Diplomarbeit

Approximating Pareto sets in multi-criteria optimization

vorgelegt am 29.September 2010 von

Sebastian Schenker
Matrikel-Nr.: 231895
asbestian@web.de

Betreuung:	Dipl.-Math. Jannik Matuschke
1.Gutachter:	Prof. Dr. Martin Skutella
2.Gutachter:	Prof. Dr. Rolf H. Möhring

Kombinatorische Optimierung & Graphalgorithmen
Institut für Mathematik
Technische Universität Berlin

Die selbständige und eigenhändige Ausfertigung versichert an Eides statt
Berlin, den

Zusammenfassung

Die folgende Diplomarbeit befasst sich mit dem Thema der Approximation von Pareto Mengen in der multikriteriellen Optimierung. Im Allgemeinen gibt es für ein multikriterielles Optimierungsproblem keine zulässige Lösung, deren Zielfunktionswerte alle betrachteten Kriterien gleichzeitig optimiert, da üblicherweise einige der Zielfunktionen in Konflikt zueinander stehen. Eine Pareto optimale Lösung zeichnet sich dadurch aus, dass keiner ihrer Zielfunktionswerte verbessert werden kann, ohne sie in einer anderen Zielfunktion zu verschlechtern. Die Menge dieser Pareto optimalen Punkte ergibt eine sinnvolle Lösungsmenge im Fall von mehreren (zueinander in Konflikt stehenden) Kriterien. Andererseits ist schon bei einfachen Beispielen die Größe der Pareto Menge nicht mehr polynomiell in der Eingabegröße der Probleminstanz. Um in solchen Fällen einen praktikablen Ansatz für die Bestimmung respektive die Charakterisierung der Pareto Menge zu erhalten, werden ϵ-approximative Pareto Mengen betrachtet. Sie zeichnen sich dadurch aus, dass es zu jeder Pareto optimalen Lösung in der originalen Pareto Menge eine Lösung in der ϵ-approximativen Pareto Menge gibt, deren Zielfunktionswert in jedem Kriterium mit einem Faktor von $1 + \epsilon$ multipliziert mindestens so gut wie der Zielfunktionswert der Pareto optimalen Lösung ist. Es stellt sich die Frage, ob diese ϵ-approximativen Pareto Mengen für multikriterielle Optimierungsprobleme generell existieren und inwiefern sie effizient berechnet werden können.

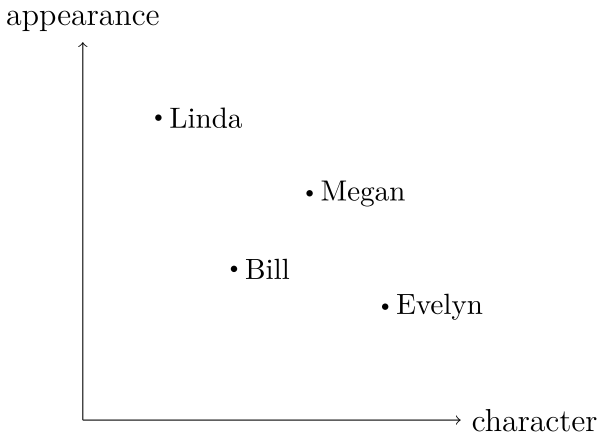
Contents

1 Introduction

1.1 Motivation and overview of the topic

Multi-criteria optimization, also known as multi-objective optimization, is the process of simultaneously optimizing multiple objectives subject to certain constraints. It is a generalization of the well-known case of single-objective optimization and a more realistic model of real-world applications. Often, the most important goal is to minimize costs or to maximize profits, respectively. On the other hand, in many cases it does not take much effort to find other objectives that play also an important role in the examined optimization task like minimization of journey/transportation times or optimizing structural properties like the reliability of an underlying (transportation) network. For instance, passengers using a certain transportation system are interested in a fast, save, and reliable transportation system. However, fast and save or fast and reliable, respectively, are objectives which are in most cases in conflict with each other. Another simple, workaday example is the task of buying groceries. In order to avoid starving one has to buy a certain amount of food. Of course, one usually wants to pay as little as possible over the counter. On the other hand, one likes to eat deliciously and healthy, and good food is probably more expensive than the cheapest available. Apparently, it is not hard to imagine that even trivial problems tend to involve several criteria. In general, models of real-world problems aim at maximizing (or minimizing, respectively) several conflicting objectives and hence do not contain a unique optimal feasible solution that optimizes all criteria at the same time. Instead, we have to deal with trade-offs. In general, improving one objective will lead to a worsening in other criteria and vice versa. Therefore, in multi-criteria optimization a decision maker will be interested in a set of solutions that reflects those inherent compromises. A feasible solution that can be improved in at least one criterion is naturally not of interest for a decision maker. However, a so-called *Pareto optimal* solution that cannot be improved in one criterion without being worsened in other criteria will be part of a reasonable solution set. The finally chosen solution, that is, the finally made decision is left to the decision maker. The mathematical interest lies in the characterization and computation of the so-called *Pareto set* - the set of all Pareto optimal solutions. Unfortunately, the size of this set will generally be exponential in the size of the input. In such cases there is, in general, no way (known) to construct efficiently running algorithms that compute and output the whole Pareto set. Hence, the concept of approximation is needed and introduced. Provided that one cannot afford to compute the whole set of Pareto optimal solutions, a reasonable approximation of the latter would be a satisfying result. Consider an approximation algorithm that runs efficiently and outputs a solution set S with the property that for every Pareto optimal solution x^* we can find a solution $\bar{x}$ in S that is within a small factor in all objectives of x^*. Then S would be a reasonable answer to a decision maker. Two questions will naturally arise. Does such an *approximate Pareto*

Figure 1.1: A simple example of a bicriteria maximization problem. Imagine that our decision maker's aim is to find his significant other. For the sake of simplicity let us assume that appearance and character are the two considered objectives. Clearly, Bill will not be a potential candidate because Megan is more beautiful and has a better character. Hence, Bill is dominated by Megan. On the other hand, the decision between Linda, Megan, and Evelyn will imply trade-offs because an improvement in beauty will imply a decline in character and vice versa.



set exist in general? Secondly, can it be efficiently computed? In the following, the first question will be answered in the affirmative whereas the second question needs further restrictions and more structure in the underlying instances for adequate answers.

1.2 Chapter outline and contributions

Aside from the introductory chapter, this thesis consists of five main chapters. Chapter 2 formally introduces the definition of Pareto optimality which is fundamental for this thesis. Moreover, basic definitions of complexity theory are recapitulated. Section 3.1 motivates the need for approximation and formally defines ϵ-approximate Pareto sets which also play a central role in this thesis. In the remainder of Chapter 3 two more notions of Pareto optimality are defined which will again occur in Chapter 4. For the definitions in Section 3.2 and in Section 3.3, we mainly draw on the book "Multicriteria Optimization" by M.Ehrgott [1]. Chapter 4 is concerned with Pareto optimal solutions and weighted sum scalarization. The results in this chapter are again mainly based on [1]. Ehrgott considers minimization problems whereas we consider by default maximization problems. However, the proofs are either equivalent or very similar. Section 4.1 gives a little more intuition about Pareto optimal solutions and states some auxiliary results. After introducing a generalizing concept of convexity in Section 4.2, Section 4.3 introduces the method of weighted sum scalarization and states results about its relationship to the Pareto set. Section 4.4 is concerned with the proper Pareto set and its relationship to weighted sum scalarization. Section 4.5 concludes Chapter 4 with a nice result about multi-criteria linear programming. Chapter 5 deals with [2]. Therein, Papadimitriou and Yannakakis stated interesting results with proof sketches about the approximability of Pareto sets. One of their main results considers a method of bounded weighted sum scalarization and its relationship to ϵ-approximate Pareto sets. After stating two basic complexity results in Section 5.1, we treat Yannakakis' and Papadimitriou's results and proofs in more detail in Section 5.2 and in Section 5.3. Chapter 6 is concerned with a framework by G.J.Woeginger guaranteeing the existence of an FPTAS (see [3]). We show that this framework is not restricted to single-objective optimization problems but also holds for the case of multiple objectives. Finally, the thesis concludes with a brief summary and a short outlook.

2 Basic definitions

2.1 Preliminaries

Throughout this thesis we will assume that the reader is familiar with mathematical methodology. Fundamental concepts of linear algebra and combinatorial optimization will not be covered in the following. However, necessary definitions and basic results will be introduced or recapitulated for the sake of self-containment.

Multi-criteria optimization problems can generally be stated as

$$\min / \max(f_1(x), \ldots, f_k(x))$$
$$\text{subject to } x \in \mathcal{X},$$

where $k \in \mathbb{N}$ is the number of objectives, $\mathcal{X} \subseteq \mathbb{R}^n$ is the feasible domain and $f_i : \mathcal{X} \to \mathbb{R}$ $(i = 1, \ldots, k)$ are the considered objective functions. In this context, the feasible domain $\mathcal{X}$ is in the so-called decision space whereas the image under $f(x) \coloneqq (f_1(x), \ldots, f_k(x))$ is in the criterion space . If $\mathcal{X}$ is a discrete set or a convex set, respectively, then we call the optimization problem a discrete optimization problem or a convex optimization problem, respectively. Furthermore, of (our) particular interest is the case in which the objective functions are linear functions. In general, in the multi-criteria case there will be no feasible solution $x \in \mathcal{X}$ that optimizes all objectives at the same time, since most of the objectives will usually be in conflict with each other. Thus, we have to state more clearly what we mean by optimality in the multi-criteria case. A convenient way and widely used scheme to deal with inherent trade-offs between objective values is the notion of Pareto optimality.

Definition 2.1 A feasible solution $x^* \in \mathcal{X}$ is called *Pareto optimal* if (in the case of maximization) there exists no $x \in \mathcal{X}$ such that $f(x) \succ f(x^*)$, that is, $f_i(x) \geq f_i(x^*)$ for $i = 1, \ldots, k$ and $f_j(x) > f_j(x^*)$ for at least one $j \in \{1, \ldots, k\}$. The set of Pareto optimal solutions is the *Pareto set* denoted by $\mathcal{X}_P$.

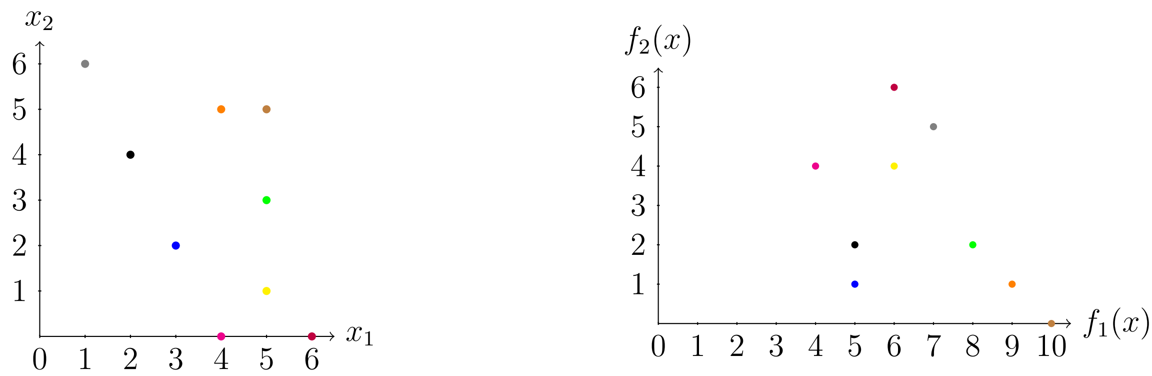
In the case of minimization the definition is similar. A feasible solution $x^* \in \mathcal{X}$ is called Pareto optimal if there is no $x \in \mathcal{X}$ such that $f(x) \prec f(x^*)$. Throughout this thesis we will assume without loss of generality that our optimization problems are maximization problems, that is, we want to maximize the objective functions $f_1, \ldots, f_k$. This is no restriction because a solution $x \in \mathcal{X}$ that maximizes a certain objective f_i will also minimize the negative of f_i. Note that forthcoming diagrams and examples will by default consider the case of maximization (except in proofs and in Example 3.1 in Section 3.1).

If $x^* \in \mathcal{X}$ is Pareto optimal, then the image $f(x^*)$ is called *non-dominated*. Equivalently, if $x \in \mathcal{X}$ is not a Pareto optimal solution, that is, there exists $\bar{x} \in \mathcal{X}$ such that $f(x) \prec f(\bar{x})$, then the image $f(x)$ is called *dominated*. Let $\mathcal{Y} = \{(f_1(x), \dots, f_k(x)) : x \in \mathcal{X}\}$ be the image of the domain $\mathcal{X}$ under the given objective function mappings $f_1, \dots, f_k$. We will denote the set of non-dominated points in criterion space by $\mathcal{Y}_N = \{y \in \mathcal{Y} :$ there exists no $\bar{y} \in \mathcal{Y}$ such that $y \prec \bar{y}\}$. Clearly, $\mathcal{Y}_N \subseteq \mathcal{Y}$.

Unfortunately, the notation of Pareto optimality and non-domination is not consistent in the multi-criteria literature. For instance, Ehrgott (see [1]) uses the term efficient instead of Pareto optimal and remarks that non-dominated points are also called non-inferior points or maximal/minimal points by some authors. Throughout this thesis we will use the term Pareto optimal referring to solutions in the decision space whereas images of Pareto optimal solutions will be called non-dominated referring to points in the criterion space.

Example 2.1 Let $\mathcal{X} = \{\binom{1}{6}, \binom{2}{4}, \binom{3}{2}, \binom{4}{0}, \binom{4}{5}, \binom{5}{1}, \binom{5}{3}, \binom{5}{5}, \binom{6}{0}\} \subset \mathbb{N}^2$ be the feasible domain and consider two objective functions $f_1(x_1, x_2) = x_1 + x_2$ and $f_2(x_1, x_2) = |x_1 - x_2|$ which are to be maximized. See Figure 2.1. For the sake of a better consideration, observe that each preimage and corresponding image are colored the same. The Pareto set is $\mathcal{X}_P = \{\binom{1}{6}, \binom{4}{5}, \binom{5}{3}, \binom{5}{5}, \binom{6}{0}\}$ and the set of non-dominated points is $\mathcal{Y}_N = \{\binom{7}{5}, \binom{9}{1}, \binom{8}{2}, \binom{10}{0}, \binom{6}{6}\}$, respectively. The feasible solution $\binom{5}{1}$, for instance, is not Pareto optimal, since its image $f(5,1) = \binom{6}{4}$ is dominated by $f(6,0) = \binom{6}{6}$.

Figure 2.1: On the left the feasible domain $\mathcal{X}$ of Example 2.1 and $\mathcal{Y}$ on the right.



Remark (The historical and literal origin of Pareto optimality)
The term of Pareto optimality goes back to and is named in honour of Vilfredo Federico Pareto (1848 – 1923) who was an Italian sociologist, philosopher and economist. He grew up in a middle-class environment, earning a degree in mathematics in 1867 and a doctorate in engineering from the Polytechnic University of Turin in 1870. Besides working as a civil engineer for some years, he was appointed the chair of political economy at the University of Lausanne in Switzerland in 1893. The title of his dissertation was "The Fundamental Principles of Equilibrium in Solid Bodies", marking the beginning of his interest in equilibrium analysis in economics and sociology. In the early years of his career he distinguished himself as a liberal criticizing the Italian government, promoting real democracy and supporting free trade and competition. Later he became more a socialist, making in 1906 the observation that (in Italy) twenty percent of the population owns eighty percent of the property. Besides the concept of Pareto optimality meaning an allocation of resources in which it is not possible to improve anyone without worsening someone else

off, his second achievement is the above mentioned concept of income distribution also known as the Pareto principle.

The following proposition will be of use in Section 5.2 and in Subsection 6.3.1, respectively.

Proposition 2.2
For any $x \geq 1$*, it holds that* $\ln(x) \geq (x-1)/x$.

Proof: The functions $\ln(x)$ and $f(x) \coloneqq (x-1)/x$ restricted to the domain $[1, \infty)$ are both continuous functions with $\ln(1) = f(1)$. Furthermore, it holds that the derivative $\ln'(x) = 1/x$ is greater than or equal to the derivative $f'(x) = 1/x^2$ on the interval $[1, \infty)$. □

The next definition is rather meant as a recapitulation for the reader than a strict formalization. Whenever we deal with algorithms we are interested in their complexity, in particular, their running times.

Definition 2.3 A *polynomial time algorithm* is an algorithm that terminates after a number of steps bounded by a polynomial in the size of the input. A step consists of performing one instruction. A problem is said to be *solvable in polynomial time* or *tractable*, respectively, if it can be solved by a polynomial time algorithm.

The size of the input of an instance I will be denoted by $|I|$. We also remind the reader of the asymptotic notation concerning the growth of functions.

Definition 2.4 For a given function $g(n)$, we denote by $\Theta(g(n))$ the set of functions $\Theta(g(n)) = \{f(n) : \exists \text{ positive constants } c_1, c_2 \text{ and } n_0 \text{ such that } 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \text{ for all } n \geq n_0\}$ and by $O(g(n))$ the set of functions
$O(g(n)) = \{f(n) : \exists \text{ positive constants } c \text{ and } n_0 \text{ s.t. } 0 \leq f(n) \leq c g(n) \text{ for all } n \geq n_0\}$.

Notice that $f(n) \in \Theta(g(n))$ implies $f(n) \in O(g(n))$, since Θ-notation is a stronger notion than O-notation.

2.2 Basic complexity theoretical definitions

We assume that the reader is familiar with the fundamentals of complexity theory. For the sake of completeness, we will sketchily recapitulate basic definitions. For a more thorough and detailed description we refer the reader to [4].

A decision problem is a problem that can be answered by “yes” or “no”. The class P contains those decision problems that can be solved in time $O(n^k)$ for some constant k, where n is the size of the input. The class NP consists of those decision problems that can be verified in polynomial time. In other words, given a certificate of a solution, we can verify the correctness of the certificate in time polynomial in the size of the input. Clearly, any problem in P is also in NP. An NP-complete problem is a problem that is contained in NP and that is as *hard* as any problem in NP. A problem that is as hard as any problem in NP but not necessarily contained in it is called NP-hard. Hardness refers in this context to the notion of reducibility. A reduction is a procedure that transforms any instance α of a problem A into some instance β of a problem B with the properties that the transformation takes polynomial time and that the answers of the problems coincide, that is, the answer for α is "yes" if and only if the answer for β is "yes". If every problem in NP can be reduced to a particular problem B, then B is NP-hard or NP-complete, respectively. In 1972 Richard M. Karp published his famous paper "Reducibility among combinatorial problems" presenting 21 NP-complete problems (see [5]). Over the intervening years the list of NP-complete problems grew enormously and continues to grow. The following two problems belong to the original list of Karp and will be encountered later on.

Definition 2.5 By the Partition problem we mean the following: Given n natural numbers $a_1, \ldots, a_n$ with $\sum_{i=1}^{n} a_i = 2b$. Does there exist a subset $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in S} a_i = b$?

For the NP-completeness proof we refer the reader to ([5] or [6]).

Another important problem in combinatorial optimization is the Knapsack problem.

Definition 2.6 The Knapsack problem is described as follows: Given non-negative numbers p_i,w_i $(i = 1, \ldots, n)$, and a weight bound W. Find a subset $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in S} w_i \leq W$ and $\sum_{i \in S} p_i$ is maximum. The decision version of Knapsack asks whether there exists a subset $S \subseteq \{1, \ldots, n\}$ fulfilling $\sum_{i \in S} w_i \leq W$ and $\sum_{i \in S} p_i \geq P$ for some given P.

Lemma 2.7
The decision version of Knapsack *belongs to the class of NP-complete problems.*

Proof: Given an instance of Knapsack, consider the subset $S \subseteq \{1, \ldots, n\}$ as a certificate. A verification algorithm inspects whether $\sum_{i \in S} w_i \leq W$ and $\sum_{i \in S} p_i \geq P$ holds. This can be done in polynomial time. Hence, Knapsack is in NP. Secondly, we use the idea of polynomial time reducibility in order to show that Knapsack is NP-complete. Consider an instance of Partition: Given n integers $a_1, \ldots, a_n$ with $\sum_{i=1}^{n} a_i = 2b$. Is there a subset $S \subseteq \{1, \ldots, n\}$ with $\sum_{i \in S} a_i = b$? Construct the following instance of Knapsack: Let $p_i \coloneqq w_i \coloneqq a_i$ for $i = 1, \ldots, n$. Furthermore, let $W \coloneqq P \coloneqq b$. Then there exists a subset $S \subseteq \{1, \ldots, n\}$ fulfilling $\sum_{i \in S} w_i \leq W$ and $\sum_{i \in S} p_i \geq P$ if and only if there is a subset $S \subseteq \{1, \ldots, n\}$ with $\sum_{i \in S} a_i = b$. □

In combinatorial optimization we often have to cope with optimization tasks. In other words, we are interested in minimizing or maximizing certain objectives. In this sense, each feasible solution has an associated objective value and we want to find a feasible solution with the optimum value. Fortunately, there is an apparent connection between optimization problems and decision problems. We can transform an optimization problem into a related decision problem by imposing a bound on the value to be optimized. For instance, consider Knapsack. Instead of asking for a subset of items fulfilling the weight limit while maximizing the sum over the p_i values, we can ask whether there exists a subset of items which does not violate the weight limit and achieves at least a certain value with respect to p_i. In this sense, the optimization problem is at least as hard as its related decision problem.

3 Approximate Pareto sets and further notions of Pareto optimality

3.1 Approximation and approximate Pareto sets

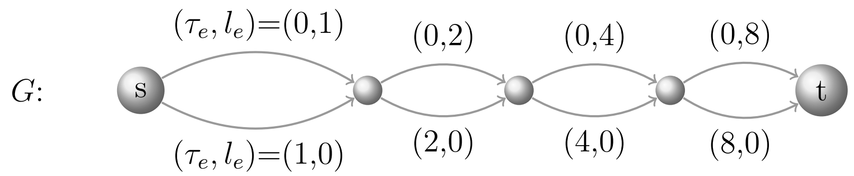
Besides the algorithmic task of an efficient computation of the problem solutions, we will encounter the influential factor that the size of the Pareto set is generally exponential in the size of the given input. Hence, an algorithm which (computes and) outputs an exponentially large solution set cannot be expected to run in polynomial time in the given input.

Example 3.1 Given a directed graph $G = (V, E)$ with two edge labels l and τ. Let $\mathcal{P}_{st}$ be the set of all s-t-paths in G. Consider the two objectives:

$$\min \sum_{e \in P} l_e \quad , \quad \min \sum_{e \in P} \tau_e$$

$$\text{subject to } P \in \mathcal{P}_{st}.$$

Furthermore, consider the following instance:



In this type of instances every s-t-path $P \in \mathcal{P}_{st}$ is Pareto optimal. Hence, the size of the solution set is exponential in the given input. In this very example the set of Pareto optimal s-t-paths contains the following tuples: (0,15),(1,14),(2,13),(3,12),(4,11),(5,10),(6,9), (7,8),(8,7),(9,6),(10,5),(11,4), (12,3),(13,2),(14,1),(15,0).

The increase of the running time of an algorithm that does not run in polynomial time is even for moderately growing instances not acceptable. The following Table 3.1 is taken from [7].

Example 3.2 Consider a problem and assume that five algorithms A_1 through A_5 exist to solve it whose complexity are illustrated in Table 3.1. Assume also that the machine running them requires 10^{-9} seconds to execute a single step. The execution times of the five algorithms are represented in Table 3.1 in terms of increasing instance sizes.

Instance size n	complexity of algorithms				
	A_1/n^2	A_2/n^3	A_3/n^5	$A_4/2^n$	$A_5/3^n$
10	0.1 μs	1 μs	0.01 ms	1 μs	59 μs
30	0.9 μs	27 μs	24.3 ms	1 s	2.4 days
50	2.5 μs	0.125 ms	0.31 s	13 days	2.3×10^5 centuries

Table 3.1: An example of polynomial and exponential times versus instance sizes

In other words, an algorithm whose running time is not bounded by a polynomial can only be applied to small instances in practice. Problems for which no polynomial time algorithm exists are regarded to be *intractable*. Notice that even faster computers and technological progress will not have a huge effect on intractable problems. Let us assume that we have a computer which is several orders of magnitude faster than the previous one. Instead of 10^9 operational steps per second, we assume our faster computer to do 10^{12} operational steps per second. If we set our computational time limit to one hour, then the largest instance that could be solved with an n^2-algorithm within this hour is approximately $n = 60000000$ ($n = \sqrt{3600 \cdot 10^{12}}$). In comparision, the largest instance computable in one hour with the "slower" computer is $n \approx 1897367$. On the other hand, if we have to deal with an algorithm running in superpolynomial time (e.g. 2^n), we cannot expect such an increase in the size of the largest instance, unfortunately. The largest instance computable in one hour with the "faster" computer is then only $n \approx 52$ ($n \approx \text{ld}(3600 \cdot 10^{12})$) whereas the largest instance on the "slower" computer was $n \approx 42$. Hence, even with an increase in the computing speed only a fractional amount is added to the size of a largest instance of an intractable problem solvable in acceptable time. Intractability is therefore somewhat independent of technological progress which makes it much more desirable to cope with polynomial time algorithms.

However, notice that from a practical point of view a polynomial time algorithm whose running time is bounded by a polynomial of high degree (e.g. n^{10}, n^{15}) cannot be regarded as practically relevant either.

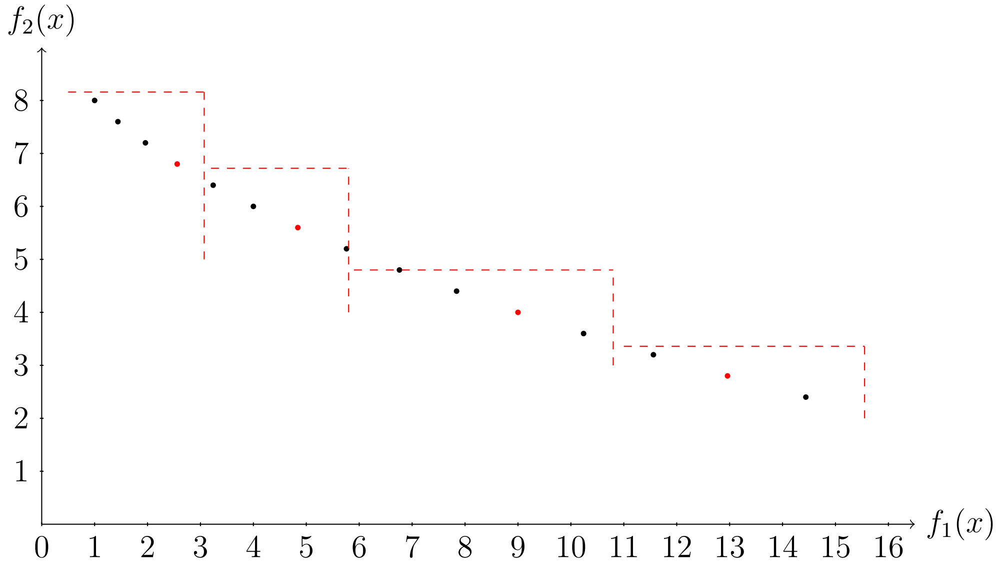
Due to the above mentioned computational difficulties the notion of approximation needs to come into play. The term approximation has two possible meanings in our context. On the one hand, consider the case in which it is *hard* to compute an optimal solution but the size of the optimal solution set does not contribute to the complexity of the problem. Then we understand by an approximation algorithm a polynomial time algorithm that computes a feasible solution with an a priori guarantee on the quality of the computed solution. On the other hand, consider the case in which the size of the Pareto set contributes to the complexity of the problem. Perhaps it is *easy* to compute a single Pareto optimal solution but it takes time which is exponential in the input to enumerate or characterize the whole set of Pareto optimal solutions. Then the term approximation is more related to the notion of covering. By choosing polynomially many representatives that *cover* certain Pareto optimal solutions, one could try to approximate and characterize the whole Pareto set. We will now introduce the notion of an ϵ-approximate Pareto set.

Definition 3.1 Let $\mathcal{X}_P$ be the Pareto set of a multi-criteria optimization problem with domain $\mathcal{X}$ and with objectives $f_1, \ldots, f_k$, $k \in \mathbb{N}$. Furthermore, let $\epsilon > 0$. $\mathcal{X}_{\epsilon P} \subseteq \mathcal{X}$ is an *ϵ-approximate Pareto set* of $\mathcal{X}_P$ if for all $x \in \mathcal{X}_P$ there exists $\bar{x} \in \mathcal{X}_{\epsilon P}$ such that (in the case of maximization)

$$f_i(x) \leq (1+\epsilon) f_i(\bar{x}) \text{ for } i = 1, \ldots, k. \tag{3.1}$$

In other words, for every Pareto optimal solution x we will find a solution $\bar{x}$ in the ϵ-approximate Pareto set that almost dominates x, that is, it is within a factor of $1+\epsilon$ in all objectives. This is a rather attractive notion of an approximation for a decision maker who is interested in the Pareto set.
If (3.1) holds for two solutions $x \in \mathcal{X}_P$ and $\bar{x} \in \mathcal{X}_{\epsilon P}$, we will say that $\bar{x}$ *ϵ-approximates* x.

Figure 3.1: The criterion space of Example 3.3. The preimages of the red colored points constitute an ϵ-approximate Pareto set for $\epsilon = 0.2$.



Example 3.3 Let $\mathcal{X} := \{1, 1.2, 1.4, 1.6, \ldots, 3.8\}$ be the feasible domain. Furthermore, let $f_1(x) := x^2$ and $f_2(x) := 10 - 2x$ and assume that we want to maximize both objectives. Clearly, the two objectives f_1 and f_2 are in conflict with each other. An increase in f_1 results in a decrease in f_2 and vice versa. Moreover, $\mathcal{X}_P$ coincides with $\mathcal{X}$. For $\epsilon = 0.2$, $\mathcal{X}_{\epsilon P} = \{1.6, 2.2, 3.0, 3.6\}$ is an ϵ-approximate Pareto set for $\mathcal{X}_P$. It holds that $1.2 \cdot f_1(1.6) = 3.072$ and $1.2 \cdot f_2(1.6) = 8.16$. Hence, the feasible solution 1.6 ϵ-approximates the Pareto optimal solutions $1.0, 1.2, 1.4$ and itself. Furthermore, $1.2 \cdot f_1(2.2) = 5.808$ and $1.2 \cdot f_2(2.2) = 6.72$ ϵ-approximating $1.8, 2.0$, itself and 2.4. The feasible solution 3.0 ϵ-approximates the feasible solutions $2.6, 2.8$, itself and 3.2, since $1.2 \cdot f_1(3.0) = 10.8$ and $1.2 \cdot f_2(3.0) = 4.8$. The feasible solutions 3.4 and 3.8 are ϵ-approximated by 3.6. Therefore, $\mathcal{X}_{\epsilon P}$ *covers* all solutions in $\mathcal{X}_P$.

Notice that Definition 3.1 does not require properties related to the size of $\mathcal{X}_{\epsilon P}$ and hence does not exclude the possibility that there exist several sets fulfilling condition (3.1). For instance, the set $\{1.4, 2.0, 2.4, 2.8, 3.2, 3.8\}$ is also an ϵ-approximate Pareto set

for Example 3.3. Furthermore, Definition 3.1 does not forbid redundant or unnecassary solutions. Clearly, it would be very satisfying if we knew that the computed approximate Pareto set is minimal with respect to the cardinality. However, since it will often be the case that the size of the Pareto set is exponential in the input, we will be satisfied with an approximate Pareto set that is polynomial in the input. Since we can get rid of redundant and unnecessary solutions in polynomial time (in the worst case by comparing all approximate solutions among each other) if the approximate Pareto set $\mathcal{X}_{\epsilon P}$ has a polynomial size, we will not care about unnecessary solutions in $\mathcal{X}_{\epsilon P}$ in the following.
Moreover, notice that Definition 3.1 does not require that $\mathcal{X}_{\epsilon P} \subseteq \mathcal{X}_P$. Clearly, if there exists a non-Pareto optimal solution $\bar{x}$ ϵ-approximating certain Pareto optimal solutions $x^1, \ldots, x^m$, then there also exists a Pareto optimal solution x^* which ϵ-approximates $x^1, \ldots, x^m$. On the other hand, it could be the case that finding a Pareto optimal solution x^* which ϵ-approximates a certain subset of the Pareto set involves much more complexity than finding a non-Pareto optimal approximate solution $\bar{x}$.

We will now define what we mean by an FPTAS in the case of multi-criteria optimization. If we cannot achieve to compute optimal or Pareto optimal solutions, respectively, in acceptable time, then we will be satisfied with algorithms that output results of bounded error but guarantee to have a running time polynomially in the input size and $\frac{1}{\epsilon}$.

Definition 3.2 Let P be a multi-criteria optimization problem. An *FPTAS* for P is a family of algorithms that outputs an ϵ-approximate Pareto set for any $\epsilon > 0$ and that has a running time which is polynomial both in the size of the input and in $1/\epsilon$.

Remark
Notice that the single-objective case is a special case of Definition 3.2. By an FPTAS in the single-objective case we mean a family of algorithms that runs in time which is polynomial in the input size and in $1/\epsilon$ and that finds a feasible solution achieving a value of at least $\frac{1}{1+\epsilon} \cdot f(x_{opt})$ where $f(x_{opt})$ is the optimal solution value. For $k = 1$, the Pareto set $\mathcal{X}_P$ can be regarded as the set containing an optimal solution x_{opt}. The corresponding ϵ-approximate Pareto set contains then a solution $\bar{x}$ such that $f(x_{opt}) \leq (1 + \epsilon) f(\bar{x})$.

3.2 Weak Pareto optimality

We will now introduce the notion of weak Pareto optimality which captures the set of solutions that cannot be improved in all objectives. We will encounter weakly Pareto optimal solutions in Section 4.3 where we characterize the solutions found by the method of weighted sum scalarization.

Definition 3.3 A feasible solution $x^* \in \mathcal{X}$ is called *weakly Pareto optimal* if there exists no $x \in \mathcal{X}$ such that $f_i(x) > f_i(x^*)$ for $i = 1, \ldots, k$. The point $y^* = f(x^*)$ is then called *weakly non-dominated*.

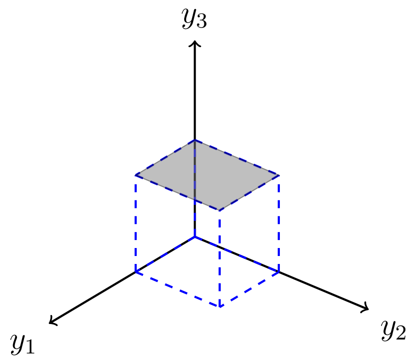
We will denote the set of weakly Pareto optimal feasible solutions by $\mathcal{X}_{wP}$. The following inclusion clearly follows from Definition 3.3: $\mathcal{X}_P \subseteq \mathcal{X}_{wP}$.

The corresponding set of weakly non-dominated points in criterion space is denoted by $\mathcal{Y}_{wN} = \{(y_1, \ldots, y_k) \in \mathcal{Y} :$ there exists no $\bar{y} \in \mathcal{Y}$ such that $y_i < \bar{y}_i$ for $i = 1, \ldots, k\}$.

In the following, we will often consider $\mathcal{Y}$ instead $\mathcal{X}$. It is reasonable to derive results on the set of non-dominated points $\mathcal{Y}_N$ and then use properties of the objective functions to make conclusions about the Pareto set $\mathcal{X}_P$. In a sense, we identify the decision space with the criterion space and $\mathcal{X}_P$ with $\mathcal{Y}_N$, respectively. In other words, we will often be satisfied with stating results on the set of non-dominated points $\mathcal{Y}_N$ instead of the Pareto set $\mathcal{X}_P$.

Example 3.4 Consider the set $\mathcal{Y} = \{(y_1, y_2, y_3) \in \mathbb{R}^3 : 0 < y_1 < 1, 0 < y_2 < 1, 0 \leq y_3 \leq 1\}$. Then $\mathcal{Y}_N = \emptyset$ because y_1 and y_2 live in the open interval $(0, 1)$. On the other hand, $\mathcal{Y}_{wN} = \{y \in \mathcal{Y} : 0 < y_1 < 1, 0 < y_2 < 1, y_3 = 1\}$ because y_3 lives in the closed interval $[0, 1]$ and there exists no point in $\mathcal{Y}$ that has coordinates strictly greater than the points in $\mathcal{Y}_{wN}$.

Figure 3.2: The set of weakly non-dominated points is indicated by the shaded area.



Example 3.5 Consider the set $\mathcal{Y} = \{(y_1, y_2, y_3) \in \mathbb{R}^3 : 0 \leq y_1 \leq 1, 0 \leq y_2 \leq 1, 0 \leq y_3 \leq 1\}$. Then $\mathcal{Y}_N = \{(1, 1, 1)\}$ and the set of weakly non-dominated points is the union of all points located on the faces of the cube $\mathcal{Y}$, that is, $\mathcal{Y}_{wN} = \{(y_1, y_2, y_3) \in \mathcal{Y} : y_1 = 1 \vee y_2 = 1 \vee y_3 = 1\}$.

3.3 Proper Pareto optimality

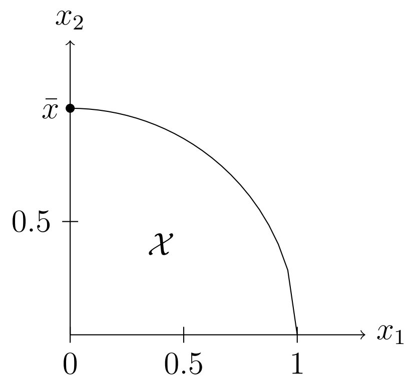
Assume that there are two objective functions f_i and f_j which are in conflict with each other. For instance, an increase in f_i leads to a decrease in f_j. These trade-offs among criteria can be measured by computing the increase in objective f_i per unit decrease in objective f_j. In some situations such relative trade-offs can be unbounded and lead to the definition of proper Pareto optimality.

Definition 3.4 A Pareto optimal solution $x^* \in \mathcal{X}_P$ is called *properly Pareto optimal* if there exists a real number $M > 0$ such that for all $i \in \{1, \ldots, k\}$ and for all $x \in \mathcal{X}$ satisfying $f_i(x) > f_i(x^*)$ there is an index $j \in \{1, \ldots, k\}$ satisfying $f_j(x^*) > f_j(x)$ such that

$$\frac{f_i(x) - f_i(x^*)}{f_j(x^*) - f_j(x)} \leq M. \tag{3.2}$$

The corresponding image $y^* = f(x^*)$ is called *properly non-dominated*. We denote the set of properly Pareto optimal solutions and the set of properly non-dominated points by $\mathcal{X}_{pP}$ and $\mathcal{Y}_{pN}$, respectively.

Figure 3.3: The decision space of Example 3.6.



Example 3.6 Consider the feasible domain $\mathcal{X} := \{(x_1, x_2) \in \mathbb{R}^2 : x_1^2 + x_2^2 \leq 1, 0 \leq x_1, x_2 \leq 1\}$ and assume that we want to maximize $f_1(x_1, x_2) = x_1$ and $f_2(x_1, x_2) = x_2$, respectively. Hence, $\mathcal{Y}$ coincides with $\mathcal{X}$. It holds that $\mathcal{Y}_N = \{(y_1, y_2) \in \mathcal{Y} : y_1^2 + y_2^2 = 1\}$. We show that $\bar{x} = (0, 1) \in \mathcal{X}_P$ is not a properly Pareto optimal solution. In other words, we show that for all $M > 0$ there exists an index $i \in \{1, 2\}$ and some $x \in \mathcal{X}$ with $f_i(x) > f_i(\bar{x})$ such that

$$\frac{f_i(x) - f_i(\bar{x})}{f_j(\bar{x}) - f_j(x)} > M$$

for all $j \in \{1, 2\}$ with $f_j(\bar{x}) > f_j(x)$.
Let $i = 1$ and choose $x^\epsilon := (\epsilon, \sqrt{1 - \epsilon^2})$, $0 < \epsilon < 1$. It holds that $x^\epsilon \in \mathcal{X}_P$, since $\epsilon^2 + \sqrt{1 - \epsilon^2}^2 = 1$. Moreover, since x^ϵ with $x_1^\epsilon > \bar{x}_1$ and $x_2^\epsilon < \bar{x}_2$, it follows that

$$\frac{f_1(x^\epsilon) - f_1(\bar{x})}{f_2(\bar{x}) - f_2(x^\epsilon)} = \frac{\epsilon - 0}{1 - \sqrt{1 - \epsilon^2}} \overset{\epsilon \to 0}{\longrightarrow} \infty,$$

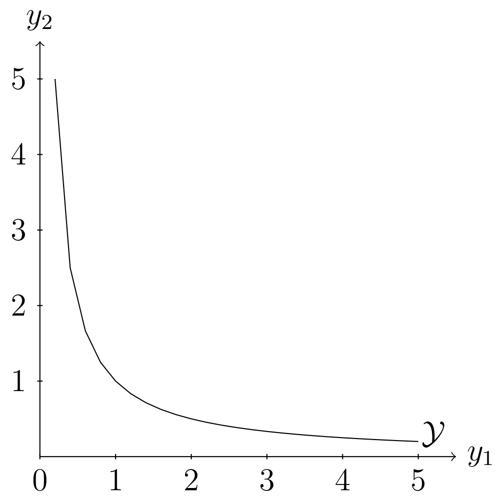
since $\lim_{\epsilon \to 0} \frac{\epsilon}{1 - \sqrt{1 - \epsilon^2}} \overset{L'H\hat{o}pital}{=} \lim_{\epsilon \to 0} \frac{(1 - \epsilon^2)^{\frac{1}{2}}}{\epsilon} = \infty$.

Example 3.7 Consider $\mathcal{Y} = \{(y_1, y_2) \in \mathbb{R}^2 : y_1 > 0, y_2 = 1/y_1\}$. (See Figure 3.4.) Then it holds that $\mathcal{Y}_N = \mathcal{Y}$. In order to see that $\mathcal{Y}_{pN} = \emptyset$, let $\bar{y} = (\bar{y}_1, 1/\bar{y}_1) \in \mathcal{Y}_N$ be arbitrary but fixed. Let $y^k \in \mathcal{Y}_N$ be a sequence with $y_1^k < \bar{y}_1$ and $y_1^k \to 0$. It holds that $y_2^k > \bar{y}_2$ for all k. Then it follows that

$$\lim_{k \to \infty} \frac{y_2^k - \bar{y}_2}{\bar{y}_1 - y_1^k} = \lim_{k \to \infty} \frac{\frac{1}{y_1^k} - \frac{1}{\bar{y}_1}}{\bar{y}_1 - y_1^k} = \lim_{k \to \infty} \left(\frac{1}{y_1^k \left(\bar{y}_1 - y_1^k\right)} - \frac{1}{\bar{y}_1 \left(\bar{y}_1 - y_1^k\right)} \right) \overset{y_1^k \to 0}{=} \infty .$$

Hence, we can conclude that in this example no feasible Pareto optimal solution is properly Pareto optimal.

Figure 3.4: The criterion space of Example 3.7.

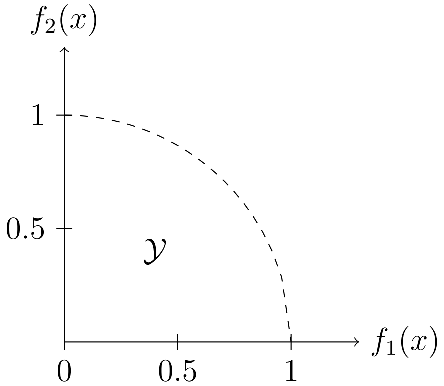


4 Pareto optimal solutions and the method of weighted sum scalarization

4.1 Existence and location of Pareto optimal solutions

Clearly, the definition of Pareto optimal solutions leads to the question whether such solutions always exist. The following example shows that even for convex sets $\mathcal{X}$ and $\mathcal{Y}$, respectively, and continuous objective functions the Pareto set and the set of non-dominated points, respectively, may be empty.

Figure 4.1: The criterion space of Example 4.1.



Example 4.1 Consider the domain $\mathcal{X} = \{(x_1, x_2) \in \mathbb{R}^2 : x_1^2 + x_2^2 < 1, 0 < x_1, x_2 < 1\}$ and the objective functions $f_1(x_1, x_2) = x_1$ and $f_2(x_1, x_2) = x_2$. $\mathcal{Y}$ coincides with $\mathcal{X}$. Since $\mathcal{X}$ and $\mathcal{Y}$ are open, there exist no Pareto optimal solutions and no non-dominated points, respectively.

As Example 4.1 indicates, if $\mathcal{Y}$ is open, then the set of non-dominated points is empty. This result will be a corollary of the forthcoming propositions. Since we are interested in solutions that maximize our objectives, we are interested in the "upper right part" of $\mathcal{Y}$. Points which are dominated will not be located on the "upper right boundary".

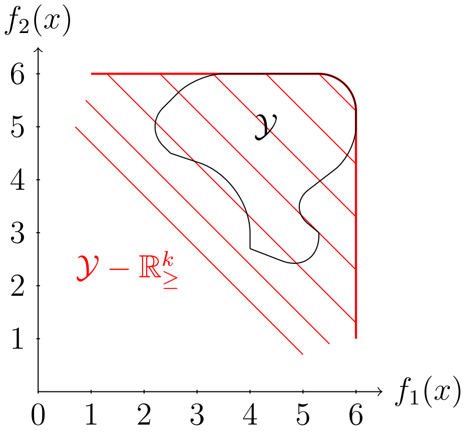
In the following, we denote the set of non-negative vectors of $\mathbb{R}^k$ by $\mathbb{R}^k_{\geq} := \{y \in \mathbb{R}^k : y_i \geq 0 \ \forall i\}$. Furthermore, we define $\mathbb{R}^k_* := \mathbb{R}^k_{\geq} \setminus \{0\}$ and $\mathbb{R}^k_+ := \{y \in \mathbb{R}^k : y_i > 0 \ \forall i\}$.

The forthcoming propositions are mainly auxiliary results for forthcoming theorems. However, they may enhance the reader's intuition behind the concept of non-dominance. The next proposition states that non-dominated points are located in the "upper right part" of $\mathcal{Y}$.

Proposition 4.1
Let $\mathcal{Y} \subseteq \mathbb{R}^k$. *Then* $\mathcal{Y}_N = (\mathcal{Y} - \mathbb{R}^k_{\geq})_N$.

Proof: "$\supseteq$": In case that $(\mathcal{Y} - \mathbb{R}^k_{\geq})$ is empty the statement clear holds. Assume that $y \in (\mathcal{Y} - \mathbb{R}^k_{\geq})_N \neq \emptyset$, but $y \notin \mathcal{Y}_N$. There are two cases. If $y \notin \mathcal{Y}$, then there exists $\bar{y} \in \mathcal{Y}$ and $0 \neq d \in \mathbb{R}^k_{\geq}$ such that $y = \bar{y} - d$. Since $\bar{y} = \bar{y} - 0 \in \mathcal{Y} - \mathbb{R}^k_{\geq}$, it follows that $y \notin (\mathcal{Y} - \mathbb{R}^k_{\geq})_N$ which contradicts the assumption. If $y \in \mathcal{Y}$, then there exists $\bar{y} \in \mathcal{Y}$ such that $\bar{y} \succ y$ due to the assumption that $y \notin \mathcal{Y}_N$. Again, since $\bar{y} - 0 \in \mathcal{Y} - \mathbb{R}^k_{\geq}$, it follows that $y \notin (\mathcal{Y} - \mathbb{R}^k_{\geq})_N$ contradicting the assumption.
"$\subseteq$": For $\mathcal{Y}_N = \emptyset$ the statement clearly holds. Assume that $y \in \mathcal{Y}_N \neq \emptyset$, but $y \notin (\mathcal{Y} - \mathbb{R}^k_{\geq})_N$. Then there is some $\bar{y} \in \mathcal{Y} - \mathbb{R}^k_{\geq}$ such that $\bar{y} = y' - d'$ with $y' \in \mathcal{Y}, d' \in \mathbb{R}^k_{\geq}$ and $\bar{y} - y = d \in \mathbb{R}^k_{*}$. Hence, $y = \bar{y} - d = y' - (d + d')$ with $(d + d') \in \mathbb{R}^k_{*}$ implying $y \notin \mathcal{Y}_N$. This contradicts the assumption. □

Figure 4.2: Non-dominated points of $\mathcal{Y}$ coincide with non-dominated points of $\mathcal{Y} - \mathbb{R}^k_{\geq}$.



The following inclusion is intuitively equivalent to the previous one. We will make use of it in Theorem 4.6 in Section 4.3.

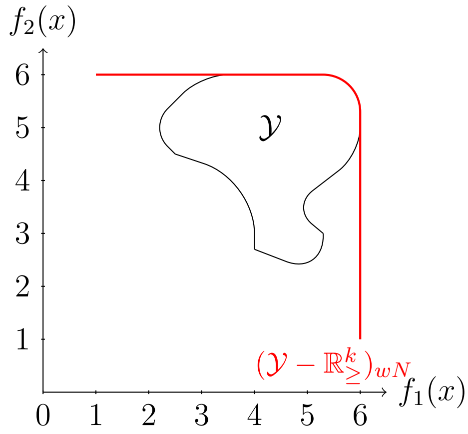
Proposition 4.2
Let $\mathcal{Y} \subseteq \mathbb{R}^k$. *Then* $\mathcal{Y}_{wN} \subseteq (\mathcal{Y} - \mathbb{R}^k_{\geq})_{wN}$.

Proof: For $\mathcal{Y}_{wN} = \emptyset$ the statement clearly holds. Now let $y \in \mathcal{Y}_{wN} \neq \emptyset$ and assume that $y \notin (\mathcal{Y} - \mathbb{R}^k_{\geq})_{wN}$. Then there exists some $\bar{y} \in \mathcal{Y} - \mathbb{R}^k_{\geq}$ such that $\bar{y} = y' - d'$ with $y' \in \mathcal{Y}, d' \in \mathbb{R}^k_{\geq}$ and $\bar{y} - y = d \in \mathbb{R}^k_{*}$. Hence, $y = \bar{y} - d = y' - (d + d')$ with $(d + d') \in \mathbb{R}^k_{*}$. This implies that $y \notin \mathcal{Y}_{wN}$ contradicting the assumption. □

However, the next example shows that the inverse inclusion does not hold in general, that is, $\mathcal{Y}_{wN} \not\supseteq (\mathcal{Y} - \mathbb{R}^k_{\geq})_{wN}$.

Example 4.2 See Figure 4.3. There are clearly points in $(\mathcal{Y} - \mathbb{R}^k_{\geq})_{wN}$, indicated by the red line, that are not contained in $\mathcal{Y}_{wN}$.

Figure 4.3: In general, $(\mathcal{Y} - \mathbb{R}^k_{\geq})_{wN} \not\subseteq \mathcal{Y}_{wN}$.



We denote the interior of a set S by $\text{int}(S)$ and the boundary of S by ∂S. The next result states that the set of non-dominated points is a subset of the boundary of the feasible set in criterion space.

Proposition 4.3
Let $\mathcal{Y} \subseteq \mathbb{R}^k$. Then it holds that $\mathcal{Y}_N \subseteq \partial\mathcal{Y}$.

Proof: For $\mathcal{Y}_N = \emptyset$ the statement clearly holds. So let $y \in \mathcal{Y}_N \neq \emptyset$ and suppose that $y \notin \partial\mathcal{Y}$. Then it must hold that $y \in \text{int}(\mathcal{Y})$. Therefore there exist an ϵ-neighbourhood $B(y, \epsilon)$ of y which is contained in $\mathcal{Y}$. In other words, $B(y, \epsilon) = y + B(0, \epsilon) \subseteq \mathcal{Y}$. Let $d \in \mathbb{R}^k_*$ and choose some sufficiently small $\alpha > 0$ such that $\alpha d \in B(0, \epsilon)$. Then $\bar{y} := y + \alpha d \in \mathcal{Y}$ dominates y. This contradicts the assumption. □

The following corollary casts the intuition behind the introductory Example 4.1 into a mould.

Corollary 4.4
If $\mathcal{Y}$ is open or if $\mathcal{Y} - \mathbb{R}^k_{\geq}$ is open, then $\mathcal{Y}_N = \emptyset$.

Proof: This result is a direct consequence of Proposition 4.1 and Proposition 4.3. □

There exist a number of theorems that state sufficient conditions for the existence of Pareto optimal solutions. For instance, the condition of *semicompactness* of the feasible set in criterion space is sufficient for the existence of non-dominated points. We refer the reader to [8] or [1] for more information. We will not elaborate on this kind of theorems, since one of the main aspects of this thesis is the approximation of Pareto sets. Hence, we want to assume that in our problem instances the set of Pareto optimal solutions is rather large than empty such that a subsequent approximation makes sense.

4.2 $\mathbb{R}^k_{\geq}$-convexity

Some of the forthcoming theorems require some kind of convexity assumption on $\mathcal{Y}$. However, to assume that $\mathcal{Y}$ is convex is very restrictive. We will see that the weaker notion of $\mathbb{R}^k_{\geq}$-convexity will be sufficient due to the previous mentioned fact that we are only interested in the "upper right part" of $\mathcal{Y}$.

Definition 4.5 A set $\mathcal{Y} \subseteq \mathbb{R}^k$ is called $\mathbb{R}^k_{\geq}$-*convex* if $\mathcal{Y} - \mathbb{R}^k_{\geq}$ is convex.

Clearly, every convex set Y is also $\mathbb{R}^k_{\geq}$-convex.

Figure 4.4: $\mathcal{Y}_1$ is $\mathbb{R}^k_{\geq}$-convex but not convex.

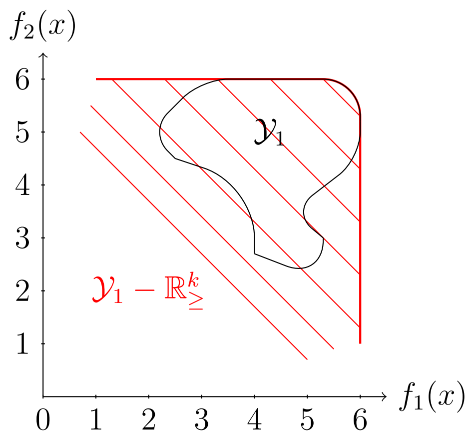
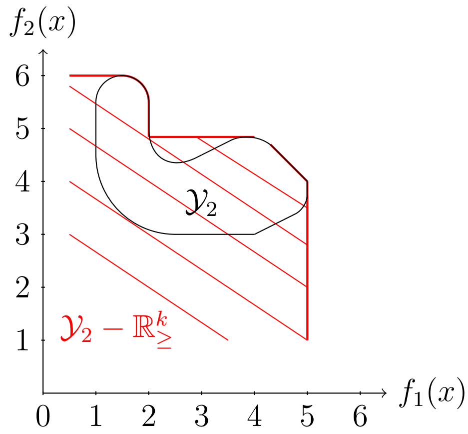


Figure 4.5: $\mathcal{Y}_2$ is neither convex nor $\mathbb{R}^k_{\geq}$-convex.



4.3 Weighted sum scalarization

In this section we consider the method of weighted sum scalarization. In other words, we solve single-objective problems of the type

$$\max_{x \in \mathcal{X}} \sum_{i=1}^{k} \lambda_i f_i(x), \tag{4.1}$$

where $\mathcal{X}$ denotes the feasible domain, the functions f_i $(i = 1, \dots, k)$ denote the objectives and λ_i $(i = 1, \dots, k)$ are chosen weights on the objective functions. Since we only consider the case in which we want to maximize the objectives, we can assume that the weights λ_i are non-negative. By applying the method of weighted sum scalarization, we transform the original k-objective optimization problem into a single-objective optimization problem. If we are originally interested in the Pareto set or an approximate Pareto set, respectively, then the most apparent question arising after the transformation concerns the relationship between optimal solutions of the weighted sum scalarization (4.1) and elements of the Pareto set. In other words, can we construct the whole Pareto set by applying the method of weighted sum scalarization for certain weights? Do optimal solutions of (4.1) always correspond to Pareto optimal solutions?
Let $\mathcal{Y} \subseteq \mathbb{R}^k$. For a fixed $\lambda \in \mathbb{R}^k_*$, we denote the set of optimal solutions of $\mathcal{Y}$ with respect to λ by

$$\mathcal{S}(\lambda, \mathcal{Y}) \coloneqq \{y^* \in \mathcal{Y} : \sum_{i=1}^{k} \lambda_i y_i^* = \max_{y \in \mathcal{Y}} \sum_{i=1}^{k} \lambda_i y_i\}.$$

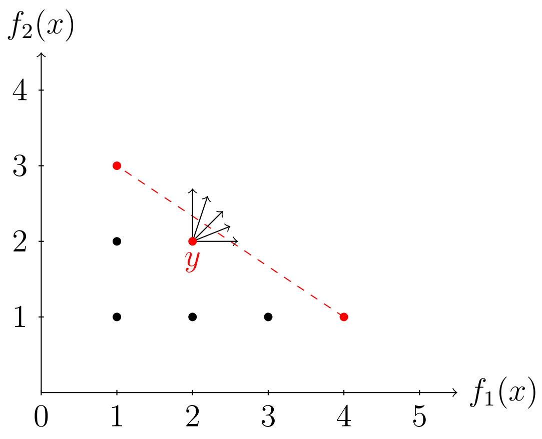
Since $\mathcal{S}(0, \mathcal{Y}) = \mathcal{Y}$, we will exclude the case $\lambda = 0$. The two above mentioned questions read then in terms of $\mathcal{S}(\lambda, \mathcal{Y})$:

1. Is $\mathcal{Y}_N \subseteq \bigcup_{\lambda \in \mathbb{R}^k_*} \mathcal{S}(\lambda, \mathcal{Y})$?

2. Is $\mathcal{S}(\lambda, \mathcal{Y}) \subseteq \mathcal{Y}_N$?

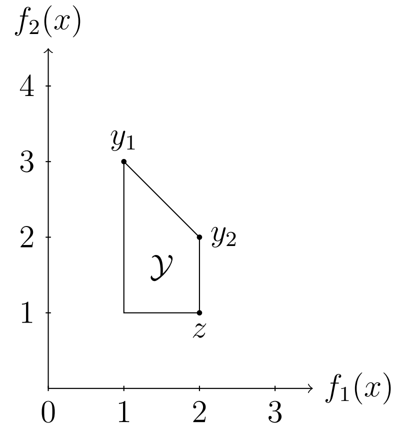
In discrete optimization problems non-dominated points that cannot be found by weighted sum scalarization can easily occur. (See Figure 4.6.)

Figure 4.6: The point y is non-dominated but cannot be found by weighted sum scalarization.



Hence, $\mathcal{Y}_N \not\subseteq \bigcup_{\lambda \in \mathbb{R}^k_*} \mathcal{S}(\lambda, \mathcal{Y})$ in general.

Figure 4.7: The criterion space of Example 4.3.



Example 4.3 Consider the bicriteria maximization problem given by Figure 4.7. All convex combinations of y_1 and y_2 correspond to non-dominated points. Let $\bar{\lambda} = (1, 0)$ and consider $\max\limits_{x \in \mathcal{X}} \sum\limits_{i=1}^{2} \bar{\lambda}_i f_i(x) = \max\limits_{x \in \mathcal{X}} f_1(x)$. Then for all convex combinations z' of z and y_2 it holds that $z' = \max\limits_{x \in \mathcal{X}} f_1(x)$. However, only y_2 is non-dominated and all other convex combinations on the segment between z and y_2 are dominated by y_2. Hence, by allowing weights that nullify one of the objectives, it can easily occur that $\mathcal{S}(\lambda, \mathcal{Y})$ contains dominated points.

Hence, $\bigcup\limits_{\lambda \in \mathbb{R}^k_*} \mathcal{S}(\lambda, \mathcal{Y}) \nsubseteq \mathcal{Y}_N$ in general. The distinction between strictly positive and non-negative weights will turn out to be essential. Thus, we distinguish between the set of points that can be detected by strictly positive weights and the set of points that can be detected by non-negative weights. We define

$$\mathcal{S}(\mathcal{Y}) \coloneqq \bigcup_{\lambda \in \mathbb{R}^k_+} \mathcal{S}(\lambda, \mathcal{Y}) = \bigcup_{\{\lambda \in \mathbb{R}^k_{\geq} : \lambda_i > 0 \forall i, \sum_{i=1}^{k} \lambda_i = 1\}} \mathcal{S}(\lambda, \mathcal{Y})$$

and

$$\mathcal{S}_0(\mathcal{Y}) \coloneqq \bigcup_{\lambda \in \mathbb{R}^k_*} \mathcal{S}(\lambda, \mathcal{Y}) = \bigcup_{\{\lambda \in \mathbb{R}^k_{\geq} : \lambda_i \geq 0 \forall i, \sum_{i=1}^{k} \lambda_i = 1\}} \mathcal{S}(\lambda, \mathcal{Y}).$$

It directly follows that $\mathcal{S}(\mathcal{Y}) \subseteq \mathcal{S}_0(\mathcal{Y})$. Moreover, we can assume without loss of generality that the weight vector λ is normalized, that is, $\sum\limits_{i=1}^{k} \lambda_i = 1$, since normalizing a vector does not change its orientation.

We have now enough ingredients to return to our transformed optimization problem and state results that relate optimal solutions found by weighted sum scalarization to weakly Pareto optimal and Pareto optimal solutions, respectively. The next theorem tells us that points found by weighted sum scalarization always correspond to weakly Pareto optimal solutions.

Theorem 4.6
For any set $\mathcal{Y} \subseteq \mathbb{R}^k$, it holds that $\mathcal{S}_0(\mathcal{Y}) \subseteq \mathcal{Y}_{wN}$.

Proof: Let $\lambda \in \mathbb{R}^k_*$ with $\sum_{j=1}^{k} \lambda_j = 1$ be arbitrary but fixed and let $y^* \in \mathcal{S}(\lambda, \mathcal{Y}) \neq \emptyset$. Then $\sum_{i=1}^{k} \lambda_i y_i^* \geq \sum_{i=1}^{k} \lambda_i y_i$ for all $y \in \mathcal{Y}$. Suppose that $y^* \notin \mathcal{Y}_{wN}$. Then there exists some $\bar{y} \in \mathcal{Y}$ with $\bar{y}_i > y_i^*$ for $i = 1, \ldots, k$. Then $\sum_{i=1}^{k} \lambda_i \bar{y}_i > \sum_{i=1}^{k} \lambda_i y_i^*$ due to the fact that all components of λ are non-negative and at least one of the weights λ_i must be positive. However, $\sum_{i=1}^{k} \lambda_i \bar{y}_i > \sum_{i=1}^{k} \lambda_i y_i^*$ contradicts the assumption that $y^* \in \mathcal{S}(\lambda, \mathcal{Y})$. □

We can strengthen the previous result by assuming that the feasible set in criterion space is $\mathbb{R}^k_{\geq}$-convex. In order to be able to prove the strengthened result we make use of a theorem about separation of convex sets. Before stating this auxiliary result we remind the reader of the definition of the relative interior $\text{ri}(S)$ of a set S.

Definition 4.7 The *relative interior* $\text{ri}(S)$ of a set S is the interior of S considered as a subset of its affine hull $\text{aff}(S)$.

It is clear that the interior of a set S may be empty. A one-dimensional segment in $\mathbb{R}^2$ has no interior points. However, it does have interior points in the one-dimensional affine space that it spans, that is, the line containing the segment. The following auxiliary theorem states that non-intersecting convex sets can be separated by a hyperplane.

Theorem 4.8
Let $Y_1, Y_2 \subseteq \mathbb{R}^k$ be non-empty convex sets. Then the intersection of the relative interiors $ri(Y_1) \cap ri(Y_2)$ is empty if and only if there exists some $y^* \in \mathbb{R}^k \setminus \{0\}$ such that

$$\inf_{y \in Y_1} \langle y, y^* \rangle \geq \sup_{y \in Y_2} \langle y, y^* \rangle \text{ and } \sup_{y \in Y_1} \langle y, y^* \rangle > \inf_{y \in Y_2} \langle y, y^* \rangle$$

where $\langle \cdot, \cdot \rangle$ denotes the usual scalar product in $\mathbb{R}^k$.

For a proof we refer the reader to [9].

The next theorem strengthens the result of Theorem 4.6 under the assumption that the feasible set in criterion space is $\mathbb{R}^k_{\geq}$-convex.

Theorem 4.9
Let $\mathcal{Y} \subseteq \mathbb{R}^k$. If $\mathcal{Y}$ is $\mathbb{R}^k_{\geq}$-convex, then $\mathcal{Y}_{wN} = \mathcal{S}_0(\mathcal{Y})$.

Proof: Due to Theorem 4.6 we only have to show that $\mathcal{Y}_{wN} \subseteq \mathcal{S}_0(\mathcal{Y})$. Furthermore, we know that $\mathcal{Y}_{wN} \subseteq (\mathcal{Y} - \mathbb{R}^k_{\geq})_{wN}$ due to Proposition 4.2. Therefore, if $y^* \in \mathcal{Y}_{wN}$, it follows that $(\mathcal{Y} - \mathbb{R}^k_{\geq} - y^*) \cap \mathbb{R}^k_+ = \emptyset$. In other words, $\mathcal{Y} - \mathbb{R}^k_{\geq} - y^*$ and $\mathbb{R}^k_+$ are both convex sets and the intersection of their relative interiors is empty. By Theorem 4.8 there exists some $\lambda \in \mathbb{R}^k \setminus \{0\}$ such that

$$\langle \lambda, d \rangle \geq \langle \lambda, y - d^* - y^* \rangle$$

for all $y \in \mathcal{Y}, d \in \mathbb{R}^k_+$ and $d^* \in \mathbb{R}^k_\geq$. More precisely, observe that

$$\langle \lambda, d \rangle \geq 0 \geq \langle \lambda, y - d^* - y^* \rangle$$

for all $y \in \mathcal{Y}, d \in \mathbb{R}^k_+$ and $d^* \in \mathbb{R}^k_\geq$, since $\langle \lambda, y - d^* - y^* \rangle = 0$ for $y = y^*$ and $d^* = 0$. Hence, it holds that $\lambda_i \geq 0$ for $i = 1, \ldots, k$. (Otherwise we can construct a vector $\bar{d} \in \mathbb{R}^k_+$ with $\langle \lambda, \bar{d} \rangle < 0$.) From $0 \geq \langle \lambda, y - d^* - y^* \rangle$ we get $\langle \lambda, y^* \rangle \geq \langle \lambda, y \rangle - \langle \lambda, d^* \rangle$ for all $y \in \mathcal{Y}$ and for all $d^* \in \mathbb{R}^k_\geq$. Hence, $\langle \lambda, y^* \rangle \geq \langle \lambda, y \rangle$ for all $y \in \mathcal{Y}$. In other words, $y^* \in \mathcal{S}(\frac{\lambda}{\|\lambda\|}, \mathcal{Y}) \subseteq \mathcal{S}_0(\mathcal{Y})$. □

We can conclude that $\mathcal{S}(\mathcal{Y}) \subseteq \mathcal{S}_0(\mathcal{Y}) \subseteq \mathcal{Y}_{wN}$ in general and $\mathcal{S}(\mathcal{Y}) \subseteq \mathcal{S}_0(\mathcal{Y}) = \mathcal{Y}_{wN}$ for $\mathbb{R}^k_\geq$-convex sets.

Next, we consider the set of non-dominated points $\mathcal{Y}_N$ and state two results that relate the set of solutions found by weighted sum scalarization to $\mathcal{Y}_N$.

Theorem 4.10
Let $\mathcal{Y} \subseteq \mathbb{R}^k$. Then $\mathcal{S}(\mathcal{Y}) \subseteq \mathcal{Y}_N$.

Proof: Let $\lambda \in \mathbb{R}^k_+$ with $\sum_{j=1}^k \lambda_j = 1$ be arbitrary but fixed and let $y^* \in \mathcal{S}(\lambda, \mathcal{Y}) \neq \emptyset$. In other words, $\sum_{i=1}^k \lambda_i y_i^* \geq \sum_{i=1}^k \lambda_i y_i$ for all $y \in \mathcal{Y}$. Suppose that $y^* \notin \mathcal{Y}_N$. Then there must be some $\bar{y} \in \mathcal{Y}$ with $\bar{y}_i \geq y_i^*$ for $i = 1, \ldots, k$ with strict inequality for at least one i. Since $\lambda_i > 0$ for $i = 1, \ldots, k$, it follows that $\sum_{i=1}^k \lambda_i \bar{y}_i > \sum_{i=1}^k \lambda_i y_i^*$, contradicting the assumption that $y^* \in \mathcal{S}(\lambda, \mathcal{Y})$. □

We remind the reader of Example 4.3 which showed that, in general, $\mathcal{S}_0(\mathcal{Y}) \not\subseteq \mathcal{Y}_N$. On the other hand, if we can assume that the feasible set in criterion space is $\mathbb{R}^k_\geq$-convex, then the reverse inclusion holds.

Corollary 4.11
If $\mathcal{Y}$ is $\mathbb{R}^k_\geq$-convex, then $\mathcal{Y}_N \subseteq \mathcal{S}_0(\mathcal{Y})$.

Proof: Since $\mathcal{Y}_N \subseteq \mathcal{Y}_{wN}$, the result is a consequence of Theorem 4.9. □

We can conclude that $\mathcal{S}(\mathcal{Y}) \subseteq \mathcal{Y}_N$ in general and $\mathcal{S}(\mathcal{Y}) \subseteq \mathcal{Y}_N \subseteq \mathcal{S}_0(\mathcal{Y})$ for $\mathbb{R}^k_\geq$-convex sets.

4.4 Proper Pareto optimality and weighted sum scalarization

In this section we consider the relationship between properly Pareto optimal solutions and optimal solutions found by weighted sum scalarization. We show that they coincide for convex sets.

The following theorem states that we find properly Pareto optimal solutions by the method of weighted sum scalarization with strictly positive weights.

Theorem 4.12
Let $\mathcal{X} \subseteq \mathbb{R}^n$ and let $f_i : \mathcal{X} \to \mathbb{R}$ objective funtions for $i = 1, \ldots, k$. Furthermore, let $\lambda \in \mathbb{R}^k_+$ with $\sum_{i=1}^{k} \lambda_i = 1$. If x^* is an optimal solution of $\max_{x \in \mathcal{X}} \sum_{i=1}^{k} \lambda_i f_i(x)$, then x^* is a properly Pareto optimal solution.

Proof: First we show that $x^* \in \mathcal{X}$ is Pareto optimal. Suppose that there exists some $x \in \mathcal{X}$ with $f_i(x) \geq f_i(x^*)$ for $i = 1, \ldots, k$ with strict inequality for at least one i. Since $\lambda_i > 0$ for all $i \in \{1, \ldots, k\}$, it follows that $\sum_{i=1}^{k} \lambda_i f_i(x) > \sum_{i=1}^{k} \lambda_i f_i(x^*)$ contradicting the assumption. Next, we show that x^* is properly Pareto optimal. We define an appropriately large M such that assuming there exists a feasible solution x such that condition (3.2) is violated contradicts the optimality of x^* for the weighted sum scalarization. Define

$$M \coloneqq (k-1) \cdot \max_{i,j \in \{1,\ldots,k\}} \frac{\lambda_j}{\lambda_i}.$$

Suppose x^* is not properly Pareto optimal. Then there exist $i \in \{1, \ldots, k\}$ and some $x \in \mathcal{X}$ with $f_i(x) > f_i(x^*)$ such that $f_i(x) - f_i(x^*) > M(f_j(x^*) - f_j(x))$ for $j \in \{1, \ldots, k\}$ with $f_j(x^*) > f_j(x)$. Hence,

$$f_i(x) - f_i(x^*) > \frac{k-1}{\lambda_i} \lambda_j (f_j(x^*) - f_j(x))$$

for all $j \in \{1, \ldots, k\} \setminus \{i\}$ by definition of M. (If $f_j(x^*) \leq f_j(x)$, then the inequality is clearly fulfilled.) Multiplying each of these inequalities by $\lambda_i/(k-1)$ and summing them over for all $j \in \{1, \ldots, k\} \setminus \{i\}$ yields

$$\lambda_i (f_i(x) - f_i(x^*)) > \sum_{\substack{j=1 \\ j \neq i}}^{k} \lambda_j (f_j(x^*) - f_j(x)).$$

Hence, it holds that

$$\lambda_i f_i(x) + \sum_{\substack{j=1 \\ j \neq i}}^{k} \lambda_j f_j(x) > \lambda_i f_i(x^*) + \sum_{\substack{j=1 \\ j \neq i}}^{k} \lambda_j f_j(x^*)$$

which contradicts the assumption that x^* is an optimal solution of $\max_{x \in \mathcal{X}} \sum_{i=1}^{k} \lambda_i f_i(x)$. □

The following corollary is a direct consequence of Theorem 4.12.

Corollary 4.13
Let $\mathcal{Y} \subseteq \mathbb{R}^k$. Then $\mathcal{S}(\mathcal{Y}) \subseteq \mathcal{Y}_{pN}$.

The next theorem strengthens the previous result under the assumption that the objective functions are convex. In order to be able to prove this strengthened version we make use of an auxiliary theorem which states a useful property of convex functions. We remind the reader that a function f defined on a convex domain is called convex if for any two points x_1 and x_2 in its domain and any $t \in [0,1]$ it holds that $f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$.

Theorem 4.14
Let $\mathcal{X} \subseteq \mathbb{R}^n$ be a convex domain. Let $h_i : \mathcal{X} \to \mathbb{R}$ be convex functions for $i = 1, \ldots, k$. If the system $h_i(x) < 0$ $(i = 1, \ldots, k)$ has no solution $x \in \mathcal{X}$, then there exist $\lambda_i \geq 0$ $(i = 1, \ldots, k)$ with $\sum\limits_{i=1}^{k} \lambda_i = 1$ such that all $x \in \mathcal{X}$ satisfy $\sum\limits_{i=1}^{k} \lambda_i h_i(x) \geq 0$.

For a proof we refer the reader to [10].

Theorem 4.15
Let $\mathcal{X} \subseteq \mathbb{R}^n$ be convex and let $f_i : \mathcal{X} \to \mathbb{R}$ be convex functions for $i = 1, \ldots, k$. Then $x^* \in \mathcal{X}$ is properly Pareto optimal if and only if x^* is an optimal solution of $\max\limits_{x \in \mathcal{X}} \sum\limits_{i=1}^{k} \lambda_i f_i(x)$ with $\lambda \in \mathbb{R}^k_+$.

Proof: Due to Theorem 4.12 we only have to prove the only-if-part. Let $x^* \in \mathcal{X}$ be properly Pareto optimal. Then there exists $M > 0$ such that for each $i \in \{1, \ldots, k\}$, the system of k inequalities

$$\begin{aligned} f_i(x) &> f_i(x^*) \\ f_i(x) + M f_j(x) &> f_i(x^*) + M f_j(x^*) \;\; j = 1, \ldots, k; j \neq i \end{aligned} \tag{4.2}$$

has no solution. Rearranging (4.2) and applying Theorem 4.14 yields for the i-th such system $\lambda^i_l \geq 0$ $(l = 1, \ldots, k)$ with $\sum\limits_{l=1}^{k} \lambda^i_l = 1$ such that for all $x \in \mathcal{X}$ the following inequalities hold:

$$\lambda^i_i \left(f_i(x^*) - f_i(x)\right) + \sum_{\substack{l=1 \\ l \neq i}}^{k} \lambda^i_l (f_i(x^*) - f_i(x) + M\left(f_l(x^*) - f_l(x)\right)) \geq 0$$

$$\Longleftrightarrow$$

$$\lambda^i_i f_i(x^*) + \sum_{\substack{l=1 \\ l \neq i}}^{k} \lambda^i_l \left(f_i(x^*) + M f_l(x^*)\right) \geq \lambda^i_i f_i(x) + \sum_{\substack{l=1 \\ l \neq i}}^{k} \lambda^i_l (f_i(x) + M f_l(x))$$

$$\Longleftrightarrow$$

$$\lambda^i_i f_i(x^*) + \sum_{\substack{l=1 \\ l \neq i}}^{k} \lambda^i_l f_i(x^*) + M \sum_{\substack{l=1 \\ l \neq i}}^{k} \lambda^i_l f_l(x^*) \geq \lambda^i_i f_i(x) + \sum_{\substack{l=1 \\ l \neq i}}^{k} \lambda^i_l f_i(x) + M \sum_{\substack{l=1 \\ l \neq i}}^{k} \lambda^i_l f_l(x)$$

$$\Longleftrightarrow$$

$$\sum_{l=1}^{k} \lambda^i_l f_i(x^*) + M \sum_{\substack{l=1 \\ l \neq i}}^{k} \lambda^i_l f_l(x^*) \geq \sum_{l=1}^{k} \lambda^i_l f_i(x) + M \sum_{\substack{l=1 \\ l \neq i}}^{k} \lambda^i_l f_l(x)$$

$$\Longleftrightarrow$$

$$f_i(x^*) + M \sum_{\substack{l=1 \\ l \neq i}}^{k} \lambda^i_l f_l(x^*) \geq f_i(x) + M \sum_{\substack{l=1 \\ l \neq i}}^{k} \lambda^i_l f_l(x) \tag{4.3}$$

Inequality (4.3) holds for each $i \in \{1, \dots, k\}$. Hence, by summing over $i = 1, \dots, k$, we obtain

$$\sum_{i=1}^{k} f_i(x^*) + M \sum_{i=1}^{k} \sum_{\substack{l=1 \\ l \neq i}}^{k} \lambda_l^i f_l(x^*) \geq \sum_{i=1}^{k} f_i(x) + M \sum_{i=1}^{k} \sum_{\substack{l=1 \\ l \neq i}}^{k} \lambda_l^i f_l(x)$$

$$\Longleftrightarrow$$

$$\sum_{i=1}^{k} f_i(x^*) + M \sum_{i=1}^{k} \sum_{\substack{l=1 \\ l \neq i}}^{k} \lambda_i^l f_i(x^*) \geq \sum_{i=1}^{k} f_i(x) + M \sum_{i=1}^{k} \sum_{\substack{l=1 \\ l \neq i}}^{k} \lambda_i^l f_i(x)$$

$$\Longleftrightarrow$$

$$\sum_{i=1}^{k} \left(1 + M \sum_{\substack{l=1 \\ l \neq i}}^{k} \lambda_i^l \right) f_i(x^*) \geq \sum_{i=1}^{k} \left(1 + M \sum_{\substack{l=1 \\ l \neq i}}^{k} \lambda_i^l \right) f_i(x)$$

By setting $\lambda_i := \frac{1+M \sum_{\substack{l=1 \\ l \neq i}}^{k} \lambda_i^l}{\sum_{j=1}^{k} (1+M \sum_{\substack{l=1 \\ l \neq j}}^{k} \lambda_j^l)}$ for $i = 1, \dots, k$, we obtain positive, normalized λ_i for which x^* is an optimal solution of $\max_{x \in \mathcal{X}} \sum_{i=1}^{k} \lambda_i f_i(x)$. □

4.5 Multi-criteria linear programming

Under the assumption that the considered domain $\mathcal{X}$ is convex and the objective functions $f_1, \dots, f_k$ are convex as well, Theorem 4.15 stated that we can detect all properly Pareto optimal solutions by the method of weighted sum scalarization with strictly positive weights. Since, in general, $\mathcal{X}_{pP} \subset \mathcal{X}_P$ (see Example 3.6), Theorem 4.15 does not imply that we can detect the whole Pareto set. However, in the case of multi-criteria linear programming it can be shown that for every Pareto optimal solution x^* there exists a weight vector $\lambda \in \mathbb{R}_+^k$ such that $x^* \in \arg\max_{x \in \mathcal{X}} \sum_{i=1}^{k} f_i(x)$ implying that every Pareto optimal solution is also properly Pareto optimal.

Definition 4.16 A multi-criteria linear program is defined as follows:

$$\begin{aligned} &\max Cx = \max(C_1 x, \dots, C_k x) \\ &\text{s.t. } Ax = b, x \geq 0 \end{aligned} \tag{4.4}$$

where C is a $k \times n$ objective matrix, C_i denotes the i-th row of C, A is an $m \times n$ constraint matrix and $b \in \mathbb{R}^m$.

The feasible set in decision space is $\mathcal{X} = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$. The feasible set in criterion space is $\mathcal{Y} = \{Cx : x \in \mathcal{X}\}$. Notice that $\mathcal{X}$ and $\mathcal{Y}$ are closed and convex. Referring to Definition 2.1 and Definition 3.4 we define Pareto optimality and proper Pareto optimality as follows:

Definition 4.17 Let $x^* \in \mathcal{X}$ be a feasible solution and let $y^* = Cx^*$.

1. x^* is called *Pareto optimal* if there is no $x \in \mathcal{X}$ such that $Cx \succ Cx^*$, that is, $C_i x \geq C_i x^*$ for all $i \in \{1, \dots, k\}$ with strict inequality for at least one i.

2. x^* is called *properly Pareto optimal* if x^* is Pareto optimal and if there exists a real number $M > 0$ such that for $i \in \{1, \dots, k\}$ and for $x \in \mathcal{X}$ satisfying $C_i x > C_i x^*$ there is an index $j \in \{1, \dots, k\}$ with $C_j x^* > C_j x$ such that

$$\frac{C_i x - C_i x^*}{C_j x^* - C_j x} \leq M.$$

Next, we consider the weighted sum scalarization LP

$$\begin{aligned} \max \lambda^T C x \\ \text{s.t. } Ax = b, x \geq 0 \end{aligned} \tag{4.5}$$

where $\lambda \in \mathbb{R}^k_+$. Referring to Theorem 4.10 the following statement holds.

Theorem 4.18
Let $\lambda \in \mathbb{R}^k_+$ and let $x^* \in \mathcal{X}$ be an optimal solution of (4.5). Then x^* is Pareto optimal.

Proof: Suppose there is $x \in \mathcal{X}$ that dominates x^*, that is, $C_i x \geq C_i x^*$ for $i = 1, \dots, k$ with strict inequality for at least one i. Then it holds that $\lambda_i C_i x \geq \lambda_i C_i x^*$ for $i = 1, \dots, k$ with strict inequality for at least one i, since $\lambda \in \mathbb{R}^k_+$. Summing over i yields $\lambda^T C x > \lambda^T C x^*$ which contradicts the assumption. □

In order to show the coincidence between $\mathcal{X}_P$ and $\mathcal{X}_{pP}$ in multi-criteria linear programming we will make use of duality theory. Therefore, we shortly remind the reader of it. Let

$$\begin{aligned} \max c^T x \\ \text{s.t. } Ax = b, x \geq 0 \end{aligned} \tag{4.6}$$

be a single-objective linear program. For (4.6) a dual linear program is defined as

$$\begin{aligned} \min b^T u \\ \text{s.t. } A^T u \geq c, u \in \mathbb{R}^m. \end{aligned} \tag{4.7}$$

We denote the feasible set of the dual linear program by $\mathcal{U} \coloneqq \{u \in \mathbb{R}^m : A^T u \geq c\}$. Weak duality states that if $x \in \mathcal{X}$ and $u \in \mathcal{U}$ are feasible solutions of (4.6) and (4.7), respectively, it holds that $c^T x \leq b^T u$. The so-called strong duality result states that if both the primal linear program (4.6) and the dual linear program (4.7) are feasible, that is, $\mathcal{X} \neq \emptyset$ and $\mathcal{U} \neq \emptyset$, then $\max\limits_{x \in \mathcal{X}} c^T x = \min\limits_{u \in \mathcal{U}} b^T u$.

Lemma 4.19
A feasible solution $x^* \in \mathcal{X} = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$ *is Pareto optimal if and only if the linear program*

$$\begin{aligned} &\max e^T z \\ s.t.\ &Ax = b \\ &Cx - Iz = Cx^* \\ &x, z \geq 0, \end{aligned} \tag{4.8}$$

where $e^T = (1, \dots, 1) \in \mathbb{R}^k$ *and* I *is the* $k \times k$ *identity matrix, has an optimal solution* $(\hat{x}, \hat{z})$ *with* $\hat{z} = 0$*.*

Proof: Notice that (4.8) is always feasible since $(x, z) = (x^*, 0)$ is always a feasible solution.
"$\Rightarrow$": Let $(x, z) \in \mathcal{X} \times \mathbb{R}^k_{\geqq}$ be a feasible solution of (4.8). Then $Cx - Iz = Cx^*$ is fulfilled. Therefore, $z = Cx - Cx^* \geqq 0$ by the non-negativity of z. If x^* is Pareto optimal, then there exists no $x' \in \mathcal{X}$ with $Cx' \succ Cx^*$. Hence, $z = 0$ for every feasible solution (x, z).
"$\Leftarrow$": Assume that $x^* \in \mathcal{X}$ is not Pareto optimal. Then there exists $x' \in \mathcal{X}$ with $Cx' \succ Cx^*$. Therefore, $z' = Cx' - Cx^* \succ 0$ implying that there exists $i \in \{1, \dots, k\}$ with $z'_i > 0$. This contradicts the optimality of $(\hat{x}, 0)$. □

Lemma 4.20
A feasible solution $x^* \in \mathcal{X}$ *is Pareto optimal if and only if the linear program*

$$\begin{aligned} &\min u^T b + w^T C x^* \\ s.t.\ &A^T u + C^T w \geq 0 \\ &-w \geq e \\ &u \in \mathbb{R}^m \end{aligned} \tag{4.9}$$

has an optimal solution $(\hat{u}, \hat{w})$ *with* $\hat{u}^T b + \hat{w}^T C x^* = 0$*.*

Proof: Notice that (4.9) is the dual of (4.8). Therefore, $(\hat{x}, \hat{z})$ is an optimal solution of (4.8) if and only if (4.9) has an optimal solution $(\hat{u}, \hat{w})$ such that $e^T \hat{z} = \hat{u}^T b + \hat{w}^T C x^* = 0$. □

The following theorem goes back to Isermann dating from 1974. It states that each Pareto optimal feasible solution x of (4.4) can be found by solving a weighted sum scalarization LP with strictly positive weights.

Theorem 4.21
A feasible solution $x^* \in \mathcal{X}$ *is a Pareto optimal solution of (4.4) if and only if there exists* $\lambda \in \mathbb{R}^k_+$ *such that*

$$\lambda^T C x^* \geq \lambda^T C x$$

for all $x \in \mathcal{X}$*.*

Proof: "$\Leftarrow$": This is the statement of Theorem 4.18.
"$\Rightarrow$": Let $x^* \in \mathcal{X}_P$. We construct an appropriate weight vector $\lambda \in \mathbb{R}^k_+$ such that $x^* \in \arg\max_{x \in \mathcal{X}} \sum_{i=1}^{k} \lambda_i C_i x$. Lemma 4.20 implies that (4.9) has an optimal solution $(\hat{u}, \hat{w})$ such that

$$\hat{u}^T b = -\hat{w}^T C x^*. \tag{4.10}$$

Then it holds that this same $\hat{u}$ is also an optimal solution of

$$\min\{u^T b : A^T u \geq -C^T \hat{w}\} \tag{4.11}$$

which is (4.9) with $w = \hat{w}$ fixed. Hence, an optimal solution of the dual of (4.11)

$$\max\{-\hat{w}^T C x : Ax = b, x \geq 0\} \tag{4.12}$$

exists. By weak duality it holds that $-\hat{w}^T C x \leq u^T b$ for all feasible solutions x of (4.12) and for all feasible solutions u of (4.11). Since $\hat{u}^T b = -\hat{w}^T C x^*$ due to (4.10), it follows that x^* is an optimal solution of (4.12). By the constraints of (4.9) it holds that $-\hat{w} \geq e$. Setting $\lambda := -\hat{w} \in \mathbb{R}^k_+$ implies that x^* is an optimal solution of the weighted sum scalarization linear program $\max\{\lambda^T C x : Ax = b, x \geq 0\}$. □

Since $\mathcal{Y}_{pN} = \mathcal{S}(\mathcal{Y})$ by Theorem 4.15, Theorem 4.21 implies that $\mathcal{X}_P = \mathcal{X}_{pP}$ and $\mathcal{S}(\mathcal{Y}) = \mathcal{Y}_N = \mathcal{Y}_{pN}$ for multi-criteria linear programs.

5 Complexity, existence and computation of ϵ-approximate Pareto sets

5.1 Complexity issues

At first glance, it could seem helpful to tackle optimization problems with multiple objectives by means of optimizing one objective while keeping the others bounded. However, the next result shows that, in general, we cannot overcome NP-hardness by this approach even for well-known combinatorial optimization problems that are tractable in the single-objective case.

Proposition 5.1
Given the bicriteria Shortest Path *problem. Optimizing one objective while keeping the other objective bounded yields the NP-hard* Constrained Shortest Path *problem.*

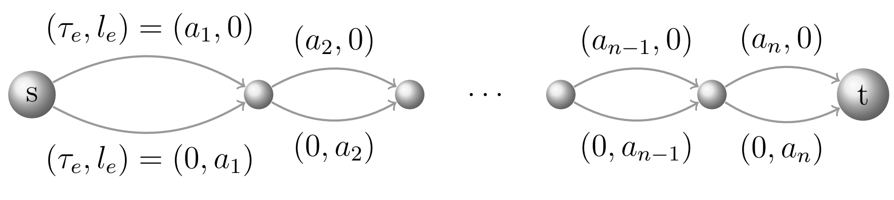
Proof: Consider the bicriteria Shortest Path problem: Given a directed graph $G = (V, E)$, two distinguished vertices $s, t \in V$ and two edge labels $l : E \to \mathbb{R}$ and $\tau : E \to \mathbb{R}$, respectively. Let $\mathcal{P}_{st}$ be the set of s-t-paths in G. Then the two considered objectives are $\min\limits_{P \in \mathcal{P}_{st}} \sum\limits_{e \in P} l(e)$ and $\min\limits_{P \in \mathcal{P}_{st}} \sum\limits_{e \in P} \tau(e)$, respectively. Optimizing one objective while keeping the other bounded yields the so-called Constrained Shortest Path problem (CSP):

$$\min_{P \in \mathcal{P}_{st}} \sum_{e \in P} \tau(e)$$

$$\text{s. t.} \sum_{e \in P} l(e) \leq L \text{ for some given } L \in \mathbb{R}.$$

Consider the NP-complete Partition problem: Given n positive integers $a_1, \ldots, a_n$ with $\sum\limits_{j=1}^{n} a_j = 2b$. Is there a subset $S \subseteq \{1, \ldots, n\}$ such that $\sum\limits_{j \in S} a_j = b$? We show that Partition can be reduced to the decision version of the Constrained Shortest Path problem.
Reduction: Let I be an instance of Partition. We construct an instance I' of CSP as follows:

G:



The question is now whether there exists an s-t-path P with $\tau(P) = \sum_{e \in P} \tau(e) \leq b$ and $l(P) = \sum_{e \in P} l(e) \leq b$. This question can be answered in the affirmative if and only if there exists a subset $S \subseteq \{1, \dots, n\}$ with $\sum_{j \in S} a_j = b$. Thus, PARTITION can be polynomially reduced to the CONSTRAINED SHORTEST PATH problem and hence the (decision version of the) latter is NP-hard. □

Proposition 5.2
Given the bicriteria MINIMUM SPANNING TREE *problem. Optimizing one objective while keeping the other objective bounded yields an NP-hard problem.*

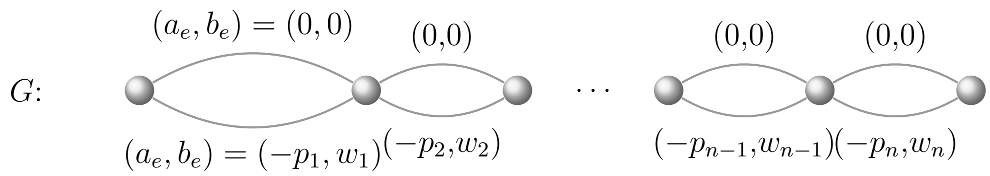
Proof: Consider the bicriteria MINIMUM SPANNING TREE problem. Let $G = (V, E)$ be an undirected graph in which each edge e has two attributes $(a_e, b_e) \in \mathbb{R}^2$. Let $\mathcal{T}$ be the set of all spanning trees in G. Consider the problem:

$$\min \sum_{e \in T} a_e \ , \ \ \min \sum_{e \in T} b_e$$
$$\text{subject to } T \in \mathcal{T}.$$

Imposing a bound $B \in \mathbb{R}$ on the second objective yields

$$\min \sum_{e \in T} a_e$$
$$\text{s. t. } \sum_{e \in T} b_e \leq B \text{ and } T \in \mathcal{T}.$$

We show that KNAPSACK can be reduced to the latter problem. Consider an instance of KNAPSACK: Given non-negative numbers p_j, w_j $(j = 1, \dots, n)$ and a weight limit W. Find a subset $S \subseteq \{1, \dots, n\}$ such that $\sum_{j \in S} w_j \leq W$ and $\sum_{j \in S} p_j$ is maximum. Construct a graph G consisting of $n + 1$ vertices as follows:



Set $B \coloneqq W$ and observe that a minimum spanning tree on G subject to the constraint $\sum_{e \in T} b_e \leq B$ corresponds to a feasible KNAPSACK selection S of items j maximizing $\sum_{j \in S} p_j$. □

The next result shows that, in general, the problem of determining whether a given solution is Pareto optimal is NP-hard even in the bicriteria case of well-known combinatorial optimization problems which are tractable in the single-objective case.

Proposition 5.3
Given the bicriteria SHORTEST PATH *problem. Determining whether a given s-t-path P is Pareto optimal is NP-hard.*

Proof: Consider the decision version of the CONSTRAINED SHORTEST PATH problem: Given a directed graph $G = (V, E)$, two distinguished vertices $s, t \in V$, two edge labels $l : E \to \mathbb{R}$ and $\tau : E \to \mathbb{R}$, respectively, and a given bound $L \in \mathbb{R}$. Is there an s-t-path P with

$$\sum_{e \in P} \tau(e) \leq B$$

$$\text{s. t. } \sum_{e \in P} l(e) \leq L?$$

In the proof of Proposition 5.1 it was shown that CSP is NP-hard. Construct for a given instance of CSP the following instance of the bicriteria SHORTEST PATH problem: $G = (V, E \cup \{\bar{e}\})$ where $\bar{e} = (s, t)$ is an edge connecting s with t. Furthermore, set $\tau(\bar{e}) \coloneqq B + \epsilon$ and $l(\bar{e}) \coloneqq L + \epsilon$ (for a tiny $\epsilon > 0$). Let $\mathcal{P}_{st}$ be the set of s-t-paths in G. Then $\bar{P} = \{\bar{e}\} \in \mathcal{P}_{st}$ is a feasible solution for the considered instance of the bicriteria SHORTEST PATH problem. Being able to determine whether $\bar{P}$ is Pareto optimal yields an answer to CSP. If $\bar{P}$ is Pareto optimal, then for all s-t-paths $P \in \mathcal{P}_{st}$ it holds that either $\tau(P) \geq B + \epsilon$ or $l(P) \geq L + \epsilon$. Hence, if $\bar{P}$ is Pareto optimal, then the answer to CSP is "no", since there exist no s-t-path P with $\sum_{e \in P} \tau(e) \leq B$ subjet to $\sum_{e \in P} l(e) \leq L$. On the other hand, if $\bar{P}$ is not Pareto optimal, then there exists an s-t-path $P \in \mathcal{P}_{st}$ with $\tau(P) \leq B$ and $l(P) \leq L$ and hence the answer to CSP is "yes". □

5.2 Existence of approximate Pareto sets

The main results presented in the remainder of this chapter were noted by Papadimitriou and Yannakakis in [2]. However, the corresponding proofs were only sketched. In the following, we present the results and the proofs in more detail.

In the remainder of this chapter we assume that the objective functions $f_1, \ldots, f_k$ are non-negative. Furthermore, we assume that the considered optimization problems are subject to the following Condition C^*: Let I be an instance of an optimization problem P. If the objective value of a feasible solution x is positive, that is, $f_i(x) > 0$, then it holds that $f_i(x) \in [2^{-p(|I|)}, 2^{p(|I|)}]$ for some polynomial p depending on the size of the instance.

The following theorem is a basic existence result noted by Papadimitriou and Yannakakis (see [2]). It states that an ϵ-approximate Pareto set always exists for any multi-criteria optimization problem (fulfilling Condition C^*).

Theorem 5.4
Given a maximization problem P with domain $\mathcal{X}$, non-negative objectives $f_1, \ldots, f_k$ and $\epsilon > 0$. There exists an ϵ-approximate Pareto set $\mathcal{X}_{\epsilon P}$ consisting of a number of solutions that is polynomial in the input size and $\frac{1}{\epsilon}$ (but exponential in the number of objectives).

Proof: Consider the k-dimensional space of all objectives. By our prerequisite we assume that the positive objective values of solutions range from $2^{-p(|I|)}$ to $2^{p(|I|)}$ for some polynomial p. We define a partition of intervals in each dimension such that the ratio of the larger to the smaller coordinate is $1+\epsilon$. This leads to a subdivision of the k-dimensional cube. Let n be the last index of a partition, that is, $n \coloneqq \min\{k \in \mathbb{N} : 2^{p(|I|)} \leq 2^{-p(|I|)}(1+\epsilon)^k\}$. Then it holds that $2^{-p(|I|)}(1+\epsilon)^{n-1} \leq 2^{p(|I|)}$. Therefore

$$n \leq \frac{2p(|I|)}{\mathrm{ld}(1+\epsilon)} + 1.$$

By Proposition 2.2 and a change to the dual logarithm, it holds that $\frac{x}{1+x} \leq \ln(1+x) \leq \mathrm{ld}(1+x)$ for $x \geq 0$ implying that

$$n \leq \frac{2p(|I|)(1+\epsilon)}{\epsilon} + 1.$$

Hence, the number of subdivisions is in $O(\frac{(2p(|I|))^k}{\epsilon^k})$. We define $\mathcal{X}_{\epsilon P}$ by choosing one point of $\mathcal{X}_P$ in each hyperrectangle that contains such a point. It follows then that for every $x \in \mathcal{X}_P$ there exists $\bar{x} \in \mathcal{X}_{\epsilon P}$ such that $f_i(x) \leq (1+\epsilon)f_i(\bar{x})$ for $i = 1, \ldots, k$. Hence, $\mathcal{X}_{\epsilon P}$ is an ϵ-approximate Pareto set for $\mathcal{X}_P$. □

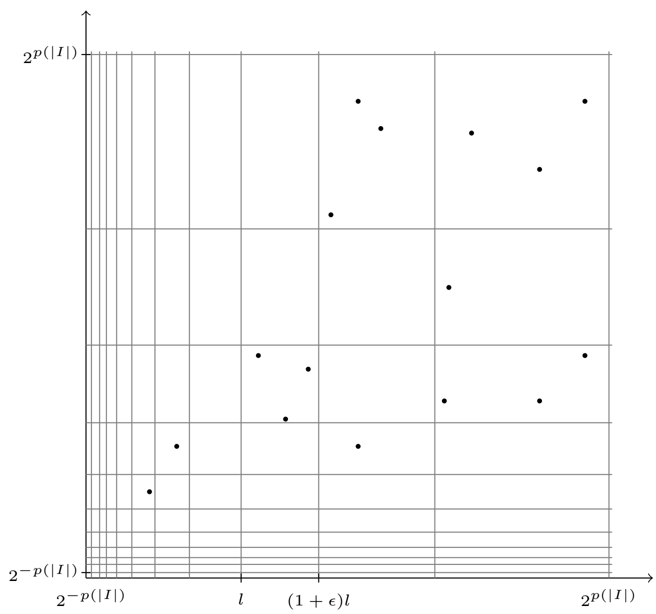


Figure 5.1: Schematic representation of the subdivision of the k-dimensional cube.

5.3 Computational issues

After the general existence of approximate Pareto sets is shown, the problem and question arising now concerns the possibility of an efficient computation of the latter. Unfortunately, we cannot derive a general efficiently running algorithm for constructing $\mathcal{X}_{\epsilon P}$ from Theorem 5.4, since there are even in the single-objective case optimization problems for which no FPTAS can exist unless P = NP. The next result gives a condition for the existence of a polynomial algorithm constructing an ϵ-approximate Pareto set $\mathcal{X}_{\epsilon P}$.

Definition 5.5 Given a multi-criteria optimization problem P with domain $\mathcal{X}$ and objective functions f_i $(i = 1, \dots, k)$.
By the Gap problem we mean the following: Given a k-vector $(c_1, \dots, c_k)$ and $\epsilon > 0$, either return a solution $x \in \mathcal{X}$ with $f_i(x) \geq c_i$ for all i or answer that there is no solution $x' \in \mathcal{X}$ with $f_i(x') \geq c_i(1 + \epsilon)$.

Theorem 5.6
Given a maximization problem P with non-negative objective functions $f_1, \dots, f_k$ and $\epsilon > 0$. There is an algorithm for constructing an ϵ-approximate Pareto set $\mathcal{X}_{\epsilon P}$ in time polynomial in $|I|$ and $\frac{1}{\epsilon}$ if and only if the Gap *problem is tractable.*

Proof: "$\Leftarrow$": We wish to find $\mathcal{X}_{\epsilon P}$. Define $\epsilon' := \sqrt{1+\epsilon} - 1$. Subdivide the k-dimensional space of objectives using ϵ' into hyperrectangles as shown in the proof of Theorem 5.4. Moreover, let $n \coloneqq \min\{l \in \mathbb{N} : 2^{p(|I|)} \leq 2^{-p(|I|)}(1+\epsilon')^l\}$. Then define $\mathcal{C} \coloneqq \bigcup_{i=0}^{n} \{2^{-p(|I|)}(1+\epsilon')^i\}$. Call the Gap problem for ϵ' and for each corner point $C \in \mathcal{C}^k$ of the subdivision. Notice that the number of corner points is $(n+1)^k$ which is polynomial in n, since we consider the number of objective functions to be fixed.

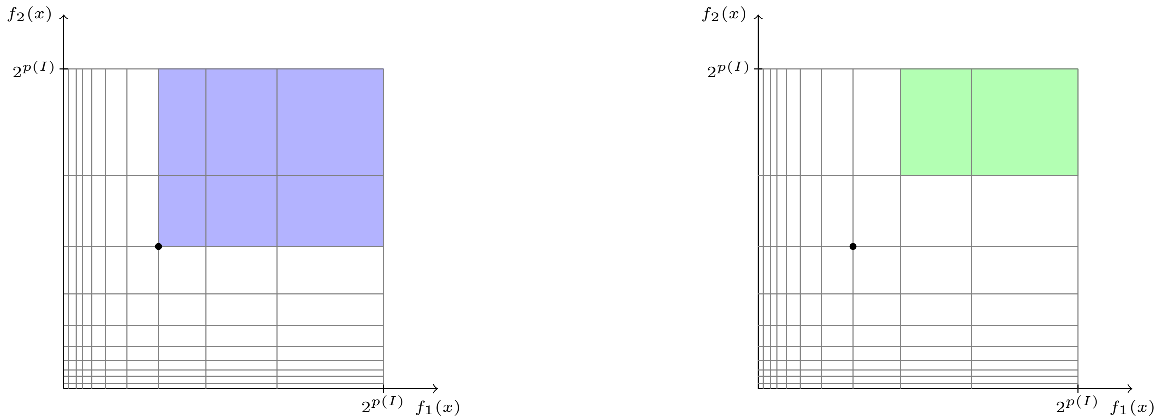


Figure 5.2: Calling the Gap problem for the highlighted black corner point. Either a solution within the purple area is returned or there is no solution within the green area.

We call the Gap problem for each corner point $C \in \mathcal{C}^k$ and keep (an undominated subset $\mathcal{X}_{\epsilon P}$ of) all returned solutions. Let $x^* \in \mathcal{X}$ be a Pareto optimal solution and let $H(x^*)$ be the corresponding hyperrectangle in which the objective values of x^* can be found. Furthermore, let $C_{H(x^*)}$ be the "lower left" corner of $H(x^*)$ with coordinates $C_{H(x^*)} = (2^{-p(|I|)}(\sqrt{1+\epsilon})^{j_1}, \dots, 2^{-p(|I|)}(\sqrt{1+\epsilon})^{j_k}) \in \mathcal{C}^k$ where $(j_1, \dots, j_k) \in \{0, \dots, n-1\}^k$. Since, x^* is Pareto optimal, calling the GAP problem for $C_{H(x^*)} = (c_{H1}, \dots, c_{Hk})$ and ϵ' could yield as answer that there is no solution x' with $f_i(x') \geq c_{Hi}(1+\epsilon') \forall i$ without returning an approximate solution for x^*. On the other hand, due to the fact that we call the Gap problem for each corner point $C \in \mathcal{C}^k$ we eventually call it for the lower diagonal neighbour $C'_{H(x^*)}$ of $C_{H(x^*)}$. Notice the following: Consider an arbitrary point $y^1 \in H(x^*)$ and an arbitrary point y^2 located in the lower diagonal hyperrectangle of $H(x^*)$. Then it holds that $y^1_i \leq (1+\epsilon) y^2_i$ for all $i = 1, \dots, k$, since the ratio of a larger coordinate to its smaller neighbouring coordinate in the subdivision was $1 + \epsilon' = \sqrt{1+\epsilon}$. Calling the GAP problem for $C'_{H(x^*)} = (c'_{H1}, \ldots, c'_{Hk})$ does not admit as answer that there is no solution x' with $f_i(x') \geq c'_{Hi}(1+\epsilon')$, since x^* is a solution fulfilling the latter inequality. Notice that $C'_{H(x^*)}$ has coordinates $(2^{-p(|I|)}(\sqrt{1+\epsilon})^{j_1-1}, \ldots, 2^{-p(|I|)}(\sqrt{1+\epsilon})^{j_k-1})$, since $C_{H(x^*)} = (2^{-p(|I|)}(\sqrt{1+\epsilon})^{j_1}, \ldots, 2^{-p(|I|)}(\sqrt{1+\epsilon})^{j_k})$. Hence, calling the GAP problem for $C'_{H(x^*)}$ will return a solution $\bar{x} \in \mathcal{X}$ with

$$f_i(\bar{x}) \geq 2^{-p(|I|)}(\sqrt{1+\epsilon})^{j_i-1}$$

for all $i \in \{1, \ldots, k\}$. Since $f(x^*) \in H(x^*)$ and $C_{H(x^*)}$ is the lower diagonal corner of $H(x^*)$, it holds that

$$f_i(x^*) \leq 2^{-p(|I|)}(\sqrt{1+\epsilon})^{j_1}\sqrt{1+\epsilon} = 2^{-p(|I|)}(\sqrt{1+\epsilon})^{j_1-1}(1+\epsilon) \leq f_i(\bar{x})(1+\epsilon)$$

for all $i \in \{1, \ldots, k\}$. In other words, the returned solution $\bar{x}$ ϵ-approximates the Pareto optimal solution x^*. Hence, an ϵ-approximate Pareto set $\mathcal{X}_{\epsilon P}$ is constructed by the above mentioned algorithm which is polynomial in the size of the instance I and in $\frac{1}{\epsilon}$.

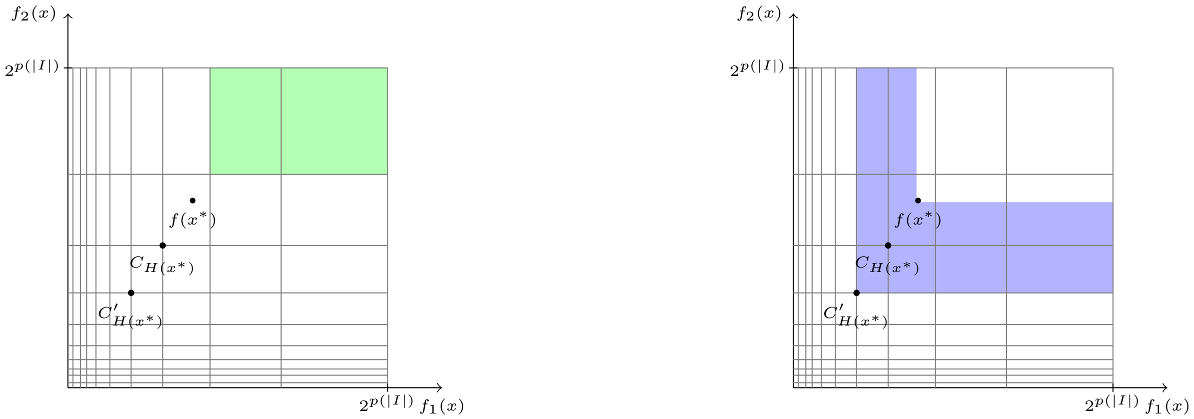


Figure 5.3: Calling the GAP problem for $C_{H(x^*)}$ with answer that no solution within the green area exists. Calling the GAP problem for $C'_{H(x^*)}$ returns a solution within the purple area.

"⇒": Suppose there is an algorithm for constructing an ϵ-approximate Pareto set polynomial in $|I|$ and in $\frac{1}{\epsilon}$. If we have an ϵ-approximate Pareto set $\mathcal{X}_{\epsilon P}$, then we can solve the GAP problem by looking only at solutions in $\mathcal{X}_{\epsilon P}$. Let $(c_1, \ldots, c_k)$ be an arbitrary but fixed k-tuple. We iterate through $\mathcal{X}_{\epsilon P}$ checking whether a solution x exists with $f_i(x) \geq c_i$ for all $i \in \{1, \ldots, k\}$. If we find such a solution x, then we return it. If we do not find such a solution fulfilling above inequalities, then we know that no solution x' exists with $f_i(x') \geq (1+\epsilon)c_i$ for $i = 1, \ldots, k$ due to the fact that $\mathcal{X}_{\epsilon P}$ is an ϵ-approximate Pareto set. In order to see this, suppose that there is a Pareto optimal solution x' with $f_i(x') \geq (1+\epsilon)c_i$ for $i = 1, \ldots, k$. Then by the definition of an ϵ-approximate Pareto set there exists an $\bar{x} \in \mathcal{X}_{\epsilon P}$ with $(1+\epsilon)f_i(\bar{x}) \geq f_i(x') \geq (1+\epsilon)c_i$ implying $f_i(\bar{x}) \geq c_i$ for $i = 1, \ldots, k$. □

5.3.1 The linear convex case

Consider a maximization problem P with convex domain $\mathcal{X}$ and non-negative linear objectives $f_1, \dots, f_k$. We assume that we are able to solve the weighted sum scalarization problem $\max_{x \in \mathcal{X}} \sum_{i=1}^{k} \lambda_i f_i(x)$ with $\lambda \in \{0, \dots, M\}^k$ for $M \in \mathbb{N}$ efficiently.

Let $M := \lceil 4k^4/\epsilon \rceil$ for a given $\epsilon > 0$. Then consider the following algorithm for choosing a subset of the weakly Pareto set $\mathcal{X}_{wP}$.

Algorithm 1:

set $S = \{\}$
foreach $\lambda \in \{0, \dots, M\}^k \setminus \{0\}$ **do**
  find an optimum $x^* \in \mathcal{X}$ for the weighted sum scalarization $\max_{x \in \mathcal{X}} \sum_{i=1}^{k} \lambda_i f_i(x)$
  insert x^* into S
end
return the set S of all optima thus found

Theorem 4.9 told us that $\mathcal{Y}_{wN} = \mathcal{S}_0(\mathcal{Y})$ if $\mathcal{Y}$ is $\mathbb{R}^k_{\geq}$-convex. This will be the case if the domain $\mathcal{X}$ is a convex set and the objective functions f_i $(i = 1, \dots, k)$ are linear. Hence, the returned solution set of Algorithm 1 consists of weakly Pareto optimal solutions. By Definition 3.3, it holds that $\mathcal{X}_P \subseteq \mathcal{X}_{wP}$. The definition of an ϵ-approximate Pareto set $\mathcal{X}_{\epsilon P}$ (Definition 3.1) requires that for every Pareto optimal solution $x^* \in \mathcal{X}_P$ there is a solution $\bar{x} \in \mathcal{X}_{\epsilon P}$ that ϵ-approximates x^*. On the other hand, we do not violate Definition 3.1 if there are "more" points in the ϵ-approximate Pareto set than necessary as long as we find an approximate solution for every Pareto optimal solution. In other words, if we have an approximation for $\mathcal{X}_{wP}$, then it is also an approximation for $\mathcal{X}_P$ and we will not care about unneeded points as long as our approximation remains of polynomial size.

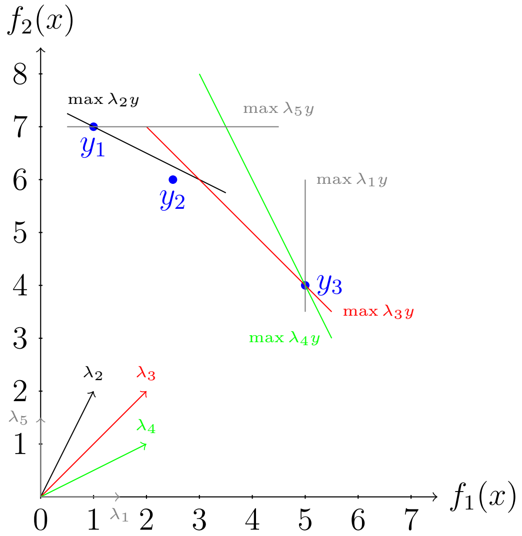
Returning to Algorithm 1, it is clear that the larger we choose M, which means that we loop over more distinct weights, the more accurate becomes the returned solution set in terms of an approximate Pareto set. Of course, it is not possible to loop over all non-negative weights due to complexity issues. Hence, we need to restrict to bounded weights which causes a loss of accuracy in the returned set. However, the main result of this section will state that by using bounded weights we can find an ϵ-approximate Pareto set.

The following definition is concerned with solutions that are optimal for weights whose components are bounded by a positive integer M.

Definition 5.7 Let $M \in \mathbb{N}$. A feasible solution $x^* \in \mathcal{X}$ is called *M-enabled* if there exists a vector $\lambda \in \{0, \dots, M\}^k \setminus \{0\}$ such that x^* is an optimal solution for the weighted sum scalarization $\max_{x \in \mathcal{X}} \sum_{i=1}^{k} \lambda_i f_i(x)$. In this case, we will also say that the image $f(x^*)$ in criterion space is M-enabled.

Example 5.1 Consider the feasible set in criterion space $\mathcal{Y}_N = \{y_1, y_2, y_3\}$ with $y_1 = (1, 7)$, $y_2 = (2.5, 6)$ and $y_3 = (5, 4)$. (See Figure 5.4.) (At this moment we do not care about $\mathcal{X}$, but assume that we also know the preimages $x_1, x_2, x_3 \in \mathcal{X}$ with $y_i = (f_1(x_i), f_2(x_i))$.) Furthermore, let $M := 2$. Then the bounded weight vectors are $\lambda_1 = (1, 0), \lambda_2 = (1, 2), \lambda_3 = (1, 1), \lambda_4 = (2, 1)$ and $\lambda_5 = (0, 1)$. For $M = 2$, the non-dominated point y_2 is not M-enabled, since $\lambda_1 \cdot y_3 = 5 > \lambda_1 \cdot y_2 = 2.5$, $\lambda_2 \cdot y_1 = 15 > 14.5 = \lambda_2 \cdot y_2$, $\lambda_3 \cdot y_3 = 9 > 8.5 = \lambda_3 \cdot y_2, \lambda_4 \cdot y_3 = 14 > 11 = \lambda_4 \cdot y_2$ and $\lambda_5 \cdot y_1 = 7 > \lambda_5 \cdot y_2 = 6$. Hence, by the method of weighted sum scalarization with bounded weight vectors $\lambda \in \{0, 1, 2\}^2 \setminus \{0\}$ we would not find the Pareto optimal solution x_2. On the other hand, letting $M = 7$ with $\bar{\lambda} = (5, 7)$ we would find x_2, since $\bar{\lambda} \cdot y_2 = 54.5 > 54 = \bar{\lambda} \cdot y_1 > 53 = \bar{\lambda} \cdot y_3$.

Figure 5.4: y_2 is not M-enabled for $M = 2$.



The next definition is concerned with solutions whose components do not differ to much from each other.

Definition 5.8 Let $y = (y_1, \ldots, y_k) \in \mathbb{R}^k_+$. Then y is called 2-*balanced* if all components of y are within a ratio of 2 of each other, that is, if $y_i/y_j \leq 2$ for all $i, j \in \{1, \ldots, k\}$.

Considering an optimization problem with objective functions f_i $(i = 1, \ldots, k)$ we will say that a feasible solution $x \in \mathcal{X}$ is 2-balanced if all components of $f(x)$ are within a ratio of 2 of each other. These 2-balanced feasible solutions will play an important role in the forthcoming Lemma 5.12. However, before we are able to state and prove Lemma 5.12 we need a little more preparatory work. We begin with the definition of a cone.

Definition 5.9 A subset $C \subseteq \mathbb{R}^k$ is called a *cone* if $\alpha c \in C$ for all $c \in C$ and for all $\alpha > 0$. A cone $C \subseteq \mathbb{R}^k$ is called *convex* if $\lambda c_1 + (1 - \lambda)c_2 \in C$ for all $c_1, c_2 \in C$ and for all $0 < \lambda < 1$. A *rotational cone* $\bar{R} \subseteq C$ of a convex cone C contains all vectors inscribed in C whose angle with the axis R of $\bar{R}$ is less than or equal to a fixed angle α. A *maximum rotational cone* $\bar{R} \subseteq C$ of a convex cone C is a rotational cone such that α is maximum.

Figure 5.5: In $\mathbb{R}^2$ the maximum rotational cone $\bar{R}$ of a convex cone C coincides with C.

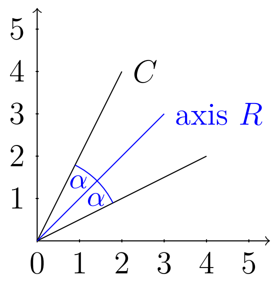
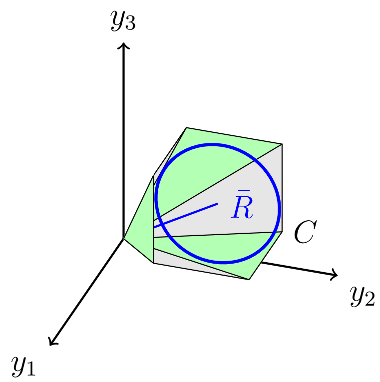


Figure 5.6: In general, in $\mathbb{R}^3$ the maximum rotational cone $\bar{R}$ of a convex cone C does not coincide with C.



Next, we state the so-called Carathéodory theorem which will also be of use in the proof of Lemma 5.12.

Theorem 5.10
Let $S \subseteq \mathbb{R}^k$. If $\bar{x}$ is a point in the convex hull of S, then there exist $x_1, \ldots, x_{k+1} \in S$ such that $\bar{x}$ can be represented as a convex combination of $x_1, \ldots, x_{k+1}$.

For a proof we refer the reader to [11].

Furthermore, we will make use of the following inequality in the forthcoming Lemma.

Proposition 5.11
Let $y^1, \ldots, y^{k+1} \in \mathbb{R}^k_{\geq}$ and let $a_1, \ldots, a_{k+1} \in \mathbb{R}_{\geq}$ with $\sum\limits_{j=1}^{k+1} a_j = 1$. Then $\|\sum\limits_{j=1}^{k+1} a_j y^j\| \geq \frac{1}{\sqrt{k+1}}\|y^J\|$ where $J \in \underset{j\in\{1,\ldots,k+1\}}{\arg\min} \|y^j\|$.

Proof:

$$\|\sum_{j=1}^{k+1} a_j y^j\|^2 = \sum_{i=1}^{k}\left(\sum_{j=1}^{k+1} a_j y_i^j\right)^2 \geq \sum_{i=1}^{k}\sum_{j=1}^{k+1}(a_j y_i^j)^2 =$$
$$\sum_{j=1}^{k+1} a_j^2 \sum_{i=1}^{k}(y_i^j)^2 = \sum_{j=1}^{k+1} a_j^2\|y^j\|^2 \geq \sum_{j=1}^{k+1} a_j^2\|y^J\|^2 \geq \frac{1}{k+1}\|y^J\|^2$$

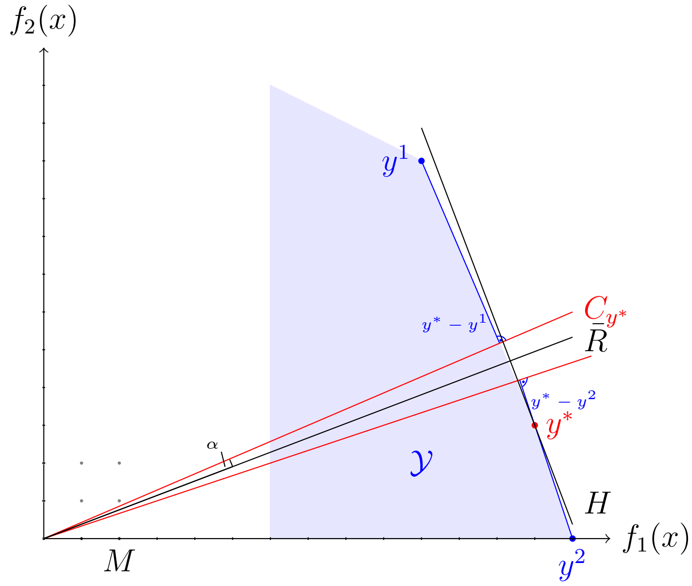
where $J \in \underset{j\in\{1,\ldots,k+1\}}{\arg\min} \|y^j\|$. The last inequality is due to the following result. Consider the problem $\min \sum_{i=1}^{k+1} a_i^2$ s.t. $\sum_{i=1}^{k+1} a_i = 1, a_i \geq 0$. Then an optimal solution is given by $a_i = \frac{1}{k+1}$ for $i = 1, \ldots, k+1$. In order to see this consider the function $L(a_1, \ldots, a_{k+1}, \lambda) = \sum_{i=1}^{k+1} a_i^2 + \lambda(\sum_{i=1}^{k+1} a_i - 1)$. Then $\nabla L(a_1, \ldots, a_{k+1}, \lambda) = 0$ yields $a_i = \frac{1}{k+1}$ for $i = 1, \ldots, k+1$. □

We have now enough ingredients to state the announced lemma.

Lemma 5.12
Given a convex domain $\mathcal{X}$, non-negative linear objectives $f_1, f_2, \ldots, f_k$ and $\epsilon > 0$. Set $\bar{\epsilon} \coloneqq 1 - \frac{1}{1+\epsilon}$ and let $M \coloneqq \lceil \frac{4k^4}{\bar{\epsilon}} \rceil$. If $x^* \in \mathcal{X}_P$ is a feasible Pareto optimal solution that is 2-balanced, then there are M-enabled solutions in $\mathcal{X}$ that have a convex combination $\bar{x} \in \mathcal{X}$ which ϵ-approximates x^*.

Proof: If $x^* \in \mathcal{X}$ is M-enabled itself, we are done. So, let us consider the case in which x^* is not M-enabled. For the sake of convenience, we set $y^* \coloneqq (f_1(x^*), \ldots, f_k(x^*))$. Let $\mathcal{Y}_M$ be the set of all M-enabled points in criterion space, that is, $\mathcal{Y}_M = \{(f_1(x'), \ldots, f_k(x')) : x' \in \underset{x\in\mathcal{X}}{\arg\max} \sum_{j=1}^{k} \lambda_j f_j(x)$ for some $\lambda \in \{0, \ldots, M\}^k \setminus \{0\}\}$. Next, we consider the k-dimensional convex cone $C_{y^*} := \{\lambda \in \mathbb{R}^k_* : \sum_{j=1}^{k} \lambda_j y_j^* \geq \sum_{j=1}^{k} \lambda_j y_j \forall y \in \mathcal{Y}_M\}$, the set of all non-negative weight vectors (excluding the zero vector) on the objectives for which x^* is at least as good as the preimages of all elements in $\mathcal{Y}_M$.

Figure 5.7: The two points y^1 and y^2 are M-enabled whereas y^* is non-dominated but not M-enabled.



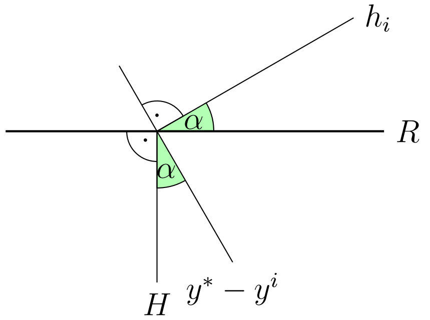
Since the image of a convex set under a linear mapping is again a convex set, it holds that the feasible set in criterion space is $\mathbb{R}^k_{\geq}$-convex. Furthermore, since x^* is Pareto optimal,

C_{y^*} is non-empty due to Corollary 4.11. On the other hand, since x^* is not M-enabled, C_{y^*} does not contain a integral weight vector whose coordinates are all between 0 and M. Consider now the maximum rotational cone $\bar{R}$ that is inscribed in the convex cone C_{y^*}.

It follows that $\tan(\alpha) < \frac{1}{M}$. Otherwise the maximum rotational cone $\bar{R}$ contains an integer point whose coordinates are all between 0 and M, and so does the cone C_{y^*}, which yields a contradiction. It clearly holds that $\frac{1}{M} \leq \frac{\sqrt{k}}{M}$ for $k \in \mathbb{N} \setminus \{0\}$ and $\arctan(x) \leq x$ for $x \in \mathbb{R}_{\geq}$. It follows that $\alpha = \arctan(\tan(\alpha)) \leq \arctan(\frac{\sqrt{k}}{M}) \leq \frac{\sqrt{k}}{M}$, that is, the angle α of the rotational cone $\bar{R}$ is less than $\frac{\sqrt{k}}{M}$.

Next, we consider the hyperplanes of C_{y^*}, denoted by $h_1, \ldots, h_m$, that are touched by $\bar{R}$. For each hyperplane h_i $(i = 1, \ldots, m)$ there exist a point $y^i = (y_1^i, \ldots, y_k^i) \in \mathcal{Y}_M$ and a weight vector $\lambda^i = (\lambda_1^i, \ldots, \lambda_k^i) \in \mathbb{R}_*^k$ such that $\sum_{j=1}^{k} \lambda_j^i y_j^i = \sum_{j=1}^{k} \lambda_j^i y_j^*$ implying that the line (segment) between y^* and y^i is normal to the hyperplane h_i for each $i \in \{1, \ldots, m\}$. Now consider the hyperplane, denoted by H, that includes y^* and that is normal to the axis R of the maximum rotational cone $\bar{R}$. Then it holds that the angle between the line $y^* - y^i$ and H is less than $\frac{\sqrt{k}}{M}$ for $i = 1, \ldots, m$. (See Figure 5.8.)

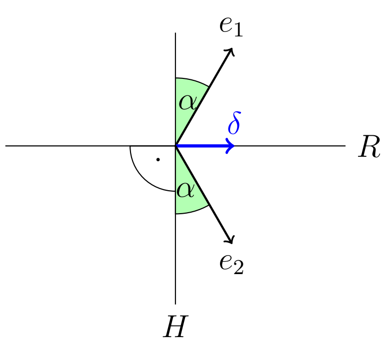
Figure 5.8



Due to the fact that H and R are perpendicular as well as h_i and $y^* - y^i$, the angle between H and $y^* - y^i$ coincides with α.

The claim is now that there is a point $\bar{y}$ that is a convex combination of the non-dominated points $y^1, \ldots, y^m$ and that is very close to y^*. To obtain this result we consider the unit vectors e_i which are collinear with the line (segment) between y^* and y^i $(i = 1, \ldots, m)$.

Figure 5.9



Since $\|e_1\| = \|e_2\| = 1$ and since the angle α is less than $\frac{\sqrt{k}}{M}$, we can estimate the upper bound of $\|\delta\|$ by $\frac{\sqrt{k}}{M}$.

They have a convex combination $\delta = \sum_{j=1}^{m} a_j e_j$ (with $\sum_{j=1}^{m} a_j = 1$ and $a_j \geq 0$ for $j = 1, \ldots, m$) that is a vector along the axis R of the maximum rotational cone $\bar{R}$. (See Figure 5.9.) It holds that $\|\delta\| \leq \sin(\frac{\sqrt{k}}{M}) \leq \frac{\sqrt{k}}{M}$. Scaling the vectors $y^* - y^j$ ($j = 1, \ldots, m$) yields a convex combination $\Delta = \sum_{j=1}^{m} \frac{a_j(y^*-y^j)}{\|y^*-y^j\|C}$, where C is a normalizing constant $C = \sum_{j=1}^{m} \frac{a_j}{\|y^*-y^j\|}$ and a_j ($j = 1, \ldots, m$) comes from the convex combination δ. Notice that Δ is also a vector along the axis of $\bar{R}$ and observe that $\|\Delta\| = \frac{\|\delta\|}{C}$. Furthermore, notice that $\sum_{j=1}^{m} \frac{a_j}{\|y^*-y^j\|C} = 1$. We set $\bar{y} := y^* - \Delta$. Then it holds that

$$\begin{aligned}
\bar{y} &= y^* - \sum_{j=1}^{m} \frac{a_j(y^* - y^j)}{\|y^* - y^j\|C} \\
&= y^* - \sum_{j=1}^{m} \frac{a_j y^*}{\|y^* - y^j\|C} + \sum_{j=1}^{m} \frac{a_j y^j}{\|y^* - y^j\|C} \\
&= y^* - y^* \sum_{j=1}^{m} \frac{a_j}{\|y^* - y^j\|C} + \sum_{j=1}^{m} \frac{a_j y^j}{\|y^* - y^j\|C} \\
&= \sum_{j=1}^{m} \frac{a_j y^j}{\|y^* - y^j\|C}.
\end{aligned}$$

Observe that $\bar{y}$ is a convex combination of the M-enabled points $y^1, \ldots, y^m$. It will be the desired approximation of y^*. In order to obtain the approximation result consider the relative error

$$\frac{\|y^* - \bar{y}\|}{\|\bar{y}\|} = \frac{\|\Delta\|}{\|y^* - \Delta\|} = \frac{\|\delta\|}{C\|\sum_{j=1}^{m} \frac{a_j y^j}{\|y^*-y^j\|C}\|} \leq \frac{\sqrt{k}}{M} \frac{1}{\|\sum_{j=1}^{m} \frac{a_j y^j}{\|y^*-y^j\|}\|} \tag{5.1}$$

Consider the vector $v := \sum_{j=1}^{m} a_j \frac{y^j}{\|y^*-y^j\|}$. Since $a_1, \ldots, a_m \in \mathbb{R}^k_{\geq}$ with $\sum_{j=1}^{k} a_j = 1$, it holds that the vector v is contained in the convex hull of $\frac{y^1}{\|y^*-y^1\|}, \ldots, \frac{y^m}{\|y^*-y^m\|} \in \mathbb{R}^k_{\geq}$. Then Caratheodory's theorem (Theorem 5.10) implies that v can be represented as a convex combination of $k+1$ of the $\frac{y^j}{\|y^*-y^j\|}$'s. Without loss of generality let these $k+1$ vectors be $\frac{y^1}{\|y^*-y^1\|}, \ldots, \frac{y^{k+1}}{\|y^*-y^{k+1}\|}$. Rewriting and applying Proposition 5.11 yields then

$$\|\sum_{j=1}^{m} a_j \frac{y^j}{\|y^* - y^j\|}\| = \|\sum_{j=1}^{k+1} \bar{a}_j \frac{y^j}{\|y^* - y^j\|}\| \geq \frac{1}{\sqrt{k+1}} \frac{\|y^J\|}{\|y^* - y^J\|}$$

where $J \in \underset{j \in \{1,\ldots,k+1\}}{\arg\min} \frac{\|y^j\|}{\|y^*-y^j\|}$. Returning to (5.1), we get

$$\frac{\|y^* - \bar{y}\|}{\|\bar{y}\|} \leq \frac{\sqrt{k}}{M} \frac{1}{\|\sum_{j=1}^{m} \frac{a_j y^j}{\|y^*-y^j\|}\|} \leq \frac{\sqrt{k}}{M} \frac{\sqrt{k+1}\|y^* - y^J\|}{\|y^J\|} \leq \frac{k+1}{M}(1 + \frac{\|y^*\|}{\|y^J\|}) \tag{5.2}$$

Since y^J is non-dominated, there is a component $L \in \{1, \dots, k\}$ with $y^*_L \leq y^J_L$. Moreover, since y^* is 2-balanced, it follows that

$$\frac{\|y^*\|}{\|y^J\|} \leq \frac{\sqrt{(k-1)(2y^*_L)^2 + (y^*_L)^2}}{\|y^J\|} \leq \frac{\sqrt{(2k-1)^2(y^J_L)^2}}{\|y^J\|} \leq (2k-1).$$

Hence, it holds that

$$\frac{\|y^* - \bar{y}\|}{\|\bar{y}\|} \leq \frac{2(k^2+k)}{M}.$$

By construction, Δ is a vector along the axis R of the maximum rotational cone $\bar{R}$. $\bar{y} = y^* - \Delta$ implies that $\bar{y}_i \leq y^*_i$ for all $i \in \{1, \dots, k\}$. Otherwise, y^* would be dominated by $\bar{y}$ yielding a contradiction. It follows that $\|\bar{y}\| \leq \|y^*\|$.
Hence, for every $i \in \{1, \dots, k\}$

$$\begin{aligned}
y^*_i - \bar{y}_i &\leq \|y^* - \bar{y}\| \\
&\leq \frac{2(k^2+k)}{M}\|\bar{y}\| \\
&\leq \frac{2(k^2+k)}{M}\|y^*\| \\
&= \frac{2(k^2+k)}{M}\sqrt{(y^*_1)^2 + \ldots + (y^*_k)^2} \\
&\leq \frac{2(k^2+k)}{M}\sqrt{(k-1)(2y^*_i)^2 + (y^*_i)^2} \\
&\leq \frac{2(k^2+k)}{M}\sqrt{4k(y^*_i)^2} \\
&\leq \frac{4k^4}{M} y^*_i
\end{aligned}$$

where $k \geq 2$. Since $M := \lceil \frac{4k^4}{\bar{\epsilon}} \rceil$, it follows that $\bar{\epsilon} \geq \frac{4k^4}{M}$. Notice that $\bar{\epsilon} \in (0, 1)$, since $\epsilon > 0$. Therefore it holds that

$$\begin{aligned}
y^*_i &\leq \frac{1}{1 - 4k^4/M}\bar{y}_i \\
&\leq \frac{1}{1 - \bar{\epsilon}}\bar{y}_i \\
&= (1+\epsilon)\bar{y}_i \text{ for all } i = 1, \dots, k.
\end{aligned}$$

□

In [2], Papadimitriou and Yannakakis claim in (5.2) a tighter bound of $\frac{\sqrt{k}}{M}\frac{\|y^* - y^J\|}{\|y^J\|}$. They state that

$$\|\sum_{j=1}^{m} \frac{a_j y^j}{\|y^* - y^j\|}\| \geq \min_{j \in \{1, \dots, m\}} \frac{\|y^j\|}{\|y^* - y^j\|}.$$

We could not follow their reasoning at this point, since this inequality does not hold for arbitrary vectors. Consider $a_1 = a_2 = 0.5$ and $\frac{y^1}{\|y^* - y^1\|} = (1, 0)$ and $\frac{y^2}{\|y^* - y^2\|} = (0, 1)$ as a counterexample.

Lemma 5.12 stated that for every 2-balanced Pareto optimal solution $x^* \in \mathcal{X}$ we can find an ϵ-approximate solution $\bar{x} \in \mathcal{X}$ by only using weight vectors whose components are bounded by $M(\epsilon)$. Of course, it is a restriction to assume that all Pareto optimal solutions are 2-balanced. Nevertheless, the following Theorem 5.14 will show that we can find an ϵ-approximate Pareto set by looping over more sophisticated chosen weight vectors whose number is nonetheless polynomially bounded. The intuition behind this result is the following: If x^* is a Pareto optimal solution that is not 2-balanced, then we can turn $(f_1(x^*), \dots, f_k(x^*))$ into a 2-balanced point $(w_1 f_1(x^*), \dots, w_k f_k(x^*))$ by scaling. Thus, Lemma 5.12 can then be applied to $(w_1 f_1(x^*), \dots, w_k f_k(x^*))$ implying that there is a convex combination $\bar{x}$ of M-enabled solutions $x^1, \dots, x^m$ which ϵ-approximates $(f_1(x^*), \dots, f_k(x^*))$. The total weights needed to find these M-enabled solutions $x^1, \dots, x^m$ are the scaling weights w_i times the scalarization factors λ_i. Then, by taking the convex hull of the x^i's, we find $\bar{x}$. Moreover, we will only need a polynomial number of weight vectors to approximate all Pareto optimal solutions.

The following proposition guarantees the existence of the scaling weights for solutions with strictly positive objective values.

Proposition 5.13
Given a multi-criteria maximization problem P with domain $\mathcal{X}$ and non-negative objective functions $f_1, \dots, f_k$. Assume that the objectives f_i $(i = 1, \dots, k)$ fulfill the prerequisite condition C^*. Let $x^* \in \mathcal{X}$ be a feasible solution whose objective values $f_i(x^*)$ $(i \in \{1, \dots, k\})$ are all strictly positive. Then there exist weigths $w_i \in \{2^m : m = 0, 1, \dots, 2p(|I|)\}$ for $i = 1, \dots, k$ such that

$$\frac{w_i f_i(x^*)}{w_j f_j(x^*)} \leq 2$$

for all $i, j \in \{1, \dots, k\}$.

Proof: It holds that $f_i(x^*)$ $(i = 1, \dots, k)$ is bounded from below by $2^{-p(|I|)}$ and from above by $2^{p(|I|)}$ for some polynomial p by C^*. Let $l \in \underset{i \in \{1,\dots,k\}}{\arg\max} f_i(x^*)$. Then there exists $r_j \in [0, 2p(|I|)]$ such that

$$1 \leq \frac{f_l(x^*)}{f_j(x^*)} = 2^{r_j}$$

for $j = 1, \dots, k$. Furthermore, it holds that $f_i(x^*)/f_j(x^*) = 2^{r_j - r_i}$ for all $i, j \in \{1, \dots, k\}$. Set the weights $w_i \coloneqq 2^{\lfloor r_j \rfloor}$ for $i = 1, \dots, k$. Then it holds that

$$\frac{w_i f_i(x^*)}{w_j f_j(x^*)} = 2^{\lfloor r_i \rfloor - \lfloor r_j \rfloor} \cdot 2^{r_j - r_i} \leq 2^{\lfloor r_i \rfloor - \lfloor r_j \rfloor} \cdot 2^{\lfloor r_j \rfloor + 1 - \lfloor r_i \rfloor} = 2$$

for all $i, j \in \{1, \dots, k\}$. Observe that $w_i \in \{2^m : m = 0, 1, \dots, 2p(|I|)\}$ for all $i \in \{1, \dots, k\}$. □

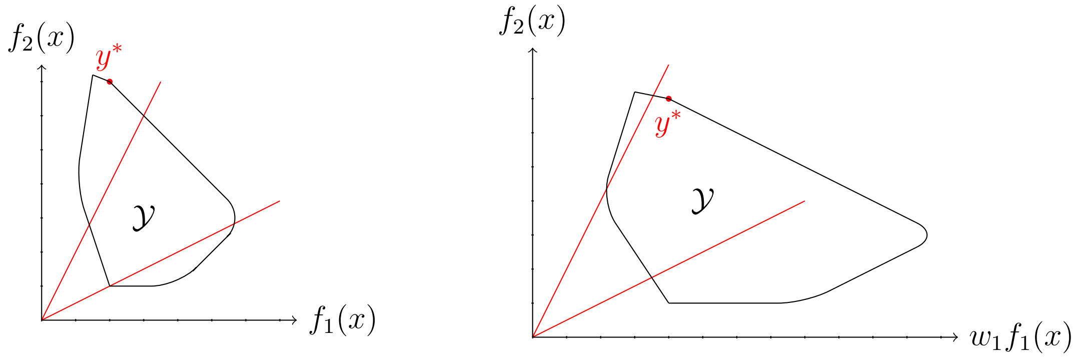


Figure 5.10: A non-dominated point that is not 2-balanced can be turned into a 2-balanced point by multiplying its components with convenient weights.

Let $M \in \mathbb{N}$. Then construct the set of weights W as follows:

$$W \coloneqq \{0, 1, \dots, M\} \cup \bigcup_{w \in \{2^l : l = 1,2,3,\dots,2p(|I|)\}} w \cdot \{\lceil M/2 \rceil, \dots, M\}. \tag{5.3}$$

The intuition behind the set of weights W is given by the idea of applying Lemma 5.12 on a scaled point $(w_1 f_1(x^*), \dots, w_k f_k(x^*))$. Observe that optimal solutions for the weighted sum scalarization problem $\max_{x \in \mathcal{X}} \sum_{i=1}^{k} \lambda_i w_i f_i(x)$ with $\lambda \in \{0, 1, \dots, M\}^k$ are also optimal solutions to $\max_{x \in \mathcal{X}} \sum_{i=1}^{k} \bar{\lambda}_i f_i(x)$ where $\bar{\lambda}_i \coloneqq w_i \lambda_i$ for $i \in \{1, \dots, k\}$.

Notice that the smallest $\lceil \frac{M}{2} \rceil$ elements of $2^l \cdot \{0, 1, \dots, M\}$ are already contained in $2^{l-1} \cdot \{0, 1, \dots, M\}$ for $l \in \mathbb{N} \setminus \{0\}$. Hence, $w \cdot \{\lceil M/2 \rceil, \dots, M\}$ in (5.3) suffices.

Algorithm 2:

---

let $\epsilon > 0$ and set $M \coloneqq \lceil \frac{4k^4}{1 - \frac{1}{1+\epsilon}} \rceil$
construct the set of weights W as described in (5.3)
set $S = \{\}$
foreach $\lambda \in W^k \setminus \{0\}$ **do**
    find an optimum x^* for the weighted sum scalarization $\max_{x \in \mathcal{X}} \sum_{i=1}^{k} \lambda_i f_i(x)$
    insert x^* into S
end
return the set S of all optima x^* thus found

Theorem 5.14
Given a convex maximization problem P with domain $\mathcal{X}$, non-negative linear objectives $f_1, f_2, \dots, f_k$ and $\epsilon > 0$. Then, by applying Algorithm 2, we can efficiently find a set S which consists of weakly Pareto optimal solutions and whose size is bounded by a polynomial in the input size and in $\frac{1}{\epsilon}$. Moreover, S has the property that (the upper envelope of) its convex hull constitutes an ϵ-approximate Pareto set $\mathcal{X}_{\epsilon P}$ for $\mathcal{X}_P$.

Proof: At first we show that the number of weight vectors $|W^k|$ in Line 2 of Algorithm 2 is polynomial in $\frac{1}{\epsilon}$ and in the input size (where k is assumed to be fixed). The set of weights W is defined as

$$W \coloneqq \{0, 1, \ldots, M\} \cup \bigcup_{w \in \{2^l : l=1,2,3,\ldots,2p(|I|)\}} w \cdot \{\lceil M/2 \rceil, \lceil M/2 \rceil + 1, \ldots, M\}.$$

Hence, $|W| = (M+1) + 2p(|I|)\lceil M/2 \rceil$ which is in $O(2p(|I|)M) = O(2p(|I|)\frac{4k^2}{1-\frac{1}{1+\epsilon}}) = O(8p(|I|)\frac{k^2(1+\epsilon)}{\epsilon})$. In Line 2 of Algorithm 2 we consider the k-ary Cartesian product W^k. By the estimate of $|W|$, it follows that the size of W^k is in $O((8p(|I|)\frac{k^2(1+\epsilon)}{\epsilon})^k)$ which is polynomial in the input size and in $\frac{1}{\epsilon}$, since we consider the number of objectives k to be fixed.

Next, consider a Pareto optimal solution $x^* \in \mathcal{X}$.
First case: Assume that the objective values $(f_1(x^*), \ldots, f_k(x^*))$ of x^* are all strictly positive. Then Proposition 5.13 yields scaling weights $w_i \in \{2^m : m = 0, 1, \ldots, 2p(|I|)\}$ for $i = 1, \ldots, k$ such that $(w_1 f_1(x^*), \ldots, w_k f_k(x^*))$ is 2-balanced. Applying Lemma 5.12 yields M-enabled solutions $x^1, \ldots, x^m$. For each $i \in \{1, \ldots, m\}$, x^i is an optimum solution to $\max_{x\in\mathcal{X}} \sum_{j=1}^{k} \lambda_i w_i f_i(x)$ where $\lambda \in \{0, 1, \ldots, M\}^k$. For each $i \in \{1, \ldots, m\}$, it holds that $w_i \lambda_i \in W$, since $w_i \in \{2^m : m = 0, 1, \ldots, 2p(|I|)\}$ and $\lambda_i \in \{0, 1, \ldots, M\}$. Hence, for each $i \in \{1, \ldots, m\}$, a solution $\hat{x}^i$ with the same objective values as x^i is found at some point (see Line 2 and Line 2 in Algorithm 2). Let $y^i \coloneqq (w_1 f_1(x^i), \ldots, w_k f_k(x^i))$ for $i = 1, \ldots, m$. From Lemma 5.12 we know that there is a convex combination $\bar{y}$ of the M-enabled points $y^1, \ldots, y^m$ such that

$$w_i f_i(x^*) \leq (1+\epsilon)\bar{y}_i \tag{5.4}$$

for all $i \in \{1, \ldots, k\}$. Since $f_1, \ldots, f_k$ are linear and $\mathcal{X}$ is convex, there is a convex combination $\bar{x} \in \mathcal{X}$ of $x^1, \ldots, x^m$ such that $w_i f_i(\bar{x}) = \bar{y}_i$ for $i = 1, \ldots, k$. From (5.4) it follows that

$$f_i(x^*) \leq (1+\epsilon) f_i(\bar{x})$$

for $i = 1, \ldots, k$, since w_i is strictly positive for all $i \in \{1, \ldots, k\}$. Hence, $\bar{x}$ ϵ-approximates the Pareto-optimal solution x^*. We get $\bar{x}$ by taking the (upper envelope of the) convex hull of the returned solution set S.
Second case: Assume that the Pareto optimal solution x^* takes a value of zero for some objectives, that is, $f_j(x^*) = 0$ for $j \in Z \subset \{1, \ldots, k\}$. (Clearly, we neglect the case where $Z = \{1, \ldots, k\}$ which means that the zero vector is a non-dominated point.) Let $\bar{Z} \coloneqq \{1, \ldots, k\} \setminus Z$. Since $f_1, \ldots, f_k$ are assumed to be non-negative, it follows that every feasible solution $x \in \mathcal{X}$ ϵ-approximates x^* in objective f_j $(j \in Z)$ because $f_j(x^*) = 0 \leq (1+\epsilon) f_j(x)$ for all $x \in \mathcal{X}$. Therefore, a solution $\bar{x} \in \mathcal{X}$ which ϵ-approximates x^* in objective f_j $(j \in \bar{Z})$ ϵ-approximate x^* in all objectives. Consider the restricted problem in which we maximize only the objectives f_j $(j \in \bar{Z})$ and neglect the other objectives. For $j \in \bar{Z}$ we find scaling weights $w_j \in W$ such that $\frac{w_i f_i(x^*)}{w_j f_j(x^*)} \leq 2$ for all

$i, j \in \bar{Z}$. Without loss of generality let $\bar{Z} = \{1, \ldots, h\}$ with $h < k$. Applying Lemma 5.12 on $(w_1 f_1(x^*), \ldots, w_h f_h(x^*))$ yields a convex combination $\bar{x}$ of M-enabled solutions x^i $(i = 1, \ldots, m)$ which ϵ-approximates x^* in objectives $f_1, \ldots, f_h$ and hence in all objectives. Since x^i $(i = 1, \ldots, m)$ is M-enabled, it is optimal for

$$\max_{x \in \mathcal{X}} \sum_{j=1}^{h} \lambda_j^i w_j f_j(x) = \max_{x \in \mathcal{X}} (\sum_{j=1}^{h} \lambda_j^i w_j f_j(x) + \sum_{j=h+1}^{k} 0 f_j(x))$$

where $\lambda^i \in \{0, \ldots, M\}^h$. For $i = 1, \ldots, m$, the k-dimensional weight vector $\hat{w}^i = (\lambda_1^i w_1, \ldots, \lambda_h^i w_h, 0, \ldots, 0)$ is an element of W^k. Hence, by looping over W^k in Algorithm 2, we will find an M-enabled solution $\hat{x}^i$ with the same objective value as x^i for $i = 1, \ldots, m$. Again, since we take (the upper envelope of) the convex hull of the returned solution set S, we get $\bar{x}$. □

5.3.2 The linear discrete case

The following result was also noted in [2]. We present it here for the sake of completeness. Let P be a discrete optimization problem. For any instance I of P there is a corresponding set of feasible solutions $\mathcal{X}$. In the following, we assume that a feasible solution $x \in \mathcal{X}$ is a non-negative n-dimensional vector whose entries are bounded by a polynomial in n. Notice that for many combinatorial problems the entries are usually 0 or 1.

Definition 5.15 Let P be a discrete optimization problem. By the *exact version of* P we mean the following: Given an instance I of P, an objective function f and an integer $B \in \mathbb{N}$. Does there exist a feasible solution $x \in \mathcal{X}$ with $f(x) = B$?

Theorem 5.16
Given a discrete maximization problem P with domain $\mathcal{X} \subseteq \mathbb{N}^n$ and non-negative linear objectives $f_1, \ldots, f_k$. There is an FPTAS for P if there is an algorithm A for the exact version of P that runs in polynomial time in the magnitude of the coefficients of the given objective function.

Proof: By Theorem 5.6, there exists an FPTAS for P if and only if there is an FPTAS for the GAP problem. Given a k-tuple $(c_1, \ldots, c_k)$ and $\epsilon > 0$, either return a solution $x \in \mathcal{X}$ with $f_i(x) \geq c_i$ for $i = 1, \ldots, k$, or answer that there is no solution $x' \in \mathcal{X}$ with $f_i(x') \geq c_i(1 + \epsilon)$ for $i = 1, \ldots, k$. Since f_i $(i = 1, \ldots, k)$ is linear, f_i can be represented as $f_i(x) = \sum_{j=1}^{n} f_{ij} x_j$. Let $m := \max_{i=1,\ldots,n} \{x_i : x \in \mathcal{X}\}$ be the largest entry of a feasible solution. Notice that m is polynomial in n by our prerequisite. Let $\epsilon > 0$. Furthermore, let $r := \lceil nm/\epsilon \rceil$. For each $i = 1, \ldots, k$, define a new objective function

$$g_i := \begin{pmatrix} \min\{\lfloor f_{i1} r / c_i \rfloor, r\} \\ \vdots \\ \min\{\lfloor f_{in} r / c_i \rfloor, r\} \end{pmatrix}.$$

Consider a feasible solution $x \in \mathcal{X}$. If $g_i(x) = \sum_{j=1}^{n} \min\{\lfloor f_{ij} r / c_i \rfloor, r\} x_j \geq r$, then, a fortiori, $\sum_{j=1}^{n} \frac{f_{ij} r}{c_i} x_j \geq r$ and hence $\sum_{j=1}^{n} f_{ij} x_j \geq c_i$. On the other hand, if $g_i(x) =$

$\sum\limits_{j=1}^{n} \min\{\lfloor f_{ij}r/c_i \rfloor, r\}x_j < r$ holds, then $\sum\limits_{j=1}^{n} \frac{f_{ij}r}{c_i}x_j < r+n$ implying that $\sum\limits_{j=1}^{n} f_{ij}x_j < c_i(1+\frac{n}{r})$. Since $r \geq nm/\epsilon$, it follows that $f_i(x) < c_i(1+\epsilon)$.

Hence, if $g_i(x) \geq r$, then $f_i(x) \geq c_i$ and if $f_i(x) \geq c_i(1+\epsilon)$, then $g_i(x) \geq r$.
If there exists a feasible solution $x \in \mathcal{X}$ with $g_i(x) \geq r$ for all $i \in \{1, \ldots, k\}$, then $f_i(x) \geq c_i$ holds for $i = 1, \ldots, k$. If there exists no feasible solution $x \in \mathcal{X}$ with $g_i(x) \geq r$ for all $i \in \{1, \ldots, k\}$, then there exists no $x \in \mathcal{X}$ with $f_i(x) \geq c_i(1+\epsilon)$ for $i = 1, \ldots, k$. Thus, it suffices to determine whether there exists a solution $x \in \mathcal{X}$ with $g_i(x) \geq r$ for all $i \in \{1, \ldots, k\}$.
Observe that $g_i(x) \leq rnm$ for all $x \in \mathcal{X}$ and for all $i \in \{1, \ldots, k\}$. Let $M \coloneqq rnm + 1$. Each of the inequalities $g_i(x) \geq r$ $(i = 1, \ldots, k)$ can be reduced to the disjunction of polynomially many equalities.

$$g_i(x) \geq r \Leftrightarrow \neg(g_i(x) = 0 \vee g_i(x) = 1 \vee \ldots \vee g_i(x) = r-1)$$

This makes r^k combinations to check whether $g_1(x) \geq r, \ldots, g_k(x) \geq r$. Notice that r is polynomial in n and $1/\epsilon$.
k equalities can be combined into one by multiplying the i-th equality by M^{i-1} and adding the results. In other words, if $g_i(x) = l_i$ for all $i \in \{1, \ldots, k\}$ with $l_i \in \{0, \ldots, M-1\}$, then

$$g_i(x) = l_i \;\forall i \in \{1, \ldots, k\} \Leftrightarrow \sum_{i=1}^{k} M^{i-1}g_i(x) = \sum_{i=1}^{k} M^{i-1}l_i. \tag{5.5}$$

It is clear that "$\Rightarrow$" holds. In order to see that "$\Leftarrow$" also holds, suppose that there exists $\emptyset \neq J \subseteq \{1, \ldots, k\}$ with $g_j(x) \neq l_j$ for $j \in J$. From $\sum\limits_{i=1}^{k} M^{i-1}g_i(x) = \sum\limits_{i=1}^{k} M^{i-1}l_i$ it follows that $\sum\limits_{j \in J} M^{j-1}g_j(x) = \sum\limits_{j \in J} M^{j-1}l_j$ with $g_j(x) \neq l_j$ for all $j \in J$. Without loss of generality let $J = \{1, \ldots, h\}$ for some $h \leq k$. In other words,

$$M^0(g_1(x) - l_1) + \ldots + M^{h-2}(g_{h-1}(x) - l_{h-1}) = -M^{h-1}(g_h(x) - l_h). \tag{5.6}$$

On the other hand, since $|g_i(x) - l_i| \in \{1, \ldots, M-1\}$ for all $i \in \{1, \ldots, k\}$, we get for $M \geq 2$

$$\begin{aligned} &|(g_1(x) - l_1) + M(g_2(x) - l_2) + \ldots + M^{h-2}(g_{h-1}(x) - l_{h-1})| \leq \\ &(M-1)(M + M^2 + \ldots + M^{h-2}) < M^{h-1} \end{aligned}$$

which contradicts (5.6). Hence, (5.5) holds and we can combine k equalities into one. The objective function $\sum\limits_{i=1}^{k} M^{i-1}g_i(x)$ is a linear function with polynomially bounded coefficients. By our assumption there exists an algorithm A for the exact version of P that runs polynomial in the magnitude of the coefficients of the given objective function. By calling A r^k times with the integer $\sum\limits_{i=1}^{k} M^{i-1}l_i$ and the objective function $\sum\limits_{i=1}^{k} M^{i-1}g_i(x)$ as input, we can solve the GAP in polynomial time in n and in $1/\epsilon$. □

6 A dynamic programming framework and multi-criteria approximation scheme

The following chapter is based on [3]. G.J.Woeginger proposed a dynamic programming framework that guarantees the existence of a fully polynomial time approximation scheme for single-objective optimization problems. If a combinatorial optimization problem can be formulated via the proposed dynamic programming framework and additionally fulfills certain structural properties, then there exists an FPTAS for the considered optimization problem. In each iteration, the dynamic program maintains a set of states, the so-called state space. This state spaces represent feasible solutions computed so far. Then, in simple terms, at the end of each iteraton these state spaces are covered by boxes similar to the existence proof of approximate Pareto sets (see Theorem 5.4). In the course of the computaton only one state in each box is chosen for the next iteration. Thus, a superpolynomial growth of the size of the state spaces is avoided. An additional fulfillment of structural properties and conditions ensures that the occuring errors can be bounded as well. In the following, we will show that the result of G.J.Woeginger can be carried over to the multi-criteria case guaranteeing the existence of an FPTAS for multi-criteria optimization problems fitting into the proposed framework.

6.1 Preliminaries

Covering a set of states by boxes and selecting only one state among several out of each box makes it necessary to be able to compare the states among each other. Hence, relations play an important role in the proposed framework. The following definitions and results on the subject of relations are regarded as a recapitulation for the reader.

Definition 6.1 Let $\mathcal{S}$ be a set. A binary relation $\preceq \subseteq \mathcal{S} \times \mathcal{S}$ is called

reflexive if for any $s \in \mathcal{S}$ it holds that $s \preceq s$.

symmetric if for any $s, s' \in \mathcal{S} : s \preceq s' \Rightarrow s' \preceq s$.

anti-symmetric if for any $s, s' \in \mathcal{S} : s \preceq s' \land s' \preceq s \Rightarrow s = s'$.

transitive if for any $s, s', s'' \in \mathcal{S}$: $s \preceq s' \land s' \preceq s'' \Rightarrow s \preceq s''$.

A reflexive, anti-symmetric, and transitive relation is called a *partial order*. A relation is called *quasi order* if it is reflexive and transitive. A quasi order on $\mathcal{S}$ is called a *quasi-linear* *order* if any two elements of $\mathcal{S}$ are comparable.

An element $s \in \mathcal{S}$ is called *maximal* with respect to $\preceq$ if there is no element $s' \in \mathcal{S} \setminus \{s\}$ with $s \preceq s'$. An element $s \in \mathcal{S}$ is called *maximum* if $s' \preceq s$ holds for all $s' \in \mathcal{S}$. Notice that if a maximum element of a partially ordered set exists, then it is unique due to the anti-symmetry. The *trivial relation* on $\mathcal{S}$ is the relation $\{(s, s) : s \in \mathcal{S}\}$. The *universal relation* on $\mathcal{S}$ is the relation $\{(s, s') : s, s' \in \mathcal{S}\}$.

Example 6.1 Consider the imparity relation $\neq$ on $\mathbb{N}$. For all $x, y \in \mathbb{N}$ it holds that $x \neq y$ implies $y \neq x$. Hence, $\neq$ is a symmetric relation on $\mathbb{N}$.
Let $\mathcal{S} := \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{2, 3\}\}$ and consider the "subset of" relation $\subseteq$. Then for every $S \in \mathcal{S}$ it holds that $S \subseteq S$. Therefore $\subseteq$ is a reflexive relation on $\mathcal{S}$. For every $S, S' \in \mathcal{S}$ it holds that $S \subseteq S'$ and $S' \subseteq S$ implies $S = S'$. Hence, $\subseteq$ is an anti-symmetric relation on $\mathcal{S}$. Certainly, $\subseteq$ is also an example for a transitive relation. $\{1, 2\}$ and $\{2, 3\}$ are the two maximal elements of $\mathcal{S}$ with respect to $\subseteq$. However, there is no maximum element in $\mathcal{S}$ with respect to $\subseteq$.

The following result on partial orders and quasi-linear orders, respectively, will be of use in Section 6.3.

Proposition 6.2
For any binary relation $\preceq$ on a set $\mathcal{S}$, and for any finite subset $\mathcal{S}'$ of $\mathcal{S}$ the following holds:

(i) If $\preceq$ is a partial order, then there exists a maximal element in $\mathcal{S}'$.

(ii) If $\preceq$ is a quasi-linear order, then there exists at least one maximum in $\mathcal{S}'$.

Proof: (i) Let $\mathcal{S}'$ be an arbitrary finite but fixed subset of $\mathcal{S}$ and let $a \in \mathcal{S}'$ an arbitrary element of $\mathcal{S}'$. If there exists no element b in $\mathcal{S}' \setminus \{a\}$ with $a \preceq b$, then a is a maximal element. Otherwise take b and check if there exists an element $c \in \mathcal{S}' \setminus \{a, b\}$ with $b \preceq c$. This approach will lead to a maximal element because $\mathcal{S}'$ is finite and $\preceq$ is transitive.
(ii) Let $\mathcal{S}'$ be an arbitrary finite but fixed subset of $\mathcal{S}$ and let $a \in \mathcal{S}'$ an arbitrary element of $\mathcal{S}'$. We can distinguish two cases: a is either a maximum and we are done, or there is an element $b \in \mathcal{S}'$ such that $b \not\preceq a$. Since a and b are comparable due to the fact that $\preceq$ is a quasi-linear order, it holds then that $a \preceq b$. Then b is either a maximum or there is an element $c \in \mathcal{S} \setminus \{a\}$ with $b \not\preceq c$. Continuing this approach will lead to a maximum element because $\mathcal{S}'$ is finite and $\preceq$ is transitive. $\square$

The following proposition will be of use in Subsection 6.3.3. It will help us to achieve the desired approximation quality in terms of Pareto optimality.

Proposition 6.3
For any $0 \leq x \leq 1$ and for any real $m \geq 1$, it holds that $(1 + x/m)^m \leq 1 + 2x$.

Proof: Notice that for $m \geq 1$, $f(x) = (1 + x/m)^m$ is a twice differentiable function with $f''(x) = \frac{m-1}{m}(1 + \frac{x}{m})^{m-2} \geq 0$, that is, $f(x)$ is a convex function. The function $g(x) = 1 + 2x$ is a linear function with $f(0) = g(0)$ and $f(1) \leq g(1)$ for any real $m \geq 1$. $\square$

6.2 The dynamic programming framework

In the single-objective case a dynamic programming formulation will generally compute an optimal solution in the course of its finite iterations, although not necessarily in polynomial time due to the superpolynomial growth of the size of the state spaces. Two standard approaches exist for transforming a dynamic program into an efficiently running algorithm. The first approach is often called rounding-the-input-data technique and is probably due to Sahni [12]. The main idea in this approach is to round the input data of the instance. The aim is to bring the running time of the dynamic program down to polynomial time by making the resulting rounded problem easy to solve. The second approach is called trimming-the-state-space technique which is due to Ibarra and Kim [13]. The main idea of this approach is to iteratively reduce the cardinality of the state spaces by thinning out similar states. In other words, feasible solutions that differ only "slightly" from each other will be merged. Hence, the aim in the trimming-the-state-space technique is to reduce the size of the state spaces to polynomial size. This approach does not round any values of the input data. The presented dynamic programming framework is based on the latter approach. The crucial point in both approaches is the need to control the resulting error that occurs by rounding input values or trimming state spaces, respectively.

Throughout this section we consider a generic optimization problem P. We introduce four definitions that specify the problem representation and the relationship to its dynamic programming formulation DP.

The first definition is concerned with the structure of the input of the generic optimization problem P. It defines how the input is given.

Definition 6.4 The input is structured into n vectors $x_1, \ldots, x_n \in \mathbb{N}^\alpha$ in any instance I of P. All components of all vectors $x_i = (x_1^i, \ldots, x_\alpha^i)$ are encoded in binary.

The number α is a positive integer whose value may depend on the input I. The next definition is concerned with the structure of the dynamic program DP. It defines how the input is processed.

Definition 6.5 The dynamic program DP for problem P goes through n phases. The i-th phase ($i = 1, \ldots, n$) processes the input vector x_i and produces a set $\mathcal{S}_i$ of states. Any state in the state space $\mathcal{S}_i$ is a vector $s = (s_1, \ldots, s_\beta) \in \mathbb{N}^\beta$. The number β is a positive integer whose value depends on P but does not depend on any specific instance of P.

In other words, every state in the state space $\mathcal{S}_i$ corresponds to a solution to the subproblem specified by the partial input $x_1, \ldots, x_i$. Next, we define how the state spaces in the dynamic program DP are iteratively computed.

Definition 6.6 The set $\mathcal{F}$ is a finite set of mappings $\mathbb{N}^\alpha \times \mathbb{N}^\beta \to \mathbb{N}^\beta$. The set $\mathcal{H}$ is a finite set of mappings $\mathbb{N}^\alpha \times \mathbb{N}^\beta \to \mathbb{R}$. For every mapping $F \in \mathcal{F}$ there is a corresponding mapping $H_F \in \mathcal{F}$. In the initialization phase of DP the state space $\mathcal{S}_0$ is initialized by a finite subset of $\mathbb{N}^\beta$. In the i-th phase ($i = 1, \ldots, n$) of DP the state space $\mathcal{S}_i$ is obtained from the state space $\mathcal{S}_{i-1}$ via $\mathcal{S}_i = \{F(x_i, s) : F \in \mathcal{F}, s \in \mathcal{S}_{i-1}, H_F(x_i, s) \leq 0\}$.

The mappings in $\mathcal{F}$ compute the states of the state space $\mathcal{S}_i$ from the states of $\mathcal{S}_{i-1}$. The functions in $\mathcal{H}$ come into play in case of infeasible new states. They serve as a tool for keeping infeasible states out of the state space $\mathcal{S}_i$. (The reader is referred to Example 6.2 for an illustrating example and further details.)

The fourth definition is concerned with the occuring objective values.

Definition 6.7 The objective functions $f_i : \mathbb{N}^\beta \to \mathbb{N}$ $(i = 1, \ldots, k)$ are non-negative functions.

Definition 6.8 An optimization problem P is called *DP-simple* if it can be expressed via a dynamic programming formulation DP as described in Definition 6.4 - Definition 6.7.

The dynamic programming formulation for a DP-simple problem P looks as follows:

Algorithm 3: The dynamic programming formulation for a DP-simple optimization problem P

1 initialize $\mathcal{S}_0$
2 **for** $i = 1$ **to** n **do**
3   let $\mathcal{S}_i := \{\}$
4   **foreach** $s \in \mathcal{S}_{i-1}$ *and* $F \in \mathcal{F}$ **do**
5   **if** $H_F(x_i, s) \leq 0$ **then**
6   add $F(x_i, s)$ to $\mathcal{S}_i$
7   **end**
8   **end**
9 **end**
10 output Pareto set based on $\{s \in \mathcal{S}_n\}$ according to objective functions f_i $(i = 1, \ldots, k)$

Line 3 in Algorithm 3 just states to output a Pareto set according to the objective functions. Of course, to find a Pareto set out of a finite but probably huge set of feasible solutions is a non-trivial task. However, since the size of the state spaces $\mathcal{S}_i$ will generally grow exponentially due to the dynamic programming formulation (Line 3 - Line 3), the dynamic program of a DP-simple optimization problem cannot be expected to run in polynomial time, in general. Therefore we will not elaborate on how to find the Pareto set in Line 3 at this point because the algorithm will have already run in superpolynomial time before reaching Line 3. The problem to find a set of non-dominated points out of a set of solutions is particularly considered in the data management and database related community where it is called vector maximization problem and skyline query problem, respectively. For more information about vector maximization algorithms we refer the reader to [14] and [15].

Next, we show that the bicriteria Knapsack problem fits into the DP-simple dynamic programming formulation.

Example 6.2 Consider the bicriteria Knapsack problem: Given $p_i, v_i, w_i \in \mathbb{N}$ $(i =$

$1, \ldots, n$) and a weight limit W:

$$\max \sum_{i=1}^{n} p_i x_i, \max \sum_{i=1}^{n} v_i x_i$$
$$\text{s.t. } \sum_{i=1}^{n} w_i x_i \leq W, x_i \in \{0, 1\}.$$

The formulation in terms of the DP-simple formulation is as follows: Let $\alpha = 3$ and $\beta = 3$. Define the input vector $x_i = (p_i, v_i, w_i)$ for $i = 1, \ldots, n$. A state $s = (s_1, s_2, s_3) \in \mathcal{S}_i$ encodes a partial solution for the first i input vectors. The first coordinate s_1 stands for the total selected profit, s_2 stands for the total selected value, and s_3 corresponds to the total selected weight in the partial solution. The sets $\mathcal{F}$ and $\mathcal{H}$ contain two functions $F_1, F_2 : \mathbb{N}^\alpha \times \mathbb{N}^\beta \to \mathbb{N}^\alpha$ and $H_1, H_2 : \mathbb{N}^\alpha \times \mathbb{N}^\beta \to \mathbb{R}$, respectively.

$$F_1(p_k, v_k, w_k, s_1, s_2, s_3) = (s_1 + p_k, s_2 + v_k, s_3 + w_k)$$

$$H_1(p_k, v_k, w_k, s_1, s_2, s_3) = s_3 + w_k - W$$

$$F_2(p_k, v_k, w_k, s_1, s_2, s_3) = (s_1, s_2, s_3)$$

$$H_2 \equiv 0$$

The two objective functions f_1, f_2 are the projections on the first and second coordinate, respectively,

$$f_1(s_1, s_2, s_3) = s_1 \text{ and } f_2(s_1, s_2, s_3) = s_2.$$

For a state $s \in \mathcal{S}_{i-1}$ the newly computed state $F(x_i, s)$ will be contained in $\mathcal{S}_i$ if $H_F(x_i, s) \leq 0$ (see Definition 6.6). In terms of KNAPSACK an already choosen selection together with a currently considered new item is still feasible if the total selected weight does not violate the weight limit, that is, if $s_3 + w_i - W \leq 0$ holds. Due to the fact that feasible selections from the previous state space $\mathcal{S}_{i-1}$ are always contained in the new state space $\mathcal{S}_i$, the function H_2 is constant 0 which means that the state $F_2(x_i, s_1, s_2, s_3) = (s_1, s_2, s_3)$ will always be included in $\mathcal{S}_i$.

Consider the following instance with 4 items $x_1 = (1, 1, 1)$, $x_2 = (2, 2, 2)$, $x_3 = (5, 5, 5)$, $x_4 = (7, 7, 7)$ and weight limit $W = 12$. Then we get the following state spaces:

$$\mathcal{S}_0 = \{(0, 0, 0)\},$$

$$\mathcal{S}_1 = \{(0, 0, 0), (1, 1, 1)\},$$

$$\mathcal{S}_2 = \{(0, 0, 0), (1, 1, 1), (2, 2, 2), (3, 3, 3)\},$$

$$\mathcal{S}_3 = \mathcal{S}_2 \cup \{(5, 5, 5), (6, 6, 6), (7, 7, 7), (8, 8, 8)\},$$

$$\mathcal{S}_4 = \mathcal{S}_3 \cup \{(9, 9, 9), (10, 10, 10), (12, 12, 12)\}.$$

The dynamic programming formulation of a Knapsack instance with n items $x_i = (p_i, v_i, w_i)$ for $(i = 1, \ldots, n)$ looks as follows:

Algorithm 4: The DP-simple formulation for the considered knapsack problem

1 $\mathcal{S}_0 = \{(0, 0, 0)\}$ `/*initialize with empty selection*/`
2 **for** $i = 1$ **to** n `/*iterate through input vectors*/` **do**
3   let $\mathcal{S}_i := \{\}$
4   **foreach** $s = (s_1, s_2, s_3) \in \mathcal{S}_{i-1}$ **do**
5   **if** $s_3 + w_i - W \leq 0$ `/*test whether selection fulfills weight limit*/`
6   **then**
7   add $(s_1 + p_i, s_2 + v_i, s_3 + w_i)$ to $\mathcal{S}_i$ `/*add feasible selection to current state space*/`
8   **end**
9   **if** $0 \leq 0$ `/*always fulfilled*/`
10   **then**
11   add (s_1, s_2, s_3) to $\mathcal{S}_i$ `/*add feasible selection to current state space*/`
12   **end**
13   **end**
14 **end**
15 output Pareto set based on $\{s \in \mathcal{S}_n\}$ according to objective functions f_1 and f_2

In Line 4 of Algorithm 4 we initialize the state space $\mathcal{S}_0$. We start with the empty selection in which no item is taken into the Knapsack. Hence, $\mathcal{S}_0 = \{(0, 0, 0)\}$. In Line 4 we iterate through the input vectors x_i for $i = 1, \ldots, n$. At the beginning of each iteration i we set the new state space $\mathcal{S}_i = \{\}$. In Line 4 we iterate through all states of the previous state space $\mathcal{S}_{i-1}$. In Line 4 we check whether the currently considered state, that is, a feasible Knapsack selection together with the currently considered item x_i yields a new feasible selection. Mathematically formulated: The selection is feasible if the total selected weight of the previous selection s_3 plus the weight of the currently considered item w_i is less then or equal to the weight limit W. In this case we include the selection into $\mathcal{S}_i$ (see Line 4). Because a feasible Knapsack selection of $\mathcal{S}_{i-1}$ is always considered as a potential solution, it will be taken into the new state space $\mathcal{S}_i$ as well (see Line 4 and Line 4). Clearly, the size of the state spaces grows exponentially in each iteration. Hence, the dynamic program would not run in polynomial time in the size of the input.

6.3 Approximation by trimming

In the trimmed dynamic program we will merge states that are close to each other in order to reduce the size of the state space. While merging certain states we have to ensure that errors do not propagate uncontrollably. The next definition introduces the concept of $[D, \Delta]$-closeness which captures more formally the idea of states that differ only "slightly". We fix a vector $D = (d_1, \ldots, d_\beta)$ which will be called *degree vector*. $D \in \mathbb{N}^\beta$ depends on the generic problem P and on the DP formulation, but it does not depend on any specific instance I of P.

Definition 6.9 Let $D = (d_1, \ldots, d_\beta)$ be a vector in $\mathbb{N}^\beta$. For a real number $\Delta > 1$ and two vectors $s, s' \in \mathbb{N}^\beta$ with $s = (s_1, \ldots, s_\beta)$ and $s' = (s'_1, \ldots, s'_\beta)$, we say that s is $[D, \Delta]$-*close* to s' if

$$\Delta^{-d_i} \cdot s_i \leq s'_i \leq \Delta^{d_i} \cdot s_i \tag{6.1}$$

for $i = 1, \ldots, \beta$.

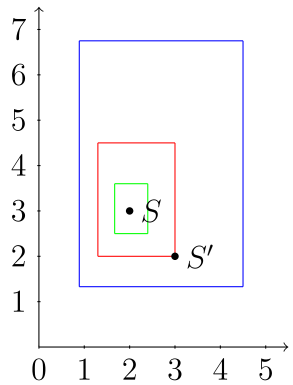
Notice that for any $\Delta > 1$ the relation of being $[D, \Delta]$-close is symmetric and reflexive. Clearly, if two vectors are $[D, \Delta]$-close to each other, then they must coincide in all coordinates $l \in \{1, \ldots, \beta\}$ for which the corresponding coordinate d_l in the degree vector is zero. Intuitively, two states s, s' are $[D, \Delta]$-close to each other if they are contained in a box whose size depends on D and Δ.

Example 6.3 Let $s = (2, 3)$, $s' = (3, 2)$, and $D = (0, 0) \in \mathbb{N}^2$. Then there exists no $\Delta > 1$ such that inequalities (6.1) hold. Hence, there is no $\Delta > 1$ such that s and s' are $[D, \Delta]$-close to each other for this particular degree vector D.
On the other hand, consider the degree vector $D = (1, 1) \in \mathbb{N}^2$. Then for $\Delta := \frac{3}{2}$ it holds that

$$\left(\frac{3}{2}\right)^{-1} \cdot s_1 \leq s'_1 \leq \frac{3}{2} \cdot s_1 \quad \text{and} \quad \left(\frac{3}{2}\right)^{-1} \cdot s_2 \leq s'_2 \leq \frac{3}{2} \cdot s_2.$$

Hence, s is $[D, \Delta]$-close to s' with $D = (1, 1)$ and $\Delta = \frac{3}{2}$, respectively. (See the red box in Figure 6.1.)

Figure 6.1: Three different boxes around the state $s = (2, 3)$: $[(1, 1), 1.2]$-box depicted by the green box, $[(1, 1), 1.5]$-box depicted by the red box and $[(2, 2), 1.5]$-box depicted by the blue box.



We will now introduce the concept of dominance between states which is common in dynamic programming. Consider two states s and s' which represent feasible solutions to the same subproblem given by the partial input $x_1, \ldots, x_i$ $(1 \leq i < n)$. Let t be any extended solution of s and t' be an extension of s'. In other words, t and t' are computed from s and s',respectively, and the residual input $x_{i+1}, \ldots, x_N$. Suppose that t' has an objective that is at least as good as the objective value of the extension t. In this case the state s is said to be dominated by the state s' and it should be clear that we would select s' in favour of s in the course of an iteration. In order to capture the concept of dominance between states, we introduce two binary relations $\preceq_{dom}$ and $\preceq_{qua}$ on $\mathbb{N}^\beta$. The dominance relation $\preceq_{dom}$ is a partial order on $\mathbb{N}^\beta$, and $\preceq_{qua}$ is a quasi-linear order on $\mathbb{N}^\beta$. In addition, $\preceq_{qua}$ is any extension of $\preceq_{dom}$, that is, $s \preceq_{dom} s'$ always implies $s \preceq_{qua} s'$. Next, we will state conditions that relate the functions in the dynamic programming framework to the concepts of $[D, \Delta]$-closeness and dominance, respectively. The following conditions on $\mathcal{F}$, on $\mathcal{H}$ and on the objective functions f_i ($i = 1, \ldots, k$) in DP will later ensure that the occuring error does not propagate uncontrollably.

The intuition behind Condition 1 indicates that two newly computed states arising from two nearby and related states should also be close and in relation to each other. If a state s' dominates a state s, then the newly computed states arising from s' and s, respectively, should also be in dominance relation to each other.

Condition 1 For any $\Delta > 1$, for any $F \in \mathcal{F}$, for any $x \in \mathbb{N}^\alpha$, and for any $s, s' \in \mathbb{N}^\beta$, the following holds:

(i) If s is $[D, \Delta]$-close to s' and if $s \preceq_{qua} s'$, then

   (a) $F(x, s) \preceq_{qua} F(x, s')$ holds and $F(x, s)$ is $[D, \Delta]$-close to $F(x, s')$, or

   (b) $F(x, s) \preceq_{dom} F(x, s')$ holds.

(ii) If $s \preceq_{dom} s'$, then $F(x, s) \preceq_{dom} F(x, s')$.

Suppose the newly computed state arising from a state s is feasible. Then, intuitively, newly computed states that arise from nearby and related states of s or dominating states of s, respectively, should also be feasible.

Condition 2 For any $\Delta > 1$, for any $H \in \mathcal{H}$, for any $x \in \mathbb{N}^\alpha$, and for any $s, s' \in \mathbb{N}^\beta$, the following holds:

(i) If s is $[D, \Delta]$-close to s' and if $s \preceq_{qua} s'$, then $H(x, s') \leq H(x, s)$.

(ii) If $s \preceq_{dom} s'$, then $H(x, s') \leq H(x, s)$.

Condition 3

(i) There exists an integer $g \geq 0$ (whose value only depends on the functions $f_1, \ldots, f_k$ and on the degree vector D) such that for any $\Delta > 1$ and for any $s, s' \in \mathbb{N}^\beta$ the following property holds:

   If s is $[D, \Delta]$-close to s' and if $s \preceq_{qua} s'$, then for all $i \in \{1, \ldots, k\}$ it holds that $f_i(s) \leq \Delta^g \cdot f_i(s')$ in case of maximization and $\Delta^{-g} \cdot f_i(s') \leq f_i(s)$ in case of minimization.

(ii) For any $s, s' \in \mathbb{N}^\beta$ with $s \preceq_{dom} s'$, it holds for all $i \in \{1, \ldots, k\}$ that $f_i(s') \geq f_i(s)$ in case of maximization and $f_i(s') \leq f_i(s)$ in case of minimization.

The following set of technical conditions ensure that the evaluation of the involved functions can be performed in polynomial time and that the lengths of the involved numbers do not mushroom.

Condition 4

(i) Every $F \in \mathcal{F}$ can be evaluated in polynomial time. Every $H \in \mathcal{H}$ can be evalutated in polynomial time. The functions f_i $(i = 1, \ldots, k)$ can be evaluated in polynomial time. The relation $\preceq_{qua}$ can be decided in polynomial time.

(ii) The cardinality of $\mathcal{F}$ is polynomially bounded in n and $\mathrm{ld}(\bar{x})$ where $\bar{x} := \sum_{i=1}^{n} \sum_{l=1}^{\alpha} x_l^i$.

(iii) For every instance I of P the state space $\mathcal{S}_0$ can be computed in time that is polynomially bounded in n and $\mathrm{ld}(\bar{x})$. As a consequence, also the cardinality of the state space $\mathcal{S}_0$ is polynomially bounded in n and $\mathrm{ld}(\bar{x})$.

(iv) For an instance I of P and for a coordinate l $(1 \leq l \leq \beta)$, let $V_l(I)$ denote the set of the values of the l-th components of all vectors in all state spaces $\mathcal{S}_i$ $(1 \leq i \leq n)$. Then the following holds for every instance I: for all coordinates l $(l = 1, \ldots, \beta)$, the natural logarithm of every value in $V_l(I)$ is bounded by a polynomial $\pi_1(n, \mathrm{ld}(\bar{x}))$. Equivalently, one may say that the length of the binary encoding of every value is polynomially bounded in the input size. Furthermore, for coordinates l with $d_l = 0$, the cardinality of $V_l(I)$ is bounded by a polynomial $\pi_2(n, \mathrm{ld}(\bar{x}))$.

The next definition combines the above mentioned conditions and the before mentioned relations of dominance and closeness.

Definition 6.10 A DP-simple optimization problem P is called *DP-benevolent* if and only if there exist a partial order $\preceq_{dom}$, a quasi-linear order $\preceq_{qua}$ and a degree vector D such that its dynamic programming formulation DP fulfills Condition 1 - Condition 4.

The following theorem states the central conclusion: DP-benevolent problems can be efficiently approximated.

Theorem 6.11
If an optimization problem P is DP-benevolent, then it has an FPTAS.

6.3.1 The trimmed dynamic program

The proof of Theorem 6.11 is based on several propositions that will be stated and combined in this section. The crucial point lies in the trimming of the state spaces. If the cardinality of the involved state spaces can be reduced to polynomial size and the loss in precision can be controlled, then we can expect that the trimmed dynamic program runs in polynomial time and outputs an ϵ-approximate Pareto set.

Algorithm 5: The dynamic programming formulation of the trimmed dynamic program TDP

```
initialize T_0 := S_0
for i = 1 to n do
    let U_i := {}
    foreach t ∈ T_{i-1} and F ∈ F do
        if H_F(x_i, t) ≤ 0 then
            add F(x_i, t) to U_i
        end
    end
    compute a trimmed copy T_i of U_i
end
output set {t ∈ T_n}
```

Aside from $\mathcal{S}_0$ there are two types of state spaces involved: $\mathcal{U}_i$ and $\mathcal{T}_i$. The state space that is produced at the end of the i-th phase in the original dynamic program DP was denoted by $\mathcal{S}_i$ (see Algorithm 3). In the trimmed dynamic program TDP, the iterative computation in the i-th phase expands the state space from the previous iteration to a new state space. This untrimmed state space is denoted by $\mathcal{U}_i$. However, before proceeding with the next iteration, the state space $\mathcal{U}_i$ is thinned out and trimmed. This yields the trimmed state space $\mathcal{T}_i$ on which the next iteration is based on. The trimming is based on the so-called *trimming parameter* $\Delta > 1$, where $\Delta := 1 + \frac{\epsilon}{2gn}$. $\epsilon > 0$ is the desired precision of approximation, n is the length of the input sequence and g is the integer constant that exists by Condition 3(i).

Moreover, we define an integer L by

$$L := \left\lceil \frac{\pi_1(n, \mathrm{ld}(\bar{x})}{\ln(\Delta)} \right\rceil \leq \left\lceil \left(1 + \frac{2gn}{\epsilon}\right) \pi_1(n, \mathrm{ld}(\bar{x})) \right\rceil . \tag{6.2}$$

$\pi_1(\cdot, \cdot)$ is the polynomial function introduced in Condition 4(iv). The upper bound on L follows by applying Proposition 2.2 with $x := \Delta$. The next step is to define $L + 1$ intervals as follows: $\mathcal{I}_0 = [0]$, $\mathcal{I}_i = [\Delta^{i-1}, \Delta^i)$ for $i = 1, \ldots, L-1$, and $\mathcal{I}_L = [\Delta^{L-1}, \Delta^L]$. Observe that every integer in the range $[0, \Delta^L]$ is contained in precisely one of these intervals.

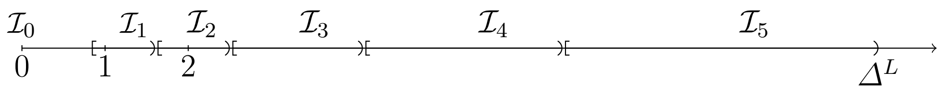


Figure 6.2: A schematic representation for interval partitioning between $[0, \Delta^L]$.

Next, we define a partition of the integer vectors in $[0, \Delta^L]^\beta$ into orthogonal, axes-parallel boxes. For every coordinate l ($1 \leq l \leq \beta$) with $d_l \geq 1$, the integer range $[0, \Delta^L]$ is partitioned into the intervals $\mathcal{I}_i$ ($i = 0, \ldots, L$). For every coordinate l ($1 \le l \le \beta$) with $d_l = 0$, the integer range $[0, \Delta^L]$ is partitioned into $\Delta^L + 1$ intervals that each contain a single integer.

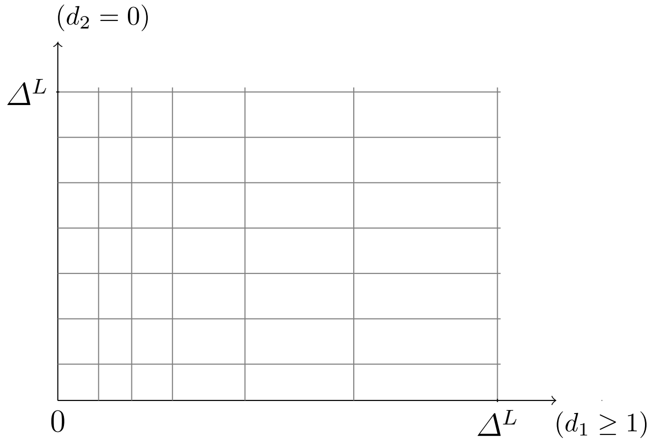


Figure 6.3: A schematic representation for a partition into axes-parallel boxes.

Proposition 6.12
Every state in every state space $\mathcal{S}_i$ in DP is contained in one of the Δ-boxes. If two states $s, s' \in \mathbb{N}^\beta$ are contained in the same Δ-box, then they are $[D, \Delta]$-close to each other.

Proof: The values of all components of all vectors in all state spaces $\mathcal{S}_i$ are integers less or equal to $e^{\pi_1(n, \mathrm{ld}(\bar{x}))}$ due to Condition 4(iv). By the definition of L in (6.2), it holds that $e^{\pi_1(n, \mathrm{ld}(\bar{x}))} \le \Delta^L$. Hence, every state in every state space is contained in one of the Δ-boxes.
If two states $s = (s_1, \ldots, s_\beta)$ and $s' = (s'_1, \ldots, s'_\beta)$ are contained in the same Δ-box, then $\Delta^{-1} \cdot s_l \le s'_l \le \Delta \cdot s_l$ holds for all coordinates l with $d_l \ge 1$ and $s_l = s'_l$ holds for all coordinates l with $d_l = 0$. This implies (6.1). □

Definition 6.13 Let $\mathcal{U}, \mathcal{T} \subseteq \mathbb{N}^\beta$ be sets of non-negative integer vectors whose components are bounded by Δ^L. $\mathcal{T}$ is a trimmed copy of $\mathcal{U}$ if the following two conditions hold:

(i) $\mathcal{T}$ is a subset of $\mathcal{U}$.

(ii) For every Δ-box $\mathcal{B}$ with $\mathcal{U} \cap \mathcal{B} \neq \emptyset$, the set $\mathcal{T}$ contains exactly one state from $\mathcal{U} \cap \mathcal{B}$. This state is a maximum of $\mathcal{U} \cap \mathcal{B}$ with respect to $\preceq_{qua}$. Such a maximum state exists by Proposition 6.2(ii).

Proposition 6.14
In the trimmed dynamic program TDP, the state spaces $\mathcal{T}_i$ and $\mathcal{U}_i$ fulfill the following properties for $i = 1, \ldots, n$.

(i) $\mathcal{T}_i \subseteq \mathcal{U}_i \subseteq \mathcal{S}_i$, where $\mathcal{S}_i$ is the i-th state space in the original dynamic programm DP (see Algorithm 3).

(ii) For every state u in the untrimmed state space $\mathcal{U}_i$, the trimmed state space $\mathcal{T}_i$ contains a state t that is $[D, \Delta]$-close to u and that fulfills $u \preceq_{qua} t$.

Proof: (i) In Line 5 of Algorithm 5 a trimmed copy $\mathcal{T}_i$ of $\mathcal{U}_i$ is computed. By Definition 6.13(i) $\mathcal{T}_i$ is a subset of $\mathcal{U}_i$. Furthermore, since $\mathcal{T}_0 \coloneqq \mathcal{S}_0$ (see Line 5 in Algorithm 5) and since Line 3 - Line 3 in Algorithm 3 and Line 5 - Line 5 in Algorithm 5, respectively, correspond to each other, $\mathcal{U}_i \subseteq \mathcal{S}_i$ clearly holds for $i = 1, \dots, n$.
(ii) Every state u in $\mathcal{U}_i$ is contained in one of the Δ-boxes by Proposition 6.12. By Definition 6.13(ii) $\mathcal{T}_i$ contains exactly one state from each box that contains states of $\mathcal{U}_i$. This state will be a maximum and is $[D, \Delta]$-close, since it is located in the same box. □

Proposition 6.15
Let s, s' and s'' be vectors in $\mathbb{N}^\beta$. Let $\Delta_1, \Delta_2 \geq 1$. If s is $[D, \Delta_1]$-close to s', and if s' is $[D, \Delta_2]$-close to s'', then s is $[D, \Delta_1 \cdot \Delta_2]$-close to s''.

Proof: If s is $[D, \Delta_1]$-close to s', then it holds by definition that $\Delta_1^{-d_i} \cdot s_i \leq s'_i \leq \Delta_1^{d_i} \cdot s_i$ for $i = 1, \dots, \beta$. Furthermore, according to our assumption it holds that $\Delta_2^{-d_i} \cdot s'_i \leq s''_i \leq \Delta_2^{d_i} \cdot s'_i$ for $i = 1, \dots, \beta$. Since $\Delta_1, \Delta_2 \geq 0$, it holds that

$$(\Delta_1 \cdot \Delta_2)^{-d_i} \cdot s_i \leq \Delta_2^{-d_i} \cdot s'_i \leq s''_i \leq \Delta_2^{d_i} \cdot s'_i \leq (\Delta_1 \cdot \Delta_2)^{d_i} \cdot s_i$$

for $i = 1, \dots, \beta$. In other words, s is $[D, \Delta_1 \cdot \Delta_2]$-close to s''. □

Lemma 6.16
Let $s \in \mathcal{S}_i$ be maximal w.r.t. $\preceq_{dom}$ for some $i \in \{1, \dots, n\}$. Then there exists a state $\bar{s} \in \mathcal{S}_{i-1}$ which is maximal w.r.t. $\preceq_{dom}$ in $\mathcal{S}_{i-1}$ and a mapping $\bar{F} \in \mathcal{F}$ with corresponding mapping $\bar{H} \in \mathcal{H}$ such that

$$\bar{H}(x_i, \bar{s}) \leq 0 \quad \textit{and} \quad \bar{F}(x_i, \bar{s}) = s.$$

Proof: Let us assume that s enters the state space $\mathcal{S}_i$ when the function $\bar{F} \in \mathcal{F}$ is applied to x_i and to some state in $\mathcal{S}_{i-1}$. Let $\bar{H} \in \mathcal{H}$ be the mapping that corresponds to $\bar{F}$. Consider the non-empty subset $\hat{\mathcal{S}} \subseteq \mathcal{S}_{i-1}$ containing all states $\hat{s} \in \mathcal{S}_{i-1}$ that fulfill

$$\bar{H}(x_i, \hat{s}) \leq 0 \quad \text{and} \quad \bar{F}(x_i, \hat{s}) = s. \tag{6.3}$$

From Proposition 6.2(i) follows the existence of a state $\bar{s} \in \hat{\mathcal{S}}$ that is maximal w.r.t. $\preceq_{dom}$ in $\hat{\mathcal{S}}$. Suppose that there is a state $s' \in \mathcal{S}_{i-1} \setminus \hat{\mathcal{S}}$ such that $\bar{s} \preceq_{dom} s'$. Then Condition 2(ii) together with (6.3) yields

$$\bar{H}(x_i, s') \leq \bar{H}(x_i, \bar{s}) \leq 0. \tag{6.4}$$

implying that $\bar{F}(x_i, s')$ is a feasible state. Consequently, $\bar{F}(x_i, s') \in \mathcal{S}_i$. Condition 1(ii) implies $s = \bar{F}(x_i, \bar{s}) \preceq_{dom} \bar{F}(x_i, s')$ due to $\bar{s} \preceq_{dom} s'$. Since $s' \notin \hat{\mathcal{S}}$, $\bar{F}(x_i, s') \neq s$ holds. Hence, there exist a state $s^* \in \mathcal{S}_i$ with $s^* \neq s$ such that $s \preceq s^*$ contradicting the assumption that s is maximal w.r.t. $\preceq_{dom}$ in $\mathcal{S}_i$. Hence, $\bar{s}$ must be maximal w.r.t. $\preceq_{dom}$ in $\mathcal{S}_{i-1}$. □

Lemma 6.17
For every $i = 0, \ldots, n$ and for every state $s \in \mathcal{S}_i$ that is maximal with respect to $\preceq_{dom}$, there exists a state $t \in \mathcal{T}_i$ that is $[D, \Delta^i]$-close to s and that fulfills $s \preceq_{qua} t$.

Proof: The result is shown by induction on i. For $i = 0$, it holds that $\mathcal{T}_0 = \mathcal{S}_0$ (see Line 5 of Algorithm 5) and the statement clearly holds. As induction hypothesis we assume that for every state $\bar{s} \in \mathcal{S}_{i-1}$ that is maximal w.r.t. $\preceq_{dom}$ there exists a state $\bar{t} \in \mathcal{T}_{i-1}$ which is $[D, \Delta^{i-1}]$-close to $\bar{s}$ and fulfills $\bar{s} \preceq_{qua} \bar{t}$. Let $s \in \mathcal{S}_i$ be a state that is maximal w.r.t. $\preceq_{dom}$. By Lemma 6.16 there exist a state $\bar{s} \in \mathcal{S}_{i-1}$ that is maximal w.r.t. $\preceq_{dom}$ and a mapping $\bar{F} \in \mathcal{F}$ with corresponding mapping $\bar{H} \in \mathcal{H}$ such that

$$\bar{H}(x_i, \bar{s}) \leq 0 \quad \text{and} \quad \bar{F}(x_i, \bar{s}) = s. \tag{6.5}$$

By the induction hypothesis there exists a state $\bar{t} \in \mathcal{T}_{i-1}$ which is $[D, \Delta^{i-1}]$-close to $\bar{s}$ and which fulfills $\bar{s} \preceq_{qua} \bar{t}$. Then by Condition 2(i) and (6.5) it holds that $\bar{H}(x_i, \bar{t}) \leq \bar{H}(x_i, \bar{s}) \leq 0$. By the latter inequality it follows that the state space $\mathcal{U}_i$ contains the state $u = \bar{F}(x_i, \bar{t})$ (see Line 5 and Line 5 in Algorithm 5). Since $\bar{t}$ is $[D, \Delta^{i-1}]$-close to $\bar{s}$ and $\bar{s} \preceq_{qua} \bar{t}$ by the induction hypothesis, Condition 1(i) implies one of two cases: either it holds that (a) u is $[D, \Delta^{i-1}]$-close to s and $s \preceq_{qua} u$ or it holds that (b) $s \preceq_{dom} u$.
Case (a): Proposition 6.14(ii) yields that the trimmed state space $\mathcal{T}_i$ contains a state t that is $[D, \Delta]$-close to u and that fulfills $u \preceq_{qua} t$. Proposition 6.15 yields that t is $[D, \Delta^i]$-close to s, since t is $[D, \Delta]$-close to u and since u is $[D, \Delta^{i-1}]$-close to s. Moreover, since $s \preceq_{qua} u$ and $u \preceq_{qua} t$, it holds that $s \preceq_{qua} t$.
Case (b): Proposition 6.14(i) yields that u is an element in $\mathcal{S}_i$. Since $s \in \mathcal{S}_i$ is maximal w.r.t. $\preceq_{dom}$, we infer that $s = u$. Then Proposition 6.14(ii) implies that the state space $\mathcal{T}_i$ contains a state t that is $[D, \Delta]$-close to $u = s$ and that fulfills $s \preceq_{qua} t$. t is particularly $[D, \Delta^i]$-close to s. □

6.3.2 Running time of the trimmed dynamic program

The following lemma shows that the trimmed dynamic program is theoretically efficient.

Lemma 6.18
The running time of TDP is polynomial in n, in $\text{ld}(\bar{x})$ and in $1/\epsilon$.

Proof: At first, we estimate the cardinality of $\mathcal{T}_i$. By Definition 6.13, it holds that $|\mathcal{T}_i|$ equals the number of Δ-boxes that have non-empty intersection with $\mathcal{U}_i$. In coordinates l with $d_l \geq 1$, these non-empty Δ-boxes arise from at most $L + 1$ distinct intervals in the partition. In coordinates l with $d_l = 0$, these non-empty Δ-boxes arise from at most $\pi_2(n, \text{ld}(\bar{x}))$ distinct intervals in the partition due to Condition 4(iv). Hence,

$$|\mathcal{T}_i| \leq (L + 1 + \pi_2(n, \text{ld}(\bar{x})))^\beta \leq \left\lceil \left(1 + \frac{2gn}{\epsilon}\right) \pi_1(n, \text{ld}(\bar{x})) + 1 + \pi_2(n, \text{ld}(\bar{x})) \right\rceil^\beta.$$

The upper bound on L is due to inequality (6.2). Since the values β and g only depend on DP (due to Definition 6.5 and Condition 3), β and g are constants. Since $\pi_1(\cdot, \cdot)$ and $\pi_2(\cdot, \cdot)$ are polynomials, it holds that for every $i = 1, \ldots, n$, the cardinality of $\mathcal{T}_i$ is polynomially bounded in n, in $\text{ld}(\bar{x})$, and in $1/\epsilon$.

Next, we discuss how the trimming is performed. For every state in $\mathcal{U}_i$, its Δ-box is computed and we thereby get a list of Δ-boxes that have non-empty intersection with $\mathcal{U}_i$. For every Δ-box $\mathcal{B}$ in this list, the relation $\preceq_{qua}$ on $\mathcal{U}_i \cap \mathcal{B}$ is computed and we find a maximum element with respect to $\preceq_{qua}$. This can be done in polynomial time per Δ-box by Condition 4(i). Since the list contains at most $|\mathcal{U}_i|$ Δ-boxes, the overall time needed for the trimming is polynomial in n, in $\mathrm{ld}(\bar{x})$, and in $1/\epsilon$.
Observe now that the trimmed dynamic program TDP goes through several nested for-loops. In every for-loop, the index either runs through a range of n, a range of $|\mathcal{T}_i|$, or a range of $|\mathcal{F}|$ values. By the above arguments and by Condition 4, the index range of every for-loop and the lengths of all encountered numbers are polynomially bounded in n, in $\mathrm{ld}(\bar{x})$, and in $1/\epsilon$. □

6.3.3 Completing the proof of the main theorem

Referring to the two previous subsections we now complete the proof of Theorem 6.11.

Proof: By Lemma 6.18 we know that the trimmed dynamic program TDP has a running time which is polynomial in n, in $\mathrm{ld}(\bar{x})$ and in $1/\epsilon$. Moreover, we showed that the cardinality of the trimmed state space $|\mathcal{T}_n| \leq \left\lceil \left(1 + \frac{2gn}{\epsilon}\right) \pi_1(n, \mathrm{ld}(\bar{x})) + 1 + \pi_2(n, \mathrm{ld}(\bar{x}))\right\rceil^{\beta}$ is polynomially bounded. Consider a state $s^* \in \mathcal{S}_n$ that is Pareto optimal. Assume that there is a state $s \in \mathcal{S}_n$ that is maximal w.r.t. $\preceq_{dom}$ such that $s^* \preceq_{dom} s$. By Condition 3(ii) it follows that $f_i(s) \geq f_i(s^*)$ for all $i \in \{1, \ldots, k\}$ (in the case of maximization). On the other hand, since s^* is Pareto optimal, there exist no $s \in \mathcal{S}_n$ such that $f_i(s) \geq f_i(s^*)$ for all $i \in \{1, \ldots, k\}$ with strict inequality for at least one i. Then it must hold that $f_i(s) = f_i(s^*)$ for all $i \in \{1, \ldots, k\}$. Hence, we can assume without loss of generality that $s^* \in \mathcal{S}_n$ is maximal w.r.t. $\preceq_{dom}$. By Lemma 6.17, there exists a state $t^* \in \mathcal{T}_n$ that is $[D, \Delta^n]$-close to s^* and that fulfills $s^* \preceq_{qua} t^*$. Then by Condition 3(i) we conclude (for maximization problems) that

$$f_i(s^*) \leq \Delta^{ng} f_i(t^*) = (1 + \frac{\epsilon}{2gn})^{gn} f_i(t^*) \leq (1 + \epsilon) f_i(t^*)$$

for all $i \in \{1, \ldots, k\}$. The last upper bound is due to Proposition 6.3 with $x = \epsilon/2$.
Therefore t^* ϵ-approximates s^* and the set $\mathcal{T}_n$ constitutes an ϵ-approximate Pareto set for $\mathcal{S}_n$. Notice that the same argument holds for the case in which we want to minimize the objective functions. □

6.4 A bicriteria scheduling example

Consider the bicriteria scheduling problem $P_2||C_{\max}, \sum w_j C_j$: Given two fixed machines and n jobs $J_1, \ldots, J_n$ with positive integer processing times p_j and positive integer weights w_j ($j = 1, \ldots, n$). All jobs are available for processing at time 0. The first objective corresponds to $P_2||C_{\max}$, that is, to schedule the jobs without preemption such that the largest job completion times is minimized. The second objective corresponds to $P_2|| \sum w_j C_j$, that is, to schedule the jobs without preemption on the two identical machines such that the weighted sum of job completion time is minimized. We renumber the jobs such that

$p_1/w_1 \leq p_2/w_2 \leq \ldots \leq p_n/w_n$. A job interchange argument shows that there always exists an optimal schedule for $P_2||\sum w_j C_j$ in which both machines process the jobs in increasing order of index. Furthermore, an optimal schedule will not contain any idle time.
Let $\alpha = 2$ and $\beta = 3$. For $i = 1, \ldots, n$ define the input vector $x_i = (p_i, w_i)$. A state $s = (s_1, s_2, s_3) \in \mathcal{S}_i$ encodes a partial schedule without idle time for the first i jobs. s_1 and s_2 are the total processing times on the first and second machine, respectively, and s_3 is the total weighted job completion time objective value for the partial schedule. Set $\mathcal{F} = \{F_1, F_2\}$ and $\mathcal{H} = \{H_1, H_2\}$ with

$$F_1(p_i, w_i, s_1, s_2, s_3) = (s_1 + p_i, s_2, s_3 + w_i(s_1 + p_i)),$$

$$F_2(p_i, w_i, s_1, s_2, s_3) = (s_1, s_2 + p_i, s_3 + w_i(s_2 + p_i)),$$

$$H_1 \equiv 0, H_2 \equiv 0.$$

Set the objective functions $f_1(s_1, s_2, s_3) = \max\{s_1, s_2\}$ and $f_2(s_1, s_2, s_3) = s_3$, respectively. Initialize the state space $\mathcal{S}_0 = \{(0, 0, 0)\}$ and define the degree vector $D = (1, 1, 1)$. Let $\preceq_{dom}$ be such that $s \preceq_{dom} s' \Leftrightarrow s_1^{'} \leq s_1 \wedge s_2^{'} \leq s_2 \wedge s_3^{'} \leq s_3$. Moreover, let $\preceq_{qua}$ be the universal relation. We will now argue that the dynamic programming formulation fulfills Condition 1 - Condition 4.
Condition 1: Since $\preceq_{qua}$ is the universal relation, Condition 1(i) is fulfilled if s is $[D, \varDelta]$-close to s' implies $F(x, s)$ is $[D, \varDelta]$-close to $F(x, s')$. Let $s = (s_1, s_2, s_3)$ be $[D, \varDelta]$-close to $s' = (s_1', s_2', s_3')$ for $D = (1, 1, 1)$ and an arbitrary but fixed $\varDelta > 1$, that is,

$$\frac{1}{\varDelta} \cdot s_i \leq s_i' \leq \varDelta \cdot s_i$$

for $i = 1, \ldots, 3$. Let $p, w \in \mathbb{N}$. Then the following inequalities hold:

$$\begin{aligned}
\frac{1}{\varDelta} \cdot (s_1 + p) \leq & s_1' + p \leq \varDelta \cdot (s_1 + p), \\
\frac{1}{\varDelta} \cdot s_2 \leq & s_2' \leq \varDelta \cdot s_2, \\
\frac{1}{\varDelta} \cdot (s_3 + w(s_1 + p)) \leq & s_3' + w(s_1' + p) \leq \varDelta \cdot (s_3 + w(s_1 + p)).
\end{aligned}$$

Hence, for $D = (1, 1, 1)$ it holds that if s is $[D, \varDelta]$-close to s', then $F_1(x, s)$ is $[D, \varDelta]$-close to $F_1(x, s')$ for any $x \in \mathbb{N}^2$ and $\varDelta > 1$. The same holds for $F_2 \in \mathcal{F}$. Hence, Condition 1(i) is fulfilled. Furthermore, if $s \preceq_{dom} s'$, then the following inequalities hold where w, p are again non-negative integers:

$$\begin{aligned}
s_1' \leq & s_1 \\
s_1' + p \leq & s_1 + p, \\
s_2' \leq & s_2, \\
s_2' + p \leq & s_2 + p \\
s_3' + w(s_1' + p) \leq & s_3 + w(s_1 + p) \\
s_3' + w(s_2' + p) \leq & s_3 + w(s_2 + p)
\end{aligned}$$

Hence, Condition 1(ii) is also fulfilled.
Condition 2 is fulfilled, since the functions $H_1, H_2 \in \mathcal{H}$ are constant zero.
Condition 3: Let s be $[D, \varDelta]$-close to s'. Then for $g = 1$ it holds that

$$\frac{1}{\varDelta} \cdot f_1(s') = \frac{1}{\varDelta} \cdot \max\{s'_1, s'_2\} = \max\{\frac{s'_1}{\varDelta}, \frac{s'_2}{\varDelta}\} \leq \max\{s_1, s_2\} = f_1(s) \text{ and}$$
$$\frac{1}{\varDelta} \cdot f_2(s') = \frac{s'_3}{\varDelta} \leq s_3 = f_2(s).$$

Moreover, if $s \preceq_{dom} s'$, then it holds that

$$f_1(s') = \max\{s'_1, s'_2\} \leq \max\{s_1, s_2\} = f_1(s) \text{ and}$$
$$f_2(s') = s'_3 \leq s_3 = f_2(s).$$

Therefore Condition 3 is fulfilled.
It is not difficult to see that the technical Condition 4 holds as well. Requirement (iv) is fulfilled, since all components in all states are upper bounded by $n(\sum_{i=1}^{n} w_i)(\sum_{i=1}^{n} p_i)$.

By Theorem 6.11, we can conclude that there exists an FPTAS for the bicriteria scheduling problem $P_2||C_{\max}, \sum w_j C_j$.

7 Summary and outlook

For several decades, multi-criteria optimization has been and remains an important branch in the optimization community. Its relevance derives from real-world applications that often involve several objectives and hence can be modeled and tackled by multi-criteria optimization. The recent book by Ehrgott ([1]) and the book by Sawaragi, Nakayama and Tanino ([8]), which treats the subject from a more functional analytic point of view, can be regarded as thorough introductions into the subject.

In Section 3.1, we defined an ϵ-approximate Pareto set $\mathcal{X}_{\epsilon P}$ by the condition that for every Pareto optimal solution x^* there exists $\bar{x}$ in $\mathcal{X}_{\epsilon P}$ that ϵ-approximates x^*. This definition did not require anything with respect to the size of $\mathcal{X}_{\epsilon P}$. We argued that we are satisfied with ϵ-approximate Pareto sets $\mathcal{X}_{\epsilon P}$ whose size is polynomially bounded. Moreover, in Section 5.3.1 we constructed a set S consisting of a polynomial number of Pareto optimal solutions. The (upper envelope of) the convex hull of this set S constituted then our ϵ-approximate Pareto set $\mathcal{X}_{\epsilon P}$. In this sense, $\mathcal{X}_{\epsilon P}$ was characterized by the polynomial number of solutions in S. Clearly, it is very desirable to be able to state requirements or conditions that imply tighter bounds on the actual size of the considered ϵ-approximate Pareto sets, in particular, if we consider discrete optimization problems.

Regarding the approximation results of Chapter 5, some experimental work on implementations might be of interest. The main focus needed to be put on the weighted sum scalarization for which several scalarization techniques and approaches exist (see Chapter 4 in [1]). Benson (see [19]) and Borwein (see [20]) define the notion of proper Pareto optimality with respect to cones. A closer examination on topics related to cones and orders could deliver further insights into some of the results in Chapter 5 and in Chapter 6.

Approximation results that are theoretically efficient do not generally yield approximation algorithms that are useful in practice. As mentioned in Section 3.1, an approximation algorithm with a running time bounded by a polynomial of high degree is not practically relevant for large instances. Woeginger's framework is an interesting and very general result guaranteeing the existence of an FPTAS. However, since the dynamic programming framework is very general and does not make use of some idiosyncratic structure of applicable problem instances, it does not necessarily yield a practically efficient approximation framework. (A naive Lisp-implementation for the KNAPSACK problem, done during the first steps of this thesis, corroborates this conclusion.) H.Safer and J.Orlin (see [16], [17] and [18]) also published results related to fast approximation schemes for multi-criteria combinatorial optimization problems. It might be of interest to peruse and relate their results to the results in Chapter 6.

Bibliography

[1] M. Ehrgott, *Multicriteria optimization.* Springer, 2000.

[2] C.Papadimitriou and M.Yannakakis, "On the approximability of trade-offs and optimal access of web sources," 2000.

[3] G. J. Woeginger, "When does a dynamic programming formulation guarantee the existence of an fptas?," in *Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 820–829, SIAM, 1999.

[4] M. R. Garey and D. S. Johnson, *Computers and intractability.* W.H.Freeman and company, 1979.

[5] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of computer computations* (R.Miller and J.Thatcher, eds.), pp. 85–103, 1972.

[6] B.Korte and J.Vygen, *Combinatorial optimization - theory and algorithms.* Springer, 3. ed., 2006.

[7] P. C. D. Bovet, *Introduction to the theory of complexity.* Prentice Hall, 1994.

[8] Y.Sawaragi, H.Nakayama, and T.Tanino, *Theory of multiobjective optimization.* Academic press, 1985.

[9] R. Rockafellar, *Convex analysis*, p. 97. Princeton university press, 1970.

[10] O.Mangasarian, *Nonlinear programming*, p. 65. McGraw Hill, 1969.

[11] M.Bazaraa, H.Sherali, and C.Shetty, *Nonlinear programming*, pp. 43–44. Wiley-Interscience, 2006.

[12] S.Sahni, "Algorithms for scheduling independent tasks," pp. 116–127, 1976.

[13] O.Ibarra and C.Kim, "Fast approximation algorithms for the knapsack and sum of subset problems," pp. 463–468, 1975.

[14] P.Godfrey, R.Shipley, and J.Gryz, "Algorithms and analyses for maximal vector computation," 2006.

[15] D.Papadias, Y.Tao, G.Fu, and B.Seeger, "Skyline queries and its variations," SIGMOD conference, 2003.

[16] H.Safer and J.Orlin, "Fast approximation schemes for multi-criteria combinatorial optimization," 1995.

[17] H.Safer and J.Orlin, “Fast approximation schemes for multi-criteria flow, knapsack, and scheduling problems,” 1995.

[18] M.Safer, J.Orlin, and M.Dror, “Fully polynomial approximation in multi-criteria combinatorial optimization,” 2004.

[19] H.Benson, “An improved definition of proper efficiency for vector maximization with respect to cones.,” pp. 232–241, 1979.

[20] J.Borwein, “Proper efficient points for maximization with respect to cones.,” pp. 57–63, 1977.

[21] P. Schuurman and G. Woeginger, “Approximation schemes - a tutorial,”

[22] Corman, Leiserson, Rivest, and Stein, *Introduction to algorithms.* MIT press, 2. ed., 2001.

[23] C.Papadimitriou and M.Yannakakis, “Multiobjective query optimization,” 2001.

[24] Aggarwal, Aneja, and Nair, “Minimal spanning tree subject to a side constraint,” pp. 287–296, 1982.

[25] M. Sniedovich, “Dynamic programming and principles of optimality,” pp. 586–606, 1978.

[26] T.Braun and R.Strauch, “Dynamic programming in multiplicative lattices,” 1964.

[27] R.Carraway, T.Morin, and H.Moskowitz, “Generalized dynamic programming for multicriteria optimization,” pp. 95–104, 1990.

[28] M.Henig, “The principle of optimality in dynamic programming with returns in partially ordered sets,” pp. 462–470, 1985.

[29] Y. Luo, *Multi-dimensional skyline query processing.* Dissertation, School of computer science and engineering, University of New South Wales, Sydney, Australia, 2004.

[30] F.Moulai, “Approximate results for bicriteria scheduling problem with deployment.”

[31] A. Warburton, “Approximation of pareto optima in multiple-objective, shortest-path problems,” pp. 70–79, 1987.

[32] M.Voorneveld, “Characterization of pareto dominance,” 2002.

[33] P.L.Yu, “Cone convexity, cone extreme points, and nondominated solutions in decision problems with multiobjectives,” 1974.

[34] C.Bazgan, H.Hugot, and D.Vanderpooten, “A practical efficient fptas for the 0-1 multi-objective knapsack problem.”

[35] J.Orlin and M.Mostagir, “When can one develop an fptas for a sequential decision problem?,” 2004.

[36] M. W.Kubiak, J.Cheng, "Fast fully polynomial approximation schemes for minimizing completion time variance."

[37] M.Soland, "Multicriteria optimization: A general characterization of efficient solutions."

[38] A.Engau and M.Wiecek, "Exact generation of epsilon-efficient solutions in multiple objective programming," tech. rep., Clemson university, Department of mathematical sciences, 2005.

[39] F.Guerriero and R.Musmanno, "Label correcting methods to solve multicriteria shortest path problems," pp. 589–613, 2001.

[40] J. D. Loera, R.Hemmecke, and M.Köppe, "Pareto optima of multicriteria integer linear programs."

[41] M.Rosenman and J.Gero, "Reducing the pareto optimal set in multicriteria optimization," pp. 189–206, 1985.

Nomenclature

$\bar{y} \prec y$ $\bar{y} \neq y, \bar{y}_i \leq y_i$ for $i = 1, \dots, k$

$\bar{y} \succ y$ $\bar{y} \neq y, \bar{y}_i \geq y_i$ for $i = 1, \dots, k$

$\mathrm{ld}(x)$ dual logarithm of x to the base 2

$\ln(x)$ natural logarithm of x to the base e

$\|x\| = \sqrt{\sum_{j=1}^{n} x_j^2}$, $x \in \mathbb{R}^n$

$\mathcal{S}(\lambda, \mathcal{Y}) = \{y^* \in \mathcal{Y} : \sum_{i=1}^{k} \lambda_i y_i^* = \max_{y \in \mathcal{Y}} \sum_{i=1}^{k} \lambda_i y_i\}$

$\mathcal{S}(\mathcal{Y}) \coloneqq \bigcup_{\lambda \in \mathbb{R}_+^k} \mathcal{S}(\lambda, \mathcal{Y})$

$\mathcal{S}_0(\mathcal{Y}) \coloneqq \bigcup_{\lambda \in \mathbb{R}_*^k} \mathcal{S}(\lambda, \mathcal{Y})$

$\mathcal{X}$ feasible domain in decision space

$\mathcal{X}_P$ set of Pareto optimal solutions

$\mathcal{X}_{pP}$ set of properly Pareto optimal solutions

$\mathcal{X}_{wP}$ set of weakly Pareto optimal solutions

$\mathcal{Y}$ feasible set in criterion space

$\mathcal{Y}_N$ set of non-dominated points

$\mathcal{Y}_{wN}$ set of weakly non-dominated points

$|I|$ encoding size of instance I

$\mathbb{R}_*^k = \mathbb{R}_\geq^k \setminus \{0\}$

$\mathbb{R}_+^k = \{x \in \mathbb{R}^k : x_i > 0$ for $i = 1, \dots, k\}$

$\mathbb{R}_\geq^k = \{x \in \mathbb{R}^k : x_i \geq 0$ for $i = 1, \dots, k\}$

$\subset$ $\subsetneq$

$A - B = \{a - b : a \in A, b \in B\}$ algebraic difference of two sets $A, B \subseteq \mathbb{R}^k$

S^k	k-ary Cartesian product of a set S
$\mathrm{int}(S)$	interior of a set S
$\mathrm{ri}(S)$	relative interior of a set S

Index