Combining these two 2LPs into one 2LP represented by $(f, g)$, in which double appearances of identical hidden nodes have been removed, one obtains a 2LP with $n$ inputs and $n(n+1)$ outputs, that uses the $\frac{1}{2}n(n-1)$ hidden nodes given by $\theta[x_j - x_i]$, for all $i, j = 1, \ldots, n$ with $i < j$.

In order to prove that the constructed 2LP solves $(I, F, c)$, we introduce a variant of RELATIVE SORTING that has a unique solution. To this end we define the ordering $\preceq$ on a given set of $n$ real numbers $x_1, \ldots, x_n \in \mathbb{R}$, by

$$x_i \preceq x_j \equiv (x_i < x_j) \lor (x_i = x_j \land i \leq j). \quad (7.13)$$

for all $i, j = 1, \ldots, n$. Note that $h_{ij}(x) = +1$, if and only if $x_i \preceq x_j$.

Consider the following problem.

- Given $x_1, \ldots, x_n \in \mathbb{R}$.
- Find $s \in \{1, \ldots, n\}$ such that $x_s \preceq x_i$, for all $i = 1, \ldots, n$, and a mapping $\alpha : \{1, \ldots, n\} \to \{1, \ldots, n\}$ that satisfies the conditions presented below, where $l \in \{1, \ldots, n\}$ is such that $x_l \preceq x_i$, for all $i = 1, \ldots, n$.

(i) $\alpha(i) \neq i$, for all $i \neq l$.

(ii) For all $i = 1, \ldots, n$, if $x_i \preceq x_k$, for some $k \neq i$, then $x_i \preceq x_{\alpha(i)} \preceq x_k$.

(iii) $\alpha(l) = s$.

In a previous paper, we have shown that every solution of this problem is a solution of RELATIVE SORTING; see Zwietering, Aarts & Wessels [1992a]. Below we prove that for all $n \in \mathbb{N}$ and $x_1, \ldots, x_n \in \mathbb{R}$, $s \in \mathbb{N}$ and $\alpha : \{1, \ldots, n\} \to \mathbb{N}$ defined by $s = \sum_{i=1}^{n} t_i(x)$ and $\alpha(i) = \sum_{j=1}^{i} j g_i(x)$, $i = 1, \ldots, n$, solve the above problem.
7.2 Sorting

Consequently, these $s$ and $\alpha$ solve RELATIVE SORTING, which completes the proof because of Proposition 7.2.6.

The $'a'$-part. Using (7.11), (7.10), and $h_{ik}(x) \in \{-1,+1\}$, for all $i, k = 1, \ldots, n$, we obtain

$$f_i(x) = 1 \iff \forall k = 1, \ldots, n : h_{ik}(x) \geq 1$$
$$\iff \forall k = 1, \ldots, n : x_i \leq x_k$$  \hspace{1cm} (7.14)

for all $i = 1, \ldots, n$. It follows from $h_{ik}(x) \in \{-1,+1\}$, for all $i, k = 1, \ldots, n$, $h_{ik}(x) + h_{id}(x) = 0$, for all $i \neq k$, (7.14), and (7.15), that $\sum_{i=1}^n f_i(x) = 1$. Consequently, $s = \sum_{i=1}^n jf_i(x) \in \{1, \ldots, n\}$. Furthermore, using (7.15), we find that $x_j \leq x_k$, for all $k = 1, \ldots, n$.

The $'a'$-part. The case that $n = 1$ is trivial, and we therefore assume that $n > 1$. Using (7.12), (7.10), and $h_{ik}(x) + h_{ij}(x) + h_{jk}(x) \in \{-1,+1\}$, for all $k = 1, \ldots, n$ and all $i \neq j$, we obtain

$$g_{ij}(x) = 1 \iff \forall k = 1, \ldots, n : h_{ik}(x) + h_{ij}(x) + h_{jk}(x) \geq 1$$
$$\iff \forall k = 1, \ldots, n : (x_k \leq x_i \leq x_j) \lor (x_i \leq x_j \leq x_k)$$
\hspace{1cm} \lor (x_j \leq x_k \leq x_i)$$  \hspace{1cm} (7.16)

for all $i, j = 1, \ldots, n$ with $i \neq j$. It follows from $h_{ik}(x) + h_{ij}(x) + h_{jk}(x) \in \{-1,+1\}$, for all $i \neq j$, $[h_{ik}(x) + h_{ij}(x) + h_{jk}(x)] + [h_{ij}(x) + h_{ik}(x) + h_{jk}(x)] = 0$, for all $i \neq k$, $k \neq j$, and $i \neq j$, (7.16), and (7.17), that $\sum_{i=1}^n g_{ij}(x) = 1$, for all $i = 1, \ldots, n$. Consequently, $\alpha(i) = \sum_{j=1}^n jg_{ij}(x) \in \{1, \ldots, n\}$, for all $i = 1, \ldots, n$.

Now, ($i$) follows directly from $g_{ij}(x) = 0$, for all $i = 1, \ldots, n$. Next, we prove ($ii$). Let $i \in \{1, \ldots, n\}$, $j = \alpha(i)$, and assume that $x_i \leq x_k$, for some $i \neq k$. Then $\sim(x_k \leq x_i)$ and, hence, using $j \neq i$, $g_{ij}(x) = 1$, and (7.17), we obtain that $x_i \leq x_j \leq x_k$, which proves ($ii$). Finally, since $x_i \leq x_j \leq x_k$, for all $i = 1, \ldots, n$, and $l \neq s$ because $n > 1$, we find using (7.17) that $g_{kl}(x) = 1$, which implies ($iii$).

This completes the proof of the theorem.

Note that both the functions $\theta[h_{ij}(x)]$, for $x \in \mathbb{R}^n$ and $i, j = 1, \ldots, n$ with $i \neq j$, and the functions $\theta[h_{ik}(x) + h_{ij}(x) + h_{jk}(x)]$, for $x \in \mathbb{R}^n$ and $i, j, k = 1, \ldots, n$ with $i, j, k$ all different, correspond to $n$-dimensional basic functions; see Section 5.1.3. The first functions are essentially two-dimensional, as follows from (7.10). The second functions are essentially three-dimensional, and it is a nice exercise to verify Theorem 5.5.16 for these functions; see also Section 5.5.4. A two-dimensional cut of the basic subset corresponding to the basic function $\theta[h_{12}(x) + h_{23}(x) + h_{31}(x)]$ is shown in Figure 7.1b, where one should note that, similar as in (7.17), $\theta[h_{12}(x) + h_{23}(x) + h_{31}(x)] = 1$, if and only if $x_1 \leq x_2 \leq x_3 \lor x_3 \leq x_2 \leq x_1 \lor x_2 \leq x_3 \leq x_1$.

The above implies that $f_i$ and $g_{ij}$ are members of $R^n_2$, i.e., they classify subsets that are an intersection of a number of basic subsets; see Section 5.5.3.

Generalization

In our discussion of ABSOLUTE SORTING and RELATIVE SORTING we assumed that $I = \mathbb{R}^n$, i.e., we assumed that the numbers to be sorted were arbitrary real-valued numbers. However, one can verify that the above results remain valid in case that one considers an $I$ that satisfies $\overline{I} \subseteq \mathbb{R}^n$, for some $\bar{x}$ with $\bar{x}_1 = \bar{x}_2 = \cdots = \bar{x}_n$. Note
that this $\hat{z}$ plays the opposite role compared to the role of the $\hat{z}$ distinguished in Lemma 6.3.2.

### 7.3 Minimum cost spanning tree

The second problem discussed is **minimum cost spanning tree**. It is shown in this section that the well-known 2-change neighborhood for spanning trees, is in fact the minimal exact neighborhood for this problem. This result is already reported by Savage [1973], and he generalized this result to the class of matroid problems. We present an original proof in order to demonstrate the use of Approach IIa, and to be able to show the analogy with **shortest network route** that is treated in Section 7.5. In the following 0-1 formulation of minimum cost spanning tree, we assume that the costs can be both negative and positive. However, if the costs of the edges are restricted to be positive, the obtained results remain the same.

#### A 0-1 formulation

Below we give a 0-1 formulation of **minimum cost spanning tree**.

**Definition 7.3.1. (minimum cost spanning tree (MCST))**

Given a completely connected undirected graph $G = (V, E)$ on $V = \{1, \ldots , n\}$ with a cost $x_e$ associated to edge $e$, for all $e \in E$; note that $|E| = \frac{1}{2}n(n - 1)$. The problem is to find a spanning tree of $G$ with minimal total cost. Let $y_e \in \mathbb{B}$ be a 0-1 variable such that $y_e = 1$ if the edge $e$ is part of the spanning tree and $y_e = 0$ otherwise. Then the problem can be represented as the tuple $(I, F, c)$ given by

$I = \mathbb{R}^{1 \times \frac{1}{2}n(n - 1)}$,

$F = \{ y \in \mathbb{B}^{1 \times \frac{1}{2}n(n - 1)} \mid y \text{ represents a spanning tree on } \{1, \ldots , n\}\}$,

$c(y, x) = \sum_{e \in E} y_ex_e$.

**Minimal Exact Neighborhood**

The above formulation satisfies the Conditions 1a, 1b, and 2a, with $N = K = \frac{1}{2}n(n - 1)$, $a_e(y) = y_e$, and $b(y) = 0$. Furthermore, it is easily verified that $c(y; 1 - y) < c(z; 1 - y)$, for all $z \in F \setminus \{y\}$, which proves that Condition 2b is satisfied also. Hence, it follows by Corollary 6.3.1 and Corollary 6.3.2, that $\mathcal{M}^S = \mathcal{M}^F$ is the minimal exact neighborhood for **minimum cost spanning tree**. The following theorem presents an explicit characterization of $\mathcal{M}^F$, following the lines of Approach IIa.

**Theorem 7.3.1.** The minimal exact neighborhood for **minimum cost spanning tree** is given by

$\mathcal{M}^{\text{MCST}}_y = \{ z \in F \setminus \{y\} \mid z \text{ can be obtained from } y \text{ by deleting an edge from the cycle that is obtained by adding an edge to } y \}$

for all $y \in F$.

**Proof.** First, we show that $\mathcal{M}^S_y \subseteq \mathcal{M}^{\text{MCST}}_y$, for all $y \in F$, using Approach IIa. Take $y \in F$, $z \notin \mathcal{M}^{\text{MCST}}_y$, $x \in I$ and assume $c(z; x) < c(y; x)$. Let $e$ be the smallest edge, with respect its length $x_e$, that is contained in the spanning tree represented by $y$ and not in $z$. Let $f$ be any edge of the cycle formed by adding $e$ to $z$ and which does not belong to $y$. We complete the argument by considering the following two
cases.

\[ x_j \geq x_f. \] Let \( z' = x + e - f \), then \( z' \neq z \) and \( c(z'; z) \leq c(z; z) < c(y; z) \).

\[ x_j < x_f. \] Let \( e \) be any edge of the cycle that is obtained by adding \( f \) to \( y \) and which does not belong to \( z \). Then, since \( e \) is the smallest edge contained in \( y \setminus z \), we have \( x_e \leq x_j \) and, hence, \( x_f < x_j \). Let \( z' = y + f - g \), then \( z' \in M^{\text{KST}}_y \subseteq P \setminus \{z\} \) and \( c(z'; z) < c(y; z) \).

Next, we show that \( M^{\text{KST}}_y \subseteq M^{\text{LP}}_y \), for all \( y \in F \). Let \( y \in F \) and \( z \in M^{\text{KST}}_y \). Define \( x \in P \) as follows. All edges that are part of \( y \) and part of \( z \) obtain a cost 1, the two edges that are part of \( x \) and not of \( z \) or vice versa, obtain a cost 2, and all other edges obtain a cost 3. Then \( c(y; z) = c(x; z) = n \). Furthermore, if a spanning tree \( z' \neq z \) has one edge not in common with \( y \), then either \( c(z'; z) = (n - 3) + 2 + 2 = n + 1 \) or \( c(z'; x) = (n - 2) + 3 = n + 1 \), and if it has two or more edges not in common with \( y \), then \( c(z'; x) = (n - 3) + 2 + 3 = n + 2 \). This implies that \( z \in M^{\text{LP}}_y \), which completes the proof of the theorem.

It is well-known and easily verified that \( |M^{\text{KST}}_y| = O(n^3) \), for all \( y \in F \). A more precise count gives the following result.

**Proposition 7.3.1.** For all \( y \in F \), we have

\[
(n - 1)(n - 2) \leq |M^{\text{KST}}_y| \leq \frac{1}{6}(n - 1)(n - 2)(n + 3).
\]

The lower bound in Proposition 7.3.1 is obtained for spanning trees in which one vertex has degree \((n - 1)\) and all other vertices have degree 1, corresponding to tree-shaped spanning trees, the upper bound in Proposition 7.3.1 is obtained for spanning trees in which two vertices have degree 1 and all other vertices have degree 2, corresponding to path-like spanning trees.

**Minimal size first layer of MLP**

Similar as was done for **ABSOLUTE SORTING**, we can compute the minimal number of first-layer nodes required by any MLP for solving **MINIMUM COST SPANNING TREE**. If \( z \in M^{\text{KST}}_y \), then \( c(y; z) = c(x; z) = z_e - x_f \), for some edges \( e \neq f \). This implies that \( L^{\text{KST}} \) is equal to the number of ways we can select two different edges from \( E \), which yields the following result.

**Proposition 7.3.2.** The minimal number of first-layer nodes of an MLP that solves **MINIMUM COST SPANNING TREE** is

\[
L^{\text{KST}} = \binom{|E|}{2} = \frac{1}{8}(n - 2)(n - 1)n(n + 1).
\]

From Theorem 6.3.6 in Chapter 6, we know that there exists a 3LP with \( L^{\text{KST}} \) nodes in the first hidden layer that solves **MINIMUM COST SPANNING TREE**, but this 3LP has an exponential number of nodes in the second hidden layer. We do not know whether there exists an MLP, for some fixed \( m \in \mathbb{N} \), with \( L^{\text{KST}} \) nodes in the first layer and a polynomial number of nodes in the other layers, that solves **MINIMUM COST SPANNING TREE**. However, one can show that there exists an MLP, with \( m = O(\log n) \), \( L^{\text{KST}} \) nodes in the first layer, and a polynomial number of nodes in the other layers, that solves **MINIMUM COST SPANNING TREE**. This can be done
by combining a 3LP, with a polynomial number of nodes, for sorting the weights, which is presented in Section 7.2, with the threshold circuit of depth $O(\log n)$ that solves \textsc{Integer Minimum Cost Spanning Tree}, which is discussed in Chapter 4.

\textbf{Minimal number of layers of MLP}

We did not determine the minimal number of layers required for solving \textsc{Minimum Cost Spanning Tree}.

### 7.4 Shortest network path

The next problem we consider is the shortest path problem. In general this is the problem of finding the shortest path between two given nodes in a graph, where all edges have a certain non-negative length. For the sake of notational convenience we restrict ourselves to network graphs. Then the problem can be presented with the following 0-1 formulation.

![Figure 7.2: A network graph with $n_2$ levels of $n_1$ nodes.](image)

#### A 0-1 formulation

\textbf{Definition 7.4.1. (Shortest Network Path (SNP))}

Given a directed network-graph with a source node $s$, a terminal node $t$, and $n_2$ levels of $n_1$ nodes, see Figure 3.1. There is an arc from $s$ to each node of the first level, there is an arc from each node of the last level to $t$, and there are arcs between the nodes of subsequent levels. With each arc a non-negative length is associated. The problem is to find a path through the network from $s$ to $t$ that has minimal total length.

To come to a 0-1 formulation we assume without loss of generality that the lengths of the arcs incident with the nodes $s$ and $t$ have a length zero. Let $y_k \in \{0, 1\}$ be a 0-1 variable, such that $y_k = 1$, if the path passes through node $i$ of level $k$, and $y_k = 0$ otherwise. Finally, we let $x_{ij} \geq 0$ denote the length of the arc between node $i$ of level $k$ and node $j$ of level $k + 1$. Then the problem can be represented by the
7.4 Shortest network path

tuple \( (I, F, c) \) given by

\[
I = \mathbb{R}_{+}^{n_{1} \times n_{2} \times n_{3}(n_{2} - 1)},
\]

\[
F = \{ y \in \mathbb{R}^{n_{1} \times n_{2}} : \sum_{i=1}^{n_{1}} y_{i1} = 1 \},
\]

\[
c(y; x) = \sum_{i=1}^{n_{1}} \sum_{j=1}^{n_{2}} \sum_{k=1}^{n_{3} - 1} y_{ik} y_{jk}(x_{k+1}) z_{i,j,k}.
\]

Again, one can easily verify that the above above formulation satisfies the Conditions 1a, 1b, and 2a, with \( N = n_{1} n_{2}(n_{2} - 1) \), \( K = n_{1} n_{2} \), \( a_{ij}(y) = y_{ik} y_{jk}(x_{k+1}) \), and \( b(y) = 0 \). Furthermore, let \( z_{i,j,k} = 2 - y_{ik} y_{jk}(x_{k+1}) \), for all \( i, j, k \), then \( z_{i,j,k} \) is \( F \), which proves that Condition 2b is satisfied also.

We use Approach 1b to determine the minimal exact neighborhood for \textsc{shortest network path}, as presented in Theorem 7.4.1 below.

Minimal Exact Neighborhood

We start by defining a distance between every pair of solutions \( y, z \in F \), denoting the number of connected parts of the path represented by \( y \), that differs from the corresponding part of the path represented by \( z \).

Definition 7.4.2. For all \( y, z \in F \), we define \( d = d(y, z) \in \mathbb{N} \) and \( k_{1}, \ldots, k_{2d+2} \in \mathbb{N} \), as the unique set of numbers that satisfy the following three conditions.

\( (i) \) \( 1 = k_{1} < k_{2} < \cdots < k_{2d+1} \leq k_{2d+2} = n_{2} + 1 \).

\( (ii) \) \( \forall i = 1, \ldots, n_{1} \forall r = 1, \ldots, d + 1 \forall k = k_{2r-1}, \ldots, k_{2r} - 1 : y_{ik} = z_{ik} \).

\( (iii) \) \( \forall i = 1, \ldots, n_{1} \forall r = 1, \ldots, d \forall k = k_{2r}, \ldots, k_{2r+1} - 1 : y_{ik} \neq z_{ik} \).

Note that \( d(y, z) = 0 \), if and only if \( y = z \). The following theorem shows that the next closest solutions are neighbors in the minimal exact neighborhood.

Theorem 7.4.1. The minimal exact neighborhood for \textsc{shortest network path} is given by

\[
M_{y}^{\text{SNP}} = \{ z \in F | d(y, z) = 1 \},
\]

(7.19)

for all \( y \in F \).

Proof. First, we show that \( M_{y}^{\text{SNP}} \) is exact, by proving that it satisfies (6.36). Let \( y \in F \) and \( z \in F \). Define \( z^{(r)} \in M_{y}^{\text{SNP}}, r = 1, \ldots, d(y, z) \), by

\[
z^{(r)}_{i,j,k} = \begin{cases} 
  z_{i,j,k}, & \text{if } k_{2r} \leq k < k_{2r+1} \\
  y_{i,j,k}, & \text{otherwise},
\end{cases}
\]

for all \( i = 1, \ldots, n_{1} \) and \( k = 1, \ldots, n_{2} \). Then we have

\[
c(y; z) - c(x; z) = \sum_{r=1}^{d} \sum_{i=1}^{n_{1}} \sum_{j=1}^{n_{2}} \sum_{k=1}^{n_{3}} y_{ikj}(x_{k+1}) z_{i,j,k} - z_{i,j,k} z_{i,j,k}(x_{k+1})
\]

\[
= \sum_{r=1}^{d} \sum_{i=1}^{n_{1}} \sum_{j=1}^{n_{2}} \sum_{k=1}^{n_{3}} y_{ikj}(x_{k+1}) z_{i,j,k} - z_{i,j,k} z_{i,j,k}(x_{k+1})
\]

\[
= \sum_{r=1}^{d} \sum_{i=1}^{n_{1}} \sum_{j=1}^{n_{2}} \sum_{k=1}^{n_{3}} z_{i,j,k}(y_{ikj}(x_{k+1}) - z_{i,j,k}(x_{k+1}) z_{i,j,k}(x_{k+1}))
\]

\[
+ \sum_{r=1}^{d} \sum_{i=1}^{n_{1}} \sum_{j=1}^{n_{2}} \sum_{k=1}^{n_{3}} y_{ikj}(x_{k+1}) z_{i,j,k} - z_{i,j,k} z_{i,j,k}(x_{k+1})
\]

\[
= \sum_{r=1}^{d} \sum_{i=1}^{n_{1}} \sum_{j=1}^{n_{2}} \sum_{k=1}^{n_{3}} z_{i,j,k}(y_{ikj}(x_{k+1}) - z_{i,j,k}(x_{k+1}) z_{i,j,k}(x_{k+1}))
\]

\[
+ \sum_{r=1}^{d} \sum_{i=1}^{n_{1}} \sum_{j=1}^{n_{2}} \sum_{k=1}^{n_{3}} y_{ikj}(x_{k+1}) z_{i,j,k} - z_{i,j,k} z_{i,j,k}(x_{k+1})
\]
\[
= \sum_{r=1}^{d} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{k_2 - 1} x_{ijk}(y_{ij(k+1)} - z_{ijk}^{(r)})
\]
\[
= \sum_{r=1}^{d} \left( c(y; x) - c(z^{(r)}; x) \right),
\]

where we used in the third step that \(z_{ijk}^{(r)} = y_{ij} = z_{ijk},\) for \(k = k_2 - 1,\) all \(r = 1, \ldots, d,\)
and all \(i = 1, \ldots, n_1.\)

Next, we show that \(M_{v}^{\text{SP}} \subseteq M_{y}^{\text{IP}}\) for all \(y \in F.\) Let \(y \in F\) and \(z \in M_{v}^{\text{SP}}.\) Then we define \(x \in I^*\) such that all arcs on either of the two paths represented by \(y\) and \(z\) have length \(1,\) and all other arcs have length \(2.\) Note that the arcs incident with \(s\) or \(t\) have length \(0.\) This definition of \(x\) implies that the paths represented by \(y\) and \(z\) have total length \(n_2 - 1,\) and all other paths have total length \(2,\) which implies that all other paths have total length at least \(n_2.\) Hence, it follows that \(z \in M_{y}^{\text{IP}},\) which completes the proof of the theorem.

In Theorem 7.4.1 we used Approach Ib to show that \(M_{v}^{\text{IP}} \subseteq M_{v}^{\text{SP}},\) for all \(y \in F.\) Alternatively, we could have used Approach IIb, by showing that for all \(z \in M_{v}^{\text{SP}},\) there exist \(\lambda_{v} \geq 0, z' \neq z,\) such that
\[
c(y; x) - c(z; x) \leq \sum_{x' \neq x} \lambda_{v} (c(y; x) - c(z'; x)),
\]
(7.20)
for all \(x \in I.\) This follows easily from the fact that for all \(z \in M_{v}^{\text{SP}}, z \neq y,\) there exist \(y^{(1)}, z^{(1)} \in \mathcal{B}^{n \times n} \) and \(y^{(2)}, z^{(2)} \in \mathcal{B}^{n \times (n-p)},\) for some \(p \in \{1, 2, \ldots, n-1\},\) with \(y^{(1)} \neq z^{(1)}, y^{(2)} \neq z^{(2)},\) such that \(y = y^{(1)}y^{(2)}, z = z^{(1)}z^{(2)},\) and
\[
c(y^{(1)}z^{(2)}; x) - c(z^{(1)}z^{(2)}; x) = c(y^{(1)}y^{(2)}; x) - c(z^{(1)}y^{(2)}; x),
\]
(7.21)
for all \(x \in I;\) see Figure 7.3 where an example is shown of the four paths \(y^{(1)}y^{(2)}, z^{(1)}z^{(2)}, y^{(1)}z^{(2)},\) and \(z^{(1)}y^{(2)}.\) Obviously, (7.20) follows from (7.21), since the latter implies
\[
c(y^{(1)}y^{(2)}; x) - c(z^{(1)}z^{(2)}; x) = c(y^{(1)}y^{(2)}; x) - c(y^{(1)}z^{(2)}; x) + c(y^{(1)}y^{(2)}; x) - c(z^{(1)}y^{(2)}; x),
\]
for all \(x \in I.\)

Figure 7.3: Four paths in the network, corresponding to \(y^{(1)}y^{(2)}\) in (a), \(z^{(1)}z^{(2)}\) in (b), \(y^{(1)}z^{(2)}\) in (c), and \(z^{(1)}y^{(2)}\) in (d); see the text for an explanation.

Next, we consider the size of the minimal exact neighborhood for shortest network path. One easily verifies that \(|M_{v}^{\text{SP}}| \geq (n_1 - 1)^{\alpha_0},\) for all \(y \in F.\) A precise count gives the following result.
7.4 Shortest network path

Theorem 7.4.2. For all \( y \in F \), we have

\[
|\mathcal{M}_{y}^{\text{SNP}}| = \begin{cases} 
\frac{1}{2}n_z(n_z + 1), & \text{if } n_1 = 2 \\
\left(\frac{n_z - 1}{n_z - 2}\right)^2(n_z - 1)^{n_2} = \frac{n_z - 1}{n_z - 2}n_1 - \left(\frac{n_z - 1}{n_z - 2}\right)^2, & \text{if } n_1 \neq 2.
\end{cases}
\]

(7.22)

Proof. If \( z \in \mathcal{M}_{y}^{\text{SNP}} \) is a neighbor of a given solution \( y \in F \), then \( d(y, z) = 1 \). Hence, \( z \) is equal to \( y_{z_k} \), for all \( k \in \{1, \ldots, k_2 - 1\} \cup \{k_3, \ldots, n_2\} \). Let \( p_k(k_2, k_3) \) denote the number of neighbors \( z \in \mathcal{M}_{y}^{\text{SNP}} \) for which the pair \((y, z)\) correspond to these \( k_2 \) and \( k_3 \). Then we have for all \( 1 \leq k_2 < k_3 \leq n + 1 \)

\[
|\mathcal{M}_{y}^{\text{SNP}}| = \sum_{k_2=1}^{n_2} \sum_{k_3=k_2+1}^{n_3} p_k(k_2, k_3)
\]

\[
= \sum_{k_2=1}^{n_2} \sum_{k_3=k_2+1}^{n_3} (n_1 - 1)^{k_2-k_3}.
\]

One easily verifies that this leads to (7.22).

Minimal size first layer of MLP

Combining the result of Theorem 7.4.2, with the general lower bound for the required number of first-layer nodes, \( L(I, F, c, \mathcal{M}^{(F)}) \geq \max_{y \in F} |\mathcal{M}_{y}^{\text{SNP}}| \), proved in Theorem 6.3.7, it follows that the number of first-layer nodes \( L_{\text{SNP}} \), and hence the total number of hidden nodes, of any MLP that solves \textsc{shortest network path}, is exponential in \( n_1 \), possibly except for \( n_1 = 2 \). Using a direct argument, one shows that \( L_{\text{SNP}} \) is at least half the number of pairs of disjunct paths that can be selected in the network, which is \( n_1(n_1 - 1)^{n_2} \). Hence, \( L_{\text{SNP}} \) is exponential in \( n_2 \), for all values of \( n_1 \geq 2 \). The exact value of \( L_{\text{SNP}} \) is given in Theorem 7.4.3 below.

Theorem 7.4.3. The minimal number of first-layer nodes of an MLP that solves \textsc{shortest network path} is given by

\[
2L_{\text{SNP}} = \left(\frac{e + n_1}{e}\right)^2(e + 1)^{n_2} - \frac{e + 1}{e}n_1^n_2 + \left(\frac{e + 1}{e}\right)^2(2e - n_1^2),
\]

(7.23)

where \( e = n_1(n_1 - 1) - 1 \).

Proof. The proof is similar to the proof of Theorem 7.4.2. Let \( q(k_2, k_3) \) denote the number of non-equivalent neighbor pairs \((y, z)\), with \( y \in F \) and \( z \in \mathcal{M}_{y}^{\text{SNP}} \), that satisfy the given values of \( k_2, k_3 \). Then we have for all \( 1 \leq k_2 < k_3 \leq n + 1 \)

\[
2L_{\text{SNP}} = \sum_{k_2=1}^{n_2} \sum_{k_3=k_2+1}^{n_3} 2q(k_2, k_3)
\]

\[
= \sum_{k_2=1}^{n_2} \sum_{k_3=k_2+1}^{n_3} n_1^2[n_1(n_1 - 1)]^{k_2-k_3} + \sum_{k_2=1}^{n_2} \sum_{k_3=k_2+1}^{n_3} n_1[n_1(n_1 - 1)]^{k_2-k_3}
\]

\[
+ \sum_{k_2=1}^{n_2} \sum_{k_3=k_2+1}^{n_3} n_1[n_1(n_1 - 1)]^{k_2-k_3} + [n_1(n_1 - 1)]^{n_2}.
\]

One easily verifies that this leads to (7.23).

Note that the above formulas apply to the case where the lengths of the arcs incident with \( s \) and \( t \) are zero. If this is not true the formulas have to be adjusted slightly.

Minimal number of layers of MLP

We did not determine the minimal number of layers.
Remarks
We have shown that the minimal exact neighborhood for \textsc{shortest network path} has exponential size for \( n_1 > 2 \), which implies that every parameter independent local search algorithm for \textsc{shortest network path} has a worst case running time that is exponential in \( n_2 \); see (6.13) in Section 6.3.1. This seems to contradict the well known fact that \textsc{shortest network path} can be solved in a time polynomial in \( n_2 \). Papadimitriou & Steiglitz impute this phenomenon in a similar case to the fact that we restrict ourselves to parameter independent neighborhoods; see the discussion of the job scheduling problem with deadlines in Chapter 19.3 of [Papadimitriou & Steiglitz, 1982]. In our opinion the contradiction is caused by a wrong choice of the problem formulation, or, better, a wrong choice of the problem to be solved. The polynomial time algorithms do not solve the shortest path problem, but they solve the many-to-one shortest path problem. In Section 7.5 we show that the minimal exact neighborhood for a suitable formulation of the many-to-one shortest path problem has polynomial size, and corresponds to the neighborhood 'used' in deterministic polynomial time algorithms.

7.5 Shortest network route

In this section we discuss a variant of the many-to-one shortest path problem, which we call \textsc{shortest network route}. Consider the network network given in Figure 7.2 and assume one wants to determine the shortest path from every node in the network to the terminal node \( t \). An elementary property of the collection of shortest paths is that they form a spanning in-tree of the network. An in-tree is a tree with each node having out degree 1. We therefore consider the problem of finding a collection of paths from every node to the terminal node that forms a spanning in-tree and has minimal total length. We call a collection of paths that forms a spanning in-tree a route. The problem then becomes to find the shortest route in the network, which can be given in a 0-1 formulation as follows.

A 0-1 formulation

\textbf{Definition 7.5.1. (Shortest network route (SNR))}
Let \( y_{ijk} \in \{0,1\} \) be a 0-1 variable, such that \( y_{ijk} = 1 \), if the route goes from node \( i \) of level \( k \) to node \( j \) of level \( k+1 \), and \( y_{ijk} = 0 \) otherwise, where \( i,j = 1, \ldots, n_1 \) and \( k = 1, \ldots, n_2 - 1 \). Furthermore, we introduce the auxiliary 0-1 variable \( y_{is0} \) such that \( y_{is0} = 1 \) if the route goes from \( s \) to node \( j \) of level 1 and \( j \neq 0 \) otherwise.

Let \( x_{ijk} \geq 0 \) denote the length of the arc between node \( i \) of level \( k \) and node \( j \) of level \( k+1 \). Then the length \( w(i,k) \) of the path that starts at node \((i,k)\), ends at the terminal \( t \), and travels using the route set by \( y \) can be recursively defined by

\begin{enumerate}
\item[(i)] \( w(i,n_2) = 0 \), for all \( i = 1, \ldots, n_1 \),
\item[(ii)] \( w(i,k) = \sum_{j=1}^{n_1} y_{ijk}(x_{ijk} + w(j,k+1)) \), for all \( i = 1, \ldots, n_1 \) and \( k = 1, \ldots, n_2 - 1 \), and
\item[(iii)] \( w(s,0) = \sum_{j=1}^{n_1} y_{is0} w(j,1) \).
\end{enumerate}

Then the problem can be represented by the tuple \((I,F,c)\) given by

\[ I = \mathbb{R}^{n_2 \times n_1 \times (n_2 - 1)}, \]
7.5 Shortest network route

\[ F = \{ y \in B^{n_1} \times B^{n_1 \times n_2 \times (n_2-1)} \mid \sum_{j=1}^{n_1} y_{ij} = 1 \}, \]

\[ c(y; z) = w(s, 0) + \sum_{i=1}^{n_1} \sum_{k=1}^{n_2-1} w(i, k). \]

We do not go into the details of checking all the conditions like we did for the previous three problems, we just mention that they are satisfied, similar as for Shortest Network Path. We show that Shortest Network Route has a minimal exact neighborhood of polynomial size using Approach IIa. However, before doing any calculations we can state a relation between the minimal number of first-layer nodes that is required by any MLP that solves Shortest Network Route and the minimal number of first-layer nodes that is required by any MLP that solves Shortest Network Path.

**Proposition 7.5.1.** The minimal number of first-layer nodes of an MLP that solves Shortest Network Route satisfies

\[ L_{\text{SNR}} \geq L_{\text{SNP}}. \]

A proof of the above observation is that if one has an MLP that solves Shortest Network Route with a certain number of first-layer nodes, then one easily constructs a \((m+2)\)LP with the same number of first-layer nodes that solves Shortest Network Path, by putting two extra layers on top of the original MLP.

A second remark is that there is a great resemblance between Shortest Network Route and Minimum Cost Spanning Tree. The sequential and parallel time complexity of both problems are comparable; see Papadimitriou & Steiglitz [1982] and Kindervater [1989]. Moreover, Maggs & Plotkin [1988] have demonstrated that Minimum Cost Spanning Tree can be viewed as a special kind of the all-pairs shortest path problem. This resemblance can also be found between the proofs of Theorem 7.3.1 and Theorem 7.5.1, the latter being presented below. However, although the minimal size of the first layer of any MLP that solves Minimum Cost Spanning Tree is shown to be polynomial, the minimal size of the first layer of any MLP that solves Shortest Network Route is exponential. This follows from Proposition 7.5.1 and the results of the previous section.

**Minimal Exact Neighborhood**

The following theorem describes the minimal exact neighborhood for Shortest Network Route, which is based on the Hamming distance between two routes.

**Theorem 7.5.1.** The minimal exact neighborhood for Shortest Network Route is given by

\[ M_y^{\text{SNR}} = \{ z \in F \mid h(y, z) = 2 \}, \]

for all \( y \in F \), where \( h(y, z) \) denotes the number of positions at which \( y \) and \( z \) differ, i.e., the Hamming distance between \( y \) and \( z \).

**Proof.** First, we show that \( M_y^S \subseteq M_y^{\text{SNR}} \), for all \( y \in F \), using Approach IIa. Take \( y \in F, z \notin M_y^{\text{SNR}}, x \in I \) and assume \( c(z; x) < c(y; x) \). Let \( k^* \in \{1, \ldots, n_2\} \) be the smallest level such that \( y_{ijk} = z_{ijk} \), for all \( k > k^* \) and \( i, j = 1, \ldots, n_1 \). Let \( i^* \) be such that \( y_{i^*j^*k^*} = 1 = z_{i^*j^*k^*} \), for some \( j_1 \neq j_2 \). We complete the argument by
considering the following two cases.

\[ x'_{ijk} \leq x_{ijk}. \]  
Let \( z' \in F \setminus \{ z \} \) be defined by

\[ z'_{ijk} = \begin{cases} 
  z_{ijk}, & \text{if } (i,k) \neq (i^*, k^*), j = 1, \ldots, n_1 \\
  y_{ijk}, & \text{if } (i,k) = (i^*, k^*), j = 1, \ldots, n_1.
\end{cases} \]

Then, using \( y_{ijk} = z_{ijk} \), for all \( k > k^* \), one can easily show that \( c(z'; x) \leq c(z; x) \), which proves \( c(z'; x) < c(y; x) \).

\[ x'_{ijk} > x_{ijk}. \]  
Let \( z' \in M_{z'}^{SNR} \subseteq F \setminus \{ z \} \) be defined by

\[ z'_{ijk} = \begin{cases} 
  y_{ijk}, & \text{if } (i,k) \neq (i^*, k^*), j = 1, \ldots, n_1 \\
  z_{ijk}, & \text{if } (i,k) = (i^*, k^*), j = 1, \ldots, n_1.
\end{cases} \]

Then, using \( z_{ijk} = y_{ijk} \), for all \( k > k^* \), one can easily show that \( c(z'; x) < c(y; x) \).

Next, we show that \( M_{y}^{SN} \subseteq M_{y}^{IP} \), for all \( y \in F \). Let \( y \in F \) and \( z \in M_{y}^{SN} \) and assume \( y_{i0} = z_{i0} \), for all \( j = 1, \ldots, n_1 \). Then we define \( x \in F \) such that all arcs on either of the two routes represented by \( y \) and \( z \) have length 1 and all other arcs have length 2, except for the arcs incident with \( s \) or \( t \), that have length 0. Then one easily verifies that the routes represented by \( y \) and \( z \) have equal total length and all other routes have greater total length. If \( y_{i0} = z_{i0} \), for some \( i \neq j \), we have to make a small adjustment in the above argument. Increase the length of the arcs on the paths \( y \) and \( z \) between the first and the second level that do not start in \( i \) or \( j \) with \( 1 \). Then the same conclusion can be drawn and, hence, it follows that \( x \in M_{y}^{IP} \), which completes the proof of the theorem.

\[ \square \]

It follows directly that size of the minimal exact neighborhood for SHORTEST NETWORK ROUTE is polynomial.

**Proposition 7.5.2.** For all \( y \in F \) we have \( |M_{y}^{SNR}| = n_1^2 n_2 - n_1^2 + n_1 \).

**Minimal size first layer of MLP**

By Proposition 7.5.1 and the results of Section 7.4, we know that the minimal number of first-layer nodes for any MLP that solves SHORTEST NETWORK ROUTE is exponential in \( n_2 \). Combined with the above result, this proves that for the minimal size of the first hidden layer of an MLP that solves a given problem to be polynomial, it is not sufficient that the size of the minimal exact neighborhood is polynomial.

**Minimal number of layers of MLP**

We did not determine the minimal number of layers required for solving SHORTEST NETWORK ROUTE.

### 7.6 Discrete dynamic lotsizing

In this section we study the MLP-complexity of DISCRETE DYNAMIC LOTSIZING, as introduced in Chapter 3. This problem is interesting, because it originates from a practical problem, that is transformed into a 0-1 problem, which parallel complexity is not yet fully determined. As is shown in Chapter 3, it is a special case of the shortest path problem. However, although SHORTEST NETWORK PATH is shown to be
unsolvable by an MLP with a polynomial number of nodes, this does not necessarily imply that the same holds for DISCRETE DYNAMIC LOTSIZING.

The problem differs from the previous problems because its cost function $c(y; x)$ is not linear in $x$. Hence, DISCRETE DYNAMIC LOTSIZING gives an example of a problem where we cannot use Farkas’ lemma, but require the extended formulation of Lemma 6.3.4. This aspect returns in the proof of Theorem 7.6.1, where we use Approach 1b, with the strict less than sign in some cases. This implies that in DISCRETE DYNAMIC LOTSIZING, the facets of $I$ are part of some of the minimal representations of the subsets $\mathcal{P}_i$; see also Chapter 6. Theorem 7.6.1 identifies the minimal exact neighborhood for DISCRETE DYNAMIC LOTSIZING. Subsequently, we show that there exist solutions for which the size of their minimal exact neighborhood is exponential. As before, this proves that the minimal size of the first layer of any MLP that solves DISCRETE DYNAMIC LOTSIZING is exponential in the number of inputs.

A 0-1 formulation
In Chapter 3 we have shown that the general dynamic lotsizing problem can be rewritten in the form given by the following 0-1 formulation.

**Definition 7.6.1. (DISCRETE DYNAMIC LOTSIZING (DDLs))**

Let $x_1, \ldots, x_n \geq 0$ be the demand that has to be satisfied for the next $n$ periods, for some $n \in \mathbb{N}$. Except for the setup in period 0, a setup occurs in period $i$, if $y_i = 1$, and does not occur, if $y_i = 0$. The cost of a setup is 1, the cost of storing a quantity $x$ for one period is $z$, and there are no other costs. Then the problem can be represented as the tuple $(I, F, c)$ given by

\[
I = \{ x \in \mathbb{R}^n \mid x_i \geq 0 \}, \\
F = \mathbb{B}^n, \\
c(y; x) = \sum_{i=1}^{n} \sum_{j=1}^{i} \left( \prod_{k=j}^{i} (1 - y_k) \right) x_i + \sum_{i=1}^{n} y_i,
\]

where for a given $x \in I$ the problem is to find a $y \in F$ that minimizes $c(y; x)$.

**Minimal Exact Neighborhood**

One can easily verify that the above formulation satisfies the Conditions 1a, 1b, and 2a, with $N = K = n$, $a_i(y) = \sum_{j=1}^{i} \prod_{k=j}^{i} (1 - y_k)$, and $b_i(y) = \sum_{i=1}^{n} y_i$. Furthermore, we have that $c(y; 2y) < c(z; 2y)$, for all $z \in F \setminus \{y\}$. However, since $2y \notin F$, for all $y \in F$, we have not yet proved Condition 2b. One easily finds an $\bar{z} \in I^\circ$, such that $c(y; \bar{z}) \neq c(z; \bar{z})$, for all $z \in F \setminus \{y\}$. Hence, there exists a $\lambda \in (0, 1)$ such that $x = 2\lambda y + (1 - \lambda) z \in I^\circ$, and $c(y; x) < c(z; x)$, for all $z \in F \setminus \{y\}$. This proves that Condition 2b is satisfied. Hence, it follows by Corollary 6.3.1 and Corollary 6.3.2, that $\mathcal{M}^e = \mathcal{M}^e_{\text{DDLs}}$ is the minimal exact neighborhood for DISCRETE DYNAMIC LOTSIZING. The following theorem presents an explicit characterization of $\mathcal{M}^e$, following the lines of Approach 1b.

**Theorem 7.6.1.** The minimal exact neighborhood for DISCRETE DYNAMIC LOTSIZING is given by $\mathcal{M}^e_{\text{DDLs}}$, where for all $y \in F$, $\mathcal{M}^e_{\text{DDLs}}$ consists of all feasible solutions $x \in F$ for which there exist $k \in \mathbb{N}$, $0 \leq i_0 < i_1 < \cdots < i_{k+1} \leq n + 1$, and
\( \tau \in \{-1,+1\} \), that satisfy

\[
\begin{align*}
&\begin{cases}
  y_i = z_i + \tau(-1)^j, & i = i_j, j \in \{1, \ldots, k\}, \\
  y_i = z_i = 0, & i \in \bigcup_{j=0}^{k} I_j, \\
  y_i = z_i, & i \in I_{-1} \cup I_{k+1}, \\
  y_i = z_i = 1, & i \in \{i_0, i_{k+1}\},
\end{cases}
\end{align*}
\]

for all \( i = 1, \ldots, n \), where \( I_j = \{i_j + 1, i_j + 2, \ldots, i_{j+1} - 1\} \), \( j = 0, 1, \ldots, k \), \( I_{-1} = \{1, 2, \ldots, i_0\} \), and \( I_{k+1} = \{i_{k+1}, i_{k+1} + 1, \ldots, n\} \).

**Proof.** As a visual aid in the remainder of the proof, we note that if \((y, z)\) is a neighbor pair with \( z \in M^{\text{وصل}} \) or \( y \in M^{\text{وصل}} \), then this pair looks like

\[
\begin{align*}
  y &= \begin{cases}
  00 \cdots 0100 \cdots 0000 \cdots 01 \cdots 0 v \\
  00 \cdots 0000 \cdots 0100 \cdots 0000 \cdots 0 v,
\end{cases} \\
  z &= \begin{cases}
  00 \cdots 0100 \cdots 0000 \cdots 01 \cdots 0 v \\
  00 \cdots 0000 \cdots 0100 \cdots 0000 \cdots 0 v,
\end{cases}
\end{align*}
\]

(7.26)

for some \( u \in B^w \) and \( v \in B^{w-1} \) with \( u \neq \tilde{u}0 \) and \( v \neq \tilde{v}0 \).

First, we prove that \( M^w \subseteq M^\text{وصل} \), for all \( y \in F \), using Approach IIIb, i.e., we show that for all \( x \in M^\text{وصل} \), there exist \( \lambda_p \geq 0, \ z' \neq z, \) such that

\[
\begin{align*}
  c(y; z) - c(z; z) &= \sum_{(p, z) \in E} \lambda_p (c(y; z) - c(x', x)),
\end{align*}
\]

(7.27)

for all \( x \in I \). In fact, we show that for all \( z \notin M^\text{وصل} \), \( z \neq y \), there exist \( y^{(1)}, z^{(1)} \in B^w \) and \( y^{(2)}, z^{(2)} \in B^{w-1} \), for some \( p \in \{1, 2, \ldots, n-1\} \), with \( y^{(1)} \neq z^{(1)} \) and \( y^{(2)} \neq z^{(2)} \), such that

\[
\begin{align*}
  c(y^{(1)}; z^{(1)}; x) - c(z^{(1)}; z^{(1)}; x) &\leq c(y^{(1)}; y^{(2)}; z) - c(z^{(1)}; y^{(2)}; z),
\end{align*}
\]

(7.28)

for all \( x \in I \), from which (7.27) follows, since it implies

\[
\begin{align*}
  c(y^{(1)}; y^{(2)}; z) - c(z^{(1)}; z^{(2)}; z) &\leq c(y^{(1)}; y^{(2)}; z) - c(y^{(1)}; y^{(2)}; z) + c(y^{(1)}; y^{(2)}; z) - c(z^{(1)}; y^{(2)}; z),
\end{align*}
\]

(7.29)

for all \( x \in I \). Note that (7.28) can also be used directly to prove that \( M^w \subseteq M^\text{وصل} \), for all \( y \in F \).

Let \( y \in F \), and \( z \notin M^\text{وصل} \), \( z \neq y \). Then there are two possibilities for the pair \((y, z)\).

**Case 1.**

There exist \( 2 \leq i \leq n - 2 \), \( \hat{y}, \hat{z} \in B^w \), and \( \hat{y}, \hat{z} \in B^{w-1} \), with \( \hat{y} \neq \hat{z}, \hat{y} \neq \hat{z} \), and

\[
\begin{align*}
  y &= \begin{cases}
    \hat{y} 1 \hat{y} \\
    \hat{z} 1 \hat{z},
\end{cases} \\
  z &= \begin{cases}
    \hat{y} 1 \hat{y} \\
    \hat{z} 1 \hat{z}.
\end{cases}
\end{align*}
\]

(7.29)

Since we obviously have

\[
\begin{align*}
  c(\hat{y} 1 \hat{y}; x) - c(\hat{z} 1 \hat{z}; x) &= c(\hat{y} 1 \hat{y}; x) - c(\hat{z} 1 \hat{y}; x),
\end{align*}
\]

for all \( x \in I \), (7.28) follows directly by choosing \( y^{(1)} = \hat{y}, y^{(2)} = 1 \hat{y}, z^{(1)} = \hat{z}, \) and \( z^{(2)} = 1 \hat{z} \).

**Case 2.**

There exist \( 0 \leq i \leq j < n - 2 \), \( \hat{y}, \hat{z} \in B^w \), and \( \hat{y}, \hat{z} \in B^{w-1} \), such that

\[
\begin{align*}
  y &= \begin{cases}
    \hat{y} 100 \cdots 01 \hat{y} \\
    \hat{z} 000 \cdots 0 \hat{z},
\end{cases} \\
  z &= \begin{cases}
    \hat{y} 100 \cdots 01 \hat{y} \\
    \hat{z} 000 \cdots 0 \hat{z}.
\end{cases}
\end{align*}
\]

(7.30)

In this case we obtain (7.28) from

\[
\begin{align*}
  c(\hat{y} 100 \cdots 01 \hat{y}; x) - c(\hat{z} 000 \cdots 0 \hat{z}; x)
\end{align*}
\]
7.6 Discrete dynamic lotsizing

\[
\begin{align*}
&\leq c(\hat{y}(1000\ldots1110\ldots0; x) - c(\hat{z}(000\ldots0111\ldots0; x)) \\
&= c(\hat{y}(1000\ldots10\hat{y}; x) - c(\hat{z}(000\ldots01\hat{z}; x),
\end{align*}
\]
for all \( x \in I \), which is straightforwardly verified, using \( x_i \geq 0 \), for all \( i = 1, \ldots, n \).

Next, we show that \( \mathcal{M}_y^{\text{Deks}} \subseteq \mathcal{M}_y^\tau \), for all \( y \in F \). Let \( y \in F \) and \( z \in \mathcal{M}_y^{\text{Deks}} \). Then there exist \( k \in \mathbb{N}, 0 \leq i_0 < i_1 < \cdots < i_{k+1} \leq n + 1 \), and \( \tau \in \{-1, 1\} \), that satisfy the conditions posed in the theorem. Define \( r_j = i_j - i_{j-1} \geq 1 \), for all \( j = 1, \ldots, k+1 \), \( r = \max\{r_j; j = 1, \ldots, k+1\}, \beta = 1 - (r + 2)^{-1} \), and \( \alpha = (\beta l + 1)/(l + 1) \), where \( l = \lfloor k/2 \rfloor \). In the definition of \( x \in I \), we distinguish between the case that \( k \) is even and the case that \( k \) is odd.

\( k = 2l \). Let \( x \in I \) be defined by

\[
x_{i_j} = \begin{cases} 
\alpha/r_j, & i = i_j, j \in \{1, k\}, \\
\beta/r_j, & i = i_j, j \in \{2, 3, \ldots, k-1\}, \\
0, & i \in \bigcup_{j=0}^l I_j, \\
2y_{i_j}, & i \in I_{l-1} \cup I_{k+1}.
\end{cases}
\]

\( k = 2l + 1 \). Let \( x \in I \) be defined by

\[
x_{i_j} = \begin{cases} 
\alpha/r_j, & i = i_j, j \in \{1, 3, \ldots, k\}, \\
\beta/r_j, & i = i_j, j \in \{2, 4, \ldots, k-1\}, \\
0, & i \in \bigcup_{j=0}^l I_j, \\
2y_{i_j}, & i \in I_{l-1} \cup I_{k+1}.
\end{cases}
\]

Thus, in the notation of (7.26), we have

\[
x = \begin{pmatrix} 2u_0 \cdots 0 \alpha \frac{\alpha}{r_1} \cdots 0 \frac{\beta}{r_2} \cdots 0 \frac{\beta}{r_k} \cdots 0 \alpha \cdots 0 \frac{\alpha}{r_k} \cdots 0 2v \\
2u_0 \cdots 0 \frac{\alpha}{r_1} \cdots 0 \frac{\beta}{r_2} \cdots 0 \frac{\beta}{r_k} \cdots 0 \frac{\alpha}{r_k} \cdots 0 2v
\end{pmatrix}
\]

(7.31)

depending on whether \( k \) is even or odd.

Using \( 1 - (r_j + 1)^{-1} < \beta < \alpha < 1 \), for all \( j = 1, \ldots, k \), and some straightforward calculations one can show that in both cases, \( c(y; x) = c(z; x) \) and \( c(y; x) < c(y'; x) \), for all \( y' \in F \setminus \{y, z\} \). However, this does not prove that \( x \in \mathcal{M}_y^\tau \), because \( z \notin F^* \) in general. This can be mended as follows.

Let \( \hat{x} \in F^* \) be defined by

\[
\hat{x}_{i_j} = \begin{cases} 
1, & i = i_j, j \in \{2, 3, \ldots, k-1\}, \\
1/r_j, & i = i_j, \\
\frac{1}{2} + \frac{1}{r_{k+1}} 1, & i = i_k, \\
1/r_j, & i \in I_j, j \in \{1, \ldots, k-1\}, \\
1, & i \in I_{l-1} \cup I_0 \cup I_{k+1}, \\
\frac{1}{2}(r_{k+1} - 1)^{-1}, & i \in I_k.
\end{cases}
\]

Then one can verify that \( c(y; \hat{x}) = c(z; \hat{x}) \). Combining \( x \in I \) with \( \hat{x} \in F^* \) yields an \( \hat{z} \in F^* \) such that \( c(y; \hat{z}) = c(z; \hat{z}) \) and \( c(y; x) < c(y'; x) \), for all \( y' \in F \setminus \{y, z\} \). Hence, \( x \in \mathcal{M}_y^\tau \), which completes the proof of the theorem.

Next, we consider the size of the minimal exact neighborhood for discrete dynamic lotsizing. It turns out that \( \mathcal{M}_y^{\text{Deks}} \) depends on \( y \in F \). For instance, one easily verifies that \( \mathcal{M}_y^{\text{Deks}} = n \), if \( y = 0 \) or \( y = 1 \), corresponding to a vector of all zeros and a vector of all ones, respectively. Below we show that for some solutions
$y \in F$, $|\mathcal{M}_y^{\text{DOL}}|$ is exponential in $n$. To this end we present the following lemma, which gives some recursive relations for $|\mathcal{M}_y^{\text{DOL}}|$.

**Lemma 7.6.1.** Let $s_n(y) = |\mathcal{M}_y^{\text{DOL}}|$, for all $y \in F = B^n$, and for all $n \in \mathbb{N}$. Then $s_1(1) = s_1(0) = 1$, and for all $n \in \mathbb{N}$ and $y \in B^n$

\[
egin{align*}
    s_{n+2}(11y) &= s_{n+1}(1y) + 1, \quad \text{(7.32)}
    \quad \text{for all y, z in F, y \sim z, if and only if z in } \mathcal{M}_y^{\text{DOL}}.
    
    s_{n+2}(00y) &= 2s_{n+1}(0y) - s_n(y), \quad \text{(7.33)}
    
    s_{n+2}(01y) &= 2s_{n+1}(1y) - s_n(y) + 1, \quad \text{(7.34)}
    
    s_{n+2}(10y) &= 2s_{n+1}(0y) + s_{n+1}(1y) - 2s_n(y). \quad \text{(7.35)}
\end{align*}
\]

**Proof.** We prove (7.32) and (7.35), the other two are proved similarly. Define for all $y, z \in F, y \sim z$, if and only if $z \in \mathcal{M}_y^{\text{DOL}}$.

Using Theorem 7.6.1, it follows that $1y \sim abz$, for some $a, b \in B$, if and only if either $a = 1$ and $1y \sim b$, or $ab = 01y$. This proves the correctness of (7.32).

The proof of (7.35) is given using a slightly different approach. We note the following constructions of neighbors of 10y.

If $0y \sim 1z$, then $10y \sim 01z$ and $10y \sim 11z$.
If $0y \sim 0z$, then $10y \sim 10z$ and $10y \sim 10z$.
If $1y \sim 1z$, then $10y \sim 10z$.
If $1y \sim 0z$, then $10y \sim 00z$.

One can verify that these are all the possibilities of constructing neighbors of 10y. It gives $2s_{n+1}(0y) + s_{n+1}(1y)$ neighbors, except that precisely $s_n(y)$ of them are counted three times. Correcting this yields (7.33). □

Using Lemma 7.6.1 one can compute the size of the neighborhood $\mathcal{M}_y^{\text{DOL}}$ of any $y \in F$. Indeed one verifies $s_n(0) = s_n(1) = n$. In the following lemma we consider four special types of solutions, which all have an exponential minimal exact neighborhood.

**Lemma 7.6.2.** Let

\[
\begin{align*}
    a_p &= s_{3p} (010010 \cdots 010010), \\
    b_p &= s_{3p-1} (10010 \cdots 010010), \\
    c_p &= s_{3p-1} (010010 \cdots 010010), \\
    d_p &= s_{3p-2} (10010 \cdots 010010).
\end{align*}
\]

for all $p \in \mathbb{N}$. Then

\[
\begin{align*}
    a_p &= 25 \cdot 2^{p-1} - 7p - 12, \quad \text{(7.36)}
    
    b_p &= 20 \cdot 2^{p-1} - 7p - 10, \quad \text{(7.37)}
    
    c_p &= 20 \cdot 2^{p-1} - 7p - 10, \quad \text{(7.38)}
    
    d_p &= 16 \cdot 2^{p-1} - 7p - 8, \quad \text{(7.39)}
\end{align*}
\]

for all $p \in \mathbb{N}$.

**Proof.** Firstly, using Lemma 7.6.1, one can easily show that $a_p, b_p$ satisfy the following equations.

\[
\begin{align*}
    a_p &= 6s_{p-1} - 5b_{p-1} + 3, \\
    b_p &= 4s_{p-1} - 3b_{p-1} + 1.
\end{align*}
\]
Furthermore, \( a_1 = 6 \) and \( b_1 = 3 \). Solving this inhomogeneous matrix equation yields (7.36) and (7.37).

Secondly, using Lemma 7.6.1, one can easily show that \( c_n, d_n \) satisfy the following equations.

\[
\begin{align*}
c_n &= 6c_{n-1} - 5d_{n-1} + 3, \\
d_n &= 4c_{n-1} - 3d_{n-1} + 1.
\end{align*}
\]

Furthermore, \( c_1 = 3 \) and \( d_1 = 1 \). Solving this inhomogeneous matrix equation yields (7.38) and (7.39).

We conjecture that the solutions distinguished in Lemma 7.6.2, are the solutions with maximal neighborhood size, for the respective values of \( n \). This can be formulated as follows.

**Conjecture 7.6.1.** Let \( u : \mathbb{N} \to \mathbb{N} \) be defined by.

\[
u(n) = \begin{cases} 
25 \cdot 2^{n-1} - 7p - 12, & n = 3p, \\
20 \cdot 2^{n-1} - 7p - 10, & n = 3p - 1, \\
16 \cdot 2^{n-1} - 7p - 8, & n = 3p - 2.
\end{cases}
\]

Then \( n \leq |\mathcal{M}(\mathcal{G})| \leq u(n) \), for all \( y \in F = \mathbb{B}^p \).

**Minimal size first layer of MLP**

So far, we have not been able to compute the minimal size of the first layer of any MLP that solves **Discrete Dynamic Lotsizing**. However, Lemma 7.6.2 and Theorem 6.3.7 imply that \( l_{\text{max}} \geq \max_{\mathcal{G} \in F} |\mathcal{M}(\mathcal{G})| \geq u(n) \), for all \( n \in \mathbb{N} \), where \( u \) is the function defined in Conjecture 7.6.1. Hence, **Discrete Dynamic Lotsizing** cannot be solved by an MLP with a polynomial number of nodes.

**Minimal number of layers of MLP**

One can verify that **Discrete Dynamic Lotsizing** can be solved by a 2LP, in case that \( n \leq 2 \); see also Zwieker, Van Kraaij, Aarts & Wessels [1991]. Using the twisted bow-tie condition for COPs derived in Chapter 6, we prove that **Discrete Dynamic Lotsizing** cannot be solved by a 2LP, in case that \( n \geq 3 \). First, we consider the case that \( n = 3 \).

Let \( x^{(1)} = (\frac{3}{5}, \frac{1}{5}, \frac{1}{5}) \), \( x^{(2)} = (\frac{3}{5}, \frac{3}{5}, \frac{1}{5}) \), \( x^{(3)} = (\frac{3}{5}, \frac{3}{5}, \frac{1}{5}) \), \( x^{(4)} = (\frac{3}{5}, \frac{3}{5}, \frac{1}{5}) \), \( y^{(1)} = (0, 0, 1) \), \( y^{(2)} = (0, 1, 0) \), \( y^{(3)} = (1, 0, 0) \), and \( y^{(4)} = (1, 0, 1) \). Then one easily verifies that the following conditions are satisfied.

- \( c(y^{(1)}; x^{(1)}) < c(x^{(1)}; z^{(1)}) \), for all \( z \in F \setminus \{y^{(1)}\} \), and \( j = 1, 2, 3, 4 \).
- \( y_1^{(1)} = y_2^{(2)} = 0 \) and \( y_1^{(3)} = y_4^{(4)} = 1 \).
- \( z_{12} = \frac{x^{(1)} + x^{(2)}}{2} = z_{34} = \frac{x^{(3)} + x^{(4)}}{2} = (\frac{3}{5}, \frac{3}{5}, \frac{1}{5}) \in F^* \).
- \( c(y^{(1)}; x^{(3)}) = c(y^{(2)}; x^{(1)}) \leq c(z; x_{12}) \), for all \( z \in F \setminus \{y^{(1)}, y^{(2)}\} \), and \( c(y^{(3)}; x_{34}) \leq c(z; x_{34}) \), for all \( z \in F \setminus \{y^{(3)}, y^{(4)}\} \).

Hence, using Theorem 6.4.2, it follows that **Discrete Dynamic Lotsizing** cannot be solved by a 2LP, in case that \( n = 3 \).

The idea behind the above argument can be illustrated using Figure 7.4, in which we have shown the subset \( V \subseteq \mathbb{R}^2 \), that corresponds to all \( (x_1, x_2) \in \mathbb{R}^2 \), for which
Figure 7.4: The instances \((y_1^*, x_2, x_3)\) for which the optimal solution \((y_1, y_2, y_3)\) satisfies \(y_1 = 0\). The points (1), (2), (3), and (4), correspond to the points \(x^{(1)}\), \(x^{(2)}\), \(x^{(3)}\), and \(x^{(4)}\), that are introduced in the text. The point (0) corresponds to \(x_1 = x_3\).

The optimal solution \(y = (y_1, y_2, y_3)\) of the instance defined by \((y^*_1, x_2, x_3)\) satisfies \(y_1 = 0\). Since this subset does not satisfy the twisted bow-tie condition, it cannot be classified with a 2LP, and, hence, DISCRETE DYNAMIC LOT SIZING cannot be solved by a 2LP, in case that \(n = 3\).

That DISCRETE DYNAMIC LOT SIZING cannot be solved by a 2LP in case that \(n > 3\), follows directly from the fact that it cannot be solved by a 2LP in case that \(n = 3\), and the fact that if \(f : \mathbb{R}_+^n \rightarrow \mathbb{B}^n\) represents a 2LP that solves DISCRETE DYNAMIC LOT SIZING in case \(n > 3\), then \(\bar{f} : \mathbb{R}_+^n \rightarrow \mathbb{B}^3\), defined by \(\bar{f}(x_1, x_2, x_3) = f_i(x_1, x_2, x_3, 0, 0, \ldots, 0), i = 1, 2, 3\), represents a 2LP that solves DISCRETE DYNAMIC LOT SIZING in case that \(n = 3\).

7.7 Concluding remarks

In this chapter we have used the results of Chapter 6 to determine the minimal number of first-layer nodes required by any MLP that solves SORTING, MINIMUM COST SPANNING TREE, SHORTEST NETWORK PATH, SHORTEST NETWORK ROUTE, and DISCRETE DYNAMIC LOT SIZING, respectively. The minimal number of first-layer nodes of any MLP that solves SORTING or MINIMUM COST SPANNING TREE was shown to be polynomial in the number of inputs. Furthermore, we presented constructions of MLPs with a polynomial number of nodes for both SORTING and MINIMUM COST SPANNING TREE. For the other three problems we could show that the minimal number of nodes was exponential in the number of inputs. Finally, we proved that the minimal number of layers required by any MLP that solves SORTING or DISCRETE DYNAMIC LOT SIZING is three.

The results on SORTING appeared previously in a series of papers; see Zwietering, Aarts & Wessels [1991b], Zwietering, Aarts & Wessels [1993], and Zwietering, Aarts
7.7 Concluding remarks

& Wessels [1992b]. In [Zwietering, Aarts & Wessels, 1992b], we discuss the possibility that there exists a 2LP for INTEGER SORTING, if one considers the case where the numbers to be sorted are taken from a bounded set of integers. The study of an MLP approach for solving DISCRETE DYNAMIC LOTSIZING formed the starting point for our study of the complexity of MLPs in general. In the future we hope to extend the analysis to more practical settings, including decision support systems; see also Zwietering, Van Kraaij, Aarts & Wessels [1991] and Aarts, Wessels & Zwietering [1993].
Chapter 8

Conclusions

In this thesis we have studied the complexity of multi-layered perceptrons (MLPs). Particularly, we considered the complexity of MLPs with real-valued inputs and hard-limiting response functions, required for solving Combinatorial Optimization Problems (COPs). We examined three approaches for obtaining insight in the MLP-complexity of COPs, with respect to the required number of nodes and the required number of layers.

The first approach was to examine the MLP-complexity of various problems that are available in the literature on neural networks. A large part considers MLPs for solving finite Combinatorial Classification Problems (CCPs), where a finite number of points in $\mathbb{R}^N$ has to be classified with respect to a finite number of classes. The discussed results show that only a small fraction of all finite CCPs can be solved by a one-layered perceptron (1LP), although the weights of a 1LP can be found in polynomial time in case that it exists. On the other hand, all finite CCPs can be solved by a two-layered perceptron (2LP) with a sufficiently large number of hidden nodes. However, the problem of finding the weights of a 2LP with a given number of hidden nodes is $\mathcal{NP}$-hard. Both the complexity of the problem of finding the required number of hidden nodes as the complexity of the problem of finding a set of weights given the number of hidden nodes are not reduced in case that one considers MLPs with more than two layers. A second part of the neural network literature considers the approximation capabilities of MLPs that use some kind of sigmoidal response functions. We have discussed some results that show that 2LPs can approximate with arbitrary precision any reasonably well-behaved function, if the number of hidden nodes is sufficiently large. Recent results provide bounds on the required number of hidden nodes for special classes of functions.

The above results can be used to show that finite COPs, where the parameter set $I$ is a finite subset of $\mathbb{R}^N$, can be solved by a 2LP with a sufficiently large number of hidden nodes. However, we have considered COPs where $I$ is an arbitrary subset of $\mathbb{R}^N$. In this case the above results indicate that two layers may not be sufficient to solve the problem at hand, although two layers may be sufficient to approximately solve the problem with an MLP.
The second approach was based on the introduction of complexity classes for MLPs, similar like it is done for Boolean circuits. We first proposed a computational model that captures both MLPs and Boolean circuits as special cases. For this Unidirectional Computational Model (UCM), we introduced a number of complexity classes. These complexity classes contain problems corresponding to families of functions that can be solved by a family of UCMs, one UCM for each function, satisfying some restrictions. The considered restrictions concern the number of nodes, the number of layers, the type of node functions, the set of possible inputs, the set of possible outputs, and the maximum fan-in. In some cases the accuracy of the computation of the node function was also taken into account. Studying the hierarchy of the various complexity classes gave insight in the relative power of the MLP model when compared for different settings of size, depth, and fan-in. Furthermore, when the inputs of the MLP are restricted to Boolean values, the capabilities of the MLP model can be compared to that of the well studied classes for Boolean circuits. The results show that the capabilities of MLPs are comparable to those of the threshold circuits, even when using unbounded real-valued weights or arbitrary sigmoidal response functions. Indirectly, this shows that the capabilities of MLPs with hard-limiting response functions are equivalent to those of MLPs that use arbitrary sigmoidal response functions.

In contrast to the first two approaches the third approach is based on a completely new theoretical framework for the mathematical description of the capabilities of MLPs that use real-valued inputs and hard-limiting response functions. First, we considered the capabilities of this type of MLPs in general. We studied the MLP-complexity required to represent an arbitrary function from $\mathbb{R}^N$ to $\mathbb{B}$, which corresponds to a Binary Classification Problem (BCP), where the two subsets to be classified are each others complement with respect to $\mathbb{R}^N$. We formulated necessary and sufficient conditions for such a BCP to be solvable with a MLP, for some integer $m$. By far the largest part of the work concerned the case that $m$ equals 2, where the necessary and sufficient conditions do not match in general, although we gave arguments that indicate that the gap is small, at least for low values of $N$, the dimension of the considered space. The necessary conditions are based on some local geometrical properties of the subsets corresponding to the BCP at hand. The sufficient conditions are based on construction, by taking the intersection and union of certain special types of subsets, that are such that the result is classifiable. For one of the obtained sufficient conditions we presented a decomposition algorithm that yields a description of this construction. A second set of results concerned the number of first-layer nodes that is required to solve a given BCP with an MLP. We have presented a lower bound that was shown to be tight in case that the BCP consists of a full-dimensional polyhedron and its complement.

Subsequently, the obtained general results on the capabilities of this type of MLPs were applied to COPs with real-valued parameter set. This yielded a sufficient condition for such a COP to be solvable by a 3LP, and a set of necessary conditions for such a COP to be solvable by a 2LP. The sufficient condition is satisfied if the cost function of the COP at hand is an affine function of the problem parameter $x$, which denotes the input data of the problem, and if the set of feasible solutions does not depend on the parameter $x$. For the subclass of COPs that satisfy two
Conclusions

additional constraints, we obtained an expression for the minimal number of first-layer nodes that is required by any MLP that solves the COP at hand, and showed that a 3LP exists with this number of first-layer nodes that solves the problem. This expression is based on the Minimal Exact Neighborhood (MEN) of the considered problem, a notion that stems from local search. To this end, we establish some relations between the MEN of a COP and minimal representations of the subsets that are formed when the problem is transformed into an equivalent CCP. Finally, we describe some approaches for determining the MEN of a given COP, which is the first step in the determination of minimal number of first-layer nodes that is required by an MLP that solves the problem, and give a lower and upper bound for this number.

Finally, we applied the general results to the real-valued data version of five well-known COPs, namely sorting, minimum cost spanning tree, shortest network path, shortest network route, and discrete dynamic lotsizing. The obtained results are summarized below.

We showed that for sorting both the MEN and the minimal number of first-layer nodes are polynomial in the number of inputs. Furthermore, there exists a 3LP with minimal sized first layer and a total number of hidden nodes that is polynomial in the number of inputs that solves sorting. Finally, sorting cannot be solved by a 2LP, although there is a variant of sorting, which we denoted relative sorting, that can be solved by a 2LP. Secondly, we proved that for minimum cost spanning tree both the MEN and the minimal number of first-layer nodes are polynomial in the number of inputs. Furthermore, there exists an MLP with minimal sized first layer and a total number of hidden nodes that is polynomial in the number of inputs that solves minimum cost spanning tree.

In contrast to the above two problems shortest network path has a exponentially sized MEN, which implies that the minimal number of first layer nodes, and hence the total number of nodes of any MLP that solves shortest network path is exponential in the number of inputs. The same holds for shortest network route, although this problem has a polynomially sized MEN. The last problem, discrete dynamic lotsizing, is special because it is the only considered problem for which the cost function is not a linear function, but an affine function of the parameter \( z \). This complicates the analysis and as a result we have not been able to determine the size of the MEN for this problem. However, we have shown that the size of the MEN is exponential in the number of inputs, which implies that the minimal number of first layer nodes, and hence the total number of nodes of any MLP that solves discrete dynamic lotsizing is exponential in the number of inputs. Finally, we have shown that this problem cannot be solved by a 2LP.
Bibliography

BAUM, E.B. [1991], Neural net algorithms that learn in polynomial time from examples and queries, IEEE Trans. on Neural Networks 2, 5-19.
BLUM, A.L. AND R.L. RIVEST [1992], Training a 3-node neural network is NP-complete, Neural Networks 5, 117-127.
CHANDRA, A.K., L. STOCKMEYER, AND U. VISHKIN [1984], Constant depth re-


Funahashi, K. [1989], On the approximate realization of continuous mappings by neural networks, Neural Networks 2, 183-192.


Gallant, S.I. [1990], A connectionist learning algorithm with provable generalization and scaling bounds, Neural Networks 3, 191-201.


Hampson, S.E., and D.J. Volper [1986], Linear function neurons: Structure and training, Biological Cybernetics 53, 203-217.


Hecht-Nielsen, R. [1990], Neurocomputing, Addison-Wesley, New York.

BIBLIOGRAPHY


HOPFIELD, J.J. [1984], Neural networks with graded response have collective computational properties like those of two-state neurons, Proc. of the National Academy of Science 81, 3088–3092.


HORN, K., M. STINCHCOMBE, AND H. WHITE [1989], Multilayer feedforward networks are universal approximators, Neural Networks 2, 359–366.


KOSKO, B. [1992], Neural Networks and Fuzzy Systems, Prentice-Hall, Englewood
Cliffs, New Jersey.


Bibliography


Savage, S., P. Weiner, and A. Bagchi [1976], Neighborhood search algorithms for guaranteeing optimal traveling salesman tours must be inefficient, J. of Computer and System Sciences 12, 25-35.
Stork, D.G., and J.D. Allen [1992], How to solve the n-bit parity problem with two hidden units, Neural Networks 5, 923-926.
Valiant, L.G. [1984], A theory of the learnable, Communications of the Association for Computing Machinery 27, 1134-1142.
Wegener, I. [1987], The complexity of boolean circuits, John Wiley & Sons, Chichester.


ZWIETERING, P.J., E.H.L. AARTS, AND J. WESSELS [1991b], **Sorting With A Neural Net**, Memorandum COSOR 91-10, Eindhoven Univ. of Techn.


ZWIETERING, P.J., E.H.L. AARTS, AND J. WESSELS [1992b], The minimal number of layers of a perceptron that sorts, Memorandum COSOR 92-06, Eindhoven Univ. of Techn., To appear in *J. of Parallel and Distributed Computing*.


Samenvatting

In dit proefschrift wordt de complexiteit van het meerlaags perceptron onderzocht, met betrekking tot het oplossen van combinatorische optimaliserings problemen.

Een meerlaags perceptron (MLP) bestaat uit een netwerk van verbonden knopen, geraagd schikt in lagen. Er zijn alleen verbindingen tussen twee opvolgende lagen, en de verbindingen zijn gericht, hetgeen impliceert dat de informatie slechts in één richting langs een verbinding kan gaan. Iedere knoop vormt een reken-eenheid, die een vaste berekening kan uitvoeren; de inputs van een knoop, zijn de waarden op de inkomende verbindingen, worden vermenigvuldigd met gewichten geassocieerd met de verbindingen, en vervolgens opgeteld. De output van de knoop is de uitkomst van een in het algemeen niet-lineaire response functie, toegepast op de verkregen som. De inputs van de knopen in de eerste laag vormen de input van het netwerk, de outputs van de knopen in een bepaalde laag vormen de inputs van de volgende laag, en de outputs van de knopen in de laatste laag vormen de output van het netwerk. De nadruk in dit proefschrift ligt op MLP's waarbij de inputs en de gewichten reëel waardig zijn, de outputs binaire waarden aannemen, en de response functies gelijk zijn aan de zogenaamde hard-limiting response functie, welke 1 oplevert als zijn invoer tenminste 0 bedraagt, en 0 oplevert in alle andere gevallen.

We beschouwen problemen gedefinieerd door middel van een rij functies $f_1, f_2, f_3, \ldots$, waarbij voor elk geheel getal $N$ de functie $f_N$ een domein heeft in $\mathbb{R}^N$ en een bereik in $\mathbb{R}^{K(N)}$, voor zeker geheel getal $K(N)$. Een dergelijk probleem wordt opgelost door een familie van MLP's, als voor iedere $N$ er een MLP in de familie zit dat $N$ inputs en $K(N)$ outputs heeft en $f_N$ berekent. Gegeven een probleem $f_1, f_2, f_3, \ldots$, dan is de vraag of we dit probleem op kunnen lossen met een familie van MLP's, en zo ja, wat de complexiteit is van deze MLP's, bijvoorbeeld met betrekking tot het aantal knopen, het aantal lagen, het aantal inkomende verbindingen per knoop, de grootte van de gewichten, het benodigde type response functie, de gewenste nauwkeurigheid van de berekening van de response functie, etc. De bedoeling van dit proefschrift is om deze vragen te beantwoorden voor combinatorische optimaliserings problemen met reëelwaardige data (COPs), met de beperking dat we alleen kijken naar MLP's met reëelwaardige inputs en hard-limiting response functies. Daarbij concentreren we ons op het benodigde aantal knopen en het benodigde aantal lagen.

Ter inleiding, en om een inschatting te krijgen van de mogelijkheden en moeilijkheden omtrent MLP's, bekijken we allerlei enkele resultaten uit de literatuur die betrekking hebben op problemen waarbij het functie domein een eindige verzameling is. Daarna onderzoeken we de mogelijkheden van een aanpak gebaseerd op de
introductie van een groot aantal complexiteitsklassen. Deze complexiteitsklassen be- vatten de problemen die opgelost kunnen worden met een familie van MLPs van een precies gedefinieerde complexiteit. Door de verschillende complexiteitsklassen met elkaar te vergelijken en te ordenen, ontstaat een indruk omtrent de capaciteiten van de verschillende types van MLPs in het algemeen, en in het bijzonder met betrekking tot een aantal bekende problemen.

Vervolgens worden problemen bekeken waarbij \( f_N \) de gehele \( \mathbb{R}^N \) als domein heeft en zijn bereik gelijk is aan \( B = \{0, 1\} \). Deze problemen noemen we binaire classificatie problemen omdat de input-ruimte \( \mathbb{R}^N \) in twee verzamelingen \( f_N^{-1}(0) \) en \( f_N^{-1}(1) \) gesplitst wordt. Het probleem wordt nu als volgt. Gegeven een deelverzameling \( V \) van \( \mathbb{R}^N \), bestaat er een MLP met \( N \) inputs, één output, en hard-limiting response functions, gereserveerd door de functie \( g : \mathbb{R}^N \rightarrow B \), zodat dat \( g^{-1}(1) \) gelijk is aan \( V \), en zo ja, bepaal de minimale afmetingen van zo'n MLP. Deze vraag wordt gedeeltelijk opgelost. Een nodige en voldoende voorwaarde voor het bestaan van een MLP is dat \( V \) de vereniging is van een eindig aantal pseudo polyeders, welke op hun beurt bestaan uit de doorsnede van een eindig aantal open of gesloten affine halfruimten. Als \( V \) een dergelijke vereniging is, bestaat er een 3-laags perceptron dat het bijbehorende classificatie probleem oplost. Verder formuleren we nodige en voldoende voorwaarden voor het bestaan van een 2-laags perceptron dat \( V \) classificeert. De nodige voorwaarden zijn echter niet gelijk aan de voldoende voorwaarden. De nodige voorwaarden zijn gebaseerd op voorwaarden voor het bestaan van 1-laags perceptrons voor eindige classificatie problemen, en beschouwen lokale geometrische eigenschappen van de onderzochte deelverzameling. De voldoende voorwaarden zijn gebaseerd op constructies bestaande uit verenigingen en doorsneden, en beschouwen daarom globale eigenschappen. Een ander deel van de resultaten voor binaire classificatie problemen is gebaseerd op technieken uit de polyedrische theorie. Ze geeft een ondergrens voor het aantal knopen in de eerste laag van een willkeurig MLP dat \( V \) classificeert. Deze ondergrens wordt bepaald door het aantal verschillende grensvlakken van \( V \), en is scherp als \( V \) een volledig dimensionaal polyeder is.

Vervolgens worden deze resultaten vertaald in resultaten voor het oplossen van COPs. We verkrijgen daarmee nodige en voldoende voorwaarden voor het kunnen oplossen van COPs met families van MLPs, en, mits aan enkele additionele voorwaarden is voldaan, een antwoord op de vragen omtrent het minimale aantal benodigde knopen in de eerste laag en het minimale aantal benodigde lagen. Het minimale aantal knopen in de eerste laag blijkt gerelateerd aan de grootte van de minimale exacte buurruimte voor de met het beschouwde COP corresponderende lokale zoekprobleem. De meeste antwoorden zijn constructief van aard, zo wordt een constructie gegeven van een familie van 3-laags perceptrons met een minimaal aantal knopen in de eerste laag dat het beschouwde COP oplost.

Tenslotte passen we de ontwikkelde theorie toe op een vijftal bekende COPs, sorting, minimum cost spanning tree, shortest network path, shortest network route, en discrete dynamic lotsizing, welke allen met een 3-laags perceptron opgelost kunnen worden. Voor elk van deze problemen bepalen we een expliciete uitdrukking voor de minimale exacte buurruimte. Hiermee kan in principe de grootte van de minimale exacte buurruimte en de minimale grootte van de eerste
Samenvatting

laag van een oplossend MLP bepaald worden. We laten zien dat de minimale exacte buurruimte een polynomial grootte heeft voor SORTING, MINIMUM COST SPANNING TREE, en SHORTEST NETWORK ROUTE, en een exponentiële grootte heeft voor SHORTEST NETWORK PATH en DISCRETE DYNAMIC LOTSIZING. Verder bewijzen we dat minimale grootte van de eerste laag van een oplossend MLP polynomiaal is voor SORTING en MINIMUM COST SPANNING TREE, en exponentieel voor SHORTEST NETWORK PATH, SHORTEST NETWORK ROUTE, en DISCRETE DYNAMIC LOTSIZING. Tevens totoen we aan dat SORTING en DISCRETE DYNAMIC LOTSIZING niet opgelost kunnen worden met een 2-laags perceptron.
Curriculum Vitae

4 november 1962 Geboren te Sittard.
   Eindexamen juni 1981.
   10 februari 1983 Propaedeutisch Examen Wiskunde.
   28 februari 1985 Kandidaats Examen Wiskunde (met lof).
   3 november 1988 Doctoraal Examen Wiskunde.
1988–1993 Assistent in opleiding aan de faculteit Wiskunde en Informatica
   van de Technische Universiteit Eindhoven, bij de vakgroep
   Besliskunde en Stochastiek.
Stellingen

behorende bij het proefschrift

THE COMPLEXITY OF
MULTI-LAYERED PERCEPTRONS

van

Patrick Zwietering
1. Er bestaat een niet-triviale bovengrens op de hoeveelheid parallelisme die in een parallelle Boltzmann machine gebruikt kan worden, waaronder de consensus nog steeds gemaximaliseerd wordt; zie [1].


\[ a_1 \]

\[ a_2 \]

\[ b_1 \]

\[ b_2 \]

\[ p \]

\[ q \]

Figuur 1. Het gecentreerd gebied, exclusief de grenzen, correspondeert met de verzameling behorende bij Stelling 2.

2. Beschouw de open verzameling zoals weergegeven in Figuur 1, voor zekere \( 0 < p, q < \infty \) en \( 0 \leq a_1, a_2, b_1, b_2 \leq \infty \). Met behulp van de resultaten uit [1] kan worden afgeleid dat deze verzameling te classificeren is met een twee-laags perceptron met twee inputs, één output, en \( 7 + 2 \min(K_1, K_2, K_3, K_4) \) knopen in de eerste laag, waarbij

\[
\forall i = 1, \ldots, 4 : K_i = \frac{\ln(1 + a_i/p) - \ln(1 + b_i/q)}{\ln(1 + q/b_i) - \ln(1 + p/a_i)}
\]

en \( a_3 = a_1, a_4 = a_2, b_3 = b_2, b_4 = b_1 \).


3. In Stelling 5.3 van [1] presenteert Parberry een één-laags perceptron met \( N \) inputs, gewichten \( a_i = 2^{(i-1)/2} \), \( i = 1, \ldots, N \), en drempelwaarde \( b = 2^{(N-1)/2} \), voor alle oneven gehele getallen \( N \). De bewering van Parberry is dat \( \max(a_1, \ldots, a_N, b) \) niet kan worden verlaagd zonder het invoer/uitvoer-getrag van dit perceptron op de verzameling \( \{0,1\}^N \) te veranderen. Ofschoon de bewering waar is, is het bewijs gegeven in [1] onjuist, voornamelijk omdat de “zonder verlies van algemeenheid” aannrme niet terecht is.


4. Genetische algoritmen zoals beschreven in [1], kunnen wellicht verbeterd worden door het invoeren van een “verdiefheidscriterium”, waarbij paren worden geselecteerd op grond van bepaalde overeenkomsten tussen de individuen, en, eventueel hiermee gecombineerd, het bijhouden van een stamboom.

5. De bewering van Lin en Vitter in [1] dat OPTIMAL CONSISTENT NET in \( NP \) zit is niet juist, ten eerste omdat de verzameling van leervoorbeelden in het algemeen een verzameling van reële vectoren is, en ten tweede omdat de gewenste bovengrens op de grootte van het neurale netwerk met een binair variabele wordt gereguleerd.


6. De definitie van \( NP \) zoals gegeven in Korst [1] is niet juist, omdat ze voor een gegeven beslissingsprobleem geen uitspraak doet over de "nee" instanties; zie ook [2].


7. Het is bekend dat CONNECTIVITY niet in \( TC^0 \) zit; zie [1]. Hoogstwaarschijnlijk zit CONNECTIVITY zelfs niet in \( TC^0 \); zie [1,2]. Dit maakt het onwaarschijnlijk dat er een twee-laags perceptron bestaat dat een stelling van HEX juist evalueren; zie [3] voor een definitie van het spel HEX, [4] voor definities van \( TC^0 \), \( TC^0 \), en meer-laags perceptrons, en ook [5].


9. Local search en polyhedral technieken kunnen als duale methoden beschouwd worden voor het oplossen van combinatorische optimaliseringsproblemen.

10. In plaats van het voeren van licht, ook overdag, kan de overheid beter het hebben van achter-spatlappen, voor alle auto’s, verplicht stellen.

11. De $N \times N \times N$ versie van de Hongaarse kubus is in theorie oplosbaar in een tijd polynomial in $N$. Hij is echter niet fysiek realiseerbaar als $N$ groter is dan zes.

12. Het verschil tussen promoverende onderzoekers en olympische sporters is de hoogte van de te ontvangen premie.