Outi Montonen

On Multiobjective Optimization from the Nonsmooth Perspective

Turku Centre *for* Computer Science

TUCS Dissertations
No 256, September 2020

On Multiobjective Optimization from the Nonsmooth Perspective

Outi Montonen
(née Wilppu)

To be presented, with the permission of the Faculty of Science and Engineering of the University of Turku, for public criticism in Auditorium Edu 2 on September 18, 2020, at 12 noon.

University of Turku
Department of Mathematics and Statistics
FI-20014 Turku
Finland

2020

Supervisors

Professor Marko Mäkelä
Department of Mathematics and Statistics
University of Turku
Finland

Docent Napsu Karmitsa
Department of Mathematics and Statistics
University of Turku
Finland

Docent Yury Nikulin
Department of Mathematics and Statistics
University of Turku
Finland

Reviewers

Research Director Emeritus Jean-Antoine Désidéri
INRIA Sophia Antipolis-Méditerranée Centre
France

Associate Professor Antonio Fuduli
Department of Mathematics and Computer Science
University of Calabria
Italy

Opponent

Professor Marco Antonio López Cerdá
Department of Mathematics
University of Alicante
Spain

ISBN 978-952-12-3966-3
ISSN 1239-1883

Abstract

Practical applications usually have multiobjective nature rather than having only one objective to optimize. A multiobjective problem cannot be solved with a single-objective solver as such. On the other hand, optimization of only one objective may lead to an arbitrary bad solutions with respect to other objectives. Therefore, special techniques for multiobjective optimization are vital. In addition to multiobjective nature, many real-life problems have nonsmooth (i.e. not continuously differentiable) structure. Unfortunately, many smooth (i.e. continuously differentiable) methods adopt gradient-based information which cannot be used for nonsmooth problems. Since both of these characteristics are relevant for applications, we focus here on nonsmooth multiobjective optimization. As a research topic, nonsmooth multiobjective optimization has gained only limited attraction while the fields of nonsmooth single-objective and smooth multiobjective optimization distinctively have attained greater interest. This dissertation covers parts of nonsmooth multiobjective optimization in terms of theory, methodology and application.

Bundle methods are widely considered as effective and reliable solvers for single-objective nonsmooth optimization. Therefore, we investigate the use of the bundle idea in the multiobjective framework with three different methods. The first one generalizes the single-objective proximal bundle method for the nonconvex multiobjective constrained problem. The second method adopts the ideas from the classical steepest descent method into the convex unconstrained multiobjective case. The third method is designed for multiobjective problems with constraints where both the objectives and constraints can be represented as a difference of convex (DC) functions. Beside the bundle idea, all three methods are descent, meaning that they produce better values for each objective at each iteration. Furthermore, all of them utilize the improvement function either directly or indirectly. A notable fact is that none of these methods use scalarization in the traditional sense. With the scalarization we refer to the techniques transforming a multiobjective problem into the single-objective one.

As the scalarization plays an important role in multiobjective optimization, we present one special family of achievement scalarizing functions as a representative of this category. In general, the achievement scalarizing functions suit well in the interactive framework. Thus, we propose the interactive method using our special family of achievement scalarizing functions. In addition, this method utilizes the above mentioned descent methods as tools to illustrate the range of optimal solutions. Finally, this interactive method is used to solve the practical case studies of the scheduling the final disposal of the spent nuclear fuel in Finland.

Tiivistelmä

Käytännön optimointisovellukset ovat usein luonteeltaan ennemmin moni- kuin yksitavoitteisia. Erityisesti monitavoitteisille tehtäville suunnitellut menetelmät ovat tarpeen, sillä monitavoitteista optimointitehtävää ei sellaisenaan pysty ratkaisemaan yksitavoitteisilla menetelmillä eikä vain yhden tavoitteen optimointi välttämättä tuota mielekästä ratkaisua muiden tavoitteiden suhteen. Monitavoitteisuuden lisäksi useat käytännön tehtävät ovat myös epäsileitä siten, etteivät niissä esiintyvät kohde- ja rajoitefunktiot välttämättä ole kaikkialla jatkuvasti differentioituvia. Kuitenkin monet optimointimenetelmät hyödyntävät gradienttiin pohjautuvaa tietoa, jota ei epäsileille funktioille ole saatavissa. Näiden molempien ominaisuuksien ollessa keskeisiä sovelluksia ajatellen, keskitytään tässä työssä epäsileään monitavoiteoptimointiin. Tutkimusalana epäsileä monitavoiteoptimointi on saanut vain vähän huomiota osakseen, vaikka sekä sileä monitavoiteoptimointi että yksitavoitteinen epäsileä optimointi erikseen ovat aktiivisia tutkimusaloja. Tässä työssä epäsileää monitavoiteoptimointia on käsitelty niin teorian, menetelmien kuin käytännön sovelluksien kannalta.

Kimppumenetelmiä pidetään yleisesti tehokkaina ja luotettavina menetelminä epäsileän optimointitehtävän ratkaisemiseen ja siksi tätä ajatusta hyödynnetään myös tässä väitöskirjassa kolmessa eri menetelmässä. Ensimmäinen näistä yleistää yksitavoitteisen proksimaalisen kimppumenetelmän epäkonveksille monitavoitteiselle rajoitteiselle tehtävälle sopivaksi. Toinen menetelmä hyödyntää klassisen nopeimman laskeutumisen menetelmän ideaa konveksille rajoitteettomalle tehtävälle. Kolmas menetelmä on suunniteltu erityisesti monitavoitteisille rajoitteisille tehtäville, joiden kohde- ja rajoitefunktiot voidaan ilmaista kahden konveksin funktion erotuksena. Kimppuajatuksen lisäksi kaikki kolme menetelmää ovat laskevia eli ne tuottavat joka kierroksella paremman arvon jokaiselle tavoitteelle. Yhteistä on myös se, että nämä kaikki hyödyntävät parannusfunktiota joko suoraan sellaisenaan tai epäsuorasti. Huomattavaa on, ettei yksikään näistä menetelmistä hyödynnä skalarisointia perinteisessä merkityksessään. Skalarisoinnilla viitataan me-

netelmiin, joissa usean tavoitteen tehtävä on muutettu sopivaksi yksitavoitteiseksi tehtäväksi.

Monitavoiteoptimointimenetelmien joukossa skalarisoinnilla on vankka jalansija. Esimerkkinä skalarisoinnista tässä työssä esitellään yksi saavuttavien skalarisointifunktioiden perhe. Yleisesti saavuttavat skalarisointifunktiot soveltuvat hyvin interaktiivisten menetelmien rakennuspalikoiksi. Täten kuvaillaan myös esiteltyä skalarisointifunktioiden perhettä hyödyntävä interaktiivinen menetelmä, joka lisäksi hyödyntää laskevia menetelmiä optimaalisten ratkaisujen havainnollistamisen apuna. Lopuksi tätä interaktiivista menetelmää käytetään aikatauluttamaan käytetyn ydinpolttoaineen loppusijoitusta Suomessa.

Acknowledgements

I consider myself privileged since I have had an opportunity to work with my awesome supervisors Professor Marko Mäkelä, Docent Napsu Karmitsa, and Docent Yury Nikulin. I am sincerely thankful all of you for encouragement, guidance, and convincing me when I am in doubt. I am extremely grateful to Marko for the constant support during these years. I truly tried my best by generating problems but Marko managed to solve them all. The feedback received from Napsu has been irreplaceable when I have explored my own style to write papers. I wish to express my deepest gratitude to Yury who has broadened my knowledge in the different areas of optimization and operation research by teaching the majority of the courses I took during my PhD studies.

Besides my supervisors, I have had other co-authors whose contribution toward this dissertation should be acknowledged. First, I want to thank Doctor Kaisa Joki from the bottom of my heart. Working with you is not rewarding only from the professional point of view and it has been enjoyable to have someone to share the peculiarities of the PhD studies. I have had also a great opportunity to work with Ville-Pekka Eronen, Doctor Jani Huttunen, and Doctor Timo Ranta amongst the others involved in Posiva Ydinoptimointi project. Their help have been valuable when I have deepened my understanding on the field of the final disposal.

I am very grateful for Research Director Emeritus Jean-Antoine Désidéri and Associate Professor Antonio Fuduli for spending their time to carefully read and review my dissertation. I am honoured that Professor Marco Antonio López Cerdá agreed to act as my opponent.

I appreciate the financial support received from the Doctoral Programme in Mathematics and Computer Sciences (MATTI) of University of Turku, Department of Mathematics and Statistics, the Vilho, Yrjö, and Kalle Väisälä Foundation of the Finnish Academy of Science and Letters, Emil Aaltonen Foundation, the Academy of Finland (Project No. 294002), and the project Posiva Ydinoptimointi (Project No. 2610102511 of University of Turku).

I would like to address my thanks to the current and former head of the department Professor Iiro Honkala and Professor Juhani Karhumäki as well as all the staff in the department for providing a pleasant working environment. I want to thank the current and former administrative staff for making my life easier. Especially, I would like to thank Doctor Arto Lepistö for solving my technical problems even before I knew I have them. Furthermore, I wish to acknowledge people making huge impact on my every-day life at the department: members of TOp Group, fellow PhD students, my previous room mates, and everyone around the coffee table. Obviously, I cannot be mentioning my high-quality (remote) lunch team: Anni Hakanen, Doctor Kaisa Joki, and Doctor Jarkko Peltomäki (and the occasional guest stars). Thank you for being who you are.

I am fortunate having loving and supportive family and friends around me. The support I gained from my husband Martti is beyond description, and there is no praise great enough describing my gratitude to Martti. Finally, I want to express my deepest adoration to Inka for putting things in the right perspective.

Turku, June 2020 Outi Montonen

List of Original Publications

This dissertation consist of the summary and the following original publications:

I M. M. Mäkelä, N. Karmitsa, and O. Wilppu. Proximal Bundle Method for Nonsmooth and Nonconvex Multiobjective Optimization. In T. Tuovinen, S. Repin, and P. Neittaanmäki, editors, *Mathematical Modeling and Optimization of Complex Structures*, volume 40 of *Computational Methods in Applied Sciences*, pages 191–204. Springer, 2016.

II M. M. Mäkelä and O. Montonen. New Multiobjective Proximal Bundle Method with Scaled Improvement Function. In A. M. Bagirov, M. Gaudioso, N. Karmitsa, M. M. Mäkelä, and S. Taheri, editors, *Numerical Nonsmooth Optimization: State of the Art Algorithms*, pages 461–479, Springer, 2020.

III O. Montonen, N. Karmitsa, and M. M. Mäkelä. Multiple subgradient descent bundle method for convex nonsmooth multiobjective optimization. *Optimization*, 67(1):139–158, 2018.

IV O. Montonen and K. Joki. Bundle-based descent method for nonsmooth multiobjective DC optimization with inequality constraints. *Journal of Global Optimization*, 72(3):403–429, 2018.

V O. Montonen and K. Joki. Multiobjective Double Bundle Method for DC Optimization. In A. M. Bagirov, M. Gaudioso, N. Karmitsa, M. M. Mäkelä, and S. Taheri, editors, *Numerical Nonsmooth Optimization: State of the Art Algorithms*, pages 481–497, Springer, 2020.

VI O. Wilppu, M. M. Mäkelä, and Y. Nikulin. New Two-slope Parameterized Achievement Scalarizing Functions for Nonlinear Multiobjective Optimization. In N. J. Daras and T. M. Rassias, editors, *Operations Research, Engineering, and Cyber Security*, volume 113 of *Optimization and Its Applications*, pages 403–422. Springer, 2017.

VII O. Montonen, T. Ranta, and M. M. Mäkelä. Planning the schedule for the disposal of the spent nuclear fuel with interactive multiobjective optimization. *Algorithms*, 12(12):252, 2019.

VIII O. Montonen, V.-P. Eronen, T. Ranta, J. A. S. Huttunen, and M. M. Mäkelä. Multiobjective mixed integer nonlinear model to plan the schedule for the final disposal of the spent nuclear fuel in Finland. *Mathematics*, 8:528, 2020.

Prior to my marriage, I published under my maiden surname Wilppu.

Contents

List of Symbols and Acronyms

$\mathbb{R}^n$	n-dimensional Euclidean space
$\mathbb{N}$	natural numbers
$\boldsymbol{x}$, $\boldsymbol{y}$	vectors
$\mathbf{0}$	zero vector
x_i	ith component of the vector $\boldsymbol{x}$
$\boldsymbol{x}^T$	transposed vector
$\boldsymbol{x}^T\boldsymbol{y}$	inner product of vectors $\boldsymbol{x}$ and $\boldsymbol{y}$
$\|\boldsymbol{x}\|$	Euclidean norm of the vector $\boldsymbol{x}$, $\|\boldsymbol{x}\| = (\boldsymbol{x}^T\boldsymbol{x})^{\frac{1}{2}}$
$\boldsymbol{x} \leq \boldsymbol{y}$, $\boldsymbol{x} < \boldsymbol{y}$	$x_i \leq y_i$, $x_i < y_i$ for all $i = 1, \ldots, n$
$\{\boldsymbol{x}_h\}$	sequence of vectors $\boldsymbol{x}_h$
(a, b)	open interval
$[a, b]$	closed interval
$[a, b)$, $(a, b]$	half-open intervals
$B(\boldsymbol{x}; \varepsilon)$	open ball with the center $\boldsymbol{x}$ and the radius ε
S	set
$\|S\|$	number of elements in the set S
$\bigcup_{i=1}^{k} S_i$	union of sets S_i, $i = 1, \ldots, k$
$\operatorname{conv} S$	convex hull of the set S
$\operatorname{ray} S$	ray of the set S
$\operatorname{cone} S$	conic hull of the set S
$\operatorname{cl} S$	closure of the set S
$K_S(\boldsymbol{x})$	contingent cone of the set S at the point $\boldsymbol{x}$
$S^{\leq}$	polar cone of the set S
$N_S(\boldsymbol{x})$	normal cone of the set S at the point $\boldsymbol{x}$
$T_S(\boldsymbol{x})$	tangent cone of the set S at the point $\boldsymbol{x}$
$f(\boldsymbol{x})$	value of the function f at the point $\boldsymbol{x}$
$\nabla f(\boldsymbol{x})$	gradient of the function f at the point $\boldsymbol{x}$
$t \downarrow 0$	$t \to 0_+$

$\frac{\partial f(\boldsymbol{x})}{\partial x_i}$	partial derivative of the function f at the point $\boldsymbol{x}$ with respect to the component x_i
$f'(\boldsymbol{x};\boldsymbol{d})$	directional derivative of the function f at the point $\boldsymbol{x}$ in the direction $\boldsymbol{d}$
$f^\circ(\boldsymbol{x};\boldsymbol{d})$	Clarke generalized directional derivative of the function f at the point $\boldsymbol{x}$ in the direction $\boldsymbol{d}$
$\partial f(\boldsymbol{x})$	subdifferential of the function f at the point $\boldsymbol{x}$
$\boldsymbol{\xi} \in \partial f(\boldsymbol{x})$	subgradient of the function f at the point $\boldsymbol{x}$
$\partial_\varepsilon^G f(\boldsymbol{x})$	Goldstein ε-subdifferential of the function f at the point $\boldsymbol{x}$
I, L, J	index sets
k	number of objective functions
m	number of constraint functions
$\boldsymbol{f}(\boldsymbol{x})$	objective function values of the multiobjective problem at the point $\boldsymbol{x}$
Z	feasible objective region
X	set of feasible solutions
$F(\boldsymbol{x})$	union of subdifferentials of objectives
$G(\boldsymbol{x})$	union of subdifferentials of active constraints
$\mathcal{B}$	bundle
$\hat{f}_h(\boldsymbol{x})$	cutting plane model for the function f at the point $\boldsymbol{x}$
$\operatorname{argmin} f(\boldsymbol{x})$	point giving the minimum for the function f
min	minimize
s.t.	subject to
MOP	multiobjective optimization problem
SOP	single-objective optimization problem
ASF	achievement scalarizing function
MSGDB	multiple subgradient descent bundle method
MPB	multiobjective proximal bundle method
DC	difference of convex functions
MDBDC	multiobjective double bundle method
MITSPA	multiobjective interactive method utilizing two-slope parameterized ASFs
MINLP	mixed integer nonlinear programming

Part I

Summary

Chapter 1

Introduction

In this chapter, we lead and motivate the reader to the world of nonsmooth multiobjective optimization. Section 1.1 gives an overview to the field of multiobjective optimization and the role of nonsmoothness in optimization is discussed. In Section 1.2, the contribution of this dissertation for the field is considered and the outline of this dissertation is given.

1.1 Background

People make decisions every day, and usually, there exist several aspects affecting the selection between different alternatives. In these daily situations, the person's own intuition generally yields an outcome being good with sufficient accuracy. However, decisions are made also in the broader perspective, like in industry, and then decisions may have a huge impact, for example, on costs, schedule, or safety. Therefore, mathematical optimization methods are widely needed to support the decision making process.

In optimization, we are interested in finding the best, or the most desirable, outcome among different alternatives. The best outcome depends on what is our goal or objective. However, in many practical cases, we have more than one aspect to take care of and these multiple goals depend on each other. Thus, by optimizing only one goal, the yielding outcome might not be the best for every aspect. Indeed, in order to obtain a relatively good solution for every aspect, we have to make compromises. When these goals are taken simultaneously into account, we are talking about the branch of optimization called multiobjective optimization. Despite multiobjective characteristic, many real-world applications are modelled as single-objective problems. The

advantage of this is the availability of efficient single-objective methods. Unfortunately, by optimizing only one of the several objectives, we may end up with a solution being an arbitrary bad solution with respect to the other objectives.

The area of multiobjective optimization is widely studied, especially in the last five decades [40, 70, 111, 142, 146]. The great interest towards multiobjective optimization is due to its applicability in numerous practical applications. To give some hints about where the multiobjective problems arise, we mention the fields of economics [123], engineering [118], management science [43], mechanics [119], chemical engineering [134], bioinformatics [60], machine learning [38], and environmental analysis [98]. Some more specific applications are the cancer treatment planning [28,69], the space exploration [149], and the humanitarian aid [57]. Actually, almost any practical application has more than one conflicting objectives to offer. That is why there is an increasing demand for efficient multiobjective optimization methods.

As it was mentioned, the solutions of the multiobjective problem are trade-offs and they are not unique. We say that these compromises are Pareto optimal solutions. A solution is Pareto optimal if we cannot improve any objective without deteriorating some other at the same time. A solution is weakly Pareto optimal if there does not exist any other solution having better values for all the objectives. In practice, only one solution is usually preferred, and to select this preferred solution among all the mathematically equally good Pareto optimal solutions, we need some external help in the form of the decision maker having more insight into the application field.

The concept of the Pareto optimality dates back to the works of Edgeworth [39] and Pareto [124] whom are considered as the pioneers of the multiobjective way of thinking with their works in economics at the late 19th and the early 20th century, respectively. While the ideas for multiobjective optimization existed, the first multiobjective methods saw the light of the day much later. Traditional solution approaches involved only single-objective optimization: either one objective is selected to be optimized and other objectives are used as constraints with suitable levels or the objectives are multiplied with some weights and summed up. One of the earliest works considering the optimization of multiple objectives in terms of mathematics was introduced in 1951 when Kuhn and Tucker published their famous optimality conditions [88]. These conditions where formulated not only for the single-objective but also for the multiobjective problem. During the same decade, the goal programming [24], where the aspiration levels for objectives are stated, was introduced. Another notable development phase with respect to

this dissertation is the introduction of achievement scalarizing functions (ASFs) by Wierzbicki [158–162] at the turn of the 1980's. For instance, ASFs are widely used as a part of interactive methods that reared their head in 1980's. During decades, alongside the traditional approaches more sophisticated and computationally efficient methods are presented, and nowadays, multiobjective optimization is a vivid and active research area.

In addition to multiobjective characteristic, many of the real-world problems have nonsmooth (i.e. not continuously differentiable) nature [7, 84, 103]. The nonsmoothness may arise from several different reasons [7,106]. First of all, an objective function itself may be nonsmooth like the piecewise linear tax models in economics [95], or the nonsmooth objective may be a result of some method for constrained optimization, like the exact penalty function method [7]. Furthermore, even if the objective is smooth some constraints may cause a nonsmooth dependence between variables, for example, in obstacle problems in optimal shape design [64] or the analytically smooth problem may numerically behave like a nonsmooth problem [95]. Nonsmooth optimization problems occur in many areas of practical applications, like computational chemistry [92], data mining [145] and classification [5], optimal shape design [64,110], and mechanics [119].

A nonsmooth function is not necessarily continuously differentiable in the classical sense. This usually causes difficulties, or depending on the perspective, arises fascinating research topics. The function may have some kink points where the gradient does not exist. In many cases, a kink point of this kind coincides with the minimizer of the function. That is why the gradient-based methods [3,10], like the steepest descent method, conjugate direction methods, and quasi-Newton methods, are not applicable to nonsmooth problems. The failure of classical gradient-based methods in nonsmooth problems is exemplified, for instance, in [95]. However, we may extend the concept of the gradient as in the works by Rockafellar [139] and Clarke [26]. This generalized gradient is called the subdifferential. Compared with the gradient being a vector, the subdifferential is a set of vectors whose elements are called subgradients. If a function is smooth, then its subdifferential contains only one element being the gradient.

Single-objective nonsmooth optimization methods can be roughly divided into two groups: subgradient methods and bundle methods. In subgradient methods, the idea is to replace the gradient by an arbitrary subgradient. The simplicity is the reason for the wide usage of subgradient methods. However, since the arbitrary subgradient does not preserve all the nice properties of the gradient, subgradient methods suffer seri-

ous drawbacks: unlike the negative gradient, the negative of an arbitrary subgradient does not ensure the descent direction, stepsizes have to be chosen a priori, and since the norm of the subgradient does not necessarily decrease while getting closer to the optimal solution, there does not exist implementable (sub)gradient-based stopping condition. To learn more about subgradient methods, we refer to the survey by Shor [144].

The next step in the evolution of nonsmooth methods was the cutting plane models by Cheney and Goldstein [25] and Kelley [80]. A difference with subgradient methods is that once an arbitrary subgradient is calculated, we store it. Therefore, by knowing subgradients from different iteration points, we are able to form a cutting plane model for the objective. This way we obtain more information about the behaviour of a function than in subgradient methods. As a disadvantage, we still cannot guarantee the descent search direction and the search direction finding problem may not have a finite optimum.

By combining the idea of the cutting plane model and the conjugate gradient method [163], the first bundle method called the ε-steepest descent method presented by Lemaréchal in [93,97], was obtained. In bundle methods, the idea is to approximate the subdifferential with a bundle including information from the neighbourhood of the iteration point instead of using only one subgradient. To overcome the problems caused by the selection of the tolerance ε in the ε-steepest descent method, the generalized cutting plane model was proposed by Lemaréchal [94] and later improved by Kiwiel [84]. Our story on the evolution of bundle methods accumulates in the introduction of the proximal bundle method by Kiwiel [85] and the bundle trust region method by Schramm and Zowe [143]. Actually, these two methods have the same core idea developed from the different basis. Nowadays, bundle methods are considered as an efficient way to solve nonsmooth problems and up to now, there exist lots of variations of bundle methods, like variable metric bundle methods [58, 102], level bundle methods [96], and bundle-Newton methods [101]. Even if bundle methods have their roots in convex problems, there exist many state of the art strategies to handle nonconvex problems (see e.g. [9, 48, 58, 62, 79, 84, 86, 110]). For a more discussion about different bundle methods, we refer to [106] and about different nonsmooth methods to [9].

In this dissertation, we bring together the fields of nonsmooth and multiobjective optimization. Compared with multiobjective optimization or single-objective nonsmooth optimization individually, nonsmooth multiobjective optimization has attracted significantly less attention. Even if the variety of multiobjective optimization methods is huge and there exist various nonsmooth methods for the single-objective prob-

lem, the multiobjective methods designed for nonsmooth perspective are much less frequent. However, in practical applications, there exists a great demand to solve nonsmooth problems with several objectives.

1.2 Structure of the Dissertation

This dissertation is composed of eight independent original publications. Six of them are devoted to methods for multiobjective optimization under the presence of nonsmoothness. The last two original publications involve the real-life applications where the methods from the other original publications are combined and the new method is employed to solve the nonsmooth multiobjective problem. In the purposes of this dissertation, we divide multiobjective optimization methods in two classes based on how they treat objectives: scalarizing methods aggregating multiple objectives into one objective, and descent methods improving multiple objectives simultaneously preserving the multiobjective nature.

The utilization of some scalarization technique is the traditional approach to solve multiobjective optimization problems. Via the scalarization, the multiple objectives of the original multiobjective problem are converted to one objective function and some additional constraints or variables may be introduced. With this procedure, we obtain a single-objective optimization problem, and then a suitable single-objective method can be applied. Examples of classical scalarization techniques are the STEM method [12], the GUESS method [19], and the ε-constraint method [59].

One widely used scalarization technique is the utilization of the achievement scalarizing functions (ASFs) [111,114,158–162] due to their good mathematical properties. In this approach, the ASF is optimized in order to find a solution being the closest to the decision maker's wishes. The point expressing the decision maker's wishes is called the reference point. The reference point is an intuitive way for the decision maker to indicate preferences, since only the desirable objective function values are needed. Moreover, the decision maker can be assisted to select the reference point, like in [104], where an approach to generate the set of equivalent reference points is proposed.

In the original publication VI, the family of the two-slope parameterized ASF is presented. This new family bases on the parameterized ASF presented in [122] and the two-slope ASF developed in [105]. By combining the advantages of both ASFs in the two-slope parameterized ASF, we can systematically produce several Pareto optimal solutions from the same preference information without implementing any test of achievability. However, the usage of the two-slope parameterized ASF

leads to a nonsmooth single-objective problem even if the original objectives are continuously differentiable.

As a contrast to the scalarization, in the original publications I–V, three different descent methods are discussed. Compared with the scalarization, these methods preserve multiobjective nature of the problem during the solution process. In literature, there exist several different descent methods for smooth multiobjective optimization like the generalizations of the steepest descent method [11, 36, 46, 55, 61, 154], the projected gradient methods [49, 56], the Newton methods [45], and the trust region strategies [20, 131]. A comprehensive study of three descent methods for smooth multiobjective optimization is given in [50]. Some descent methods exist also for convex nonsmooth optimization like the generalization of the steepest descent method for stochastic problems in [128], the projected subgradient methods in [17,30], the proximal point method [15,130] and the bundle-based method in [83]. For nonconvex problems, methods of this kind are, for example, the bundle-based methods [113, 157], the quasi-Newton method in [133], and the trust region strategy in [131].

The first descent method under the scope is the multiobjective proximal bundle method (MPB) discussed in the original publications I and II. MPB extends the single-objective proximal bundle method for nonconvex multiobjective framework by applying the technique basing on the use of the improvement function [83, 157]. In the original publication I, MPB is analyzed under some generalized convexity assumptions in addition to the general nonconvex case. When these generalized convexity assumptions hold, MPB produces a globally weakly Pareto optimal solution instead of a stationary solution guaranteed in the general nonconvex case. Additionally, the original publication II develops MPB further by presenting the scaled improvement function enabling the scaling of the objective functions to improve the numerical performance of the standard improvement function.

The discussion about descent methods continues with the multiple subgradient descent bundle method (MSGDB) proposed in the original publication III. This method is aimed for convex nonsmooth multiobjective problems that do not involve constraints. MSGDB is the extension of the well-known smooth single-objective steepest descent method to the nonsmooth and multiobjective framework. The method utilizes the proximal bundle approach which is the connective feature of all the descent methods described in this dissertation. Compared with the nonsmooth single-objective steepest descent method, in MSGDB, only one arbitrary subgradient needs to be evaluated, and unlike in subgradient methods, the descent direction can be ensured. Even if these drawbacks

are overcome, MSGDB is yet rather simply to implement.

Multiobjective DC (i.e. difference of convex functions) optimization has drawn only a little attention even if the class of DC functions is wide and the convex analysis can be applied. Some optimality conditions are studied, for instance, in [51, 132, 148] and the proximal point methods, exact and inexact, are presented in [73, 74]. The contribution of the original publications IV and V for the field of multiobjective DC optimization is threefold. First, a new method for constrained multiobjective DC optimization producing the theoretically better quality of solutions than the existing methods given in [73] is presented. Second, compared with the methods in [73], the new method is of the descent type. Third, the new method is numerically compared with a method designed for a general nonconvex function and observed that by taking the DC structure into account, we may benefit from it by obtaining better solutions. The new method is called the multiobjective double bundle method for DC functions (MDBDC). Like MPB, MDBDC utilizes the improvement function and instead of the proximal bundle method, we utilize the modified double bundle method for DC functions [77] due to its good ability to find global solutions even if it is only a local method.

Having a suitable optimization method is only the half of the truth. As we all know, the real-world phenomena are often too complicated to model accurately as a mathematical problem. Thus, the modelling is struggling between two goals: how to form a model being simple enough for the methods available and yet enough true to life. As discussed previously, the nonsmoothness and the several contradicting goals are common features in practical applications. Thus, we obtain more freedom to form mathematical models when we are able to utilize a nonsmooth multiobjective solver.

The original publications VII and VIII take a challenge to treat the practical application about scheduling the final disposal of the spent nuclear fuel as a multiobjective optimization problem. A similar topic is discussed in [135], where the disposal process is modelled as a single-objective linear mixed integer optimization problem minimizing the total costs. In the original publication VII, the process is modelled as a multiobjective nonsmooth mixed integer optimization problem for one fuel type and the problem is solved by the interactive method utilizing the two-slope parameterized ASF discussed in the original publication VI. The same method is used when the process is modelled in the original publication VIII involving all three fuel types used in Finland.

The rest of this dissertation is organized as follows. In Chapter 2, we give a fundamental theoretical background to nonsmooth multiobjective optimization and discuss some of the different methodologies. Chapters

3, 4, and 5 are devoted to summarize the most crucial results of this dissertation. In Chapters 3 and 4, ideas of the descent methods and the scalarization presented in the original publications are sketched, respectively. Chapter 5 consists of application for the method described in Chapter 4, and in Chapter 6, some final conclusions are given.

Chapter 2

Nonsmooth Multiobjective Optimization

This chapter introduces some notations and preliminary results used throughout the rest of the dissertation. First, we focus on nonsmooth analysis and define the generalized gradient amongst other things. Second, a nonsmooth constrained multiobjective optimization problem is presented. The chapter culminates in Section 2.3 where a Pareto optimal solution is defined and the multiobjective optimality condition is given. Finally, the methodology of multiobjective optimization is discussed in two sections. In Section 2.4, a brief overview on different types of methods for multiobjective optimization is given and in Section 2.5 the distinctive features of scalarization approaches and descent methods are pondered. More extensive introduction to nonsmooth analysis and optimization can be found, for example, in textbooks by Bagirov et al. [7], Clarke [26], Hiriart-Urruty and Lemaréchal [66], Mäkelä and Neittaanmäki [110], and Rockafellar [139] and for multiobjective optimization in textbooks by Ehrgott [40], Miettinen [111], Sawaragi [142], and Steuer [146].

2.1 Nonsmooth Analysis

This section covers the fundamental concepts and results regarding to nonsmooth analysis. We begin with some concepts related to sets. First, we say that a set $S \subseteq \mathbb{R}^n$ is compact if it is closed and bounded. A set $S \subset \mathbb{R}^n$ is said to be *convex* if

$$\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y} \in S \text{ for all } \boldsymbol{x}, \boldsymbol{y} \in S \text{ and } \lambda \in [0,1].$$

The *convex combination* of points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k \in \mathbb{R}^n$ is a point of the form $\sum_{i=1}^k \lambda_i \boldsymbol{x}_i$ such that $\lambda_i \geq 0$ for all $i = 1, \ldots, k$ and $\sum_{i=1}^k \lambda_i = 1$. A set containing all the convex combinations of the points in a set $S \subseteq \mathbb{R}^n$ is called the *convex hull* of a set S. That is,

$$\operatorname{conv} S = \left\{ \boldsymbol{x} \in \mathbb{R}^n \,\middle|\, \boldsymbol{x} = \sum_{i=1}^k \lambda_i \boldsymbol{x}_i,\ \sum_{i=1}^k \lambda_i = 1,\ \lambda_i \geq 0,\ \boldsymbol{x}_i \in S,\ k > 0 \right\}.$$

The convex hull of a set S is the smallest convex set containing the set S. The closure of a set S is denoted by $\operatorname{cl} S$.

A set $S \subseteq \mathbb{R}^n$ is a *cone* if $\lambda \boldsymbol{x} \in S$ for all $\lambda \geq 0$ and $\boldsymbol{x} \in S$. A cone is *convex* if S is a convex set. The smallest cone containing a set S, called the *ray* of the set S, is defined by

$$\operatorname{ray} S = \{\lambda \boldsymbol{x} \,|\, \lambda \geq 0, \boldsymbol{x} \in S\}.$$

Therefore, the smallest convex cone containing a set S is

$$\operatorname{cone} S = \operatorname{conv} \operatorname{ray} S.$$

Furthermore, we define a *contingent cone* at $\boldsymbol{x} \in S$ by

$$K_S(\boldsymbol{x}) = \{\boldsymbol{d} \in \mathbb{R}^n \,|\, \text{ there exist } t_i \downarrow 0 \text{ and } \boldsymbol{d}_i \to \boldsymbol{d} \text{ with } \boldsymbol{x} + t_i \boldsymbol{d}_i \in S\},$$

a *tangent cone* at $\boldsymbol{x} \in S$ by

$$\begin{aligned} T_S(\boldsymbol{x}) = \{\boldsymbol{d} \in \mathbb{R}^n \,|\, &\text{for all } t_i \downarrow 0 \text{ and } \boldsymbol{x}_i \to \boldsymbol{x} \text{ with } \boldsymbol{x}_i \in S, \\ &\text{there exists } \boldsymbol{d}_i \to \boldsymbol{d} \text{ with } x_i + t_i \boldsymbol{d}_i \in S\}, \end{aligned}$$

and a *polar cone* by

$$S^{\leq} = \{\boldsymbol{d} \in \mathbb{R}^n \,|\, \boldsymbol{s}^T \boldsymbol{d} \leq 0, \text{ for all } \boldsymbol{s} \in S\}.$$

Additionally, we define the *normal cone* of a nonempty set S at $\boldsymbol{x} \in S$ as the polar cone of the tangent cone

$$N_S(\boldsymbol{x}) = T_S(\boldsymbol{x})^{\leq}.$$

Throughout the dissertation, we assume that the functions under consideration are *locally Lipschitz continuous* if other is not stated. A function $f : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz continuous at the point $\boldsymbol{x} \in \mathbb{R}^n$ if there exist a *Lipschitz constant* $K > 0$ and a scalar $\varepsilon > 0$ such that

$$|f(\boldsymbol{y}) - f(\boldsymbol{z})| \leq K \|\boldsymbol{y} - \boldsymbol{z}\| \text{ for all } \boldsymbol{y}, \boldsymbol{z} \in B(\boldsymbol{x}; \varepsilon),$$

where $B(\boldsymbol{x};\varepsilon) = \{\boldsymbol{y} \in \mathbb{R}^n \,|\, \|\boldsymbol{x}-\boldsymbol{y}\| < \varepsilon\}$ is an open ball with a center $\boldsymbol{x}$ and a radius ε. Additionally, a function $f : \mathbb{R}^n \to \mathbb{R}$ is *upper semicontinuous* at the point $\boldsymbol{x}$ if

$$\limsup_{h\to\infty} f(\boldsymbol{x}_h) \leq f(\boldsymbol{x})$$

holds for every sequence $\{\boldsymbol{x}_h\}$ converging to the point $\boldsymbol{x}$ and *lower semicontinuous* if

$$f(\boldsymbol{x}) \leq \liminf_{h\to\infty} f(\boldsymbol{x}_h).$$

If a function is both upper and lower semicontinuous, it is also *continuous*. Additionally, every locally Lipschitz continuous function is also continuous.

One nice feature of locally Lipschitz continuous functions is that they are differentiable almost everywhere by Rademacher's theorem [110]. A function f is said to be *differentiable* at the point $\boldsymbol{x} \in \mathbb{R}^n$ if

$$f(\boldsymbol{x}+\boldsymbol{d}) = f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T\boldsymbol{d} + \|\boldsymbol{d}\|\alpha(\boldsymbol{x},\boldsymbol{d}),$$

where $\alpha(\boldsymbol{x},\boldsymbol{d}) \to 0$, when $\|\boldsymbol{d}\| \to 0$ and $\nabla f(\boldsymbol{x})$ is a *gradient* at the point $\boldsymbol{x}$. The gradient is a vector containing *partial derivatives* of a function f at the point $\boldsymbol{x}$ as its components, or in other words,

$$\nabla f(\boldsymbol{x}) = \left(\frac{\partial f(\boldsymbol{x})}{\partial x_1}, \ldots, \frac{\partial f(\boldsymbol{x})}{\partial x_n}\right).$$

Furthermore, a function f is called *smooth* or *continuously differentiable* if all the partial derivatives are continuous, and a function is *nonsmooth* if it is not continuously differentiable.

The *directional derivative* of a function $f : \mathbb{R}^n \to \mathbb{R}$ at the point $\boldsymbol{x} \in \mathbb{R}^n$ in the direction $\boldsymbol{d} \in \mathbb{R}^n$ is

$$f'(\boldsymbol{x};\boldsymbol{d}) = \lim_{t\downarrow 0} \frac{f(\boldsymbol{x}+t\boldsymbol{d}) - f(\boldsymbol{x})}{t}.$$

If a function f is differentiable at the point $\boldsymbol{x}$, then $f'(\boldsymbol{x};\boldsymbol{d}) = \nabla f(\boldsymbol{x})^T\boldsymbol{d}$ for any direction $\boldsymbol{d}$.

However, the directional derivative does not necessarily exist for a general locally Lipschitz continuous function. Thus, we consider a *Clarke generalized directional derivative* always existing for locally Lipschitz continuous functions. The Clarke generalized directional derivative of a function $f : \mathbb{R}^n \to \mathbb{R}$ at the point $\boldsymbol{x} \in \mathbb{R}^n$ in the direction $\boldsymbol{d} \in \mathbb{R}^n$ is defined by

$$f^\circ(\boldsymbol{x};\boldsymbol{d}) = \limsup_{\substack{\boldsymbol{y}\to\boldsymbol{x} \\ t\downarrow 0}} \frac{f(\boldsymbol{y}+t\boldsymbol{d}) - f(\boldsymbol{y})}{t},$$

where $\boldsymbol{y} \in \mathbb{R}^n$ and $t > 0$. A locally Lipschitz continuous function is *subdifferentially regular* at the point $\boldsymbol{x}$ if the directional derivative $f'(\boldsymbol{x}; \boldsymbol{d})$ exists for all $\boldsymbol{d}$ and $f'(\boldsymbol{x}; \boldsymbol{d}) = f^\circ(\boldsymbol{x}; \boldsymbol{d})$.

A locally Lipschitz continuous function $h : \mathbb{R} \to \mathbb{R}$ is *sign preserving* if $\operatorname{sign}(h(z)) = \operatorname{sign}(z)$, or in other words,

$$h(z) \begin{cases} < 0, & \text{when } z < 0 \\ = 0, & \text{when } z = 0 \\ > 0, & \text{when } z > 0. \end{cases}$$

Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be points in $\mathbb{R}^n$. We say that a function $f : \mathbb{R}^n \to \mathbb{R}$ is *increasing* if $f(\boldsymbol{x}) \leq f(\boldsymbol{y})$ when $\boldsymbol{x} \leq \boldsymbol{y}$ and *strictly increasing* if $f(\boldsymbol{x}) < f(\boldsymbol{y})$ when $\boldsymbol{x} < \boldsymbol{y}$. Moreover, a function f is *strongly increasing* if $f(\boldsymbol{x}) < f(\boldsymbol{y})$ when $\boldsymbol{x} \leq \boldsymbol{y}$, and $\boldsymbol{x}$ and $\boldsymbol{y}$ are distinct.

Convexity plays a remarkable role in the optimization theory. In particular, convexity is handy when we discuss optimality conditions. Indeed, under the convexity assumption, we can usually ensure that a method produces optimal solutions, while in the nonconvex case, we can only say that a solution is stationary. The stationarity means that some necessary optimality condition is satisfied but the optimality cannot be guaranteed. Thus, we say some words about classifying functions by their convexity. First, a function $f : \mathbb{R}^n \to \mathbb{R}$ is *convex* if

$$f(\lambda \boldsymbol{x} + (1 - \lambda)\boldsymbol{y}) \leq \lambda f(\boldsymbol{x}) + (1 - \lambda) f(\boldsymbol{y}), \quad \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n, \ \lambda \in [0, 1].$$

If this inequality is strict, $\boldsymbol{x} \neq \boldsymbol{y}$, and $\lambda \in (0, 1)$, then a function f is called *strictly convex* and a function f is *concave* if $-f$ is convex. If a function f is not convex, it is said to be *nonconvex*. Furthermore, every convex function is locally Lipschitz continuous and subdifferentially regular at any point $\boldsymbol{x} \in \mathbb{R}^n$ [26]. It is worth of noticing that some operations, like the addition or taking maximum, preserve convexity.

However, convexity is a demanding condition in many practical cases. That is why there have been several attempts to generalize convexity [126] such that we are able to obtain the same results as in the convex case but under the milder assumptions. Here we consider two of these generalizations of convexity. A locally Lipschitz continuous function $f : \mathbb{R}^n \to \mathbb{R}$ is f°-*pseudoconvex* if for all points $\boldsymbol{x}$ and $\boldsymbol{y}$ in $\mathbb{R}^n$

$$f(\boldsymbol{y}) < f(\boldsymbol{x}) \quad \text{implies} \quad f^\circ(\boldsymbol{x}; \boldsymbol{y} - \boldsymbol{x}) < 0$$

and f°-*quasiconvex* if for all points $\boldsymbol{x}$ and $\boldsymbol{y}$ in $\mathbb{R}^n$

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) \quad \text{implies} \quad f^\circ(\boldsymbol{x}; \boldsymbol{y} - \boldsymbol{x}) \leq 0.$$

Every convex function is f°-pseudoconvex and every f°-pseudoconvex function is f°-quasiconvex. Similar to the convex case, the maximum taken over f°-pseudoconvex functions is f°-pseudoconvex. Furthermore, if we consider a function $g : \mathbb{R}^n \to \mathbb{R}$ such that $g = h \circ f$, where $f : \mathbb{R}^n \to \mathbb{R}$ is f°-pseudoconvex and $h : \mathbb{R} \to \mathbb{R}$ is f°-pseudoconvex and strictly increasing, then g is also f°-pseudoconvex. On the other hand, if f is f°-quasiconvex and h is f°-quasiconvex and increasing, then g is f°-quasiconvex.

The benefits of convex analysis can also be utilized with nonconvex functions by using other distinctive structure. One subclass of nonconvex functions is formed by functions that can be represented as the difference of two convex functions, called *DC functions*. A function $f : \mathbb{R}^n \to \mathbb{R}$ is a DC function if it has a *DC decomposition* $f = p - q$, where functions $p : \mathbb{R}^n \to \mathbb{R}$ and $q : \mathbb{R}^n \to \mathbb{R}$ are convex *DC components*. Based on this DC structure, DC functions are locally Lipschitz continuous.

As mentioned previously, a locally Lipschitz continuous function is differentiable almost everywhere by Rademacher's theorem [110], and thus, it also has the gradient almost everywhere. In those points, where the gradient does not exist, we can utilize the generalized gradient. In this dissertation, we utilize the Clarke's generalized gradient [26]. The generalized gradient is called a *subdifferential* $\partial f(\boldsymbol{x})$ being a nonempty, convex and compact set and an element of this set is a vector $\boldsymbol{\xi} \in \mathbb{R}^n$, called a *subgradient*. First, we give two equivalent definitions for the subdifferential of a convex function $f : \mathbb{R}^n \to \mathbb{R}$ at the point $\boldsymbol{x} \in \mathbb{R}^n$

$$\begin{aligned}\partial_c f(\boldsymbol{x}) &= \{\boldsymbol{\xi} \in \mathbb{R}^n \mid f'(\boldsymbol{x}; \boldsymbol{d}) \geq \boldsymbol{\xi}^T \boldsymbol{d} \text{ for all } \boldsymbol{d} \in \mathbb{R}^n\} \\ &= \{\boldsymbol{\xi} \in \mathbb{R}^n \mid f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{\xi}^T(\boldsymbol{y} - \boldsymbol{x}) \text{ for all } \boldsymbol{y} \in \mathbb{R}^n\}.\end{aligned}$$

To simplify the notations, the subscript c is omitted while only convex functions are considered.

The *subdifferential* of a nonconvex locally Lipschitz continuous function $f : \mathbb{R}^n \to \mathbb{R}$ at the point $\boldsymbol{x}$ is defined by

$$\begin{aligned}\partial f(\boldsymbol{x}) &= \{\boldsymbol{\xi} \in \mathbb{R}^n \mid f^\circ(\boldsymbol{x}; \boldsymbol{d}) \geq \boldsymbol{\xi}^T \boldsymbol{d} \text{ for all } \boldsymbol{d} \in \mathbb{R}^n\} \\ &= \operatorname{conv}\{\lim_{i \to \infty} \nabla f(\boldsymbol{x}_i) \mid \boldsymbol{x}_i \to \boldsymbol{x} \text{ and } \nabla f(\boldsymbol{x}_i) \text{ exists}\}.\end{aligned}$$

Since $f^\circ(\boldsymbol{x}; \boldsymbol{d}) = f'(\boldsymbol{x}; \boldsymbol{d})$ if a function f is convex, the definition of the subdifferential of a nonconvex locally Lipschitz continuous function given above is the generalization of the subdifferential of a convex function. To conclude, if a function f is convex, then $\partial f(\boldsymbol{x})$ coincides with $\partial_c f(\boldsymbol{x})$ and if a function f is continuously differentiable, then the subdifferential is a singleton containing only the gradient, or in other words,

$\partial f(\boldsymbol{x}) = \{\nabla f(\boldsymbol{x})\}$. In addition, a function f is *weakly semismooth* if the directional derivative $f'(\boldsymbol{x}; \boldsymbol{d})$ exists for all $\boldsymbol{x}$ and $\boldsymbol{d}$, and

$$f'(\boldsymbol{x}; \boldsymbol{d}) = \lim_{t \downarrow 0} \boldsymbol{\xi}(\boldsymbol{x} + t\boldsymbol{d})^T \boldsymbol{d},$$

where $\boldsymbol{\xi}(\boldsymbol{x} + t\boldsymbol{d}) \in \partial f(\boldsymbol{x} + t\boldsymbol{d})$.

Finally, we define a *Goldstein ε-subdifferential* for a locally Lipschitz continuous function $f : \mathbb{R}^n \to \mathbb{R}$ at the point $\boldsymbol{x}$ by

$$\partial_\varepsilon^G f(\boldsymbol{x}) = \operatorname{conv} \{\partial f(\boldsymbol{y}) \,|\, \boldsymbol{y} \in B(\boldsymbol{x}; \varepsilon),\, \varepsilon \geq 0\}$$

generalizing the subdifferential of a nonconvex function. In some nonconvex cases, the Goldstein ε-subdifferential $\partial_\varepsilon^G f(\boldsymbol{x})$ has turned out to be more useful than the subdifferential $\partial f(\boldsymbol{x})$. As a more general notion, Goldstein ε-subdifferential can be used to approximate the subdifferential. Indeed, the smaller the value of the parameter ε is, the more accurate is the approximation. Furthermore, if $\varepsilon = 0$, then $\partial_\varepsilon^G f(\boldsymbol{x})$ coincides with $\partial f(\boldsymbol{x})$.

2.2 Multiobjective Optimization Problem

This dissertation focuses on multiobjective optimization. That is, we either minimize or maximize several objectives simultaneously. For simplicity, we formulate our optimization problems as minimization problems. However, this does not restrict our notations, since the maximization of a function f can easily be transformed to the minimization by multiplying the maximized function f by -1. The *multiobjective optimization problem*, where several goals are optimized simultaneously, is mathematically denoted as following:

$$\begin{aligned} \min \quad & \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})) \\ \text{s.t.} \quad & \boldsymbol{x} \in X, \end{aligned} \tag{MOP}$$

where s.t. is the abbreviation for subject to. The functions $f_i : \mathbb{R}^n \to \mathbb{R}$, $i \in I$, where $I = \{1, \ldots, k\}$, are the *objective functions* and k is the number of these objective functions. We denote by Z the *feasible objective region* being the image of the feasible solutions in the *objective space* $\mathbb{R}^k$, or in other words,

$$Z = \boldsymbol{f}(X) = \{\boldsymbol{z} \in \mathbb{R}^k \,|\, \boldsymbol{z} = \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x} \in X \subseteq \mathbb{R}^n\} \subseteq \mathbb{R}^k.$$

In the problem (MOP), the set $X \subseteq \mathbb{R}^n$ denotes the *set of feasible solutions* being a subset of the *decision space* $\mathbb{R}^n$. An element $\boldsymbol{x}$ of the

decision space $\mathbb{R}^n$ in the problem (MOP) is called the *decision vector*. If $X = \mathbb{R}^n$, then the problem is *unconstrained*. In this dissertation, if we specify the set of feasible solutions, it is defined by

$$X = \left\{ \boldsymbol{x} \in \mathbb{R}^n \,|\, g_l(\boldsymbol{x}) \leq 0, l \in L = \{1, \ldots, m\} \right\} \subseteq \mathbb{R}^n. \tag{2.1}$$

The functions $g_l : \mathbb{R}^n \to \mathbb{R}$, $l \in L$ are called *constraint functions*, and m is the number of these constraints. We assume that $X \neq \emptyset$ implying that there exists at least one feasible solution for the problem (MOP).

We make some assumptions regarding to the objective and constraint functions of the problem (MOP). First, in order to make the multiobjective problem sensible, we assume that our objective functions are at least partially conflicting. This means that there exists no solution yielding the best outcome for all the objectives simultaneously. Even though the objective functions are conflicting, none of them is more important than other. Another assumption is that all the objective and constraint functions are locally Lipschitz continuous. The problem (MOP) is the *nonsmooth multiobjective optimization problem* if at least one of the objective or constraint functions is nonsmooth. Moreover, the problem is convex if all the objective functions and the set X are convex.

While multiobjective optimization plays a crucial role in this dissertation, we need some knowledge of single-objective optimization as well. The *single-objective optimization problem* can be denoted (cf. (MOP)):

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & \boldsymbol{x} \in X, \end{aligned} \tag{SOP}$$

where the function $f : \mathbb{R}^n \to \mathbb{R}$ is an objective function and the set of feasible solutions $X \subseteq \mathbb{R}^n$. Again, we assume that all the objective and constraint functions are locally Lipschitz continuous. Like the multiobjective case, the problem (SOP) is the *nonsmooth single-objective optimization problem* if either objective or at least one constraint function is nonsmooth and the problem (SOP) is convex if the objective function and X are convex. Later in this dissertation, we are interested in solving the problem (SOP) with a quadratic objective and linear constraint functions. A problem of this kind is called the *quadratic problem*.

2.3 Optimality in Multiobjective Optimization

As suggested previously, in optimization we are eager to find the most desirable solutions. In single-objective optimization, this is a rather intuitive aim. However, in multiobjective optimization, we usually have a set of optimal solutions and different optimal solutions yield different

values for objectives. An optimal solution can be seen as a compromise between the objectives and the most desirable solution is a subjective concept. Thus, the comparison between different optimal solutions is difficult, since the relative importance of objectives is unknown. Now we define these concepts of optimality for both single-objective and multiobjective optimization.

A point $\boldsymbol{x}^* \in X$ is a *globally optimal solution* of the problem (SOP) if

$$f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}) \quad \text{for all} \quad \boldsymbol{x} \in X. \tag{2.2}$$

Furthermore, if the point $\boldsymbol{x}^*$ satisfies the condition (2.2) in some small neighbourhood such that $\boldsymbol{x} \in B(\boldsymbol{x}^*; \varepsilon)$, then the point $\boldsymbol{x}^*$ is a *locally optimal solution* of the problem (SOP). If the solution is globally optimal, there exists no other feasible solution yielding better value for the objective, unlike in the case of the locally optimal solution, where this is true only in the small neighbourhood of $\boldsymbol{x}^*$ and the better solution may exist outside this neighbourhood. Therefore, every global optimum is also a local optimum, but the inverse does not necessarily hold.

Theorem 2.3.1. [7] *If $\boldsymbol{x}^* \in X$ is a local minimum of a locally Lipschitz continuous function $f : \mathbb{R}^n \to \mathbb{R}$, then*

$$\mathbf{0} \in \partial f(\boldsymbol{x}^*) + N_X(\boldsymbol{x}^*). \tag{2.3}$$

Additionally, the point $\boldsymbol{x}^*$ is the global optimum if the function f is f°-pseudoconvex and X is convex.

Since every convex function is also f°-pseudoconvex function, Theorem 2.3.1 gives the sufficient condition for the global optimum of a convex function as well. A point $\boldsymbol{x} \in \mathbb{R}^n$ satisfying the condition (2.3) is later referred as a *Clarke stationary solution*. If a stationary solution is obtained for a convex function, then the solution is optimal as well.

The most desirable solution of the multiobjective optimization problem (MOP) could be such a solution that every objective attains its individual optimum. However, we assume that our objectives are at least partially conflicting, and thus, such a solution cannot exist. Even if this point cannot be the solution for the problem (MOP), we can utilize it to approximate the lower bound of the solution for the problem (MOP), and we call this point as an *ideal vector* $\boldsymbol{f}^I \in \mathbb{R}^k$. The components of the ideal vector are obtained by minimizing every objective of the problem (MOP) separately and the i-th component of the ideal vector $\boldsymbol{f}^I$ is obtained by minimizing the i-th objective globally. Another vector to approximate the lower bound of the solution is the *utopian vector* $\boldsymbol{f}^U \in \mathbb{R}^k$. This vector contains strictly better solutions for individual

objectives than the ideal vector. The components of the utopian vector are defined by $f_i^U = f_i^I - \varepsilon_i$ for all $i \in I$, where $\varepsilon_i > 0$ is a sufficient small constant.

While the ideal and the utopian vector give the lower bound for the set of solutions, the *nadir vector* $\boldsymbol{f}^N \in \mathbb{R}^k$, expressing the worst values of objectives among the optimal solutions, gives the upper bound for a solution. Components of the nadir vector are obtained by maximizing objectives over the set of optimal solutions. However, in practice, optimization over the set of optimal solutions is a hard task, and thus, it is complicated to define the nadir vector accurately, even in linear multiobjective optimization with three or more objectives [71, 138]. One rather simple method for approximating the nadir vector is the *payoff table* (see e.g. [40, 111]), which is obtained by optimizing all objectives separately. Then, each objective value is calculated at those optima obtained and wrote down to the payoff table. The approximation of the nadir vector is determined by selecting the worst value for each objective from the payoff table. Since the quality of the approximation obtained from the payoff table is not necessarily very good, some more sophisticated methods are described, for instance, in [14, 34, 42, 87]. The knowledge of the lower and upper bounds for the solution of the multiobjective problem is useful in many methods. For example, they can be used to scale objectives on different magnitudes to ease the computation. Second, many methods utilize the ideal and the nadir vectors as such, or they can help to understand the range of the optimal solution. Third, these vectors can be utilized in the illustration of the solution obtained.

We say that an optimal solution of the problem (MOP) is a Pareto optimal solution being a point such that none of the objectives can be improved without impairing some other objective simultaneously. Thus, to change the Pareto optimal solution, the trade-offs are needed. Mathematically speaking, a point $\boldsymbol{x}^* \in X$ is the *globally Pareto optimal solution* for the problem (MOP) if there exists no solution $\boldsymbol{x} \in X$ such that

$$f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{x}^*) \text{ for all } i \in I \text{ and } f_j(\boldsymbol{x}) < f_j(\boldsymbol{x}^*) \text{ for at least one } j \in I.$$

A solution $\boldsymbol{x}^* \in X$ is a *locally Pareto optimal solution* if there exists a radius $\varepsilon > 0$ such that $\boldsymbol{x}^*$ is a globally Pareto optimal solution on $X \cap B(\boldsymbol{x}^*; \varepsilon)$. In order to guarantee the existence of a Pareto optimal solution, we have to assume that the objectives are lower semicontinuous, which is true for any locally Lipschitz continuous function, and that the set of feasible solutions X is a nonempty compact set [142].

Many of the existing multiobjective optimization methods find a solution being *weakly Pareto optimal*. This means that there does not

exist any other solution such that all the objectives attain better values. More precisely, a solution $\boldsymbol{x}^* \in X$ is the *globally weakly Pareto optimal solution* of the problem (MOP) if there does not exist another solution $\boldsymbol{x} \in X$ such that

$$f_i(\boldsymbol{x}) < f_i(\boldsymbol{x}^*) \text{ for all } i \in I.$$

Moreover, a solution $\boldsymbol{x}^* \in X$ is a *locally weakly Pareto optimal solution* if it is a globally weakly Pareto optimal solution on some neighbourhood $X \cap B(\boldsymbol{x}^*; \varepsilon)$, where $\varepsilon > 0$. Based on these definitions, every Pareto optimal solution is also weakly Pareto optimal, but a weakly Pareto optimal solution is not necessarily Pareto optimal. We emphasize that the multiobjective problem (MOP) has several mathematically equally good solutions forming the set of (weakly) Pareto optimal solutions or the *Pareto set*.

The concepts described above are illustrated in Figure 2.1, where the multiobjective problem is depicted in the objective space. The Pareto front is emphasized with red and the ideal and nadir vectors are illustrated with black points. Due to the nature of the multiobjective optimization, some consider the multiobjective problem solved when the whole Pareto set is found. However, in many practical cases obtaining only one final solution is preferred. In this dissertation we consider the problem solved when some satisfying (weakly) Pareto optimal solution is found.

Next we state the necessary optimality condition for the constrained multiobjective problem of the form (MOP). In order to simplify the notations, we denote by

$$F(\boldsymbol{x}) = \bigcup_{i \in I} \partial f_i(\boldsymbol{x}) \tag{2.4}$$

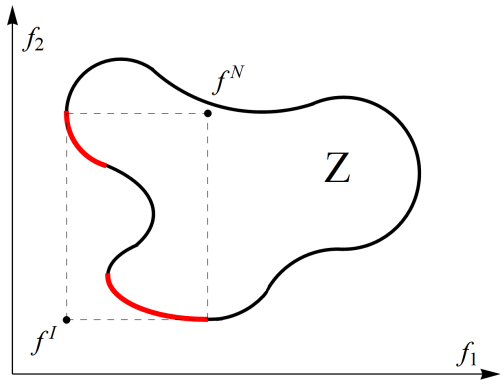


Figure 2.1: Pareto optimal solution in the biobjective case

and

$$G(\boldsymbol{x}) = \bigcup_{l \in L(\boldsymbol{x})} \partial g_l(\boldsymbol{x}), \text{ where } L(\boldsymbol{x}) = \{l \in L \,|\, g_l(\boldsymbol{x}) = 0\}.$$

Once constraints are involved, we usually have to make some regularity assumptions for the constraints. In this dissertation, we utilize the following constraint qualification (see e.g. [99, 108]):

$$G^{\leq}(\boldsymbol{x}) \subseteq K_X(\boldsymbol{x}). \tag{2.5}$$

Theorem 2.3.2. [108] *If $\boldsymbol{x}^* \in X$ is a locally weakly Pareto optimal solution for the problem* (MOP)*, and the constraint qualification* (2.5) *holds, then*

$$\mathbf{0} \in \text{conv } F(\boldsymbol{x}^*) + \text{cl cone } G(\boldsymbol{x}^*). \tag{2.6}$$

Additionally, if in Theorem 2.3.2 the objectives f_i for all $i \in I$ are f°-pseudoconvex and the constraints g_l for all $l \in L$ are f°-quasiconvex, then $\boldsymbol{x}^*$ is the globally weakly Pareto optimal solution. Since every Pareto optimal solution is a weakly Pareto optimal solution, this condition is the necessary condition for a Pareto optimal solution as well. If a point $\boldsymbol{x} \in \mathbb{R}^n$ satisfies the condition (2.6), then it is called *weakly Pareto stationary*.

2.4 Methods for Multiobjective Optimization

A good share of this dissertation considers methods for nonsmooth multiobjective optimization. The aim of this section is to give the wider perspective of the field of multiobjective optimization methods, while in Chapters 3 and 4 we discuss with details a few specific methods. Obviously, there does not exist any method being the best one for all problems, and every method has its own advantages and disadvantages.

A natural starting point for the multiobjective method development is to utilize efficient single-objective methods. However, solving a multiobjective problem with respect to only one objective may lead to an arbitrary bad solution regarding to the other objectives. Therefore, the multiobjective problem has to be somehow transformed into a single-objective problem. Approaches, where multiple objectives are converted into one and applied a single-objective optimization method, are called *scalarization*. In fact, scalarization is widely used technique for handling several objectives. One scalarization technique, in its simplicity, is just to sum up all the objectives.

We recall that the multiobjective problem has several mathematically equally good solutions. In order to select only one solution among the Pareto set, we need more insight into the problem. Therefore, the solution process of the multiobjective problem usually involves two parts: one for taking care of the mathematical side of the solution process and one having expertise to select the most preferred solution among the Pareto set. The first is referred to an *analyst* and the latter one is called a *decision maker*. The decision maker has the strong influence on the final solution. Different persons may consider different solutions to be the best one and even a single person may vary an opinion about the best solution under different circumstances.

We exemplify the different type of methods for multiobjective optimization by using the classification which is based on the role of the decision maker given in [70] and later in [111]. This classification has four classes: no-preference, a posteriori, a priori, and interactive methods. Note that this classification is not unique and a method may belong to several classes based on the way it is used.

The class of *no-preference methods* consists of methods which do not require the expertise of the decision maker. The aim in these methods is to produce one solution and the decision maker either accept it or not. These methods are usually fast, since only one solution needs to be found and the decision maker is not consulted. Additionally, they suit well for the situations when there are no special requirements for the solution. However, if the decision maker has any expectations for the solution, they probably will not be satisfied. These kind of methods are, for example, compromise programming [164], where the distance between the ideal point and the feasible objective region is minimized and the descent methods described in Chapter 3.

The other three classes involve the interaction with the decision maker. In *a posteriori methods*, a subset of the Pareto set is generated, and then, the decision maker chooses a desirable element of that subset as the solution. The benefit of this approach is that the wider picture of the Pareto set is obtained, but at the expense of the computation. Moreover, the visualization of the multiobjective problem with more than two objectives is difficult, and it is a complicated task for the decision maker to select one solution among a large set. Classical examples of this class are different scalarization techniques like the weighted sum method [52], where the linear combination of the objectives is minimized, or the ε-constraint method [59], where one objective is selected and the others are turned into constraints. Nowadays one widely studied class of heuristics is evolutionary algorithms [33, 166, 167] which are representing this class as well.

Unlike a posteriori methods, *a priori methods* first collect some preliminary information from the decision maker and then solve the problem. Thus, the solutions obtained correspond to the preliminary preconception of the decision maker. The disadvantage is that at the beginning of the solution process the decision maker may not be aware what (s)he actually wants. Some examples of a priori methods are to order objectives by their importance like in the lexicographic ordering [44] or to give aspiration levels for objectives and find a solution close to those as in the goal programming [23]. Additionally, the achievement scalarizing functions (ASFs) [158–162], discussed more in Chapter 4, can be considered as either a posteriori methods or a priori methods.

The last class of methods is *interactive methods*. In interactive methods, the decision maker is involved throughout the solution process. The idea is that first some preliminary solution is found and the decision maker's opinion about the solution is asked. Then the solution is modified based on the decision maker's preconceptions and again the opinion of the decision maker is asked. This process is continued until the satisfactory solution for the decision maker is found. Thus, the process gives the decision maker a chance to learn about the problem and to change preferences. However, the success of the interactive method is dependent on the decision maker. The process takes time and the decision maker may forget something from previous iterations or may get bored. Other notable issue is that if the solution in the early state of the solution process is close to the decision maker's premonition, then the decision maker may be willing to stop without taking advantage of the possibility to learn about the problem [19]. Some examples of interactive methods are one of the first interactive methods, called the step method STEM [12], where the decision maker selects an objective to be impaired to find the satisfactory solution. Other example is the NIMBUS method [113], where the decision maker classifies objectives based on which need to be improved and which are allowed to deteriorate. Furthermore, the interactive method in Section 4.3 is the member of this class.

2.5 Contrasting Scalarization and Descent Methods

As an active research area, there exists a countless number of methods to solve multiobjective problems. In the previous section, methods were divided by the role of the decision maker. Here we divide the methods roughly into two groups by how they treat the multiobjectiveness: the ones that handle the multiobjective problem by transforming them into a single-objective problem, and the ones that preserve multiobjec-

tive nature during the solution process. Therefore, the characteristic of methods from the different groups is completely different. The methodologies in Chapter 4 obviously belong to the first group and the descent methods in Chapter 3 belong to the latter one. As an introduction to the part of the dissertation focusing on methods, we discuss benefits and drawbacks related to these two groups of solution approaches in the general level. The discussion concentrates on scalarization approaches aggregating several objectives and descent methods considering all the objectives simultaneously as the representatives of these two groups.

Among all the multiobjective optimization methods, different scalarization techniques are widely used. Indeed, they have an intuitive core idea and a strong historical background. The unquestionable advantage of the scalarization is that the efficient single-objective methods can be applied. Especially with nonsmooth problems, with a suitable scalarization technique, the single-objective solver can be left in the charge to handle the nonsmoothness. However, in order to be the appropriate solution approach, the scalarized function needs to capture the multiobjective nature of the problem well.

The descent property yields some benefits over the scalarization, although, the descent property has not been seen as essential in multiobjective optimization as in the single-objective case. As a consequence, the number of descent multiobjective methods is significantly smaller than scalarization techniques. In fact, many descent methods are developed only quite recently (see e.g. [15,30,36,45,46,49,55,61,128,129,133, 154]). Many of these methods extend some ideas from single-objective optimization for the multiobjective framework instead of making the multiobjective problem suitable for a single-objective method.

The scalarization techniques need to take a stand on the importance of different objectives by specifying parameters. The tuning of suitable parameters is a hard task and sometimes it may cause difficulties. It is possible that even if the original multiobjective problem has a Pareto optimal solution, the scalarized problem becomes unbounded with a bad selection of parameters. This kind of behaviour is exemplified in [45], where the weighting method is used and almost all the parameters selected yield an unbounded single-objective problem. After the previous reasoning, a notable benefit of descent methods is that we do not have to specify any parameters related to the importance of objectives nor consider objectives' order of importance since they are all taken into account as equally important. That is why the solutions obtained with descent methods are more neutral.

Besides the information about the order of the importance of objectives, the scalarization techniques usually involve some preference

information from the decision maker. Contrary to the scalarization, preference information is not necessarily needed in descent methods. Examples of the preference information are the reference point and the classification of objectives based on which objectives will be improved and which impaired. The benefit of preference information is that the solutions obtained are more likely to be pleasant for the decision maker. However, if the scalarization is performed interactively, new preference information is needed at every iteration. Even if the decision maker's wishes can be better fulfilled at each iteration, providing the new preference information takes time and effort. Moreover, the decision maker has to consider trade-offs and it might be hard to give up the objective value already obtained.

The role of the decision maker is less significant in descent methods. Since preference information is not needed in descent methods, the decision maker takes the easy way out. As a matter of fact, a descent method gives a more preferred solution and a more content decision maker at each iteration. In some cases, it is preferred to obtain just one Pareto optimal solution without more wishes. Then, a descent method is a good choice.

It is worth noticing that the scalarized problem may be harder to solve than the objectives individually. This may happen, for instance, with ASFs discussed later in Chapter 4. Unfortunately, even with linear objectives the scalarized problem may be nonsmooth. These kind of theoretical changes in the degree of difficulty will not happen if the objectives are considered as they are. Especially in the case of integer variables, solving the scalarized problem may computationally become an exhaustive task, as was exemplified in [41].

In their turn, descent methods are sensitive to the selection of the starting point. Indeed, we obtain only solutions lying on the negative orthant from the starting point in the objective space. Therefore, it might be that every Pareto optimal solution cannot be obtained from every starting point. On the other hand, the starting point can be adjusted by the desired properties of the final solution. This property can be utilized in the cases where an approximated solution is already found and we wish to find a Pareto optimal solution dominating the approximated solution. The example can be found in [16], where a procedure of this kind is used with an evolutionary algorithm whose generally known drawback is that they cannot guarantee that the solution obtained is optimal. Some scalarization techniques suffer also from the inability to find every Pareto optimal solution. The classical example of this is the weighted sum method in the case of the nonconvex Pareto set (see e.g. [40]). Nevertheless, many more sophisticated scalarization tech-

niques, such as ASFs, can be designed such that every Pareto optimal solution may be obtained.

After all the differences, we collect some connective features between the methods in Chapters 3 and 4. First of all, they are all designed to solve the problem (MOP) and they all yield a solution being a weak Pareto optimal solution, or a weak Pareto stationary solution in the nonconvex case. In addition, they can all be performed repeatedly to obtain the approximation of the Pareto set. It is not mandatory to consider the importance of different objectives directly in any of these methods

To conclude, every method has its own advantages and disadvantages and it depends on the case whether the property is a drawback or a benefit. All in all, none of the methods is superior to others in the general case. Therefore, it is important to select a suitable method for every situation separately. By combining several solution approaches to one, the benefits from several methods can be utilized and possible overcome some disadvantages of other method. A synergy gain of this kind is obtained, for example, in the interactive method in Section 4.3 where descent methods are used to aid the decision maker in the method otherwise applying ASFs.

Chapter 3

Bundle-based multiobjective optimization

In this chapter, we focus on descent methods for multiobjective optimization. We discuss with details about three descent methods from the original publications: the multiobjective proximal bundle method (MPB) in I and II, the multiple subgradient descent bundle method (MSGDB) in III, and the multiobjective double bundle method for DC functions (MDBDC) in IV and V. Many of the descent multiobjective methods are generalizations of some single-objective method. Among these methods, there exist some descent methods presented for the differentiable multiobjective problem, for example, the generalizations of the single-objective steepest descent method [36,46,55,61,154], the Newton type method [45, 129], and the projected gradient method [49]. For convex nonsmooth problems, descent methods of this kind are: the projected subgradient type method [30], the proximal type method [15], the generalization of the steepest descent method for stochastic multiobjective problems [128], and the bundle-based method [83]. For nonconvex multiobjective problems, there exist descent methods of the Newton type method in [133] and the bundle type methods in [113, 157].

A distinctive feature of descent methods is that the objective function values decrease at every iteration. Indeed, descent methods project the starting point to the Pareto front without any preference information. Due to their descent property, descent methods obtain solutions from the negative orthant from the starting point in the objective space. Thus, a Pareto optimal solution is not a sensible starting point. The lack of preference information is an advantage to descent methods. A descent method can be used as it is to solve only one solution or by running it from several starting points, and thus, giving an approximation of the

Pareto set. Another possibility is to use a descent method together with some heuristic method by using a heuristic solution as the starting point to guarantee (weak) Pareto optimality. Moreover, descent methods can be used as a part of an interactive method as in [111, 113, 120] and in Section 4.3.

A direction $\boldsymbol{d} \in \mathbb{R}^n$ is a *descent direction* for a function $f : \mathbb{R}^n \to \mathbb{R}$ at the point $\boldsymbol{x} \in \mathbb{R}^n$ if there exists $\varepsilon > 0$ such that

$$f(\boldsymbol{x} + t\boldsymbol{d}) < f(\boldsymbol{x}) \quad \text{for all} \quad t \in (0, \varepsilon], \tag{3.1}$$

and if a direction $\boldsymbol{d}$ satisfies the condition

$$\boldsymbol{\xi}^T \boldsymbol{d} < 0 \quad \text{for all} \quad \boldsymbol{\xi} \in \partial f(\boldsymbol{x}),$$

then it is the descent direction for a function f. The other option is to verify that $f^\circ(\boldsymbol{x}; \boldsymbol{d}) < 0$. In addition, we say that a direction $\boldsymbol{d}$ is the common descent direction for all functions $f_i : \mathbb{R}^n \to \mathbb{R}$, $i \in I$ if the condition (3.1) holds for all f_i, $i \in I$.

A general framework of descent methods is to select a starting point $\boldsymbol{x}_0 \in X$ and then solve some subproblem to obtain a common descent direction $\boldsymbol{d} \in \mathbb{R}^n$ for all the objectives. After that, a *stepsize* $t > 0$ is determined and a new iteration point $\boldsymbol{x}_{h+1} = \boldsymbol{x}_h + t\boldsymbol{d}$ is calculated. The descent direction is a local property, and at every nonstationary point, the descent direction exists [7]. This implies that at every iteration there either exists a descent direction or the current iteration point is a stationary solution. Thus, we generate the sequence of solutions $\{\boldsymbol{x}_h\}$ converting to a solution such that $f_i(\boldsymbol{x}_h) > f_i(\boldsymbol{x}_{h+1})$ for all $i \in I$.

As mentioned, this chapter discusses about three different descent methods: MPB, MSGDB, and MDBDC. These three methods have three things in common:

1. they are all descent;
2. they all utilize the idea of the single-objective proximal bundle method [85, 110, 143];
3. in order to guarantee the descent property, they utilize the improvement function [83, 113, 157].

Thus, the first two sections of this chapter are devoted to define some essential ingredients for the methods such as the use of the improvement function and the standard single-objective proximal bundle method. After that, MPB, MSGDB, and MDBDC are discussed in the following three sections, respectively.

3.1 Improvement Function

As a foretaste of the multiobjective methods, we introduce a technique for handling several objectives and constraints simultaneously, called the improvement function. This technique has its origin in single-objective smooth [127] and nonsmooth [78, 84, 96, 110, 117, 141] optimization as a constraint handling technique. With the aid of the improvement function, we can control several functions as one, and thus, the single-objective constrained problem can be formulated as an unconstrained problem. Besides several constraint functions, the improvement function can handle several objectives without the scalarization in the classical sense. Additionally, we obtain a nice connection between the Pareto optimal solution and the solution of the single-objective problem with the improvement function as its objective. Thus, by solving the single-objective problem, we obtain a solution for the multiobjective problem. As a multiobjective technique, the use of the improvement function is described, for instance, in [83,113,157] in addition to the original publications I–V. Next, we collect some properties of the improvement function.

Given the problem (MOP), we define the *standard improvement function* $H : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ by

$$H(\boldsymbol{x}, \boldsymbol{y}) = \max\{f_i(\boldsymbol{x}) - f_i(\boldsymbol{y}),\, g_l(\boldsymbol{x}) \,|\, i \in I, l \in L\}. \tag{3.2}$$

The improvement function works well in theory, but the practice has revealed that the improvement function is sensitive for scaling, as is noticed in the numerical experiments in the original publication IV. There the simple linear scaling is utilized in order to reduce the computational efforts. In the original publication II, this scaling was further generalized and properly formulated by introducing the generalization of the standard improvement function known as the *scaled improvement function* $H_s : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$

$$H_s(\boldsymbol{x}, \boldsymbol{y}) = \max\{\mu_i(f_i(\boldsymbol{x})) - \mu_i(f_i(\boldsymbol{y})),\, \delta_l(g_l(\boldsymbol{x})) \,|\, i \in I, l \in L\}, \tag{3.3}$$

where μ_i, $i \in I$ and δ_l, $l \in L$ are scaling functions such that they are assumed to be sign preserving, and for all $i \in I$, μ_i is supposed to be strictly increasing. Note that $H_s(\,\cdot\,, \boldsymbol{y})$ reduces back to $H(\,\cdot\,, \boldsymbol{y})$ if $\mu_i(z) = \delta_l(z) = z$ for all $i \in I$, $l \in L$, and $z \in \mathbb{R}$. Some examples of the suitable scaling functions are

linear scaling	$\mu(z) = \lambda z,$
polynomial scaling	$\mu(z) = \lambda \operatorname{sign}(z)((\lvert z\rvert + 1)^p - 1),$
logarithmic scaling	$\mu(z) = \lambda \operatorname{sign}(z) \ln(\lvert z\rvert + 1),$
exponential scaling	$\mu(z) = \lambda \operatorname{sign}(z)(e^{\lvert z\rvert} - 1),$

where $\lambda > 0$ and $p \in \mathbb{N}_+$.

Besides the constraint handling, the charm of this technique in multiobjective framework springs up, on the one hand, from the fact that if we find a descent direction for the scaled improvement function, then we have a common descent direction for all the objective functions of the problem (MOP). On the other hand, an optimal solution for the scaled improvement function is at least weakly Pareto stationary for the problem (MOP). These properties are collected in the following theorem. Similar properties for the standard improvement function can be found in [83, 157] and in the original publications I and IV.

Theorem 3.1.1. (Original publication II) *Let μ_i and δ_l be sign preserving and μ_i be strictly increasing for all $i \in I$ and $l \in L$. The scaled improvement function $H_s(\,\cdot\,, \boldsymbol{y})$ (3.3) has the following properties:*

(i) *If $H_s(\boldsymbol{x}, \boldsymbol{y}) < H_s(\boldsymbol{y}, \boldsymbol{y})$, $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in X$, then $f_i(\boldsymbol{x}) < f_i(\boldsymbol{y})$ for all $i \in I$ and $g_l(\boldsymbol{x}) < 0$ for all $l \in L$.*

(ii) *If the solution $\boldsymbol{x}^* \in X$ is a globally weakly Pareto optimal solution of the problem* (MOP)*, then*

$$\boldsymbol{x}^* = \operatorname*{argmin}_{\boldsymbol{x} \in \mathbb{R}^n} H_s(\boldsymbol{x}, \boldsymbol{x}^*). \tag{3.4}$$

(iii) *If f_i and μ_i for all $i \in I$ are f°-pseudoconvex, g_l and δ_l for all $l \in L$ are f°-quasiconvex, δ_l for all $l \in L$ is increasing, and the constrain qualification* (2.5) *is valid, then the condition* (3.4) *is sufficient for a point $\boldsymbol{x}^* \in X$ to be a globally weakly Pareto optimal solution of the problem* (MOP)*.*

(iv) *If $\mathbf{0} \in \partial H_s(\boldsymbol{x}^*, \boldsymbol{x}^*)$, then the solution $\boldsymbol{x}^* \in X$ of the problem* (MOP) *is weakly Pareto stationary.*

As mentioned, the standard improvement function is the special case of the scaled improvement function, and therefore, the properties given in Theorem 3.1.1 are valid for the standard improvement function as well. Note that the improvement function is the maximum function, implying that if all the objective and constraint functions are convex, f°-pseudoconvex or DC functions, so is the improvement function. This property comes handy when the methods of this chapter are discussed. The scaled improvement function is a core ingredient for MPB described in Section 3.3 and the standard improvement function for MDBDC discussed in Section 3.5. Even if MSGDB in Section 3.4 does not utilize the concept of the improvement function as such, the convergence proof of MSGDB relies on the properties of MPB utilizing the improvement function.

3.2 Single-objective Proximal Bundle Method

In this section, we describe the general idea of the proximal bundle method for single-objective optimization, and simultaneously, emphasize the characteristic features of the general proximal bundle method. To simplify the presentation, the idea of the proximal bundle method is presented for the convex unconstrained single-objective problem of the form (SOP) and the modifications needed in order to solve nonconvex problems are commented at the end. For more details about bundle methods we refer to [9, 84, 85, 97, 103, 106, 110, 143].

Bundle methods are considered as efficient and reliable solvers for single-objective nonsmooth problems and they fall into the category of descent methods. Many descent methods use gradient-based information, but since the problem is nonsmooth, the subdifferentials are employed in bundle methods. Due to the fact that the evaluation of the whole subdifferential is a hard task, the first distinctive feature of bundle methods is to approximate the subdifferential with the aid of a *bundle* gathering information from the neighbourhood of the current iteration point. Based on the information in the bundle, we can formulate a model for the objective function and with this model the search direction can be determined. If the search direction is descent enough, we take a serious step by generating the new iteration point. On the other hand, if the direction is not descent enough or is not descent at all, we take a null step being the second distinctive feature of bundle methods. In the null step, the model is improved by adding further information from an auxiliary point into the bundle.

We begin by considering the problem (SOP) with $X = \mathbb{R}^n$. In order to get the idea of the local behaviour of the objective function $f : \mathbb{R}^n \to \mathbb{R}$, in bundle methods, the subdifferential $\partial f(\boldsymbol{x}_h)$ is approximated with the subgradients belonging to the bundle

$$\mathcal{B}^h = \Big\{ \big(\boldsymbol{y}_j, f(\boldsymbol{y}_j), \boldsymbol{\xi}_j \in \partial f(\boldsymbol{y}_j)\big) \mid j \in J_h \Big\} \tag{3.5}$$

containing information from the previous iterations. In the h-th iteration, a point $\boldsymbol{x}_h \in \mathbb{R}^n$ is the current iteration point and points $\boldsymbol{y}_j \in \mathbb{R}^n$ are auxiliary points from the past iterations. Thus, we generate two sequence of points: a sequence of auxiliary points $\{\boldsymbol{y}_j\}$ and a sequence of iteration points $\{\boldsymbol{x}_h\} \subseteq \{\boldsymbol{y}_j\}$. Finally, a nonempty set of indices $J_h \subseteq \{1, \ldots, h\}$ contains indices selected to the bundle. Therefore, we assume that we are able to evaluate a function value $f(\boldsymbol{y}_j)$ and one arbitrary subgradient $\boldsymbol{\xi} \in \partial f(\boldsymbol{y}_j)$ at each point $\boldsymbol{y}_j$.

In the following, we consider the h-th iteration of the proximal bundle method. First, we linearize the function f with the classical *cutting plane*

model defined by

$$\begin{aligned}\hat{f}^h(\boldsymbol{x}) &= \max\{f(\boldsymbol{y}_j) + \boldsymbol{\xi}_j^T(\boldsymbol{x} - \boldsymbol{y}_j) \,|\, j \in J_h\} \\ &= \max\{f(\boldsymbol{x}_h) + \boldsymbol{\xi}_j^T(\boldsymbol{x} - \boldsymbol{x}_h) - \alpha_j^h \,|\, j \in J_h\}\end{aligned} \tag{3.6}$$

being the piecewise linear approximation of the function f. This approximation improves every time when more information is added to the bundle $\mathcal{B}^h$. The term α_j^h is the *linearization error* calculated at the iteration point $\boldsymbol{x}_h$, that is,

$$\alpha_j^h = f(\boldsymbol{x}_h) - f(\boldsymbol{y}_j) - \boldsymbol{\xi}_j^T(\boldsymbol{x}_h - \boldsymbol{y}_j) \quad \text{for all} \quad j \in J_h. \tag{3.7}$$

This linearization error defines how good the model is by describing the difference between the actual function value $f(\boldsymbol{x}_h)$ and the value of the linearization calculated at the auxiliary point $\boldsymbol{y}_j$. If the function f is convex, then $\alpha_j^h \geq 0$ by the definition of the subdifferential. Therefore, the cutting plane model $\hat{f}^h$ gives the lower approximation for the function f, in other words, $f(\boldsymbol{x}) \geq \hat{f}^h(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{R}^n$. The cutting plane model and the linearization error are illustrated in Figure 3.1.

In order to find the search direction $\boldsymbol{d}_h \in \mathbb{R}^n$, we solve the problem

$$\min_{\boldsymbol{d} \in \mathbb{R}^n} \quad \hat{f}^h(\boldsymbol{x}_h + \boldsymbol{d}) + \frac{1}{2} u_h \|\boldsymbol{d}\|^2. \tag{3.8}$$

Here u_h is the positive weighting parameter keeping the auxiliary point $\boldsymbol{y}_{h+1}$ in the region where $\hat{f}^h$ is close to f. The weighting parameter is the special feature of the proximal bundle method enabling the omission of the line search in the convex case, since the stepsize can be adjusted by chancing the weighting parameter. The role of the *stabilizing term* $\frac{1}{2} u_h \|\boldsymbol{d}\|^2$ is twofold. First, a piecewise linear function $\hat{f}^h$ might not have

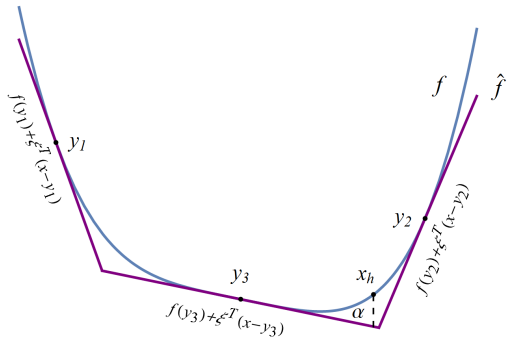


Figure 3.1: Cutting plane model and linearization error

a finite minimum, but by adding the stabilizing term, the existence and the uniqueness of a search direction is guaranteed. Second, the stabilizing term keeps the approximation local enough. This is important since the farther away the current iteration point $\boldsymbol{x}_h$ we get, the rougher and more unreliable the approximation of the function f becomes.

Due to piecewise linear nature of the function $\hat{f}^h$ in the problem (3.8), it is possible to rewrite the nonsmooth unconstrained problem (3.8) as the smooth, quadratic, constrained problem

$$\begin{aligned} \min \quad & v + \frac{1}{2} u_h \left\| \boldsymbol{d} \right\|^2 \\ \text{s.t.} \quad & -\alpha_j^h + \boldsymbol{\xi}_j^T \boldsymbol{d} \leq v \quad \text{for all } j \in J_h \\ & \boldsymbol{d} \in \mathbb{R}^n,\ v \in \mathbb{R} \end{aligned} \tag{3.9}$$

involving $n+1$ variables and $|J_h|$ constraints. However, it is often easier to solve the quadratic dual problem of the problem (3.9) given in the form

$$\begin{aligned} \min \quad & \frac{1}{2u_h} \Big\| \sum_{j \in J_h} \lambda_j \boldsymbol{\xi}_j \Big\|^2 + \sum_{j \in J_h} \lambda_j \alpha_j^h \\ \text{s.t.} \quad & \sum_{j \in J_h} \lambda_j = 1 \\ & \lambda_j \geq 0 \quad \text{for all } j \in J_h. \end{aligned} \tag{3.10}$$

If λ_j for all $j \in J_h$ yields a solution for the problem (3.10), then the unique solution of the problem (3.9) is of the form

$$\begin{aligned} \boldsymbol{d}_h &= -\frac{1}{u_h} \sum_{j \in J_h} \lambda_j \boldsymbol{\xi}_j \\ v_h &= -\left(\frac{1}{u_h} \Big\| \sum_{j \in J_h} \lambda_j \boldsymbol{\xi}_j \Big\|^2 + \sum_{j \in J_h} \lambda_j \alpha_j^h \right). \end{aligned} \tag{3.11}$$

A notable advantage of bundle methods from the practical point of view is that the size of the quadratic problem (3.9) can be controlled by restricting the size of the bundle. A simple strategy is to update the set of bundle indices by $J_{h+1} = J_h \cup \{h+1\}$ until the maximum size of the bundle $J_{\max}$ is reached. After that, we may update the bundle by $J_h = J_{h-1} \cup \{h\} \setminus \{h - J_{\max}\}$ or we may remove the index $j \in J_h$ with the largest linearization error α_j^h. However, the index related to the current iteration point must be included in the bundle. One more sophisticated technique to control the size of the bundle is the subgradient aggregation strategy proposed in [82] accumulating information by aggregating

subgradients and linearization errors from previous iterations. All the methods implemented in this dissertation utilize the quadratic solver by Lukšan [100] to solve these quadratic direction finding problems.

Once the search direction $\boldsymbol{d}_h$ is found, we decide whether we want to use it to generate a new iteration point or not. For that, a new auxiliary point $\boldsymbol{y}_{h+1} = \boldsymbol{x}_h + \boldsymbol{d}_h$ is calculated and we test whether

$$f(\boldsymbol{y}_{h+1}) \leq f(\boldsymbol{x}_h) + mv_h \tag{3.12}$$

or not, where the constant $m \in (0, \frac{1}{2})$ in (3.12) is the decrease parameter supplied by the user. We validate the condition by observing that we can write v_h in the form

$$v_h = \hat{f}^h(\boldsymbol{x}_h + \boldsymbol{d}_h) - f(\boldsymbol{x}_h),$$

due to the fact that v_h is the solution of the problem (3.9). Since $v_h < 0$, it is justified to consider v_h as the *predicted descent* of the function f at the point $\boldsymbol{x}_h$. Therefore, if the condition (3.12) holds, the model is good and the function value at the new auxiliary point $\boldsymbol{y}_{h+1}$ is significantly better than the function value at the current iteration point $\boldsymbol{x}_h$. Hence, we select the auxiliary point $\boldsymbol{y}_{h+1}$ as the new iteration point $\boldsymbol{x}_{h+1} = \boldsymbol{y}_{h+1}$. This selection is called the *serious step*. Therefore, the significant decrease is obtained at every serious step. Moreover, if $-v_h < \varepsilon$, where $\varepsilon > 0$ is the stopping tolerance, we may stop with $\boldsymbol{x}^* = \boldsymbol{x}_h$ as a solution.

Even if the direction $\boldsymbol{d}_h$ is descent to the model $\hat{f}_h$, it is not necessarily descent to the function f. If the condition (3.12) does not hold, the approximation of the function f given by the cutting plane model is not good enough, and either $\boldsymbol{d}_h$ is not the descent direction to the function f or we obtain only a slight decrease to the function f. In order to improve the approximation, we perform the *null step*. The current iteration point $\boldsymbol{x}_h$ remains unchanged, in other words $\boldsymbol{x}_{h+1} = \boldsymbol{x}_h$, and only the search direction will differ. Furthermore, we update the bundle $\mathcal{B}^h$ by adding more information about the function f around the current iteration point $\boldsymbol{x}_h$, or in other words, we add the triplet $(\boldsymbol{y}_{j+1}, f(\boldsymbol{y}_{j+1}), \boldsymbol{\xi}_{j+1} \in \partial f(\boldsymbol{y}_{j+1}))$ into the bundle $\mathcal{B}^h$ and the index $j+1$ into the set J_h. The null step yields the significant modification for the approximation of the function f and the direction $\boldsymbol{d}_{h+1}$ obtained after the null step differs from the previous direction $\boldsymbol{d}_h$ [84].

In the convex proximal bundle method, a line search strategy to define the stepsize is not mandatory and is thus omitted here since we can control the stepsize by adjusting the weighting parameter u_h. The appropriate selection of the weighting parameter is important since with too large fixed u_h we may end up taking too short steps, while with a

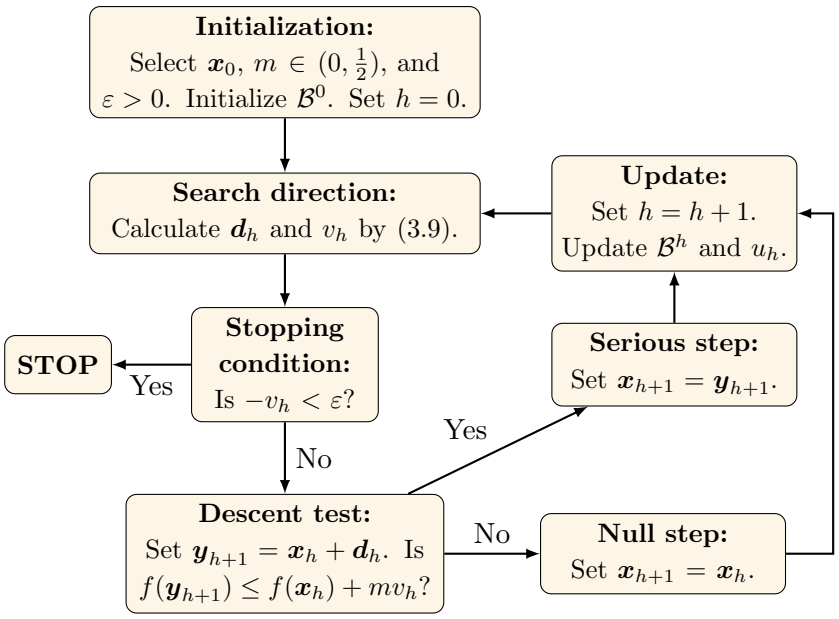


Figure 3.2: Flowchart of a general proximal bundle method

too small fixed u_h we may require unnecessarily many null steps. The implementations of the methods in this dissertation utilize mainly the weighting update procedure presented by Kiwiel in [85] or its modification. However, there exist several strategies to update the weighting parameter u_h (see e.g. [86, 143]). To conclude the discussion about the proximal bundle method, the simplified flowchart is given in Figure 3.2.

Finally, we say a few words about difficulties caused by nonconvexity in the proximal bundle method. First, a nonconvex function may have several locally optimal solutions and the optimality condition (2.3) is only necessary. Thus, only a Clarke stationary solution can be guaranteed. Second, for a convex function, the cutting plane model (3.6) gives the lower approximation of the function and the linearization error (3.7) is nonnegative. Nevertheless, in the nonconvex case, these are not true, and the linearization error might be very small or negative even if an auxiliary point is far from the current iteration point. Thus we do not learn any useful information. To overcome this, we may replace the linearization errors with the *subgradient locality measures*

$$\beta_j^h = \max\{|\alpha_j^h|\,,\, \gamma\|\boldsymbol{x}_h - \boldsymbol{y}_j\|^2\},$$

where $\gamma \geq 0$ is the *distance measure parameter* supplied by the user. In the convex case, $\gamma = 0$ and the subgradient locality measure coincides

with the linearization error α_j^h. Third, in the nonconvex case it might be that the bundle update with the subgradient from the auxiliary point of the form $\boldsymbol{y}_{h+1} = \boldsymbol{x}_h + \boldsymbol{d}_h$ does not improve the model significantly, and thus, the line search procedure is mandatory to ensure the global convergence. The two-point line search strategy [110] is discussed more in Section 3.3 and some others are presented in [84].

To summarize the discussion about the single-objective proximal bundle method, we mention that by assuming the objective function to be weakly semismooth in addition to local Lipschitz continuity, the proximal bundle method can be proven to be globally convergent [85, 110, 143]. Furthermore, the number of null steps performed can be shown to be finite, unless the optimal solution is reached [84].

3.3 Multiobjective Proximal Bundle Method

In the nutshell, the multiobjective proximal bundle method (MPB) is a version of the single-objective proximal bundle method using the improvement function (3.2) as the objective function. This method is able to solve nonconvex constrained multiobjective problems. The essential features of MPB are that it improves all the objective functions simultaneously and treats the objectives as they are without involving the scalarization in the classical sense. A version of the proximal bundle method utilizing the improvement function for a single-objective nonconvex constrained problem was presented by Mäkelä and Neittaanmäki [110]. This single-objective version is then extended for the multiobjective case, and MPB as it is, was used in [113] as a part of the interactive NIMBUS method. Our discussion in this section bases on two original publications: in the publication I, the standard MPB was investigated especially with respect to different generalized convexities and in the publication II, MPB was further extended by involving the scaled improvement function.

We begin by describing the idea of the method with the scaled improvement function. Notice that with the suitable selection of scaling functions, this presentation reduces back to the standard MPB. Consider the problem (MOP) with the set of feasible solutions $X \subseteq \mathbb{R}^n$ defined as in (2.1). As a bundle-based method, we assume that we can evaluate arbitrary subgradients $\boldsymbol{\xi}_{i,j} \in \partial f_i(\boldsymbol{y}_j)$ and $\zeta_{i,j} \in \partial \mu_i(f_i(\boldsymbol{y}_j))$ for all $i \in I$ and $\boldsymbol{\xi}_{l,j} \in \partial g_l(\boldsymbol{y}_j)$ and $\zeta_{l,j} \in \partial \delta_l(g_l(\boldsymbol{y}_j))$ for all $l \in L$ at every auxiliary point $\boldsymbol{y}_j \in \mathbb{R}^n$ and $j \in J_h$ in addition to the corresponding function values.

In the spirit of Theorem 3.1.1, we search the direction $\boldsymbol{d}_h \in \mathbb{R}^n$ as a

solution of the problem

$$\begin{aligned} \min \quad & H_s(\boldsymbol{x}_h + \boldsymbol{d}, \boldsymbol{x}_h) \\ \text{s.t.} \quad & \boldsymbol{d} \in \mathbb{R}^n \end{aligned} \tag{3.13}$$

at the h-th iteration of MPB. As in the single-objective proximal bundle method, we make a convex piecewise linear approximation of an objective function $H_s(\,\cdot\,, \boldsymbol{x}_h)$ by linearizing objective and constraint functions like in (3.6). Thus, we obtain the approximation for the scaled improvement function of the form

$$\widehat{H}^h(\boldsymbol{x}) = \max\left\{\widehat{\mu f}^{\,h}_{i,j}(\boldsymbol{x}) - \mu_i(f_i(\boldsymbol{x}_h)), \widehat{\delta g}^{\,h}_{l,j}(\boldsymbol{x}) \,\middle|\, i \in I, l \in L, j \in J_h\right\},$$

where the cutting plane models for the scaled objective and constraint functions are

$$\begin{aligned} \widehat{\mu f}^{\,h}_{i,j}(\boldsymbol{x}) &= \mu_i(f_i(\boldsymbol{y}_j)) + \zeta_{i,j}(\boldsymbol{\xi}_{i,j})^T(\boldsymbol{x} - \boldsymbol{y}_j) \quad \text{for all} \quad i \in I,\, j \in J_h \quad \text{and} \\ \widehat{\delta g}^{\,h}_{l,j}(\boldsymbol{x}) &= \delta_l(g_l(\boldsymbol{y}_j)) + \zeta_{l,j}(\boldsymbol{\xi}_{l,j})^T(\boldsymbol{x} - \boldsymbol{y}_j) \quad \text{for all} \quad l \in L,\, j \in J_h. \end{aligned}$$

Note that in the convex case, we can ensure that $\zeta_{i,j}(\boldsymbol{\xi}_{i,j}) \in \partial(\mu_i \circ f_i)(\boldsymbol{y}_j)$ and $\zeta_{l,j}(\boldsymbol{\xi}_{l,j}) \in \partial(\delta_l \circ g_l)(\boldsymbol{y}_j)$. However, in the nonconvex case, these are not necessarily valid. Then we can either assume the subdifferential regularity for f_i and g_l for all $i \in I$ and $l \in L$ or to use smooth scaling functions like exemplified in Section 3.1 (see the original publication II).

Furthermore, we obtain the approximation of the problem (3.13) by

$$\begin{aligned} \min \quad & \widehat{H}^h(\boldsymbol{x}_h + \boldsymbol{d}) + \frac{1}{2}u_h\|\boldsymbol{d}\|^2 \\ \text{s.t.} \quad & \boldsymbol{d} \in \mathbb{R}^n, \end{aligned}$$

and as was seen in Section 3.2, this problem can be rewritten as a smooth quadratic problem

$$\begin{aligned} \min \quad & v + \frac{1}{2}u_h\|\boldsymbol{d}\|^2 \\ \text{s.t.} \quad & -\beta^h_{i,j} + \zeta_{i,j}(\boldsymbol{\xi}_{i,j})^T\boldsymbol{d} \le v, \;\; i \in I,\, j \in J_h \\ & -\beta^h_{l,j} + \zeta_{l,j}(\boldsymbol{\xi}_{l,j})^T\boldsymbol{d} \le v, \;\; l \in L,\, j \in J_h \\ & \boldsymbol{d} \in \mathbb{R}^n, \end{aligned} \tag{3.14}$$

where subgradient locality measures are

$$\begin{aligned} \beta^h_{i,j} &= \max\left\{|\mu_i(f_i(\boldsymbol{x}_h)) - \widehat{\mu f}^{\,h}_{i,j}(\boldsymbol{x}_h)|,\, \gamma_i\|\boldsymbol{x}_h - \boldsymbol{y}_j\|^2\right\} \\ \beta^h_{l,j} &= \max\left\{|-\widehat{\delta g}^{\,h}_{l,j}(\boldsymbol{x}_h)|,\, \gamma_l\|\boldsymbol{x}_h - \boldsymbol{y}_j\|^2\right\}. \end{aligned}$$

Here the distance measure parameters γ_i and γ_l for all $i \in I$ and $l \in L$ are nonnegative, and if the objective or constraint function is convex, then the corresponding distance measure parameter is zero. Similarly to the proximal bundle method, instead of the problem (3.14), we can solve its dual problem of the similar form than the problem (3.10).

The last thing to discuss about the algorithm of MPB is the line search. In MPB, the two-point line search strategy [110] is applied. The benefit of this line search strategy is its ability to detect the discontinuities in the gradients of objective functions. The aim is to determine the stepsize $0 < t_h \leq 1$ such that the value of $H_s(\boldsymbol{x}_h + t_h \boldsymbol{d}_h, \boldsymbol{x}_h)$ is minimized when $\boldsymbol{x}_h + t_h \boldsymbol{d}_h \in \mathbb{R}^n$. Let the line search parameters $m_L \in (0, \frac{1}{2})$, $m_R \in (m_L, 1)$ and $\bar{t} \in (0, 1]$ be fixed. We begin by searching the largest $t_L^h \in [0, 1]$ such that

$$\max\{\mu_i(f_i(\boldsymbol{x}_h + t_L^h \boldsymbol{d}_h)) - \mu_i(f_i(\boldsymbol{x}_h)) \,|\, i \in I\} \leq m_L t_L^h v^h \quad \text{and}$$
$$\max\{\delta_l(g_l(\boldsymbol{x}_h + t_L^h \boldsymbol{d}_h)) \,|\, l \in L\} \leq 0.$$

If $t_L^h \geq \bar{t}$, then we take the *long serious step* by setting $\boldsymbol{y}_{h+1} = \boldsymbol{x}_h + t_L^h \boldsymbol{d}_h$ and $\boldsymbol{x}_{h+1} = \boldsymbol{y}_{h+1}$. Since we obtain the significant improvement for the objective function with the long serious step, there is no need to investigate the discontinuities in the gradients.

If the significant improvement is not obtained, we take either the *short serious step* or the null step. If $0 < t_L^h < \bar{t}$, then the short serious step is taken by setting $\boldsymbol{x}_{h+1} = \boldsymbol{x}_h + t_L^h \boldsymbol{d}_h$ and $\boldsymbol{y}_{h+1} = \boldsymbol{x}_h + t_R^h \boldsymbol{d}_h$. Furthermore, if $t_L^h = 0$, we take the null step by setting $\boldsymbol{x}_{h+1} = \boldsymbol{x}_h$ and $\boldsymbol{y}_{h+1} = \boldsymbol{x}_h + t_R^h \boldsymbol{d}_h$. For both of these steps we need to define $t_R^h > t_L^h$ such that

$$-\beta_{i,h+1}^{h+1} + \zeta_{i,h+1}(\boldsymbol{\xi}_{i,h+1})^T \boldsymbol{d}_h \geq m_R v_h.$$

Due to this condition, the points $\boldsymbol{x}_{h+1}$ and $\boldsymbol{y}_{h+1}$ lie in the different sides of the discontinuity of the gradient yielding the significant change for the model [110]. The simplified flowchart of MPB is given in Figure 3.3.

The solution obtained with MPB is weakly Pareto stationary. As usual, by assuming that all the objective and constraint functions of the problem (MOP) are convex, the better quality of the solution can be guaranteed. That is, the solution obtained with MPB is a globally weakly Pareto optimal solution. In fact, we can widen the class of functions yielding the globally weakly Pareto optimal solution instead of the Pareto stationary solution by recalling that Theorem 2.3.1 is the sufficient optimality condition not only for convex functions but also for f°-pseudoconvex functions. Then, by taking the maximum over f°-pseudoconvex functions, the resulting function is f°-pseudoconvex as well.

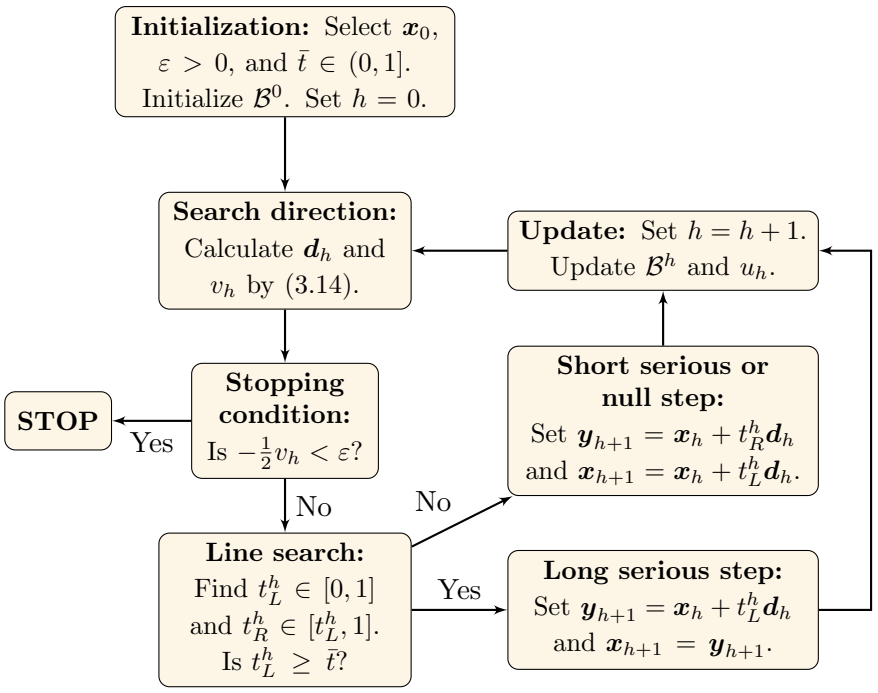


Figure 3.3: Flowchart of MPB

The convergence of MPB bases on Theorem 3.1.1 (ii)–(iv) and the convergence analysis of the proximal bundle method. Indeed, if we solve the single-objective problem with the improvement function as the objective by using the proximal bundle method and we find a solution $\boldsymbol{x}^* \in \mathbb{R}^n$ being Clarke stationary for the improvement function, or in other words, $\mathbf{0} \in \partial H_s(\boldsymbol{x}^*, \boldsymbol{x}^*)$, then the solution $\boldsymbol{x}^*$ is at least weakly Pareto stationary for the original multiobjective problem (MOP).

Let μ_i for all $i \in I$ and δ_l for all $l \in L$ be sign preserving and subdifferentially regular. Additionally, let μ_i for all $i \in I$ be f°-pseudoconvex and strictly increasing and δ_l for all $l \in L$ be f°-quasiconvex and increasing. Then, by assuming that the objective and constraint functions are weakly semismooth we can prove that

(i) if MPB stops with a finite number of iterations, then the solution is a weakly Pareto stationary solution for the problem (MOP).

(ii) any accumulation point of an infinite sequence of solutions generated by MPB is a weakly Pareto stationary solution for the problem (MOP).

Furthermore, if f_i for all $i \in I$, and g_l, δ_l for all $l \in L$ are f°-pseudoconvex and δ_l for all $l \in L$ is strictly increasing function, then a weakly Pareto stationary solution of the problem (MOP) is also a globally weakly Pareto optimal solution in the case (i). If in addition μ_i and δ_l are continuously differentiable or f_i and g_l are subdifferentially regular, a globally weakly Pareto optimal solution is obtained in the case (ii). Note that, in order to guarantee the weakly Pareto optimal solution, the improvement function needs to be f°-pseudoconvex. We emphasize that in Theorem 3.1.1 (iv), we make the assumption that constraint functions g_l for all $l \in L$ of the problem (MOP) are f°-quasiconvex. However, in order to ensure that the improvement function is f°-pseudoconvex, we have to make a stronger assumption that constraint functions are f°-pseudoconvex.

To conclude, MPB treats objectives as they are, and it is a descent method. The descent property of MPB follows from Theorem 3.1.1 (i). The method is globally convergent and by assuming the f°-pseudoconvexity of the objective and constraint functions, the solution obtained is guaranteed to be a globally weakly Pareto optimal. The numerical tests for multiobjective problems in [109] and for single-objective problems in [7] have shown that the implementation of MPB [107] is rather reliable and efficient. As a matter of fact, MPB is utilized as the reference method in the following two sections.

3.4 Multiple Subgradient Descent Bundle Method

This section is devoted to describe the idea of the multiple subgradient descent bundle method (MSGDB) presented in the original publication III. The idea of MSGDB is simple: find a direction improving all the objectives. MSGDB is designed for convex, unconstrained multiobjective optimization and it is a counterpart to the single-objective smooth steepest descent method in the convex nonsmooth multiobjective framework.

The idea in the well-known single-objective smooth steepest descent method is to move in the direction improving the objective most. This idea is rather intuitive and easy to implement, since the direction in interest is the negative direction of the gradient [10]. This fundamental idea dates back to the mid of the 19th century to the work of Cauchy [22]. There exists a couple of generalizations of the steepest descent method for smooth multiobjective optimization like the multiple gradient descent algorithm (MGDA) [35, 36] or the steepest descent method for multicriteria optimization [46].

The steepest descent method is generalized also for a single-objective

nonsmooth problem [93]. The drawback of this approach is that the whole ε-subdifferential needs to be known at every point, and as noted before, the whole subdifferential is exhaustive to calculate or may not even be possible to calculate. By using only one subgradient from the subdifferential, we end up with the single-objective subgradient method [144] but the lost of the guaranteed descent direction is the price to pay. Moreover, subgradient methods suffer the lack of the implementable (sub)gradient-based stopping condition. The idea of subgradient methods is extended also for the multiobjective framework [13, 29, 31], but they preserve the lack of a descent direction as their single-objective counterparts.

MSGDB is our share for the work related to extending the steepest descent direction idea for a more general framework. MSGDB is rather easy to implement and it utilizes the proximal bundle approach. Thus, only one subgradient needs to be evaluated at every point for each objective and we can ensure the existence of both the descent direction and the implementable stopping condition.

Some words about the idea of MSGDB are in order. First of all, MSGDB extends the ideas of MGDA presented by Désidéri in [35,36] and it is aimed for unconstrained, smooth multiobjective optimization. As a descent method for the smooth multiobjective problem, MGDA utilizes gradient-based information. As it was said, the negative direction of a gradient is a descent direction for a function. Once the individual descent directions for objectives are known, in MGDA, the common descent direction is obtained by taking a convex hull of these descent directions and finding the minimum norm element of that hull.

In order to adopt this idea for nonsmooth optimization, the first intuition suggests to replace classical gradients with subgradients. However, there is no guarantee that an opposite direction of an arbitrary subgradient would be a descent one. For the multiobjective nonsmooth convex case, we can prove that by taking the minimum norm element of the convex hull of the union of the subdifferentials $F(\boldsymbol{x})$ as in (2.4), we either obtain a common descent direction or the zero vector indicating that the point under consideration is weakly Pareto optimal. This is formulated in the following theorem.

Theorem 3.4.1. (Original publication III) *Let $f_i : \mathbb{R}^n \to \mathbb{R}$ be a convex function for all $i \in I$ and let $\boldsymbol{p}^* = \operatorname{argmin} \|\boldsymbol{p}\|$, where $\boldsymbol{p} \in \operatorname{conv} F(\boldsymbol{x})$ and $\boldsymbol{d}^* = -\boldsymbol{p}^*$. Either we have*

(i) *$\boldsymbol{d}^* = \boldsymbol{0}$ and the point $\boldsymbol{x}$ is weakly Pareto optimal, or*

(ii) *$\boldsymbol{d}^* \neq \boldsymbol{0}$ and the vector $\boldsymbol{d}^*$ is a common descent direction for every objective function.*

As noted before, calculating the whole subdifferential even for one objective is an exhaustive task, not to mention that subdifferentials would need to be calculated for several objectives. To follow the spirit of Theorem 3.4.1, in MSGDB we utilize the proximal bundle approach to approximate the subdifferentials of individual objective functions by gathering information from the neighbourhood of the current iteration point. Therefore, the only mild requirements are that values of all the objectives and one arbitrary subgradient for each objective can be evaluated at every iteration point.

In practice, we approximate the convex hull of $F(\boldsymbol{x})$ with the convex hull of the individual descent directions defined as the search directions of the single-objective proximal bundle method. Each of these individual directions is obtained by utilizing the proximal bundle method, described in Section 3.2, for the individual objective. For each objective, we form the bundle $\mathcal{B}_i^h$ for all $i \in I$ of the form (3.5) to approximate the subdifferential of the individual objective. Then the direction finding of the proximal bundle method is performed for individual objective until the descent condition (3.12) is satisfied. We denote the individual descent direction by $\boldsymbol{d}_i$ for all $i \in I$. As a result of the proximal bundle method, the direction $\boldsymbol{d}_i$ is a solution of the problem (3.9) and it is of the form (3.11).

Once individual descent directions are found separately, the next task is to determine a candidate for the common descent direction. We define the set C by

$$C = \operatorname{conv}\{\boldsymbol{d}_i \,|\, i \in I\},$$

which is used to approximate the convex hull of $F(\boldsymbol{x})$. Next, we find the minimum norm element of this set by solving the problem

$$\begin{aligned} \min \quad & \Big\| \sum_{i \in I} \lambda_i \boldsymbol{d}_i \Big\|^2 \\ \text{s.t.} \quad & \sum_{i \in I} \lambda_i = 1 \\ & \lambda_i \geq 0, \quad \text{for all } \; i \in I, \end{aligned} \tag{3.15}$$

having a unique solution λ_i for all $i \in I$ since the objective function of the problem (3.15) is strictly convex. We denote the solution candidate by $\boldsymbol{d}_h^* = -\sum_{i \in I} \lambda_i \boldsymbol{d}_i$ obtained from the solution of the problem (3.15). It is worth emphasizing that Theorem 3.4.1 does not guarantee that we actually obtain a common descent direction, as in the smooth case, but at least we obtain a good candidate for that.

If $\|\boldsymbol{d}_h^*\| = 0$, or $\|\boldsymbol{d}_h^*\| < \varepsilon$, where $\varepsilon > 0$ in practice, then we can stop and we have found a weakly Pareto optimal solution. Otherwise, we continue by checking whether our candidate is suitable for the common descent direction or not. If it is, we calculate the new iteration point $\boldsymbol{x}_{h+1} = \boldsymbol{x}_h + t\boldsymbol{d}_h^*$, where $t \geq \tau > 0$ such that t is the stepsize and τ is the stepsize tolerance. The stepsize can be calculated, for example, with the Armijo [4] like rule.

Again, since we are considering the approximation, the descent direction is not ensured and if any improvement is not achieved, we try to find another candidate by adding information from the new auxiliary point into the bundles and solving new individual descent directions. This cycle is like the null step in the single-objective proximal bundle method. If this null step does not pay dividends after some fixed number of times, we have a hint that we might have reached a weakly Pareto optimal solution even if the stopping condition is not yet satisfied. This is due to the nonsmooth nature of the problem and the formulation of the algorithm, and we might have lost some crucial information needed to obtain the zero vector as the candidate. In this case, we are willing to put in more computational effort and we improve the approximation of $F(\boldsymbol{x}_h)$ by involving not only the individual descent directions but all the subgradients stored. Thus, we now solve the problem

$$\begin{aligned} \min \quad & \left\| \sum_{i \in I} \sum_{j \in J_{i,h}} \lambda_{i,j} \boldsymbol{\xi}_{i,j} \right\|^2 + \sum_{i \in I} \sum_{j \in J_{i,h}} \lambda_{i,j} \alpha_{i,j}^h \\ \text{s.t.} \quad & \sum_{i \in I} \sum_{j \in J_{i,h}} \lambda_{i,j} = 1 \\ & \lambda_{i,j} \geq 0, \quad \text{for all } i \in I,\ j \in J_{i,h}, \end{aligned} \tag{3.16}$$

where a subgradient $\boldsymbol{\xi}_{i,j} \in \partial f_i(\boldsymbol{y}_j)$ and $\boldsymbol{y}_j$ is an auxiliary point, $\alpha_{i,j}^h$ is a linearization error of the form (3.7), and $J_{i,h}$ is a set of indices related to the bundle of the objective function $i \in I$. In fact, we have ended up with the same problem what we would solve, if we found a search direction in the single-objective proximal bundle method with the standard improvement function (3.2) as its objective. More about this is discussed in Section 3.3. The general idea of MSGDB is described in the simplified flowchart in Figure 3.4.

Under mild assumptions that the level set $\{\boldsymbol{x} \in \mathbb{R}^n \mid f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{x}_0),$ for all $i \in I\}$ is bounded and the stopping parameter ε is zero we can prove that if MSGDB stops with a finite number of iterations, then the solution is weakly Pareto optimal. Furthermore, if there exists an infinite cycle, the accumulation point generated by MSGDB is weakly Pareto

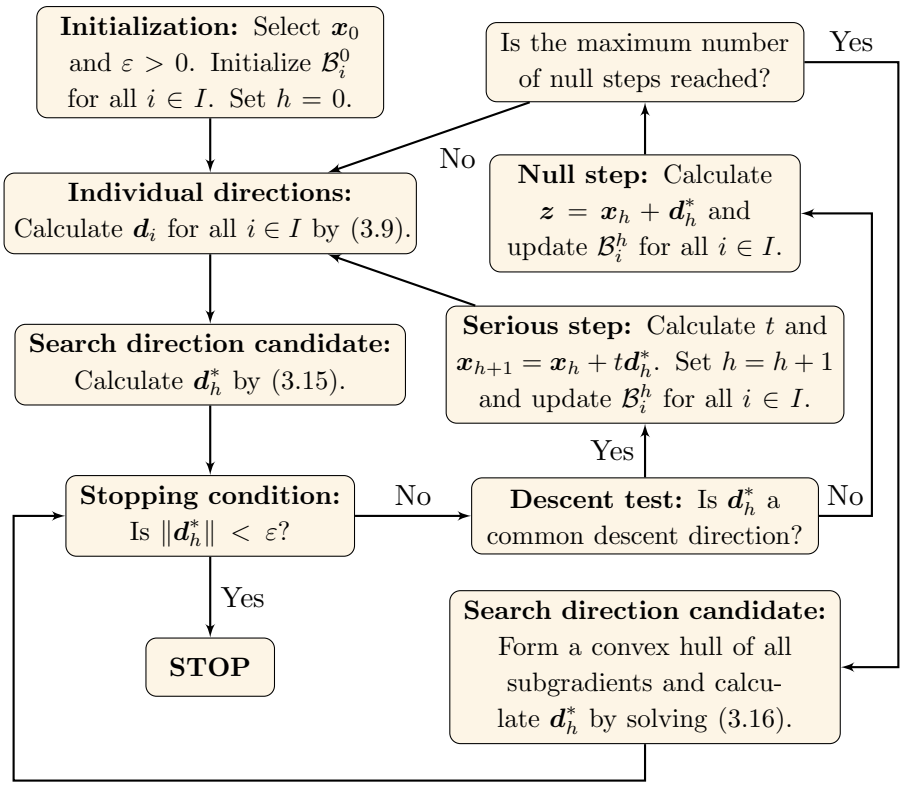
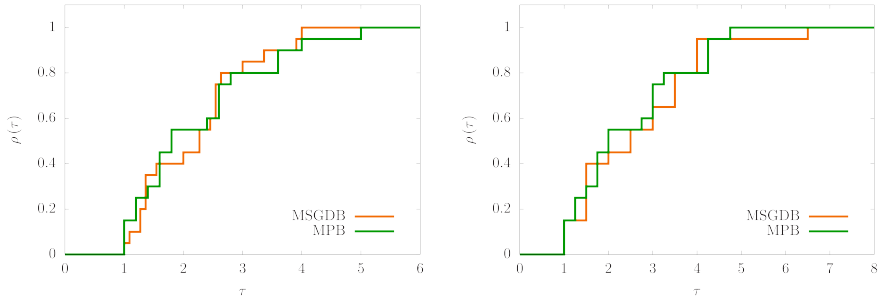


Figure 3.4: Flowchart of MSGDB

optimal. The proof of the convergence rely mainly on the convergence of the single-objective proximal bundle method.

The comparison of two methods is usually problematic and this becomes emphasized in the multiobjective case when we really cannot compare how well the known unique optimal solution is achieved as can be done in the single-objective case. We briefly comment on the numerical comparisons with MSGDB and MPB in the original publication III, where 20 unconstrained convex multiobjective problems are solved with two or three objectives. One important issue is the efficiency of the method. The computational efforts of MSGDB and MPB in terms of subgradient evaluations and iterations are quite similar, as we can notice from the performance profiles [37] given in Figure 3.5. In multiobjective optimization, it is important not only to produce solutions efficiently but also to be able to obtain different weakly Pareto optimal solutions from the same preference information. By the results in the original publication III, MSGDB and MPB produce mainly different solutions.



(a) Subgradient evaluations (b) Iterations

Figure 3.5: Performance profiles of MSGDB and MPB

To summarize, MSGDB is a descent method extending the ideas of the single-objective smooth steepest descent method for the nonsmooth, unconstrained, convex multiobjective framework. The method is rather simple to implement, since we need to evaluate only one arbitrary subgradient and function values at every point for each objective. If we compare MSGDB with MPB described in Section 3.3, we solve several quadratic subproblems at each iteration in MSGDB, but those can be solved parallel and the size of problems is smaller than the one quadratic subproblem solved in MPB at each iteration. Only exception is the problem (3.16) being actually the same problem as solved in MPB as the dual problem of the problem (3.14).

3.5 Multiobjective Double Bundle Method

The multiobjective double bundle method (MDBDC), proposed in the original publication IV and discussed more in the original publication V, is designed for multiobjective problems having objectives and constraints of the DC form meaning that they can all be represented as the difference of two convex functions. These DC functions form a wide subclass of nonconvex functions, but due to their special structure, we can still employ the convex analysis. In order to utilize the DC structure, the DC decomposition needs to be known. However, this DC decomposition is not unique, and unfortunately, it might be hard to single out even one DC decomposition for an arbitrary DC function.

Besides the ability to utilize convex analysis, the motivation to study DC optimization arises from the fact that some practical problems have objectives in the explicit DC form. The examples of this kind are clustering [8], spherical separability problems [54], production-transportation

planning [67], wireless sensor network planning [6], and data visualization [21]. All of these examples have multiobjective nature but they are solved as the single-objective problem. Additionally, in [74] a probabilistic lot sizing model was solved as the multiobjective DC problem.

As an introduction to the topic, we say some words about single-objective DC optimization. A great amount of studies dedicated to the theory of DC functions and single-objective DC optimization has been published in the past few decades (see e.g. [63,65,125,151,152]). Among the theoretical achievements, there exist many methods designed for single-objective DC optimization from different bases like DCA utilizing duality [90, 91, 125], proximal point based methods [147], bundle type methods [32,47,53,76,77], and branch-and-bound and outer approximation algorithms [68], to name just a few types of methods.

In Theorem 2.3.2, the necessary optimality condition is given when the objectives are assumed to be general locally Lipschitz continuous functions. The similar necessary optimality condition can be derived also by assuming that the objectives are DC functions as is done, for example, in [132]. From this point of view, if the objectives of the problem (MOP) are of the form $f_i = p_i - q_i$ and $\boldsymbol{x}' \in \mathbb{R}^n$ is a locally weakly Pareto optimal solution for the problem (MOP), then

$$\operatorname{conv}\{\partial q_i(\boldsymbol{x}') \,|\, i \in I\} \subseteq \operatorname{conv}\{\partial p_i(\boldsymbol{x}') \,|\, i \in I\}.$$

Unfortunately, this condition is hard to verify. However, it is easier to validated *Pareto criticality* of the solution $\boldsymbol{x}' \in \mathbb{R}^n$, that is,

$$\mathbf{0} \in \operatorname{conv}\{\partial p_i(\boldsymbol{x}') - \partial q_i(\boldsymbol{x}') \,|\, i \in I\}. \tag{3.17}$$

This is done in [73], where two proximal point based multiobjective methods for DC functions are presented. Every weakly Pareto stationary point $\boldsymbol{x}^* \in \mathbb{R}^n$ satisfies the condition (3.17), since

$$\mathbf{0} \in \operatorname{conv}\{\partial f_i(\boldsymbol{x}^*) \,|\, i \in I\} \subseteq \operatorname{conv}\{\partial p_i(\boldsymbol{x}^*) - \partial q_i(\boldsymbol{x}^*) \,|\, i \in I\},$$

but a Pareto critical point is not necessarily weakly Pareto stationary. Indeed, if we consider the biobjective problem with DC components $p_1(x) = \max\{-x, 2x\}$, $q_1(x) = \max\{-2x, x\}$, $p_2(x) = \max\{x^2, x\}$, and $q_2(x) = \max\{0.5x^2, -x\}$, where $x \in \mathbb{R}$. Let $x' = 0$. Now we can deduce from the intersection

$$\lambda\partial p_1(x') + (1-\lambda)\partial p_2(x') \cap \lambda\partial q_1(x') + (1-\lambda)\partial q_2(x')$$

that, for example, with $\lambda = 1$ this intersection equals $[-1, 1]$ and x' is Pareto critical by the condition (3.17). On the other hand, x' is

not weakly Pareto stationary, since $0 \notin \operatorname{conv}\{\partial f_1(x'), \partial f_2(x')\} = \{1\}$. Thus, the weak Pareto stationarity is stronger condition than the Pareto criticality. This happens also in the single-objective case, and therefore, the majority of DC methods produce only critical solutions.

The starting point for MDBDC is to combine ideas from the improvement function, the single-objective proximal bundle method for DC functions presented by Joki et al. in [76] and the escaping procedure described in [77] by Joki et al. In MDBDC, we utilize the improvement function to solve the problem (MOP), where objective functions are DC functions $f_i = p_i - q_i$ for all $i \in I$ and the set of feasible solutions is as in (2.1) such that $g_l = r_l - s_l$ for all $l \in L$. The improvement function $H(\,\cdot\,, \boldsymbol{x}_h)$ is now a DC function as the maximum taken over DC functions. The DC decomposition of the improvement function $H(\,\cdot\,, \boldsymbol{x}_h)$ can be obtained by rewriting functions f_i and g_l by using the technique presented in [65]. For the objectives, we get

$$f_i(\boldsymbol{x}) = p_i(\boldsymbol{x}) + \sum_{\substack{j \in I \\ j \neq i}} q_j(\boldsymbol{x}) + \sum_{t \in L} s_t(\boldsymbol{x}) - \sum_{j \in I} q_j(\boldsymbol{x}) - \sum_{t \in L} s_t(\boldsymbol{x})$$

for all $i \in I$ and the same technique can be applied for constraints as well. In order to simplify the notations, we denote

$$A_i(\boldsymbol{x}, \boldsymbol{y}) = p_i(\boldsymbol{x}) + \sum_{\substack{j \in I \\ j \neq i}} q_j(\boldsymbol{x}) + \sum_{t \in L} s_t(\boldsymbol{x}) - f_i(\boldsymbol{y}) \quad \text{and}$$
$$B_l(\boldsymbol{x}) = r_l(\boldsymbol{x}) + \sum_{\substack{t \in L \\ t \neq l}} s_t(\boldsymbol{x}) + \sum_{j \in I} q_j(\boldsymbol{x}).$$

Now the DC decomposition of $H(\,\cdot\,, \boldsymbol{y})$ can be written as

$$H(\boldsymbol{x}, \boldsymbol{y}) = H_1(\boldsymbol{x}, \boldsymbol{y}) - H_2(\boldsymbol{x}),$$

where

$$H_1(\boldsymbol{x}, \boldsymbol{y}) = \max\{A_i(\boldsymbol{x}, \boldsymbol{y}), B_l(\boldsymbol{x}) \,|\, i \in I, l \in L\} \quad \text{and}$$
$$H_2(\boldsymbol{x}) = \sum_{i \in I} q_i(\boldsymbol{x}) + \sum_{l \in L} s_l(\boldsymbol{x})$$

and both $H_1(\,\cdot\,, \boldsymbol{y})$ and H_2 are convex with respect to $\boldsymbol{x}$, since the sum of convex functions is convex as is the maximum taken over convex functions.

As a bundle method, in MDBDC we store information from previous iterations into the bundle. However, in this case, we have two bundles,

$\mathcal{B}_1^h$ and $\mathcal{B}_2^h$ of the form (3.5) related to the DC components $H_1(\,\cdot\,, \boldsymbol{y})$ and H_2, respectively. Depending on the bundle, the index j is an element of J_1^h if the bundle is for $H_1(\,\cdot\,, \boldsymbol{y})$ and $j \in J_2^h$ if the bundle is for H_2. In practice, the bundle $\mathcal{B}_1^h$ is the union of separate bundles for each $A_i(\,\cdot\,, \boldsymbol{x}_h)$ and B_l. At every iteration, we assume that besides the function values we can evaluate arbitrary subgradients of p_i, q_i, r_l, and s_l from which the values of functions $A_i(\,\cdot\,, \boldsymbol{x}_h)$, B_l, $H_1(\,\cdot\,, \boldsymbol{x}_h)$, and H_2 and their subgradients $\boldsymbol{a}_i$, $\boldsymbol{b}_l$, $\boldsymbol{h}_1$, and $\boldsymbol{h}_2$ can be composed for all $i \in I$ and $l \in L$.

By Theorem 3.1.1 we know that if $\boldsymbol{x}^* \in \mathbb{R}^n$ is weakly Pareto stationary, then $\mathbf{0} \in \partial H(\boldsymbol{x}^*, \boldsymbol{x}^*)$, and thus, $\boldsymbol{x}^*$ is Clarke stationary solution for the improvement function $H(\,\cdot\,, \boldsymbol{x}^*)$. We can now focus on the single-objective problem

$$\begin{aligned} \min \quad & H(\boldsymbol{x}_h + \boldsymbol{d}, \boldsymbol{x}_h) \\ \text{s.t.} \quad & \boldsymbol{d} \in \mathbb{R}^n. \end{aligned} \tag{3.18}$$

To solve the problem (3.18), we utilize the special cutting plane model which is based on the cutting plane model given in [76]. This cutting plane model captures both the convex and the concave behaviour of the function. First, we linearize convex functions $A_i(\,\cdot\,, \boldsymbol{y})$, B_l, and H_2 as in (3.6) and we denote them $\hat{A}_i^h$, $\hat{B}_l^h$, and $\hat{H}_2^h$, respectively. Thus we can define the cutting plane model for $H_1(\,\cdot\,, \boldsymbol{y})$ by

$$\hat{H}_1^h(\boldsymbol{x}) = \max\{\hat{A}_i^h(\boldsymbol{x}), \hat{B}_l^h(\boldsymbol{x}) \,|\, i \in I, l \in L\}$$

and finally we obtain the piecewise linear, nonconvex, DC approximation for the function $H(\,\cdot\,, \boldsymbol{y})$ by

$$\hat{H}^h(\boldsymbol{x}) = \hat{H}_1^h(\boldsymbol{x}) - \hat{H}_2^h(\boldsymbol{x}).$$

By utilizing the DC structure of the improvement function, we approximate the problem (3.18) with the quadratic problem of the form

$$\min_{\boldsymbol{d} \in \mathbb{R}^n} \quad P^h(\boldsymbol{d}) = \hat{H}_1^h(\boldsymbol{x}_h + \boldsymbol{d}) - \hat{H}_2^h(\boldsymbol{x}_h + \boldsymbol{d}) + \frac{1}{2t}\|\boldsymbol{d}\|^2, \tag{3.19}$$

where $t > 0$ is the proximity parameter. To obtain a global solution for this problem, we may utilize the solution approach described in [90, 91, 125]. In that purpose, we can reformulate the problem (3.19) and end up to solve convex, nonsmooth subproblems

$$\min_{\boldsymbol{d} \in \mathbb{R}^n} \{P_j^h(\boldsymbol{d}) = \hat{H}_1^h(\boldsymbol{x}_h + \boldsymbol{d}) - H_2(\boldsymbol{x}_h) - \boldsymbol{h}_{2,j}^T \boldsymbol{d} + \alpha_{2,j}^H + \frac{1}{2t}\|\boldsymbol{d}\|^2\}$$

for all j, where j is an index of the bundle $\mathcal{B}_2^h$ and $\boldsymbol{h}_{2,j} \in \partial H_2(\boldsymbol{y}_j)$. Thus the number of the subproblems can be controlled, since the size of the bundle can be freely chosen. This subproblem may be solved by utilizing the same dual approach described in Section 3.2. The overall global solution is denoted with $\boldsymbol{d}_t^h$ and it is the solution of the subproblem giving the smallest objective value.

Once we end up the situation, where $\|\boldsymbol{d}_t^h\| < \delta$ and $\delta > 0$, we either generate a new descent direction or Clarke stationarity of the problem (3.18) is achieved by applying the escaping procedure presented in [77]. We emphasize that ensuring the Clarke stationarity of the solution of a DC problem in general is not a trivial task. Besides information about the subdifferentials $\partial H_1(\boldsymbol{x}_h, \boldsymbol{x}_h)$ and $\partial H_2(\boldsymbol{x}_h)$ we already have, we need some information about the subdifferential $\partial H(\boldsymbol{x}_h, \boldsymbol{x}_h)$. This is due to the fact that by calculating arbitrary subgradients $\boldsymbol{h}_1 \in \partial H_1(\boldsymbol{x}_h, \boldsymbol{x}_h)$ and $\boldsymbol{h}_2 \in \partial H_2(\boldsymbol{x}_h)$, we cannot say that $\boldsymbol{h}_1 - \boldsymbol{h}_2 \in \partial H(\boldsymbol{x}_h, \boldsymbol{x}_h)$. The beauty of the applied escaping procedure lies in its ability to select $\boldsymbol{\xi}_1 \in \partial p(\boldsymbol{x})$, $\boldsymbol{\xi}_2 \in \partial q(\boldsymbol{x})$ for any DC function $f = p - q$ and $\boldsymbol{x} \in \mathbb{R}^n$ such that $\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2 \in \partial f(\boldsymbol{x})$ is ensured.

At every iteration of the escaping procedure, we approximate the Goldstein ε-subdifferential $\partial_\varepsilon^G H(\boldsymbol{x}_h, \boldsymbol{x}_h)$ with the set U_j consisting of subgradients calculated as the difference of subgradients of the DC components. Thus, the new direction can be found by evaluating $\boldsymbol{d}_{j+1} = -\bar{\boldsymbol{u}}_j / \|\bar{\boldsymbol{u}}_j\|$, where $\boldsymbol{u}_j$ is the solution of the problem

$$\min_{\boldsymbol{u} \in U_j} \frac{1}{2} \|\boldsymbol{u}\|^2.$$

If the resulting direction is not descent or Clarke stationarity is not achieved, the approximation of $\partial_\varepsilon^G H(\boldsymbol{x}_h, \boldsymbol{x}_h)$ is improved by adding a new subgradient. In order to exit from the escaping procedure, we either find a descent direction or the approximated Clarke stationary condition $\|\boldsymbol{\xi}^*\| \leq \delta$, where $\delta > 0$ and $\boldsymbol{\xi}^* \in \partial_\varepsilon^G H(\boldsymbol{x}_h, \boldsymbol{x}_h)$ is satisfied. If the latter is the case, MDBDC is terminated. The progress of MDBDC is illustrated in the simplified flowchart in Figure 3.6.

Under the mild assumptions that $\delta \in (0, 1)$, $\varepsilon > 0$, the subdifferentials $\partial H_1(\boldsymbol{x}, \boldsymbol{y})$ and $\partial H_2(\boldsymbol{x})$ are polytopes, and that the level set $\{\boldsymbol{x} \in X \mid f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{x}_0), \text{ for all } i \in I\}$ is compact, MDBDC can be proven to be finitely convergent to a weakly Pareto stationary point. This result bases on the properties of the improvement function in Theorem 3.1.1 and the convergence of the double bundle method for the single-objective DC problem given in [77].

The numerical experiments in the original publication IV performed with MDBDC and MPB have shown that MDBDC is a good alternative

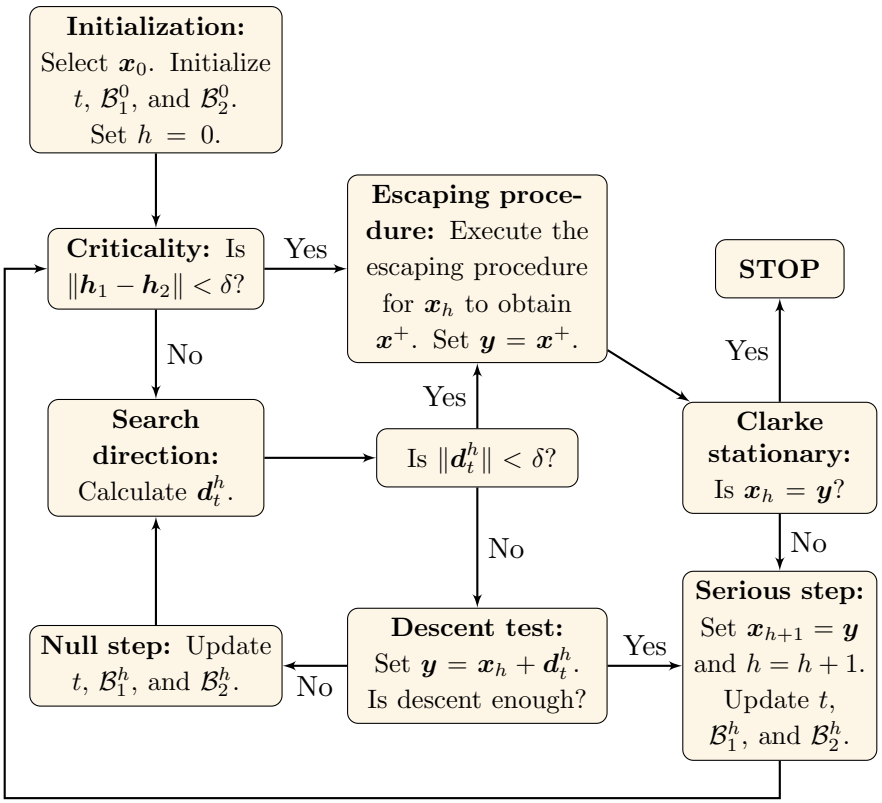
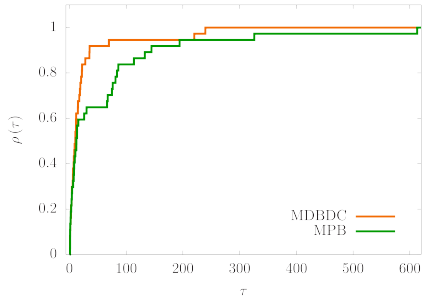


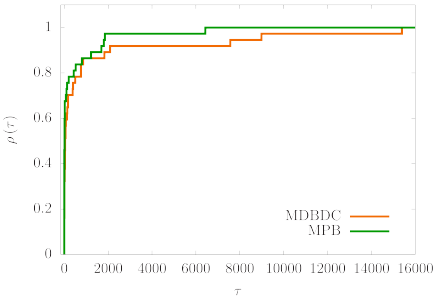
Figure 3.6: Flowchart of MDBDC

for MPB in the case of multiobjective DC optimization. In the computational point of view, the performance profiles [37] in Figure 3.7 show that MDBDC is slightly better that comes to the subgradient evaluations and MPB is better in CPU in the case of small test problems ($n \in [2, 100]$). However, in the larger test problems ($n \in [250, 500]$), MDBDC beats MPB. Another advantage of MDBDC is that it can overcome some local stationary points that MPB does not. Indeed, in these numerical tests this happened around 30% of cases.

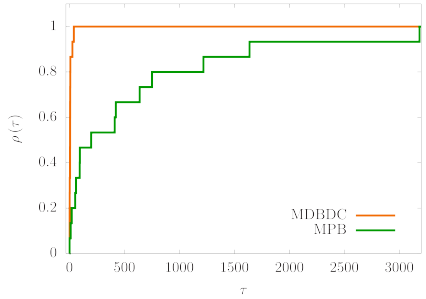
To summarize, the novelty of MDBDC is to utilize the DC structure of the nonconvex objectives in multiobjective optimization producing weakly Pareto stationary solutions. By the numerical experiments, we can conclude that with the more accurate model capturing the convex and concave behaviour of the objective, we can learn more about the objectives, and thus, obtain better solutions.



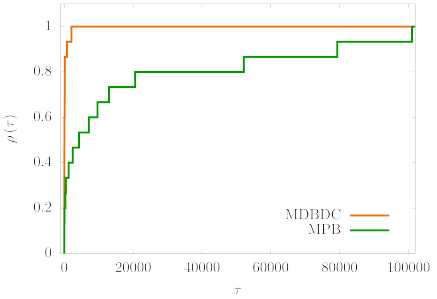
(a) Subgradient evaluations, $n \in [2, 100]$



(b) CPU, $n \in [2, 100]$



(c) Subgradient evaluations, $n \in [250, 500]$



(d) CPU, $n \in [250, 500]$

Figure 3.7: Performance profiles of MDBDC and MPB

Chapter 4

Methods Using Achievement Scalarizing Functions

This chapter focuses on a scalarization technique called the achievement scalarizing function (ASF). The class of ASFs consists of functions having certain properties, discussed more in Section 4.1, making them a reasonable and reliable way to aggregate several objectives into one objective. The use of ASFs dates back to the work of Wierzbicki [158–162] at the turn of the 1980's and the different types of ASFs are discussed, for instance, in [114, 116].

The idea of ASFs in brief is to find a solution as close as possible to the reference point by projecting the reference point to the Pareto set. The reference point involves some preference information supplied by the decision maker. Every component of the reference point describes which value the decision maker would like to obtain for each objective. Thus, it is important to find an optimal solution corresponding the reference point as well as possible. The general mathematical properties of ASFs are discussed in Section 4.1 and one specific family of ASFs, namely the two-slope parameterized ASFs, is described in Section 4.2. In Section 4.3, the interactive method utilizing two-slope parameterized ASFs is introduced.

4.1 Achievement Scalarizing Functions

When the objective functions of the problem (MOP) are scalarized by using an *achievement scalarizing function*, we are solving the following

scalarized problem

$$\min_{\boldsymbol{x} \in X} s_R(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{\lambda}), \tag{4.1}$$

where $s_R : \mathbb{R}^k \times \mathbb{R}^k_+ \to \mathbb{R}$ is an ASF, and the objective functions $f_i : \mathbb{R}^n \to \mathbb{R}$ are assumed to be lower semicontinuous. The set X is the compact set of feasible solutions, and $\mathbb{R}^k_+ = \left\{ \boldsymbol{y} \in \mathbb{R}^k \,|\, y_i \geq 0 \text{ for all } i \in I \right\}$. One widely used ASF is the Chebyshev type ASF

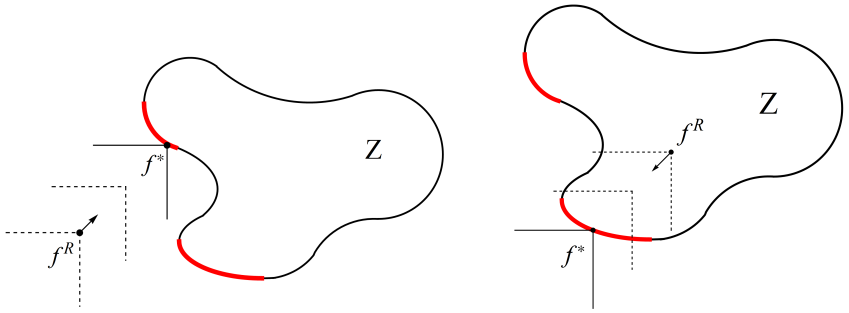
$$s_R(f(\boldsymbol{x}), \boldsymbol{\lambda}) = \max_{i \in I} \left\{ \lambda_i (f_i(\boldsymbol{x}) - f_i^R) \right\}, \tag{4.2}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^k$, and $\lambda_i > 0$ is a weighting coefficient for the objective function f_i. These coefficients specify the direction of the projection from the reference point $\boldsymbol{f}^R \in \mathbb{R}^k$ to the Pareto set. Note that, the Chebyshev type ASF is nonsmooth.

We define a *reference point* as a vector $\boldsymbol{f}^R$ consisting of desirable values for the objectives f_i, $i \in I$. The reference point is considered as an approachable way for the decision maker to express preference information. The advantage is that there is no need to order the objectives by their importance. We say that the reference point is *achievable* if $\boldsymbol{f}^R \in Z + \mathbb{R}^k_+$, where Z is an image of the feasible solutions in the objective space. Otherwise, the reference point is called *unachievable*. The unachievable reference point corresponds to the optimistic and infeasible hopes of the decision maker while the achievable reference point represents more secure and realistic wishes.

We illustrate the use of the Chebyshev type ASF in Figure 4.1. Figures 4.1a and 4.1b represent a set of feasible solutions in the objective space with two different reference points, an unachievable one in Figure 4.1a and an achievable one in Figure 4.1b. The right-angled contours of the ASF are expressed with black lines. In Figure 4.1a, the reference point is unachievable, and the ASF minimizes the distance from the reference point to the feasible set. Thus, the contours increase towards the feasible set and the solution obtained is the feasible solution touching the contour first. In Figure 4.1b, the reference point is achievable, and therefore, the ASF minimizes the maximum value of the negative difference between the reference point and the Pareto set in the objective space. In this case, the solution obtained is a feasible solution touching the contour last.

Once the multiobjective problem is converted into a single-objective one, there are two essential issues to consider. First, is the optimal solution of the scalarized problem (weakly) Pareto optimal solution for the original multiobjective problem? Second, is it possible to obtain every Pareto optimal solution with the selected scalarization?



(a) Unachievable reference point (b) Achievable reference point

Figure 4.1: Graphical illustration of Chebyshev type ASF

The charm of ASFs originates from the fact that under mild assumptions, the above mentioned properties can be guaranteed. Indeed, if the ASF is strictly increasing, then the optimal solution of the scalarized problem (4.1) is a weakly Pareto optimal solution of the problem (MOP). If ASF is strongly increasing, then the solution of the scalarized problem (4.1) is known to be a Pareto optimal solution. Additionally, by utilizing a strictly increasing ASF, every weakly Pareto optimal solution may be obtained by moving only the reference point. [161, 162]

Based on these definitions, any strongly increasing ASF is strictly increasing as well, but the inverse is not necessarily true. For example, the Chebyshev type ASF (4.2) is strictly increasing but not strongly increasing. Nevertheless, in order to obtain the good properties of the strongly increasing ASF, we can add the *augmentation term* to the strictly increasing ASF. For example, in the situation illustrated in Figure 4.1, the augmentation term added to the Chebyshev type ASF widens the angle of contours a bit, and thus, it excludes weakly Pareto optimal solutions. The scalarized problem (4.1) can be written in the augmented form

$$\min_{\boldsymbol{x}\in X} s_R(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{\lambda}) + \rho \sum_{i\in I} \lambda_i (f_i(\boldsymbol{x}) - f_i^R),$$

where the scalar $\rho > 0$. By adding the augmentation term to the ASF, the strictly increasing ASF becomes strongly increasing, and therefore, the Pareto optimal solution is ensured as a solution.

4.2 Two-slope Parametrized ASFs

In the original publication VI, a family of two-slope parameterized achievement scalarizing functions is proposed. This family of ASFs extends both parameterized ASFs introduced in [122] by Nikulin et al.

and two-slope ASFs described in [105] by Ruiz et al. Here we briefly describe the basic properties of two-slope parameterized ASFs and exemplify them.

The Chebyshev type ASFs (4.2) are widely used, since they have good mathematical properties discussed in the previous section. One example of these properties is that any weakly Pareto optimal solution can be obtained. The Chebyshev type ASF corresponds to the use of the L_∞ metric. In addition, there exist other types of metrics, for instance, the linear L_1 metric. However, unlike with the L_∞ metric, not every weakly Pareto optimal solution is necessarily obtained in the nonconvex case. This is due to the fact that there might exist nonsupported solutions. Therefore, in [140] the additive ASF basing on the L_1 metric ensuring that every weakly Pareto optimal solution can be obtained is presented. This idea of the additive ASF was further generalized in [122], where the parameterized ASFs are introduced.

Sometimes more than just one Pareto optimal solution is wanted. This task can be compiled by solving the multiobjective problem by using several different scalarizing functions. The advantage is that we might find solutions of different kind. This is exemplified in Figure 4.2 where two different scalarizing functions are used with the same reference point. However, the selection of scalarizing functions is not an easy task. Indeed, by fixing the scalarizing function, also the solution is fixed, and randomly selected scalarizing functions may lead to a badly distributed set of solutions.

The parameterization used in the parameterized ASFs gives a systematic way to produce different Pareto optimal solutions from the same preference information. The parameterization of this kind yields a better distributed sample of the Pareto set. However, a limitation of the

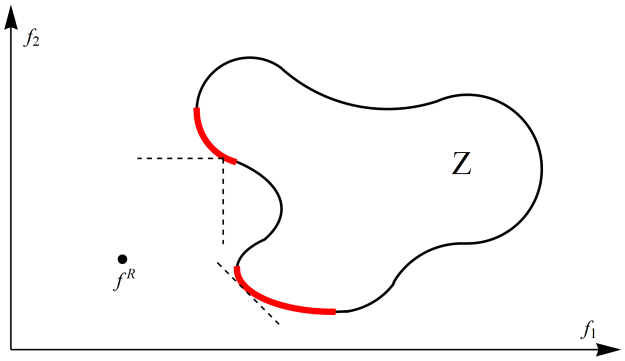


Figure 4.2: Solutions with two different ASFs and one reference point

parameterized ASFs is that the reference point should not be strictly dominated by some feasible point. To overcome this drawback, we utilize the idea of two different weighting vectors depending on the achievability of the reference point described in [105]. The use of different weighting vectors is sensible, since the decision maker usually prefers different solutions if the reference point is achievable than in the case of the unachievable reference point, as was suggested in [18]. Therefore, there is neither need for any assumptions about the reference point nor to test its achievability.

We consider a scalarized problem of the form

$$\min_{\boldsymbol{x}\in X} \hat{s}_R^q(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{\lambda}^U, \boldsymbol{\lambda}^A), \tag{4.3}$$

where $\hat{s}_R^q : \mathbb{R}^k \times \mathbb{R}_+^k \times \mathbb{R}_+^k \to \mathbb{R}$ is the *two-slope parameterized achievement scalarizing function* defined by

$$\hat{s}_R^q(f(\boldsymbol{x}), \boldsymbol{\lambda}^U, \boldsymbol{\lambda}^A) = \max_{I^q} \Big\{ \sum_{i\in I^q} \big[\max\big\{ \lambda_i^U (f_i(\boldsymbol{x}) - f_i^R), 0 \big\} + \min\big\{ \lambda_i^A (f_i(\boldsymbol{x}) - f_i^R), 0 \big\} \big] \Big\}.$$

Again, we assume that the objective functions $f_i : \mathbb{R}^n \to \mathbb{R}$ are lower semicontinuous and the set of feasible solutions X is compact. Here weighting vectors related to the unachievable reference point $\boldsymbol{\lambda}^U \in \mathbb{R}^k$ and to the achievable reference point $\boldsymbol{\lambda}^A \in \mathbb{R}^k$ are selected such that $\boldsymbol{\lambda}^U > 0$ and $\boldsymbol{\lambda}^A > 0$, the integer $q \in I$, and $I^q \subseteq I$ such that $|I^q| = q$. Indeed, the maximization is taken over all the different sets containing q integers from the interval $[1, k]$. Note that this formulation can be extended to contain also integer variables in addition to the continuous variables as in the original publication VII.

An interesting point to consider is whether the scalarized problem is harder to solve than the original multiobjective problem with one of the original objectives as the objective function or not. Due to the min-max nature of the problem (4.3), it is nonsmooth even though all the objective functions of the problem (MOP) are smooth. However, the problem (4.3) can be written as the nonlinear mixed integer problem, and therefore, preserve the possible smoothness of the original problem. But then the integer variables are introduced and special methods handling integer variables are needed.

By changing the values of the parameter q, different metrics varying in different combinations between the L_1 metric to the L_∞ metric are obtained. Extreme cases with the L_1 metric and the L_∞ metric are obtained when $q = k$ and $q = 1$, respectively, where k is the number of objective functions in the problem (MOP). Based on the formulation of the function $\hat{s}_R^q$, if $q = 1$, then $\hat{s}_R^q$ equals the two-slope ASF proposed in

[105] and with the unachievable reference point, the function $\hat{s}_R^q$ reduces back to the parameterized ASF described in [122].

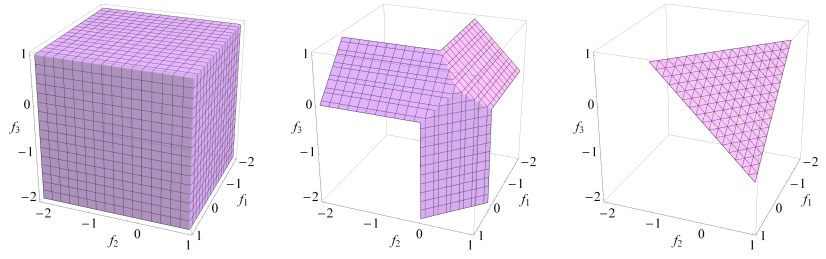
We illustrate the two-slope parameterized ASFs in the simplest non-trivial case where the problem (MOP) has three objectives, and thus, $k = 3$. In Figure 4.3, 1-level sets of a function $\hat{s}_R^q$ are presented with three different values of the parameter q. With the 1-level set we mean a set of points for which the distance from the reference point equals 1. The objectives are identity mappings $f_i(\boldsymbol{x}) = \boldsymbol{x}$, for all $i \in I$, the weighting vectors $\boldsymbol{\lambda}^U = \boldsymbol{\lambda}^A = (1, 1, 1)$ and the reference point is selected to be $\boldsymbol{f}^R = (0, 0, 0)^T$. As we see, the shapes of 1-levels vary from sharp to flat based on the choice of the parameter q.

If $q = 1$, an algebraic form of the two-slope parameterized ASF is

$$\begin{aligned}\hat{s}_R^1(f(\boldsymbol{x}), \boldsymbol{\lambda}^U, \boldsymbol{\lambda}^A) = \max\Big\{ & \max\left\{\lambda_1^U(f_1(\boldsymbol{x}) - f_1^R), 0\right\} + \min\left\{\lambda_1^A(f_1(\boldsymbol{x}) - f_1^R), 0\right\}; \\ & \max\left\{\lambda_2^U(f_2(\boldsymbol{x}) - f_2^R), 0\right\} + \min\left\{\lambda_2^A(f_2(\boldsymbol{x}) - f_2^R), 0\right\}; \\ & \max\left\{\lambda_3^U(f_3(\boldsymbol{x}) - f_3^R), 0\right\} + \min\left\{\lambda_3^A(f_3(\boldsymbol{x}) - f_3^R), 0\right\} \Big\}.\end{aligned}$$

This is graphically illustrated in Figure 4.3a. This case corresponds to the case of the Chebyshev type, or the L_∞, metric. When $q = 2$, we obtain a metric being between the L_∞ and L_1 metrics and this is exemplified in Figure 4.3b. In this case, the two-slope parameterized ASF takes the form

$$\begin{aligned}\hat{s}_R^2(f(\boldsymbol{x}), \boldsymbol{\lambda}^U, \boldsymbol{\lambda}^A) = \max\Big\{ & \max\left\{\lambda_1^U(f_1(\boldsymbol{x}) - f_1^R), 0\right\} + \min\left\{\lambda_1^A(f_1(\boldsymbol{x}) - f_1^R), 0\right\} \\ & + \max\left\{\lambda_2^U(f_2(\boldsymbol{x}) - f_2^R), 0\right\} + \min\left\{\lambda_2^A(f_2(\boldsymbol{x}) - f_2^R), 0\right\}; \\ & \max\left\{\lambda_1^U(f_1(\boldsymbol{x}) - f_1^R), 0\right\} + \min\left\{\lambda_1^A(f_1(\boldsymbol{x}) - f_1^R), 0\right\} \\ & + \max\left\{\lambda_3^U(f_3(\boldsymbol{x}) - f_3^R), 0\right\} + \min\left\{\lambda_3^A(f_3(\boldsymbol{x}) - f_3^R), 0\right\}; \\ & \max\left\{\lambda_2^U(f_2(\boldsymbol{x}) - f_2^R), 0\right\} + \min\left\{\lambda_2^A(f_2(\boldsymbol{x}) - f_2^R), 0\right\} \\ & + \max\left\{\lambda_3^U(f_3(\boldsymbol{x}) - f_3^R), 0\right\} + \min\left\{\lambda_3^A(f_3(\boldsymbol{x}) - f_3^R), 0\right\} \Big\}.\end{aligned}$$



(a) 1-level set for $q = 1$ (b) 1-level set for $q = 2$ (c) 1-level set for $q = 3$

Figure 4.3: Two-slope parameterized ASF with different values of q

Finally, when $q = 3$, the situation corresponds to the linear L_1 metric, as is seen in Figure 4.3c. The algebraic form of the two-slope parameterized ASF is now

$$\begin{aligned}\hat{s}_R^3(f(\boldsymbol{x}), \boldsymbol{\lambda}^U, \boldsymbol{\lambda}^A) = &\max\left\{\lambda_1^U(f_1(\boldsymbol{x}) - f_1^R), 0\right\} + \min\left\{\lambda_1^A(f_1(\boldsymbol{x}) - f_1^R), 0\right\} \\ &+ \max\left\{\lambda_2^U(f_2(\boldsymbol{x}) - f_2^R), 0\right\} + \min\left\{\lambda_2^A(f_2(\boldsymbol{x}) - f_2^R), 0\right\} \\ &+ \max\left\{\lambda_3^U(f_3(\boldsymbol{x}) - f_3^R), 0\right\} + \min\left\{\lambda_3^A(f_3(\boldsymbol{x}) - f_3^R), 0\right\}.\end{aligned}$$

We move on to discuss the theoretical properties of the two-slope parameterized ASFs. The essential features of scalarization discussed in Section 4.1 are claimed for the two-slope parameterized ASF.

Theorem 4.2.1. (Original publication VI)
For the scalarized problem (4.3) *it holds that:*

(i) *Any optimal solution of the scalarized problem is weakly Pareto optimal for the problem* (MOP)*, and among these optimal solutions of the scalarized problem, there exists at least one Pareto optimal solution.*

(ii) *If $\boldsymbol{x}^*$ is a weakly Pareto optimal solution for the problem* (MOP)*, then it is a solution of the scalarized problem with $\boldsymbol{f}^R = \boldsymbol{f}(\boldsymbol{x}^*)$, and an optimal value is zero.*

Based on Theorem 4.2.1, it is obvious that if an optimal solution of the scalarized problem is unique, then the obtained solution is a Pareto optimal solution for the original multiobjective problem.

As these theoretical results may give a hint, it can be established that the function $\hat{s}_R^q$ is increasing, and as discussed previously, we can add the augmentation term to the function $\hat{s}_R^q$ making it strongly increasing to guarantee a Pareto optimal solution in practice. Then, the resulting scalarized problem is

$$\min_{\boldsymbol{x} \in X} \hat{s}_R^q(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{\lambda}^U, \boldsymbol{\lambda}^A) + \rho \sum_{i \in I} \lambda_i (f_i(\boldsymbol{x}) - f_i^R), \tag{4.4}$$

where $\rho > 0$ and $\lambda_i > 0$ for all $i \in I$.

Moreover, it is straightforward to see that the function $\hat{s}_R^q$ preserves the convexity and the f°-pseudoconvexity of the objectives f_i, $i \in I$, and if all the objectives f_i, $i \in I$ are DC functions, then a function $\hat{s}_R^q$ is a DC function as well. In the case of convex or f°-pseudoconvex objective functions, a global optimum of the scalarized problem (4.3) is obtained, but in general, only a local optimum can be ensured.

To sum it up, the advantages of the two-slope parameterized ASF are that at least a weakly Pareto optimal solution is always found, and any weakly Pareto optimal solution may be obtained by changing the reference point. Moreover, there are no restrictions for the location of the reference point, since the two-slope parameterized ASF uses a suitable weighting coefficient in every case, and thus, the achievability tests of the reference point can be omitted. The parameterization used gives a systematic way to produce different (weakly) Pareto optimal solutions from the same preference information with different metrics.

In practice, this kind of systematic way to utilize preference information may be useful in some interactive methods. In order to find different (weakly) Pareto optimal solutions, the scalarized problem (4.3) can be solved with all or just some values of the parameter q, especially with the higher number of objectives. Thus, it is interesting to know more about how the value q affects the shape of the D-levels (i.e. the points whose distance from the reference point equals D). The numerical tests performed in the original publication VI have shown that the sparsity of the solutions produced with different values of the parameter q is good. In these tests, the computational times do not differ with different values of the parameter q in the convex case, but in the nonconvex case, the Chebyshev type metric turns out to be the most time-consuming metric. An example of the use of the two-slope parameterized ASF in the interactive framework is given in the following section and this interactive procedure is then applied in practice in Chapter 5.

4.3 Interactive Method with Two-slope Parameterized ASFs

In Section 2.4, the multiobjective methods were categorized in four different groups based on the role of the decision maker. In Chapter 3, three different no-preference methods from the first group were discussed, and in this chapter, the discussion continues with ASFs that can be seen as members of either the second or the third group, namely a posteriori or a priori method. Besides these two groups, ASFs fit also in the fourth group, involving interactive methods, with the suitable framework. Compared with the methods discussed previously, interactive methods need interaction between two parts: the analyst solving the mathematical problem and the decision maker solving the decision making problem during the whole solution process. This interaction is beneficial since the aim is to produce the most satisfying solution for the decision maker.

At the best, interactive methods are an active dialogue between the

decision maker and the analyst. Due to the active role of the decision maker, (s)he is able to learn about the problem what is possible to achieve and how the variables and objectives are linked together. Only after the learning phase the decision maker needs to make the final decision. Besides the learning, other advantages of the interactive methods include the realistic expectations of the decision maker and the saving in the computational time when the calculation of the whole Pareto set is not needed. Unfortunately, the benefits of interactive methods are also their weakness. Indeed, interactive methods demand much from the decision maker. At every iteration, new input and decisions are needed. During the solution process, the decision maker may get bored or forget things and the interaction takes time.

Many interactive methods have the similar core idea: first begin with the initialization phase followed by the learning phase and ending with the decision phase. In the initialization phase, some aid for the decision maker to select suitable preference information can be given by illustrating the range of the Pareto set. The learning phase consists of the following steps:

1. ask some preference information from the decision maker;
2. solve the optimization problem;
3. provide some solutions for the decision maker;
4. ask the decision maker to specify one preferred solution;
5. ask the new preference information from the decision maker.

These steps are repeated until the satisfying solution is found and the decision maker is ready to move on the decision phase to make the final selection. The main differences between interactive methods occur in how the preference information is given and how the different Pareto optimal solutions are obtained.

The communication between the decision maker and the analyst can be performed in different ways. In [112], the different types of preference information are categorised in five. The first category contains methods needing the aspiration levels for the objectives. One common form of the aspiration levels is the reference point. The second category is formed by the methods classifying the objectives such that the decision maker expresses her/his wishes by selecting the objectives to be improved and to be allowed to impair. It is noteworthy that the first two categories are quite similar, since the reference point can be formed by the classification (see e.g. [114]). The third category requires the decision maker to be

able to compare the different solutions and guide the solution process in this way. In the fourth category, the decision maker considers the trade-off information in the terms of the marginal rates of substitution, and the last category is the navigation, where the preference information is given as a direction.

Our focus is on the methods utilizing the reference point (see e.g. [19, 72, 111, 112, 121, 153, 160]) as the tool of communication. Even if preference information in these methods is similar, the solutions presented for the decision maker may differ. One option is to present only one solution obtained by solving some scalarized problem, as is done, for instance, in the GUESS method [19]. If we wish to present several alternatives as a sample of the Pareto set, we can give the decision maker a solution obtained using the reference point with some additional close solutions. Examples of this are the use of characteristic neighbours in the light beam search [72] and the perturbed reference points in the reference point method [160]. The closer the reference points are to the Pareto set, the narrower the sparse of solutions is whereas if the reference point is further from the Pareto set, the wider selection of Pareto optimal solutions is obtained [160]. In addition, as mentioned, the scalarization used defines the solution obtained. Thus, it is possible to use some different scalarizing functions and produce solutions from the same reference point, as is done in the synchronous NIMBUS method [115].

Like many other interactive methods, the multiobjective interactive method utilizing two-slope parameterized ASFs (MITSPA) presented in the original publication VII, uses reference points as the tool for handling preference information. The special features of our approach relate to the use of the two-slope parameterized ASF and the way the alternative solutions are obtained. The beauty of the two-slope parameterized ASF lies in the fact that instead of using some general ASFs, we can systematically produce different Pareto optimal solutions from the same preference information. Therefore, we are able to avoid the badly distributed set of solutions which may happen with randomly selected scalarizing functions. Next, we sketch the idea of MITSPA and discuss its properties.

At first, the ranges of the Pareto optimal solution are illustrated for the decision maker. The Pareto optimal solution $\boldsymbol{f}_0$ can be calculated with some no-preference method, like descent methods in Chapter 3, or for example with the two-slope parameterized ASF with the ideal or the utopian vector as the reference point. Therefore, somewhat neutral Pareto optimal solution can be presented as the starting point. After that, the decision maker selects the first reference point $\boldsymbol{f}_1^R$ and the scalarized problem (4.4) is solved. As stated in the previous section,

the solution of this scalarized problem with the augmentation term is Pareto optimal, and thus, the solutions presented for the decision maker are Pareto optimal. We recall that we are able to obtain as many different solutions as there are objectives. However, at each iteration only some of the solutions are presented for the decision maker. We denote the number of solutions presented with s. It is beneficial to keep s quite small since the high number of solutions presented simultaneously unnecessarily complicates the decision maker's task. Especially, if the number of objectives is large, some suitable selection of solutions is better to be presented instead of all values of the parameter q expressing how many different metrics there are available. If the decision maker is still interested to see more solutions from the same reference point, this is allowed. If the total number of original solutions selected and the supplementary solutions wanted exceeds the number of the objectives, or in other words, the number of different solutions obtained with the parameterization, more solutions can be calculated by varying the coefficients $\boldsymbol{\lambda}^U$ and $\boldsymbol{\lambda}^A$ in (4.4). The flowchart of MITSPA is given in Figure 4.4, where the abbreviation DM refers to the decision maker.

Amongst the solutions presented, the decision maker is asked to select the most preferred one to be the current solution $\boldsymbol{f}_h$. In order to enable the learning, the decision maker can select a new reference point $\boldsymbol{f}_h^R$ and new solutions are presented. Note that solutions from the previous iterations can be stored such that the decision maker can return to previous solutions if the newer solutions are less likeable.

Since the solutions and the reference points are in the same form, it is easier for the decision maker to select new reference points based on the previous solutions. In principle, a new reference point, and at the same time, a new iteration can be performed as many times as the decision maker wishes. The procedure stops when the satisfying solution for the decision maker is found or some fixed maximum number of iterations $h_{\max}$ is reached. Even if the maximum number is fixed, it is still important to confirm that the decision maker is willing to stop.

To conclude, the above framework gives an example of how to utilize the two-slope parameterized ASF in the interactive method. The preference information is given with the reference point which is considered as an approachable way for the decision maker to handle the preference information. The decision making is aided by giving a few different solutions at the time, and the solutions presented aim to reflect the decision maker's wishes. By giving the time for the learning, hopefully more satisfactory solutions are found. The practical value of this interactive framework is exemplified in Chapter 5, where MITSPA is applied for the real-life application of the disposal of the spent nuclear fuel.

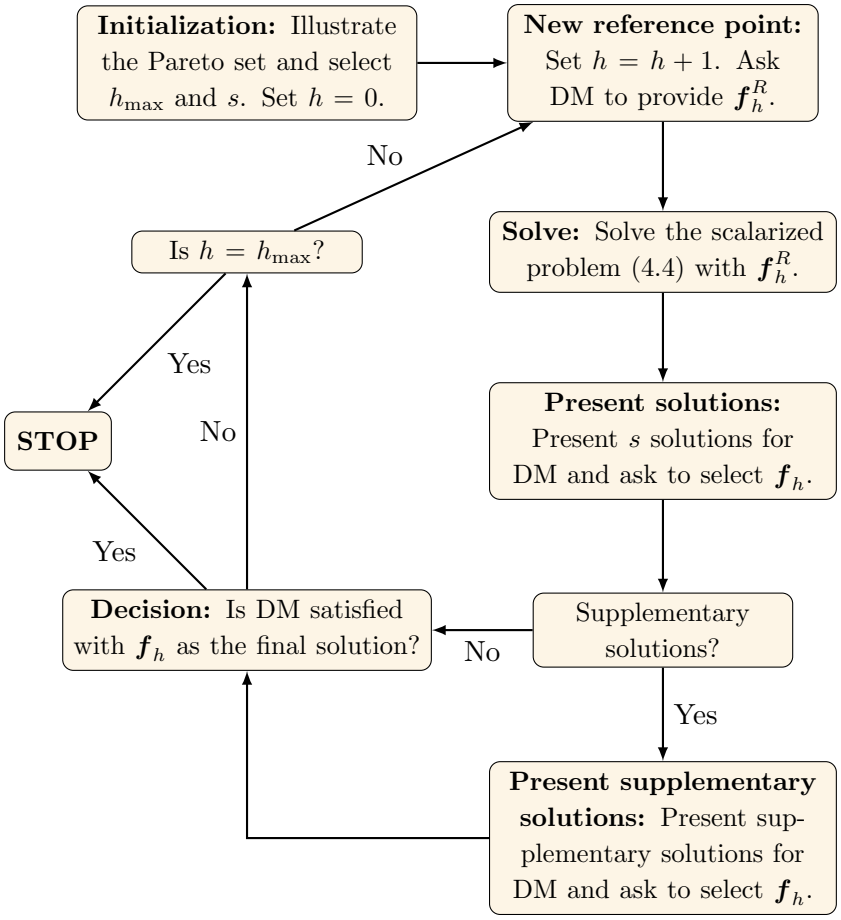


Figure 4.4: Flowchart of MITSPA

Chapter 5

Scheduling the Final Disposal of the Spent Nuclear Fuel

The final disposal of the spent nuclear fuel is a current topic, especially in Finland where the disposal will start in the near future. Indeed, Finland will be one of the first countries to begin the disposal in 2020's. The challenge in the disposal is its long term nature. Due to this, the decisions made today has long term consequences. Therefore, the careful planning and optimization of the processes are necessary.

In Finland, the geological disposal, which is widely accepted to be a safe and ethical solution, will be implemented. In the geological disposal, the assemblies removed from the reactor are first put into the water pool in the reactor hall to decrease the decay heat power and the radiation to the suitable level such that assemblies can be transferred to the interim storage to be stored in water pools for decades. Once the assemblies are cool enough, they are transferred into the encapsulation facility, where they are placed in the copper-iron canisters. After that, the canisters are placed in the vertical holes in the disposal tunnels. The disposal tunnels are connected to the central tunnel in the disposal facility situated in the bedrock. The last thing to do is to fill up and seal the disposal tunnel.

In the original publications VII and VIII, the aim is to model and solve the schedule for the disposal in Finland in terms of multiobjective optimization. The first publication considers the case when only one type of the spent nuclear fuel is disposed of and the latter one extends the model in the first publication for the case when all three fuel types used in Finland are involved. A study of the same type is discussed in [135], where the similar situation is modelled as a single-objective linear cost minimization problem. Another effort for the disposal schedule is in [137]

but there the area of the disposal facility is minimized. There exists also a research related to the placement of the spent nuclear fuel in the underground repository modelled as a multiobjective problem in [75].

In this chapter, we discuss the suitable models to optimize the disposal of the spent nuclear fuel in Finland. We sketch the models and consider their numerical challenges. Then two case studies are summarized. In these case studies, the interactive approach utilizing the two-slope parameterized ASFs (MITSPA) presented in Section 4.3 is applied.

5.1 Mathematical Model

In the original publications VII and VIII, the final disposal of the spent nuclear fuel is modelled as the multiobjective nonsmooth MINLP problems of the following form:

$$\begin{aligned} \min \quad & f_1(\boldsymbol{x}, \boldsymbol{y}), \ldots, f_k(\boldsymbol{x}, \boldsymbol{y}) \\ \text{s.t.} \quad & g_j(\boldsymbol{x}, \boldsymbol{y}) \leq 0, \quad j = 1, \ldots, m \\ & h_l(\boldsymbol{x}, \boldsymbol{y}) = 0, \quad l = 1, \ldots, p \\ & \boldsymbol{x} \in \mathbb{R}^n_+, \ \ \boldsymbol{y} \in \mathbb{Z}^r. \end{aligned} \tag{5.1}$$

The model in the original publication VII involves nine objectives and in the original publication VIII eight objectives. In the both cases the number of objectives is quite high. These objectives to be minimized in the case of one fuel type are:

1. maximum number of assemblies in the storage;
2. maximum storage time;
3. average storage time;
4. total number of canisters;
5. ending date of the encapsulation;
6. operation time of the encapsulation facility;
7. total length of disposal tunnels;
8. total length of the central tunnel;
9. total cost

and in the case of three fuel types they are:

1. number of additional water pools;
2. average storage time;

3. total number of canisters;
4. ending date of the encapsulation;
5. operation time of the encapsulation facility;
6. total length of disposal tunnels;
7. total length of the central tunnel;
8. total cost.

Note that due to the importance of safety issues, the safety aspects, like the thermal dimensioning of the spent nuclear fuel repository or the minimum cooling time of the spent fuel assembly, are considered as constraints. In the first case, the objectives 1, 2, 5, and 7–9 are nonlinear and the rest are linear. Concerning the nonlinear objectives, the objectives 1, 2, 5, and 9 are also nonsmooth. In the latter case, the objectives 4 and 6–8 are nonlinear and 4 and 8 are also nonsmooth. In the first case, the first three objectives are related to the interim storage, the next three for the encapsulation facility, the next two for the disposal facility, and the last objective considers the whole disposal process. Concerning the latter case, instead of the maximum number of assemblies in the storage the number of additional water pools is considered and the objective minimizing the maximum storage time is omitted in order to reduce the size of the model. The rest of the objectives are the same than in the first case.

The total cost is an obvious objective but also the rest of the objectives are cost factors. In the case where the actual price is not known it is reasonable to minimize the source of costs. Furthermore, these objectives have also other reasons to be optimized. For example, the faster the assemblies are disposed of the safer it is. On the other hand, the size of the bedrock is bounded, so the area needed for the disposal is minimized.

As an intuition might suggest, these objectives are conflicting. Indeed, if we try to end the disposal as early as possible, the heat load of the canisters is higher and the larger area of the underground repository is needed. In order to handle the higher heat load, we either have to increase the storage time or leave empty assembly positions in the canisters. Naturally, all of these affect the costs.

As the result, the model (5.1) gives a schedule in terms of how many canisters are disposed of in each period. The assigning of assemblies in to the canisters is left as its own problem which is studied for example in [89,135,136,156,165]. We have made some assumptions, for instance, nothing has been disposed of yet and all the assemblies are identical and obtainable.

From the theoretical and the computational point of view the problem of the form (5.1) is hard to solve. Besides the quite high number of objectives, the problem involves binary variables. Moreover, the model includes nonlinear constraints, and in the case of one fuel type, one of them is also nonsmooth. The nonsmoothness arises from the dependence between the canister and the disposal tunnel spacing, and the maximum average power of the canister. This relation is approximated with a piecewise linear function. In the second case, these relations are defined separately for each fuel type and they are approximated with nonlinear functions. Furthermore, the sizes of the problems are large. Indeed, with 19 periods of 5 years and 11 removals from the reactor used in the first case study, the dimensions of the problem are 9 objectives, 440 continuous and 475 binary variables, 3 box constraints, 1144 linear constraints, and 20 nonlinear constraints. In the case of three fuel types, 19 periods of 5 years and 13 removals from the reactor, the dimensions are 8 objectives, 963 continuous, 77 binary, and 41 integer variables, and 887 linear, 61 nonlinear, and 890 box constraints.

In order to solve the problem (5.1) in the both cases, we use MITSPA described in Section 4.3. In these cases, this method is superior to the descent methods presented in Chapter 3, since those methods cannot handle binary variables. Compared with the use of only the two-slope parameterized ASF as such, the efforts needed for the interactive approach are profitable, since in the problem of this scale and importance, everything we can learn is valuable.

In Section 4.1 it was discussed that when ASFs are used it is important to address the questions whether the optimal solution of the scalarized problem yield a Pareto optimal solution and whether we are able to obtain every Pareto optimal solution. As it is pointed out in [41], in addition to these, it is important to consider the question whether we can solve the scalarized problem in the discrete case since then the scalarized problem may easily become very challenging. In Section 4.2, we noticed that the two-slope parameterized ASF is always nonsmooth. However, the problem (5.1) already contains nonsmooth functions, and thus, this does not change the theoretical difficulty of the problem. In practice, the nonsmoothness can be overcome by rewriting the problem as a MINLP problem. This, in its turn, yields to the introduction of some auxiliary binary variables. Nevertheless, the original problem already has binary variables so neither this does yield a significant increase for the level of the theoretical difficulty. In the following sections, we see how this problem is solved in practice and what kind of results are obtained.

5.2 Optimization Results with One Fuel Type

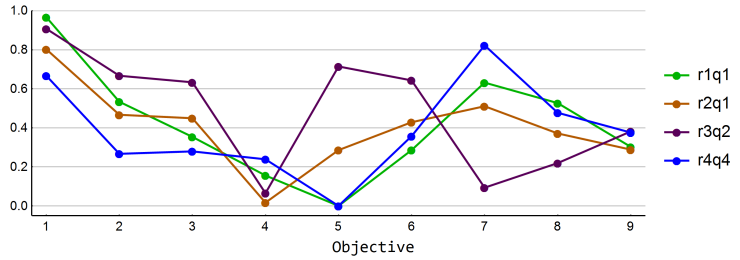
In this section, we give an example of how the interactive method in Section 4.3 is performed to solve the problem (5.1) with nine objectives from the previous section. These results are commented with more details in the original publication VII. The scalarized problem (4.4) in Step 2 of MITSPA is solved in GAMS [27] by using the method called BARON [81, 150] which is a global branch-and-cut type method for single-objective MINLP problems.

In the initialization phase of MITSPA, some ideas of the range of the Pareto set are given for the decision maker. Based on these, the decision maker selects the first reference point. In the learning phase, the decision maker guides the solution process with four different reference points, each corresponding to one iteration. For each reference point, we can calculate 9 different solutions by varying the value of the parameter q in (4.3) specifying the metric used. However, in each iteration we show only two different solutions from the nine possible solutions for the decision maker to ease the decision.

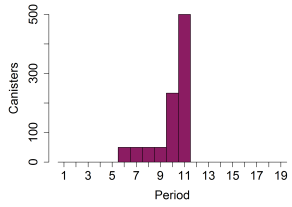
In the learning phase, it is observed that the solutions can be divided into two groups based on the starting time of the disposal. Interestingly, the solutions with early starting time are obtained with the small values of q and the solutions with later starting time with larger values of q. The solutions obtained in the learning phase are depicted in Figures 5.1 and 5.2. The notable issue is that if, for example, only the Chebyshev metric was used, no solutions with the late starting time would have been obtained.

In Figures 5.1a and 5.2a, the objective function values obtained are illustrated in the scale from 0 to 1 such that 0 is the ideal value for each objective and 1 represents the value of the component of the nadir vector for the objective in question. The different solutions are labelled based on the reference point used and the value of the parameter q. For example, the solution r1q1 is the result obtained by using the reference point 1 and $q = 1$.

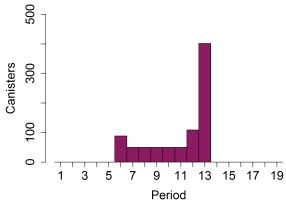
For instance, in Figure 5.1a, we can compare the solutions obtained with the reference point 1 and $q = 1$ (r1q1) and the solution obtained with the reference point 3 and $q = 2$ (r3q2). As is shown in Figure 5.1a, the solution r3q2 has slightly smaller maximum number of stored assemblies (obj. 1) and empty canister positions than the solution r1q1 (obj. 4). On the contrary, the total cost (obj. 9) and the maximum storage time (obj. 2) in the solution r3q2 are little higher than in the solution r1q1. The bigger differences can be seen, for example, in the average storage time (obj. 3) which is significantly smaller in the solution



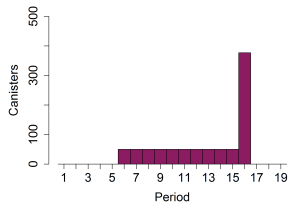
(a) The objective values for solutions



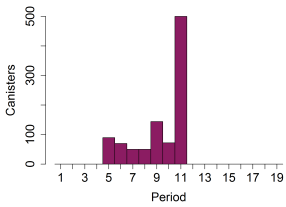
(b) The solution r1q1



(c) The solution r2q1



(d) The solution r3q2

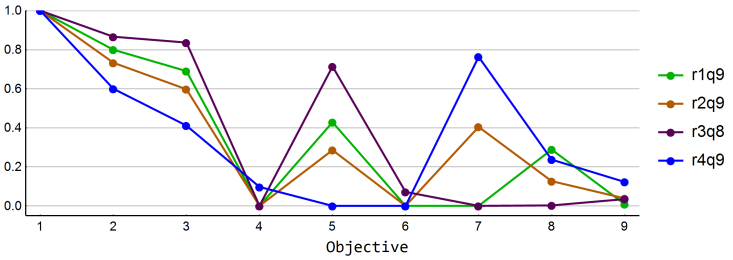


(e) The solution r4q4

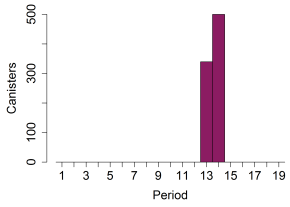
Figure 5.1: The 4 solutions where disposal starts early

r1q1 than in the solution r3q2. Similar observation can be made for the objectives related to the ending time of the encapsulation (obj. 5) and the operation time of the encapsulation facility (obj. 6). However, the dimensions of the disposal facility (obj. 7 and 8) are smaller in the solution r3q2 than in the solution r1q1.

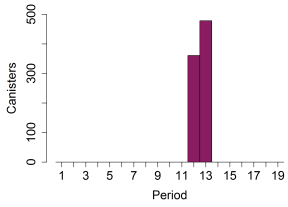
As mentioned, the result of the problem (5.1) is a schedule in terms of the number of canisters which are encapsulated at each period. This is illustrated in Figures 5.1b–5.1e and 5.2b–5.2e. We see that the schedules in Figure 5.1 have similar shape between each other as well as the schedules in Figure 5.2. In Figure 5.1, the starting and the ending times together with the total number of canisters vary, but all the schedules suggest to begin with a smaller capacity for the several periods and only at the last period raise the capacity. The opposite is recommended in the schedules in Figure 5.2, since here all the assemblies are encapsulated within two periods such that the second period is encapsulated with full capacity and in the first period the rest is encapsulated.



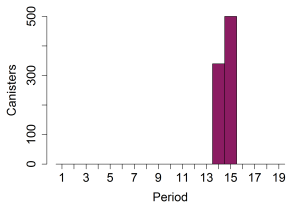
(a) The objective values for solutions



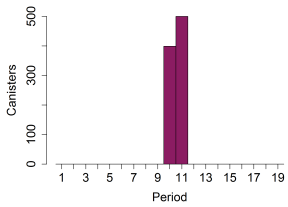
(b) The solution r1q9



(c) The solution r2q9



(d) The solution r3q8



(e) The solution r4q9

Figure 5.2: The 4 solutions where disposal starts late

From the learning phase, we can say that the solutions adopt the reference points quite well in general. The solutions can be divided into two groups by the starting time of the disposal. Compared with the late start, the early start seems to improve the objectives 1–3 and impair others. As mentioned, the early start suggests to begin with a small capacity but this deteriorates the values of the objectives 1–3, 5, 6, and 9. Nevertheless, despite the objective 9, these objectives can be improved by allowing empty canister positions, but this, in its turn, yields increase in the objectives 4 and 7–9. On the other hand, the late start implies the high capacity. The objectives 7–9 can be improved by delaying the starting time, but then the objectives 2, 3, and 5 increase. The empty canister positions have only minor effect in the case of the late start. However, the objectives 2, 3, and 5 can be improved by increasing the number of empty canister positions which again yields impairing the objectives 4 and 7–9.

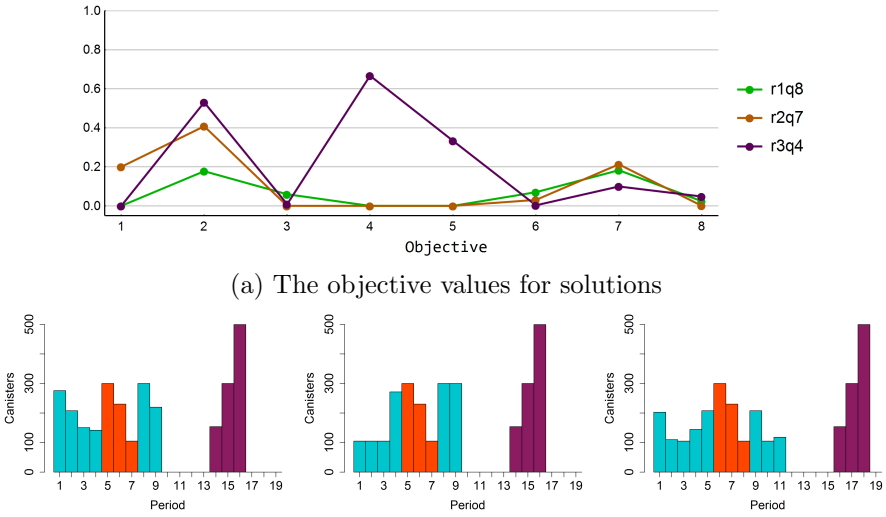
In the learning phase, the different trade-offs in the model were in-

vestigated with different reference points. During the solution process, the decision maker selects the current solutions $\boldsymbol{f}_h$ as following: the solution r1q9 for $\boldsymbol{f}_1$, the solution r2q1 for $\boldsymbol{f}_2$, the solution r3q8 for $\boldsymbol{f}_3$, and the solution r4q4 for $\boldsymbol{f}_4$. As was noticed, the solutions are distributed in two groups based on the starting time of the disposal. In the decision phase, the decision maker is willing to return the solutions obtained with the reference point 2 and consider these solutions as satisfying compromises regardless the starting time of the disposal. However, as the final solution $\boldsymbol{f}^*$ the decision maker selects the solution r2q9 due to its rather good values for other than the maximum storage capacity. Nevertheless in the learning phase, it was noticed that this is mandatory, if the cost and the disposal facility area are small. Furthermore, if we compare the solution r2q9 with the late start to the solution r2q1 with the early start, we see that both of these still end at the same disposal period.

5.3 Optimization Results with Three Fuel Types

While the case study given in the previous section has academic background, the more realistic case study presented in the original publication VIII and in this section is considered. The study is performed in association with Posiva Oy being responsible to the final disposal of the spent fuel assemblies irradiated in the reactors which are currently operating and under the construction in Finland. In total these five reactors use three different type of fuels, namely OL1-2, LO1-2, and OL3 while the case study in the previous section involved only the disposal of OL3. Since the final disposal of these three fuel types takes place for a long time interval it is reasonable to consider the possibility to include a hiatus in the disposal process. Here we include the hiatus in the base case and fix the order of the fuel types as the following: OL1-2, LO1-2, OL1-2, hiatus, OL3.

The solutions presented are obtained by using MITSPA but in this case, the solver used in GAMS [27] is SCIP [1, 2, 155] being a global branch-and-bound type solver. We have performed three iterations of MITSPA and in this case, the number of objectives is eight meaning that eight solutions are obtained at each iteration. From these eight solutions, three solutions are presented to the decision maker. The solutions selected by the decision maker at these iterations are given in Figure 5.3 where Figure 5.3a illustrates the objective values of these solutions on the interval from 0 to 1 such that 0 represents the ideal value of each objective and 1 is the nadir value obtained by the payoff table. Figures 5.3b–5.3d give the schedules such that blue refers to OL1-2, orange to LO1-2, and purple to OL3.



(a) The objective values for solutions

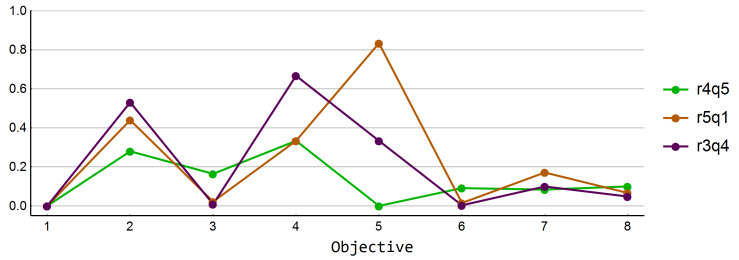
(b) The solution r1q8 (c) The solution r2q7 (d) The solution r3q4

Figure 5.3: The solutions selected at each iteration

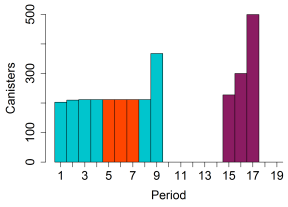
At first, the decision maker selects the ideal vector as the first reference point. From the solutions obtained, the decision maker selects the solution r1q8 since it does not involve any additional water pools (obj. 1) and it is a good compromise with respect to the other objectives. However, for the second iteration, the decision maker wish to adopt the solution r1q8 by setting the number of canisters (obj. 3) to the minimum and postponing the ending time (obj. 4) and lengthening the operation time (obj. 5).

In the second iteration, the selected solution r2q7 needs less canisters (obj. 3) than the previous one and the schedule in Figure 5.3c is more feasible from the practical point of view with the calm beginning than the other solutions. For the third reference point, the solution r2q7 is adopted by selecting the minimum central and the disposal tunnel (obj. 6 and 7) lengths and increasing the storage, the ending, and the operation time (obj. 2, 4, and 5). The third and the final solution is r3q4 due to the shorter disposal and central tunnels (obj. 6 and 7) and no additional water pools (obj. 1) are needed even if the total cost (obj. 8) is a bit higher. In general, it seems that the length of the central tunnel (obj. 7) and the total cost (obj. 8) are conflicting objectives.

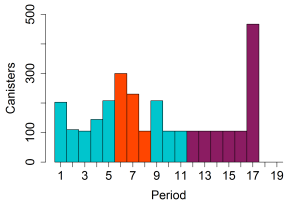
The model in the original publication VIII allows the use of the two-shift work increasing the maximum number of canisters encapsulated at one period but increasing also the cost. From the schedules in Figures 5.3b–5.3d we see that the two-shift work is utilized only when OL3 is



(a) The objective values for solutions



(b) The solution r4q5



(c) The solution r5q1

Figure 5.4: The solutions with increasing production and without hiatus

encapsulated. Throughout the interactive process, it is also noticed that when the fuel type is changed, the production tends to begin with the higher capacity in the first period and after that it decreases. However, from the practical point of view the behaviour of this kind is not beneficial. Thus, we investigate the effect of the increasing production in the terms of the objective function values and schedules. This is done by adding a constraint forcing that at every period at least as many canisters need to be encapsulated than the previous period and using the solution r3q4 as the reference point.

The results of the solution r4q5 involving the increasing production are given in Figure 5.4a and Figure 5.4b. As we see, the average storage time (obj. 2) decreases as well as the operation time of the encapsulation facility (obj. 5) and the ending date of the encapsulation (obj. 4). Nevertheless, the number of canisters (obj. 3) increases together with the length of the disposal tunnel (obj 6) and the cost (obj. 8). The obtained schedule in Figure 5.4b looks much feasible and quite even from the practical point of view compared with the solution r3q4 in Figure 5.3d. Additionally, the disposal ends one period earlier and the hiatus is one period longer.

The results presented in the original publication VIII have all the hiatus from 15 to 25 years even thought only 5 year hiatus is forced by the model. This indicates that the model gains advantage from the hiatus. Unfortunately, the hiatus has its own risks. Thus, we have also

studied how the objective function values and the schedule change if the hiatus is not allowed. The disposal schedule in Figure 5.4c is similar to the schedule in Figure 5.3d for the first nine periods. After that, the disposal continues with the minimum pace until the last period the two-shift work is employed to complete the disposal. At the end, the disposal ends one period earlier than in the solution r3q4 with the hiatus. Besides the changes in the operation time and ending time (obj. 5 and 4), the changes in the objective function values focuses on the decrease in the storage time (obj. 2) and the increase in the central tunnel length (obj. 7) and the minor increase in the cost (obj. 8).

In the case of three spent nuclear fuel types the similar division of solutions depending the value of the parameter q is not observed than in the case of one fuel type. However, the cases when q is larger, the CPU time seems to be smaller than in the case of the small values of q. The similar observation is made also in the test examples of the original publication VI. The average CPU time in the original publications VII and VIII are on the same magnitude but the median CPU time in the latter case is 13-fold compared with the first case.

In conclusion, the solutions obtained for the problem (5.1) with MITSPA are realistic in the both of the cases of one and three fuel types. Due to the interaction, the decision maker is able to select more satisfactory solution than the first reference point suggested. Additionally, the decision maker is not forced to stick with the solutions from the last reference point but is able to return to some previous solution. It is worth pointing out that depending on the case, the model (5.1) involves eight or nine objectives which is a quite large number of objectives not only from the computational point of view but also from the human comprehension. Therefore, selecting only some of these objectives may aid the decision maker's task.

Chapter 6

Conclusions

Nonsmooth multiobjective optimization is an intriguing and challenging research field with the vast range of practical applications. Nevertheless, smooth multiobjective optimization and nonsmooth single-objective optimization have gained much more attention separately. In this dissertation, nonsmooth multiobjective optimization is discussed from different perspectives. At first, a short introductory to the field is given with the necessary and relevant theory of nonsmooth and multiobjective optimization. After that, several different methods from different categories depending on the role of the decision maker are presented. Finally, two case studies of the practical application are solved with one of these methods.

The first methods are discussed in Chapter 3, where three different descent multiobjective methods are described. Besides the descent property, all of these methods utilize the proximal bundle approach and benefit from the improvement function either using it directly or indirectly. Moreover, by treating the objectives as they are, the descent methods do not need any preference information, and thus, they consider objectives equally important. These methods have potential applications in methods where some (weak) Pareto optimal solution counterpart to a nonoptimal solution needs to be found, for example, in interactive methods or with some heuristic. Another possibility is to solve the problem with several starting points in order to produce a selection of the solutions from the Pareto set.

The first descent method under the scope is the multiobjective proximal bundle method (MPB) with the scaled improvement function. In brief, the idea is to use the single-objective proximal bundle method with the scaled improvement function as its objective. With the scaling, the aim is to stabilize the numerical behaviour of MPB. The method is

designed for the general nonconvex nonsmooth multiobjective problems with constraints. It is proved to be globally convergent to the weak Pareto stationary solution, and under some generalized convexity assumptions, the global weak Pareto optimal solution can be ensured. In practice, the implementation of MPB is turned out to be efficient and reliable both in the multiobjective and the single-objective case. From the theoretical point of view, it is still an open question whether the assumptions caused by the scaling are minimal for the convergence or can they be reduced by studying the relations between the assumptions. From the computational point of view, the effects of the scaling need more research.

The second descent method is the multiple subgradient descent bundle method (MSGDB) generalizing the classical steepest descent method for the multiobjective nonsmooth convex unconstrained case. The core idea is to calculate individual descent directions for objectives with the proximal bundle approach and then calculate the candidate for the common descent direction. As a result, MSGDB gives a global weak Pareto optimal solution. Even if the fascination of the generalization of the steepest descent method is rather theoretical, MSGDB have some computational benefits. First, the individual descent directions can be calculated parallel, and second, the size of the subproblems is smaller, or in some extreme cases equal compared with the MPB. Obviously, the nonconvex version of MSGDB handling the constraints would be great to develop. Furthermore, the structure of MSGDB has potential for adding some interaction and taking advantage of the individual search directions. For example, if a candidate for the common descent direction is not descent to all the objectives, it could be possible that the decision maker selects which objective will be improved and the individual descent direction would be used instead of performing a null step or to scale directions based on the decision maker's preferences.

The third descent method is the multiobjective double bundle method for DC optimization (MDBDC). As its name suggests, the method is designed to solve constrained nonsmooth multiobjective problems whose objective and constraint functions can be represented as DC functions. Similar to MPB, the idea is to combine the improvement function with the single-objective bundle method, which in this case is namely the double bundle method for DC optimization. This method enjoys the fact that the solutions are weak Pareto stationary as usually obtained with general nonconvex methods instead of the milder Pareto critical solutions being more common in the case of DC optimization. In practice, MDBDC performs well and compared with the solutions obtained with MPB, MDBDC produces better solutions in terms of

objective function values in around 30% in the numerical test in the original publication IV. This confirms that it is beneficial to employ the convex and concave behaviour of the function. Furthermore, compared with MPB, MDBDC performed better with the larger problems. In future, some subgradient aggregation strategy could be applied with MDBDC and see its strength by applying the method for some practical application.

Chapter 4 is devoted to achievement scalarizing functions (ASFs). Contrary to the methods in Chapter 3, the core idea in ASFs is to aggregate multiple objectives into a single-objective problem such that by solving the resulting scalarized problem, the optimum is (weakly) Pareto optimal solution for the original multiobjective problem and it is possible to obtain all the Pareto optimal solutions by selecting a suitable reference point. Indeed, the solution of the scalarized problem is the reference point projected to the Pareto set as close as possible with respect to some metric.

The focus in Chapter 4 is on the family of the two-slope parameterized ASFs. The special features of this family are that first the reference point can be either unachievable or achievable and always the suitable weighting coefficient is used omitting the urge for the test of the achievability of the reference point. Secondly, with the parameterization the different metrics are used, and thus, we obtain more than one (weakly) Pareto optimal solution for one reference point. This aims to be able to systematically produce Pareto optimal solutions giving a rough but yet reasonably distributed approximation of the Pareto set to be presented for the decision maker. As ASFs are well-suited for adaptation to the interactive framework, the interactive method is also described. This method suits well for the practical problems and more practical applications would be fascinating to solve in future. Furthermore, these practical applications give more information about the role of the parameterization and whether some metric turns out to be more preferred in some cases than others by the decision maker.

Chapter 5 is dedicated to investigating the case studies of scheduling the final disposal of the spent nuclear fuel in Finland. Two cases are considered: one involving only one fuel type and one involving three fuel types. The problem is modelled as the multiobjective nonsmooth MINLP problem with nine and eight objectives, respectively. The deliciousness of these problems springs from the complexity of the problems which is due to several objectives, binary variables, and nonlinear and nonsmooth constraints. This problem is then solved by using the interactive method utilizing the two-slope parameterized ASFs (MITSPA). Thanks to the interactivity, the decision maker is able to learn about

the problem during the solution process. Therefore, the decision maker is capable of making better decisions. The further studies of this application in the real life situation include the increase in the scale by shortening the length of the disposal period and considering different scenarios, for example, by changing the order of the fuel types.

To conclude, the multiobjective optimization in different forms is discussed in the presence of the nonsmoothness. The topic is covered from various angles: theoretically, methodologically and with the practical application. The individual contributions of this dissertation intertwine together in Section 4.3. Indeed, the descent methods can be used to produce a neutral Pareto optimal solution for the decision maker to aid the selection of the good reference point. Then the two-slope parameterized ASF is used to obtain a selection of Pareto optimal solutions based on the decision makers preferences. Furthermore, this method is used to solve the practical application in Chapter 5. Besides the continuation of the method development, wider adopting of these methods in practical applications would be fascinating for the further research.

Bibliography

[1] T. Achterberg. SCIP: Solving constraint integer programs. *Mathematical Programming Computation*, 1(1):1–41, 2009.

[2] T. Achterberg, T. Berthold, T. Koch, and K. Wolter. Constraint Integer Programming: A New Approach to Integrate CP and MIP. In L. Perron and M. A. Trick, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 6–20. Springer, Berlin, Heidelberg, 2008.

[3] F. J. Aragón, M. A. Goberna, M. A. López, and M. M. L. Rodríguez. *Nonlinear Optimization*. Springer, Cham, 2019.

[4] L. Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16(1):1–3, 1966.

[5] A. Astorino, A. Fuduli, and E. Gorgone. Non-smoothness in classification problems. *Optimizaton Methods and Software*, 23(5):675–688, 2008.

[6] A. Astorino and G. Miglionico. Optimizing sensor cover energy via DC programming. *Optimization Letters*, 10(2):355–368, 2016.

[7] A. Bagirov, N. Karmitsa, and M. M. Mäkelä. *Introduction to Nonsmooth Optimization: Theory, Practice and Software*. Springer, Cham Heidelberg, 2014.

[8] A. Bagirov and J. Yearwood. A new nonsmooth optimization algorithm for minimum sum-of-squares clustering problems. *European Journal of Operational Research*, 170(2):578–596, 2006.

[9] A. M. Bagirov, M. Gaudioso, N. Karmitsa, M. M. Mäkelä, and S. Taheri. *Numerical Nonsmooth Optimization: State of the Art Algorithms*. Springer, 2020.

[10] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, Inc., New Jersey, third edition, 2006.

[11] J. Y. Bello Cruz and G. Bouza Allende. A steepest descent-like method for variable order vector optimization problems. *Journal of Optimization Theory and Applications*, 162(2):371–391, 2014.

[12] R. Benayoun, J. de Montgolfier, J. Tergny, and O. Laritchev. Linear programming with multiple objective functions: Step method (stem). *Mathematical Programming*, 1(1):366–375, 1971.

[13] G. C. Bento and J. X. Cruz Neto. A subgradient method for multiobjective optimization on Riemannian manifolds. *Journal of Optimization Theory and Applications*, 159(1):125–137, 2013.

[14] N. Boland, H. Charkhgard, and M. Savelsbergh. A new method for optimizing a linear function over the efficient set of a multiobjective integer program. *European Journal of Operational Research*, 260(3):904 – 919, 2017.

[15] H. Bonnel, A. N. Iusem, and B. F. Svaiter. Proximal methods in vector optimization. *SIAM Journal on Optimization*, 15(4):953–970, 2005.

[16] P. A. N. Bosman and E. D. de Jong. Exploiting Gradient Information in Numerical Multi–objective Evolutionary Optimization. In *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, GECCO '05, pages 755–762, 2005.

[17] A. S. Brito, J. X. Cruz Neto, P. S. M. Santos, and S. S. Souza. A relaxed projection method for solving multiobjective optimization problems. *European Journal of Operational Research*, 256(1):17–23, 2017.

[18] J. Buchanan and L. Gardiner. A comparison of two reference point methods in multiple objective mathematical programming. *European Journal of Operational Research*, 149(1):17–34, 2003.

[19] J. T. Buchanan. A naïive approach for solving MCDM problems: The GUESS method. *The Journal of the Operational Research Society*, 48(2):202–206, 1997.

[20] G. A. Carrizo, P. A. Lotito, and M. C. Maciel. Trust region globalization strategy for the nonconvex unconstrained multiobjective

optimization problem. *Mathematical Programming*, 159(1):339–369, 2016.

[21] E. Carrizosa, V. Guerrero, and D. Romero Morales. Visualizing data as objects by DC (difference of convex) optimization. *Mathematical Programming*, 169(1):119–140, 2018.

[22] A. Cauchy. Methodes generales pour la resolution des systémes dequations simultanees. *Compte rendus de l'Académie des sciences*, (25):536–538, 1847.

[23] A. Charnes and W. W. Cooper. Goal programming and multiple objective optimizations: Part 1. *European Journal of Operational Research*, 1(1):39–54, 1977.

[24] A. Charnes, W. W. Cooper, and R. O. Ferguson. Optimal estimation of executive compensation by linear programming. *Management Science*, 1(2):138–151, 1955.

[25] E. W. Cheney and A. A. Goldstein. Newton's method for convex programming and Tchebycheff approximation. *Numerische Mathematik*, 1:253–268, 1959.

[26] F. H. Clarke. *Optimization and Nonsmooth Analysis*. John Wiley & Sons, Inc., New York, 1983.

[27] GAMS Development Corporation. General Algebraic Modeling System (GAMS) Release 26.1.0. Washington, DC, USA, 2019.

[28] D. Craft, T. Halabi, H. A. Shih, and T. Bortfeld. An approach for practical multiobjective IMRT treatment planning. *International Journal of Radiation Oncology · Biology · Physics*, 69(5):1600–1607, 2007.

[29] J. Y. Bello Cruz. A subgradient method for vector optimization problems. *SIAM Journal on Optimization*, 23(4):2169–2182, 2013.

[30] J. Y. Bello Cruz and A. N. Iusem. A strongly convergent method for nonsmooth convex minimization in Hilbert spaces. *Numerical Functional Analysis and Optimization*, 32(10):1009–1018, 2011.

[31] J. X. Da Cruz Neto, G. J. P. Da Silva, O. P. Ferreira, and J. O. Lopes. A subgradient method for multiobjective optimization. *Computational Optimization and Applications*, 54(3):461–472, 2013.

[32] W. de Oliveira. Proximal bundle methods for nonsmooth DC programming. *Journal of Global Optimization*, 75(2):523–563, 2019.

[33] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Chichester, 2001.

[34] K. Deb and K. Miettinen. Nadir Point Estimation Using Evolutionary Approaches: Better Accuracy and Computational Speed Through Focused Search. In M. Ehrgott, B. Naujoks, T. J. Stewart, and J. Wallenius, editors, *Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems*, pages 339–354. Springer, Berlin, 2010.

[35] J.-A. Désidéri. Multiple-Gradient Descent Algorithm (MGDA). Technical Report 6953, INRIA Research Report, 2009.

[36] J.-A. Désidéri. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *Compte rendus de l'Académie des sciences, Ser. I*, 350:313–318, 2012.

[37] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.

[38] Y. Jin (ed.). *Multi-Objective Machine Learning*. Studies in Computational Intelligence, 16. Springer-Verlag, Berlin, Heidelberg, 2006.

[39] F. Y. Edgeworth. Mathematical psychics. *Mind*, 6(24):581–583, 1881.

[40] M. Ehrgott. *Multicriteria Optimization*. Springer, Berlin, second edition, 2005.

[41] M. Ehrgott. A discussion of scalarization techniques for multiple objective integer programming. *Annals of Operations Research*, 147(1):343–360, 2006.

[42] M. Ehrgott and D. Tenfelde-Podehl. Computation of ideal and nadir values and implications for their use in MCDM methods. *European Journal of Operational Research*, 151(1):119–139, 2003.

[43] G. W. Evans. An overview of techniques for solving multiobjective mathematical programs. *Management Science*, 30(11):1268–1282, 1984.

[44] P. C. Fishburn. Lexicographic orders, utilities and decision rules: A survey. *Management Science*, 20(11):1442–1471, 1974.

[45] J. Fliege, L. M. Graña Drummond, and B. F. Svaiter. Newton's method for multiobjective optimization. *SIAM Journal on Optimization*, 20(2):602–626, 2009.

[46] J. Fliege and B. F. Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000.

[47] A. Fuduli, M. Gaudioso, and G. Giallombardo. A DC piecewise affine model and a bundling technique in nonconvex nonsmooth minimization. *Optimization Methods and Software*, 19(1):89–102, 2004.

[48] A. Fuduli, M. Gaudioso, and G. Giallombardo. Minimizing nonconvex nonsmooth functions via cutting planes and proximity control. *SIAM Journal on Optimization*, 14(3):743–756, 2004.

[49] E. H. Fukuda and L.M. Graña Drummond. Inexact projected gradient method for vector optimization. *Computational Optimization and Applications*, 54(3):473–493, 2013.

[50] E. H. Fukuda and L.M. Graña Drummond. A survey on multiobjective descent methods. *Pesquisa Operacional*, 34(3):585–620, 2014.

[51] N. Gadhi and A. Metrane. Sufficient optimality condition for vector optimization problems under D.C. data. *Journal of Global Optimization*, 28(1):55–66, 2004.

[52] S. Gass and T. Saaty. The computational algorithm for the parametric objective function. *Naval Research Logistics Quarterly*, 2(1-2):39–45, 1955.

[53] M. Gaudioso, G. Giallombardo, G. Miglionico, and A. Bagirov. Minimizing nonsmooth DC functions via successive DC piecewise-affine approximations. *Journal of Global Optimization*, 71(1):37–55, 2018.

[54] M. Gaudioso, T. V. Gruzdeva, and A. S. Strekalovsky. On numerical solving the spherical separability problem. *Journal of Global Optimization*, 66(1):21–34, 2016.

[55] L. M. Graña Drummond and B. F. Svaiter. A steepest descent method for vector optimization. *Journal of Computational and Applied Mathematics*, 175:395–414, 2005.

[56] L.M. Graña Drummond and A.N. Iusem. A projected gradient method for vector optimization problems. *Computational Optimization and Applications*, 28(1):5–29, 2004.

[57] W. J. Gutjahr and P. C. Nolz. Multicriteria optimization in humanitarian aid. *European Journal of Operational Research*, 252(2):351–366, 2016.

[58] M. Haarala, K. Miettinen, and M. M. Mäkelä. New limited memory bundle method for large-scale nonsmooth optimization. *Optimization Methods and Software*, 19(6):673–692, 2004.

[59] Y. Haimes, L. Lasdon, and D. Wismer. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-1(3):296–297, 1971.

[60] J. Handl, D. B. Kell, and J. Knowles. Multiobjective optimization in bioinformatics and computational biology. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4(2):279–292, 2007.

[61] K. Harada, J. Sakuma, and S. Kobayashi. Local Search for Multiobjective Function Optimization: Pareto Descent Method. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2006)*, pages 659–666, 2006.

[62] W. Hare and C. Sagastizábal. A redistributed proximal bundle method for nonconvex optimization. *SIAM Journal on Optimization*, 20(5):2442–2473, 2010.

[63] P. Hartman. On functions representable as a difference of convex functions. *Pacific Journal of Mathematics*, 9(3):707–713, 1959.

[64] J. Haslinger and P. Neittaanmäki. *Finite Element Approximation for Optimal Shape, Material and Topology Design.* J. Wiley & Sons, Chichester, 1996.

[65] J.-B. Hiriart-Urruty. Generalized differentiability, duality and optimization for problems dealing with differences of convex functions. *Lecture Note in Economics and Mathematical Systems*, 256:37–70, 1985.

[66] J. B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms, I and II.* Springer Verlag, Berlin, 1993.

[67] K. Holmberg and H. Tuy. A production-transportation problem with stochastic demand and concave production costs. *Mathematical Programming*, 85(1):157–179, 1999.

[68] R. Horst and N. V. Thoai. DC programming: Overview. *Journal of Optimization Theory and Applications*, 103(1):1–43, 1999.

[69] A. Hutterer and J. Jahn. On the location of antennas for treatment planning in hyperthermia. *OR Spectrum*, 25(3):397–412, 2003.

[70] C.-L. Hwang and A. S. M. Masud. *Multiple Objective Decision Making - Methods and Applications: State-of-the-Art Survey*. Lecture Notes in Economics and Mathematical Systems 164. Springer, Berlin, 1979.

[71] H. Isermann and R. E. Steuer. Computational experience concerning payoff tables and minimum criterion values over the efficient set. *European Journal of Operational Research*, 33(1):91–97, 1988.

[72] A. Jaszkiewicz and R. Słowiński. The 'Light Beam Search' approach – an overview of methodology applications. *European Journal of Operational Research*, 113(2):300–314, 1999.

[73] Y. Ji, M. Goh, and R. de Souza. Proximal point algorithms for multi-criteria optimization with the difference of convex objective functions. *Journal of Optimization Theory and Applications*, 169(1):280–289, 2016.

[74] Y. Ji and S. Qu. Proximal point algorithms for vector DC programming with applications to probabilistic lot sizing with service levels. *Discrete Dynamics in Nature and Society*, 2017. Article ID 5675183.

[75] B. Johnson, A. Newman, and J. King. Optimizing high-level nuclear waste disposal within a deep geologic repository. *Annals of Operations Research*, 253(2):733–755, 2017.

[76] K. Joki, A. Bagirov, N. Karmitsa, and M. M. Mäkelä. A proximal bundle method for nonsmooth DC optimization utilizing nonconvex cutting planes. *Journal of Global Optimization*, 68(3):501–535, 2017.

[77] K. Joki, A. Bagirov, N. Karmitsa, M. Mäkelä, and S. Taheri. Double bundle method for finding Clarke stationary points in nonsmooth DC programming. *SIAM Journal on Optimization*, 28(2):1892–1919, 2018.

[78] E. Karas, A. Ribeiro, C. Sagastizábal, and M. Solodov. A bundle-filter method for nonsmooth convex constrained optimization. *Mathematical Programming*, 116(1):297–320, 2009.

[79] N. Karmitsa, M. Gaudioso, and K. Joki. Diagonal bundle method with convex and concave updates for large-scale nonconvex and nonsmooth optimization. *Optimizaton Methods and Software*, 34(2):363–382, 2019.

[80] J. E. Kelley. The cutting plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.

[81] M. R. Kilinc and N. V. Sahinidis. Exploiting integrality in the global optimization of mixed-integer nonlinear programming problems with BARON. *Optimization Methods and Software*, 33(3):540–562, 2018.

[82] K. C. Kiwiel. An aggregate subgradient method for nonsmooth convex minimization. *Mathematical Programming*, 27(3):320–341, 1983.

[83] K. C. Kiwiel. A descent method for nonsmooth convex multiobjective minimization. *Large Scale Systems*, 8(2):119–129, 1985.

[84] K. C. Kiwiel. *Methods of Descent for Nondifferentiable Optimization*. Springer, Berlin, 1985.

[85] K. C. Kiwiel. Proximity control in bundle methods for convex nondifferentiable optimization. *Mathematical Programming*, 46:105–122, 1990.

[86] K. C. Kiwiel. Restricted step and Levenberg–Marquardt techniques in proximal bundle methods for nonconvex nondifferentiable optimization. *SIAM Journal on Optimization*, 6(1):227–249, 1996.

[87] M. Köksalan and B. Lokman. Finding nadir points in multi-objective integer programs. *Journal of Global Optimization*, 62(1):55–77, 2015.

[88] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, Berkeley, California, 1951. University of California Press.

[89] J. Kuopanportti and T. Lahtinen. Applied study on optimizing the final disposal of Loviisa NPP spent fuel assemblies. *Kerntechnik*, 84(4):334–339, 2019.

[90] H. A. Le Thi and T. Pham Dinh. Solving a class of linearly constrained indefinite quadratic problems by D.C. algorithms. *Journal of Global Optimization*, 11(3):253–285, 1997.

[91] H. A. Le Thi and T. Pham Dinh. The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems. *Annals of Operations Research*, 133(1):23–46, 2005.

[92] A. R. Leach. *Molecular Modelling: Principles and Applications*. Pearson Education Limited, Harlow, England, second edition, 2001.

[93] C. Lemaréchal. Combining Kelley's and Conjugate Gradient Methods. In *Abstracts of IX International Symposium on Mathematical Programming*. Budapest, Hungary, 1976.

[94] C. Lemaréchal. Nonsmooth Optimization and Descent Methods. Technical report, IIASA report, Laxemburg, Austria, 1978.

[95] C. Lemaréchal. Nondifferentiable Optimization. In G. G. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd, editors, *Optimization*, pages 529–572. Elsevier, Amsterdam, 1989.

[96] C. Lemaréchal, A. Nemirovskii, and Yu. Nesterov. New variants of bundle methods. *Mathematical Programming*, 69:111–147, 1995.

[97] C. Lemaréchal, J. S. Strodiot, and A. Bihain. On a Bundle Algorithm for Nonsmooth Optimization. In O. L. Mangasarian, G. L. Meyer, and S. M. Robinson, editors, *Nonlinear Programming, Computational Methods in Applied Sciences vol. 4*, pages 245–282. Academic Press, New York, 1981.

[98] T. M. Leschine, H. Wallenius, and W. A. Verdini. Interactive multiobjective analysis and assimilative capacity-based ocean disposal decisions. *European Journal of Operational Research*, 56(2):278 – 289, 1992.

[99] W. Li, C. Nahak, and I. Singer. Constraint qualifications for semi-infinite systems of convex inequalities. *SIAM Journal on Optimization*, 11(1):31–52, 2000.

[100] L. Lukšan. Dual method for solving a special problem of quadratic programming as a subproblem at linearly constrained nonlinear minimax approximation. *Kybernetika*, 20:445–457, 1984.

[101] L. Lukšan and J. Vlček. A bundle-Newton method for nonsmooth unconstrained minimization. *Mathematical Programming*, 83:373–391, 1998.

[102] L. Lukšan and J. Vlček. Globally convergent variable metric method for convex nonsmooth unconstrained minimization. *Journal of Optimization Theory and Applications*, 102:593–613, 1999.

[103] L. Lukšan and J. Vlček. Introduction to Nonsmooth Analysis. Theory and Algorithms. Technical Report DMSIA 1/2000, University of Bergamo, 2000.

[104] M. Luque, L. A. Lopez-Agudo, and O. D. Marcenaro-Gutierrez. Equivalent reference points in multiobjective programming. *Expert Systems with Applications*, 42:2205–2212, 2015.

[105] M. Luque, K. Miettinen, A. B. Ruiz, and F. Ruiz. A two-slope achievement scalarizing function for interactive multiobjective optimization. *Computers & Operations Research*, 39:1673–1681, 2012.

[106] M. M. Mäkelä. Survey of bundle methods for nonsmooth optimization. *Optimization Methods and Software*, 17(1):1–29, 2002.

[107] M. M. Mäkelä. Multiobjective Proximal Bundle Method for Nonconvex Nonsmooth Optimization: Fortran Subroutine MPBNGC 2.0. Technical Report B 13/2003, Reports of the Department of Mathematical Information Technology, Series B, Scientific computing, University of Jyväskylä, Jyväskylä, 2003.

[108] M. M. Mäkelä, V.-P. Eronen, and N. Karmitsa. On Nonsmooth Multiobjective Optimality Conditions with Generalized Convexities. In Th. M. Rassias, C. A. Floudas, and S. Butenko, editors, *Optimization in Science and Engineering*, pages 333–357. Springer, 2014.

[109] M. M. Mäkelä, N. Karmitsa, and O. Wilppu. Multiobjective Proximal Bundle Method for Nonsmooth Optimization. Technical Report 1120, TUCS Technical Reports, Turku Centre for Computer Science, Turku, 2014.

[110] M. M. Mäkelä and P. Neittaanmäki. *Nonsmooth Optimization: Analysis and Algorithms with Applications to Optimal Control*. World Scientific Publishing Co., Singapore, 1992.

[111] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, 1999.

[112] K. Miettinen, J. Hakanen, and D. Podkopaev. Interactive Nonlinear Multiobjective Optimization Methods. In S. Greco, M. Ehrgott, and J. R. Figueira, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, pages 927–976. Springer New York, New York, NY, 2016.

[113] K. Miettinen and M. M. Mäkelä. Interactive bundle-based method for nondifferentiable multiobjective optimization: NIMBUS. *Optimization*, 34:231–246, 1995.

[114] K. Miettinen and M. M. Mäkelä. On scalarizing functions in multiobjective optimization. *OR Spectrum*, 24:193–213, 2002.

[115] K. Miettinen and M. M. Mäkelä. Synchronous approach in interactive multiobjective optimization. *European Journal of Operation Research*, 170:909–922, 2006.

[116] K. Miettinen, M. M. Mäkelä, and K. Kaario. Experiments with classification-based scalarizing functions in interactive multiobjective optimization. *European Journal of Operational Research*, 175:931–947, 2006.

[117] R. Mifflin. An algorithm for constrained optimization with semismooth functions. *Mathematics of Operations Research*, 2(2):191–207, 1977.

[118] E. S. Mistakidis and G. E. Stavroulakis. *Nonconvex Optimization in Mechanics. Smooth and Nonsmooth Algorithms, Heuristics and Engineering Applications by the F.E.M.* Kluwer Academic Publisher, Dordrecht, 1998.

[119] J. J. Moreau, P. D. Panagiotopoulos, and G. Strang (Eds.). *Topics in Nonsmooth Mechanics*. Birkhäuser, Basel, 1988.

[120] H. Mukai. Algorithms for multicriterion optimization. *IEEE Transactions on Automatic Control AC-25*, pages 177–186, 1979.

[121] H. Nakayama and Y. Sawaragi. Satisficing Trade-off Method for Multiobjective Programming. In M. Grauer and A. P. Wierzbicki,

editors, *Interactive Decision Analysis*, pages 113–122, Berlin, Heidelberg, 1984. Springer Berlin Heidelberg.

[122] Y. Nikulin, K. Miettinen, and M. M. Mäkelä. A new achievement scalarizing function based on parameterization in multiobjective optimization. *OR Spectrum*, 34:69–87, 2012.

[123] J. Outrata, M. Koĉvara, and J. Zowe. *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints. Theory, Applications and Numerical Results.* Kluwer Academic Publishers, Dordrecht, 1998.

[124] V. Pareto. *Manual d'economie politique.* Giard, Paris, 1909.

[125] T. Pham Dinh and H. A. Le Thi. Convex analysis approach to DC programming: Theory, algorithms and applications. *Acta Mathematica Vietnamica*, 22(1):289–355, 1997.

[126] R. Pini and C. Singh. A survey of recent [1985–1995] advances in generalized convexity with applications to duality theory and optimality conditions. *Optimization*, 39(4):311–360, 1997.

[127] O. Pironneau and E. Polak. Rate of convergence of a class of methods of feasible directions. *SIAM Journal on Numerical Analysis*, 10(1):161–174, 1973.

[128] F. Poirion, Q. Mercier, and J.-A. Désidéri. Descent algorithm for nonsmooth stochastic multiobjective optimization. *Computational Optimization and Applications*, 68(2):317–331, 2017.

[129] Ž. Povalej. Quasi-Newton's method for multiobjective optimization. *Journal of Computational and Applied Mathematics*, 255:765–777, 2014.

[130] S. Qu, M. Goh, Y. Ji, and R. De Souza. A new algorithm for linearly constrained c-convex vector optimization with a supply chain network risk application. *European Journal of Operational Research*, 247(2):359–365, 2015.

[131] S. Qu, M. Goh, and B. Liang. Trust region methods for solving multiobjective optimisation. *Optimization Methods and Software*, 28(4):796–811, 2013.

[132] S. Qu, M. Goh, S. Y. Wu, and R. De Souza. Multiobjective DC programs with infinite convex constraints. *Journal of Global Optimization*, 59(1):41–58, 2014.

[133] S. Qu, C Liu, M. Goh, Y. Li, and Y. Ji. Nonsmooth multiobjective programming with quasi-Newton methods. *European Journal of Operational Research*, 235(3):503–510, 2014.

[134] G. P. Rangaiah. *Multi-Objective Optimization: Techniques and Applications in Chemical Engineering*. Advances in process systems engineering. World Scientific, Singapore, 2009.

[135] T. Ranta. *Optimization in the Final Disposal of Spent Nuclear Fuel*. PhD thesis, Tampere University of Technology, 2012.

[136] T. Ranta and F. Cameron. Heuristic methods for assigning spent nuclear fuel assemblies to canisters for final disposal. *Nuclear Science and Engineering*, 171(1):41–51, 2012.

[137] C. A. Rautman, R. A. Reid, and E. E. Ryder. Scheduling the disposal of nuclear waste material in a geologic repository using the transportation model. *Operations Research*, 41(3):459–469, 1993.

[138] G. R. Reeves and R. C. Reid. Minimum values over the efficient set in multiple objective decision making. *European Journal of Operational Research*, 36(3):334–338, 1988.

[139] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, New Jersey, 1970.

[140] F. Ruiz, M. Luque, F. Miguel, and M. del Mar Muñoz. An additive achievement scalarizing function for multiobjective programming problems. *European Journal of Operational Research*, 188(3):683–694, 2008.

[141] C. Sagastizábal and M. Solodov. An infeasible bundle method for nonsmooth convex constrained optimization without a penalty function or a filter. *SIAM Journal on Optimization*, 16(1):146–169, 2005.

[142] Y. Sawaragi, H. Nakayama, and T. Tanino. *Theory of Multiobjective Optimization*. Academic Press, Orlando, 1985.

[143] H. Schramm and J. Zowe. A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results. *SIAM Journal on Optimization*, 2(1):121–152, 1992.

[144] N. Z. Shor. *Minimization Methods for Non-differentiable Functions*. Springer-Verlag, Berlin, 1985.

[145] H. Späth. *Cluster Analysis Algorithms*. Ellis Horwood Limited, Chichester, second edition, 1980.

[146] R. E. Steuer. *Multiple Criteria Optimization: Theory, Computation, and Applications*. John Wiley & Sons, New York, 1986.

[147] W. Y. Sun, R. J. B. Sampaio, and M. A. B. Candido. Proximal point algorithm for minimization of DC functions. *Journal of Computational Mathematics*, 21(4):451–462, 1979.

[148] A. Taa. Optimality conditions for vector optimization problems of a difference of convex mappings. *Journal of Global Optimization*, 31(3):421–436, 2005.

[149] M. Tavana. A subjective assessment of alternative mission architectures for the human exploration of Mars at NASA using multicriteria decision making. *Computers & Operations Research*, 31(7):1147–1164, 2004.

[150] M. Tawarmalani and N. V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103(2):225–249, 2005.

[151] J. F. Toland. On subdifferential calculus and duality in nonconvex optimization. *Mémoires de la Société Mathématique de France*, 60:173–180, 1979.

[152] H. Tuy. *Convex Analysis and Global Optimization*. Kluwer Academic Publishers, Dordrescht, 1998.

[153] D. Vanderpooten. The interactive approach in MCDA: A technical framework and some basic conceptions. *Mathematical and Computer Modelling*, 12(10):1213–1220, 1989.

[154] D. A. G. Vieira, R. H. C. Takahashi, and R. R. Saldanha. Multicriteria optimization with a multiobjective golden section line search. *Mathematical Programming*, 131(1):131–161, 2012.

[155] S. Vigerske and A. Gleixner. SCIP: global optimization of mixed-integer nonlinear programs in a branch-and-cut framework. *Optimization Methods and Software*, 33(3):563–593, 2018.

[156] E. Vlassopoulos, B. Volmert, and A. Pautz. Logistics optimization code for spent fuel assembly loading into final disposal canisters. *Nuclear Engineering and Design*, 325:246–255, 2017.

[157] S. Wang. Algorithms for Multiobjective and Nonsmooth Optimization. In P. Kleinschmidt, F.J. Radermacher, W. Sweitzer, and H. Wildermann, editors, *Methods of Operations Research, 58*, pages 131–142. Athenaum Verlag, 1989.

[158] A. P. Wierzbicki. Basic properties of scalarizing functionals for multiobjective optimization. *Optimization*, 8:55–60, 1977.

[159] A. P. Wierzbicki. The Use of Reference Objectives in Multiobjective Optimization. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making Theory and Applications. MCDM Theory and Applications Proceedings. Lecture Notes in Economics and Mathematical Systems, vol 177*, pages 468–486. Springer, Berlin, 1980.

[160] A. P. Wierzbicki. A mathematical basis for satisficing decision making. *Mathematical Modelling*, 3(5):391–405, 1982.

[161] A. P. Wierzbicki. A Methodological Approach to Comparing Parametric Characterizations of Efficient Solutions. In G. Fandel, M. Grauer, A. Kurzhanski, and A. P. Wierzbicki, editors, *Large-scale Modelling and Interactive Decision Analysis. Lecture Notes in Economics and Mathematical Systems, vol 273*, pages 27–45. Springer, Berlin, 1986.

[162] A. P. Wierzbicki. On the completeness and constructiveness of parametric characterizations to vector optimization problems. *OR Spectrum*, 8:73–87, 1986.

[163] P. Wolfe. A method of conjugate subgradients for minimizing nondifferentiable functions. *Mathematical Programming Study*, 3:145–173, 1975.

[164] P. L. Yu. A class of solutions for group decision problems. *Management Science*, 19(8):936–946, 1973.

[165] G. Žerovnik, L. Snoj, and M. Ravnik. Optimization of spent nuclear fuel filling in canisters for deep repository. *Nuclear Science and Engineering*, 163(2):183–190, 2009.

[166] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, 2011.

[167] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput.*, 8(2):173–195, 2000.

Turku Centre for Computer Science
TUCS Dissertations

1. **Marjo Lipponen**, On Primitive Solutions of the Post Correspondence Problem
2. **Timo Käkölä**, Dual Information Systems in Hyperknowledge Organizations
3. **Ville Leppänen**, Studies on the Realization of PRAM
4. **Cunsheng Ding**, Cryptographic Counter Generators
5. **Sami Viitanen**, Some New Global Optimization Algorithms
6. **Tapio Salakoski**, Representative Classification of Protein Structures
7. **Thomas Långbacka**, An Interactive Environment Supporting the Development of Formally Correct Programs
8. **Thomas Finne**, A Decision Support System for Improving Information Security
9. **Valeria Mihalache**, Cooperation, Communication, Control. Investigations on Grammar Systems.
10. **Marina Waldén**, Formal Reasoning About Distributed Algorithms
11. **Tero Laihonen**, Estimates on the Covering Radius When the Dual Distance is Known
12. **Lucian Ilie**, Decision Problems on Orders of Words
13. **Jukkapekka Hekanaho**, An Evolutionary Approach to Concept Learning
14. **Jouni Järvinen**, Knowledge Representation and Rough Sets
15. **Tomi Pasanen**, In-Place Algorithms for Sorting Problems
16. **Mika Johnsson**, Operational and Tactical Level Optimization in Printed Circuit Board Assembly
17. **Mats Aspnäs**, Multiprocessor Architecture and Programming: The Hathi-2 System
18. **Anna Mikhajlova**, Ensuring Correctness of Object and Component Systems
19. **Vesa Torvinen**, Construction and Evaluation of the Labour Game Method
20. **Jorma Boberg**, Cluster Analysis. A Mathematical Approach with Applications to Protein Structures
21. **Leonid Mikhajlov**, Software Reuse Mechanisms and Techniques: Safety Versus Flexibility
22. **Timo Kaukoranta**, Iterative and Hierarchical Methods for Codebook Generation in Vector Quantization
23. **Gábor Magyar**, On Solution Approaches for Some Industrially Motivated Combinatorial Optimization Problems
24. **Linas Laibinis**, Mechanised Formal Reasoning About Modular Programs
25. **Shuhua Liu**, Improving Executive Support in Strategic Scanning with Software Agent Systems
26. **Jaakko Järvi**, New Techniques in Generic Programming – C++ is more Intentional than Intended
27. **Jan-Christian Lehtinen**, Reproducing Kernel Splines in the Analysis of Medical Data
28. **Martin Büchi**, Safe Language Mechanisms for Modularization and Concurrency
29. **Elena Troubitsyna**, Stepwise Development of Dependable Systems
30. **Janne Näppi**, Computer-Assisted Diagnosis of Breast Calcifications
31. **Jianming Liang**, Dynamic Chest Images Analysis
32. **Tiberiu Seceleanu**, Systematic Design of Synchronous Digital Circuits
33. **Tero Aittokallio**, Characterization and Modelling of the Cardiorespiratory System in Sleep-Disordered Breathing
34. **Ivan Porres**, Modeling and Analyzing Software Behavior in UML
35. **Mauno Rönkkö**, Stepwise Development of Hybrid Systems
36. **Jouni Smed**, Production Planning in Printed Circuit Board Assembly
37. **Vesa Halava**, The Post Correspondence Problem for Market Morphisms
38. **Ion Petre**, Commutation Problems on Sets of Words and Formal Power Series
39. **Vladimir Kvassov**, Information Technology and the Productivity of Managerial Work
40. **Frank Tétard**, Managers, Fragmentation of Working Time, and Information Systems

41. **Jan Manuch**, Defect Theorems and Infinite Words
42. **Kalle Ranto**, Z_4-Goethals Codes, Decoding and Designs
43. **Arto Lepistö**, On Relations Between Local and Global Periodicity
44. **Mika Hirvensalo**, Studies on Boolean Functions Related to Quantum Computing
45. **Pentti Virtanen**, Measuring and Improving Component-Based Software Development
46. **Adekunle Okunoye**, Knowledge Management and Global Diversity – A Framework to Support Organisations in Developing Countries
47. **Antonina Kloptchenko**, Text Mining Based on the Prototype Matching Method
48. **Juha Kivijärvi**, Optimization Methods for Clustering
49. **Rimvydas Rukšėnas**, Formal Development of Concurrent Components
50. **Dirk Nowotka**, Periodicity and Unbordered Factors of Words
51. **Attila Gyenesei**, Discovering Frequent Fuzzy Patterns in Relations of Quantitative Attributes
52. **Petteri Kaitovaara**, Packaging of IT Services – Conceptual and Empirical Studies
53. **Petri Rosendahl**, Niho Type Cross-Correlation Functions and Related Equations
54. **Péter Majlender**, A Normative Approach to Possibility Theory and Soft Decision Support
55. **Seppo Virtanen**, A Framework for Rapid Design and Evaluation of Protocol Processors
56. **Tomas Eklund**, The Self-Organizing Map in Financial Benchmarking
57. **Mikael Collan**, Giga-Investments: Modelling the Valuation of Very Large Industrial Real Investments
58. **Dag Björklund**, A Kernel Language for Unified Code Synthesis
59. **Shengnan Han**, Understanding User Adoption of Mobile Technology: Focusing on Physicians in Finland
60. **Irina Georgescu**, Rational Choice and Revealed Preference: A Fuzzy Approach
61. **Ping Yan**, Limit Cycles for Generalized Liénard-Type and Lotka-Volterra Systems
62. **Joonas Lehtinen**, Coding of Wavelet-Transformed Images
63. **Tommi Meskanen,** On the NTRU Cryptosystem
64. **Saeed Salehi**, Varieties of Tree Languages
65. **Jukka Arvo**, Efficient Algorithms for Hardware-Accelerated Shadow Computation
66. **Mika Hirvikorpi**, On the Tactical Level Production Planning in Flexible Manufacturing Systems
67. **Adrian Costea**, Computational Intelligence Methods for Quantitative Data Mining
68. **Cristina Seceleanu**, A Methodology for Constructing Correct Reactive Systems
69. **Luigia Petre**, Modeling with Action Systems
70. **Lu Yan**, Systematic Design of Ubiquitous Systems
71. **Mehran Gomari**, On the Generalization Ability of Bayesian Neural Networks
72. **Ville Harkke**, Knowledge Freedom for Medical Professionals – An Evaluation Study of a Mobile Information System for Physicians in Finland
73. **Marius Cosmin Codrea**, Pattern Analysis of Chlorophyll Fluorescence Signals
74. **Aiying Rong**, Cogeneration Planning Under the Deregulated Power Market and Emissions Trading Scheme
75. **Chihab BenMoussa**, Supporting the Sales Force through Mobile Information and Communication Technologies: Focusing on the Pharmaceutical Sales Force
76. **Jussi Salmi**, Improving Data Analysis in Proteomics
77. **Orieta Celiku**, Mechanized Reasoning for Dually-Nondeterministic and Probabilistic Programs
78. **Kaj-Mikael Björk**, Supply Chain Efficiency with Some Forest Industry Improvements
79. **Viorel Preoteasa**, Program Variables – The Core of Mechanical Reasoning about Imperative Programs
80. **Jonne Poikonen**, Absolute Value Extraction and Order Statistic Filtering for a Mixed-Mode Array Image Processor
81. **Luka Milovanov**, Agile Software Development in an Academic Environment
82. **Francisco Augusto Alcaraz Garcia**, Real Options, Default Risk and Soft Applications
83. **Kai K. Kimppa**, Problems with the Justification of Intellectual Property Rights in Relation to Software and Other Digitally Distributable Media
84. **Dragoş Truşcan**, Model Driven Development of Programmable Architectures
85. **Eugen Czeizler**, The Inverse Neighborhood Problem and Applications of Welch Sets in Automata Theory

86. **Sanna Ranto**, Identifying and Locating-Dominating Codes in Binary Hamming Spaces
87. **Tuomas Hakkarainen**, On the Computation of the Class Numbers of Real Abelian Fields
88. **Elena Czeizler**, Intricacies of Word Equations
89. **Marcus Alanen**, A Metamodeling Framework for Software Engineering
90. **Filip Ginter**, Towards Information Extraction in the Biomedical Domain: Methods and Resources
91. **Jarkko Paavola**, Signature Ensembles and Receiver Structures for Oversaturated Synchronous DS-CDMA Systems
92. **Arho Virkki**, The Human Respiratory System: Modelling, Analysis and Control
93. **Olli Luoma**, Efficient Methods for Storing and Querying XML Data with Relational Databases
94. **Dubravka Ilić**, Formal Reasoning about Dependability in Model-Driven Development
95. **Kim Solin**, Abstract Algebra of Program Refinement
96. **Tomi Westerlund**, Time Aware Modelling and Analysis of Systems-on-Chip
97. **Kalle Saari**, On the Frequency and Periodicity of Infinite Words
98. **Tomi Kärki**, Similarity Relations on Words: Relational Codes and Periods
99. **Markus M. Mäkelä**, Essays on Software Product Development: A Strategic Management Viewpoint
100. **Roope Vehkalahti**, Class Field Theoretic Methods in the Design of Lattice Signal Constellations
101. **Anne-Maria Ernvall-Hytönen**, On Short Exponential Sums Involving Fourier Coefficients of Holomorphic Cusp Forms
102. **Chang Li**, Parallelism and Complexity in Gene Assembly
103. **Tapio Pahikkala**, New Kernel Functions and Learning Methods for Text and Data Mining
104. **Denis Shestakov**, Search Interfaces on the Web: Querying and Characterizing
105. **Sampo Pyysalo**, A Dependency Parsing Approach to Biomedical Text Mining
106. **Anna Sell**, Mobile Digital Calendars in Knowledge Work
107. **Dorina Marghescu**, Evaluating Multidimensional Visualization Techniques in Data Mining Tasks
108. **Tero Säntti**, A Co-Processor Approach for Efficient Java Execution in Embedded Systems
109. **Kari Salonen**, Setup Optimization in High-Mix Surface Mount PCB Assembly
110. **Pontus Boström**, Formal Design and Verification of Systems Using Domain-Specific Languages
111. **Camilla J. Hollanti**, Order-Theoretic Mehtods for Space-Time Coding: Symmetric and Asymmetric Designs
112. **Heidi Himmanen**, On Transmission System Design for Wireless Broadcasting
113. **Sébastien Lafond**, Simulation of Embedded Systems for Energy Consumption Estimation
114. **Evgeni Tsivtsivadze**, Learning Preferences with Kernel-Based Methods
115. **Petri Salmela**, On Commutation and Conjugacy of Rational Languages and the Fixed Point Method
116. **Siamak Taati**, Conservation Laws in Cellular Automata
117. **Vladimir Rogojin**, Gene Assembly in Stichotrichous Ciliates: Elementary Operations, Parallelism and Computation
118. **Alexey Dudkov**, Chip and Signature Interleaving in DS CDMA Systems
119. **Janne Savela**, Role of Selected Spectral Attributes in the Perception of Synthetic Vowels
120. **Kristian Nybom**, Low-Density Parity-Check Codes for Wireless Datacast Networks
121. **Johanna Tuominen**, Formal Power Analysis of Systems-on-Chip
122. **Teijo Lehtonen**, On Fault Tolerance Methods for Networks-on-Chip
123. **Eeva Suvitie**, On Inner Products Involving Holomorphic Cusp Forms and Maass Forms
124. **Linda Mannila**, Teaching Mathematics and Programming – New Approaches with Empirical Evaluation
125. **Hanna Suominen**, Machine Learning and Clinical Text: Supporting Health Information Flow
126. **Tuomo Saarni**, Segmental Durations of Speech
127. **Johannes Eriksson**, Tool-Supported Invariant-Based Programming

128. **Tero Jokela**, Design and Analysis of Forward Error Control Coding and Signaling for Guaranteeing QoS in Wireless Broadcast Systems
129. **Ville Lukkarila**, On Undecidable Dynamical Properties of Reversible One-Dimensional Cellular Automata
130. **Qaisar Ahmad Malik**, Combining Model-Based Testing and Stepwise Formal Development
131. **Mikko-Jussi Laakso**, Promoting Programming Learning: Engagement, Automatic Assessment with Immediate Feedback in Visualizations
132. **Riikka Vuokko**, A Practice Perspective on Organizational Implementation of Information Technology
133. **Jeanette Heidenberg**, Towards Increased Productivity and Quality in Software Development Using Agile, Lean and Collaborative Approaches
134. **Yong Liu**, Solving the Puzzle of Mobile Learning Adoption
135. **Stina Ojala**, Towards an Integrative Information Society: Studies on Individuality in Speech and Sign
136. **Matteo Brunelli**, Some Advances in Mathematical Models for Preference Relations
137. **Ville Junnila**, On Identifying and Locating-Dominating Codes
138. **Andrzej Mizera**, Methods for Construction and Analysis of Computational Models in Systems Biology. Applications to the Modelling of the Heat Shock Response and the Self-Assembly of Intermediate Filaments.
139. **Csaba Ráduly-Baka**, Algorithmic Solutions for Combinatorial Problems in Resource Management of Manufacturing Environments
140. **Jari Kyngäs**, Solving Challenging Real-World Scheduling Problems
141. **Arho Suominen**, Notes on Emerging Technologies
142. **József Mezei**, A Quantitative View on Fuzzy Numbers
143. **Marta Olszewska**, On the Impact of Rigorous Approaches on the Quality of Development
144. **Antti Airola**, Kernel-Based Ranking: Methods for Learning and Performace Estimation
145. **Aleksi Saarela**, Word Equations and Related Topics: Independence, Decidability and Characterizations
146. **Lasse Bergroth**, Kahden merkkijonon pisimmän yhteisen alijonon ongelma ja sen ratkaiseminen
147. **Thomas Canhao Xu**, Hardware/Software Co-Design for Multicore Architectures
148. **Tuomas Mäkilä**, Software Development Process Modeling – Developers Perspective to Contemporary Modeling Techniques
149. **Shahrokh Nikou**, Opening the Black-Box of IT Artifacts: Looking into Mobile Service Characteristics and Individual Perception
150. **Alessandro Buoni**, Fraud Detection in the Banking Sector: A Multi-Agent Approach
151. **Mats Neovius**, Trustworthy Context Dependency in Ubiquitous Systems
152. **Fredrik Degerlund**, Scheduling of Guarded Command Based Models
153. **Amir-Mohammad Rahmani-Sane**, Exploration and Design of Power-Efficient Networked Many-Core Systems
154. **Ville Rantala**, On Dynamic Monitoring Methods for Networks-on-Chip
155. **Mikko Pelto**, On Identifying and Locating-Dominating Codes in the Infinite King Grid
156. **Anton Tarasyuk**, Formal Development and Quantitative Verification of Dependable Systems
157. **Muhammad Mohsin Saleemi**, Towards Combining Interactive Mobile TV and Smart Spaces: Architectures, Tools and Application Development
158. **Tommi J. M. Lehtinen**, Numbers and Languages
159. **Peter Sarlin**, Mapping Financial Stability
160. **Alexander Wei Yin**, On Energy Efficient Computing Platforms
161. **Mikołaj Olszewski**, Scaling Up Stepwise Feature Introduction to Construction of Large Software Systems
162. **Maryam Kamali**, Reusable Formal Architectures for Networked Systems
163. **Zhiyuan Yao**, Visual Customer Segmentation and Behavior Analysis – A SOM-Based Approach
164. **Timo Jolivet**, Combinatorics of Pisot Substitutions
165. **Rajeev Kumar Kanth**, Analysis and Life Cycle Assessment of Printed Antennas for Sustainable Wireless Systems
166. **Khalid Latif**, Design Space Exploration for MPSoC Architectures

167. **Bo Yang**, Towards Optimal Application Mapping for Energy-Efficient Many-Core Platforms
168. **Ali Hanzala Khan**, Consistency of UML Based Designs Using Ontology Reasoners
169. **Sonja Leskinen**, m-Equine: IS Support for the Horse Industry
170. **Fareed Ahmed Jokhio**, Video Transcoding in a Distributed Cloud Computing Environment
171. **Moazzam Fareed Niazi**, A Model-Based Development and Verification Framework for Distributed System-on-Chip Architecture
172. **Mari Huova**, Combinatorics on Words: New Aspects on Avoidability, Defect Effect, Equations and Palindromes
173. **Ville Timonen**, Scalable Algorithms for Height Field Illumination
174. **Henri Korvela**, Virtual Communities – A Virtual Treasure Trove for End-User Developers
175. **Kameswar Rao Vaddina**, Thermal-Aware Networked Many-Core Systems
176. **Janne Lahtiranta**, New and Emerging Challenges of the ICT-Mediated Health and Well-Being Services
177. **Irum Rauf**, Design and Validation of Stateful Composite RESTful Web Services
178. **Jari Björne**, Biomedical Event Extraction with Machine Learning
179. **Katri Haverinen**, Natural Language Processing Resources for Finnish: Corpus Development in the General and Clinical Domains
180. **Ville Salo**, Subshifts with Simple Cellular Automata
181. **Johan Ersfolk**, Scheduling Dynamic Dataflow Graphs
182. **Hongyan Liu**, On Advancing Business Intelligence in the Electricity Retail Market
183. **Adnan Ashraf**, Cost-Efficient Virtual Machine Management: Provisioning, Admission Control, and Consolidation
184. **Muhammad Nazrul Islam**, Design and Evaluation of Web Interface Signs to Improve Web Usability: A Semiotic Framework
185. **Johannes Tuikkala**, Algorithmic Techniques in Gene Expression Processing: From Imputation to Visualization
186. **Natalia Díaz Rodríguez**, Semantic and Fuzzy Modelling for Human Behaviour Recognition in Smart Spaces. A Case Study on Ambient Assisted Living
187. **Mikko Pänkäälä**, Potential and Challenges of Analog Reconfigurable Computation in Modern and Future CMOS
188. **Sami Hyrynsalmi**, Letters from the War of Ecosystems – An Analysis of Independent Software Vendors in Mobile Application Marketplaces
189. **Seppo Pulkkinen**, Efficient Optimization Algorithms for Nonlinear Data Analysis
190. **Sami Pyöttiälä**, Optimization and Measuring Techniques for Collect-and-Place Machines in Printed Circuit Board Industry
191. **Syed Mohammad Asad Hassan Jafri**, Virtual Runtime Application Partitions for Resource Management in Massively Parallel Architectures
192. **Toni Ernvall**, On Distributed Storage Codes
193. **Yuliya Prokhorova**, Rigorous Development of Safety-Critical Systems
194. **Olli Lahdenoja**, Local Binary Patterns in Focal-Plane Processing – Analysis and Applications
195. **Annika H. Holmbom**, Visual Analytics for Behavioral and Niche Market Segmentation
196. **Sergey Ostroumov**, Agent-Based Management System for Many-Core Platforms: Rigorous Design and Efficient Implementation
197. **Espen Suenson**, How Computer Programmers Work – Understanding Software Development in Practise
198. **Tuomas Poikela**, Readout Architectures for Hybrid Pixel Detector Readout Chips
199. **Bogdan Iancu**, Quantitative Refinement of Reaction-Based Biomodels
200. **Ilkka Törmä**, Structural and Computational Existence Results for Multidimensional Subshifts
201. **Sebastian Okser**, Scalable Feature Selection Applications for Genome-Wide Association Studies of Complex Diseases
202. **Fredrik Abbors**, Model-Based Testing of Software Systems: Functionality and Performance
203. **Inna Pereverzeva**, Formal Development of Resilient Distributed Systems
204. **Mikhail Barash**, Defining Contexts in Context-Free Grammars
205. **Sepinoud Azimi**, Computational Models for and from Biology: Simple Gene Assembly and Reaction Systems
206. **Petter Sandvik**, Formal Modelling for Digital Media Distribution

207. **Jongyun Moon**, Hydrogen Sensor Application of Anodic Titanium Oxide Nanostructures
208. **Simon Holmbacka**, Energy Aware Software for Many-Core Systems
209. **Charalampos Zinoviadis**, Hierarchy and Expansiveness in Two-Dimensional Subshifts of Finite Type
210. **Mika Murtojärvi**, Efficient Algorithms for Coastal Geographic Problems
211. **Sami Mäkelä**, Cohesion Metrics for Improving Software Quality
212. **Eyal Eshet**, Examining Human-Centered Design Practice in the Mobile Apps Era
213. **Jetro Vesti**, Rich Words and Balanced Words
214. **Jarkko Peltomäki**, Privileged Words and Sturmian Words
215. **Fahimeh Farahnakian**, Energy and Performance Management of Virtual Machines: Provisioning, Placement and Consolidation
216. **Diana-Elena Gratie**, Refinement of Biomodels Using Petri Nets
217. **Harri Merisaari**, Algorithmic Analysis Techniques for Molecular Imaging
218. **Stefan Grönroos**, Efficient and Low-Cost Software Defined Radio on Commodity Hardware
219. **Noora Nieminen**, Garbling Schemes and Applications
220. **Ville Taajamaa**, O-CDIO: Engineering Education Framework with Embedded Design Thinking Methods
221. **Johannes Holvitie**, Technical Debt in Software Development – Examining Premises and Overcoming Implementation for Efficient Management
222. **Tewodros Deneke**, Proactive Management of Video Transcoding Services
223. **Kashif Javed**, Model-Driven Development and Verification of Fault Tolerant Systems
224. **Pekka Naula**, Sparse Predictive Modeling – A Cost-Effective Perspective
225. **Antti Hakkala**, On Security and Privacy for Networked Information Society – Observations and Solutions for Security Engineering and Trust Building in Advanced Societal Processes
226. **Anne-Maarit Majanoja**, Selective Outsourcing in Global IT Services – Operational Level Challenges and Opportunities
227. **Samuel Rönnqvist**, Knowledge-Lean Text Mining
228. **Mohammad-Hashem Hahgbayan**, Energy-Efficient and Reliable Computing in Dark Silicon Era
229. **Charmi Panchal**, Qualitative Methods for Modeling Biochemical Systems and Datasets: The Logicome and the Reaction Systems Approaches
230. **Erkki Kaila**, Utilizing Educational Technology in Computer Science and Programming Courses: Theory and Practice
231. **Fredrik Robertsén**, The Lattice Boltzmann Method, a Petaflop and Beyond
232. **Jonne Pohjankukka**, Machine Learning Approaches for Natural Resource Data
233. **Paavo Nevalainen**, Geometric Data Understanding: Deriving Case-Specific Features
234. **Michal Szabados**, An Algebraic Approach to Nivat's Conjecture
235. **Tuan Nguyen Gia**, Design for Energy-Efficient and Reliable Fog-Assisted Healthcare IoT Systems
236. **Anil Kanduri**, Adaptive Knobs for Resource Efficient Computing
237. **Veronika Suni**, Computational Methods and Tools for Protein Phosphorylation Analysis
238. **Behailu Negash**, Interoperating Networked Embedded Systems to Compose the Web of Things
239. **Kalle Rindell**, Development of Secure Software: Rationale, Standards and Practices
240. **Jurka Rahikkala**, On Top Management Support for Software Cost Estimation
241. **Markus A. Whiteland**, On the k-Abelian Equivalence Relation of Finite Words
242. **Mojgan Kamali**, Formal Analysis of Network Routing Protocols
243. **Jesús Carabaño Bravo**, A Compiler Approach to Map Algebra for Raster Spatial Modeling
244. **Amin Majd**, Distributed and Lightweight Meta-heuristic Optimization Method for Complex Problems
245. **Ali Farooq**, In Quest of Information Security in Higher Education Institutions: Security Awareness, Concerns, and Behaviour of Students
246. **Juho Heimonen**, Knowledge Representation and Text Mining in Biomedical, Healthcare, and Political Domains

247. **Sanaz Rahimi Moosavi**, Towards End-to-End Security in Internet of Things based Healthcare
248. **Mingzhe Jiang**, Automatic Pain Assessment by Learning from Multiple Biopotentials
249. **Johan Kopra**, Cellular Automata with Complicated Dynamics
250. **Iman Azimi**, Personalized Data Analytics for Internet-of-Things-based Health Monitoring
251. **Jaakko Helminen**, Systems Action Design Research: Delineation of an Application to Develop Hybrid Local Climate Services
252. **Aung Pyae**, The Use of Digital Games to Enhance the Physical Exercise Activity of the Elderly: A Case of Finland
253. **Woubishet Zewdu Taffese**, Data-Driven Method for Enhanced Corrosion Assessment of Reinforced Concrete Structures
254. **Etienne Moutot**, Around the Domino Problem – Combinatorial Structures and Algebraic Tools
255. **Joonatan Jalonen**, On Some One-Sided Dynamics of Cellular Automata
256. **Outi Montonen**, On Multiobjective Optimization from the Nonsmooth Perspective

Turku
Centre *for*
Computer
Science

http://www. tucs.fi

tucs@abo.fi

University of Turku
Faculty of Science and Engineering
- Department of Future Technologies
- Department of Mathematics and Statistics

Turku School of Economics
- Institute of Information Systems Science

Åbo Akademi University
Faculty of Science and Engineering
- Computer Engineering
- Computer Science

Faculty of Social Sciences, Business and Economics
- Information Systems

ISBN 978-952-12-3966-3
ISSN 1239-1883

Outi Montonen

On Multiobjective Optimization from the Nonsmooth Perspective

Outi Montonen

On Multiobjective Optimization from the Nonsmooth Perspective

Outi Montonen

On Multiobjective Optimization from the Nonsmooth Perspective