JYVÄSKYLÄ STUDIES IN COMPUTING
68

Jussi Hakanen

On Potential of Interactive Multiobjective Optimization in Chemical Process Design

JYVÄSKYLÄN YLIOPISTO

JYVÄSKYLÄ STUDIES IN COMPUTING 68

Jussi Hakanen

On Potential of Interactive Multiobjective Optimization in Chemical Process Design

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella
julkisesti tarkastettavaksi Villa Ranan Blomstedtin salissa
joulukuun 16. päivänä 2006 kello 14.

Academic dissertation to be publicly discussed, by permission of
the Faculty of Information Technology of the University of Jyväskylä,
in the building Villa Rana, Blomstedt Hall, on December 16, 2006 at 2 p.m.

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2006

On Potential of Interactive Multiobjective Optimization in Chemical Process Design

JYVÄSKYLÄ STUDIES IN COMPUTING 68

Jussi Hakanen

On Potential of Interactive Multiobjective Optimization in Chemical Process Design

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2006

Editors
Tommi Kärkkäinen
Department of Mathematical Information Technology, University of Jyväskylä
Pekka Olsbo, Marja-Leena Tynkkynen
Publishing Unit, University Library of Jyväskylä

URN:ISBN:9513926982
ISBN 951-39-2698-2 (PDF)

ISBN 951-39-2684-2 (nid.)
ISSN 1456-5390

Jyväskylä University Printing House, Jyväskylä 2006

ABSTRACT

Hakanen, Jussi
On Potential of Interactive Multiobjective Optimization in Chemical Process Design
Jyväskylä: University of Jyväskylä, 2006, 81 p.(+included articles)
(Jyväskylä Studies in Computing
ISSN 1456-5390; 68)
ISBN 951-39-2698-2
Finnish summary
Diss.

In this thesis, we study the potential of interactive multiobjective optimization (MOO) in solving chemical process design problems with several conflicting objectives. When designing real-world chemical processes, there are several performance criteria that need to be considered simultaneously, for example, economical aspects, environmental impact and process operability to name a few. These aspects are often conflicting and, therefore, the process considered can not be optimal with respect to each criteria at the same time. This means that some compromise must be made.

In this thesis, interactive MOO is used to help the designer in finding the best compromise between the conflicting performance criteria. A compromise in MOO is called a Pareto optimal solution and it means that the performance of the process can not be improved with respect to any criteria without impairment in some other criteria. Previously, chemical process design problems have been usually solved with two performance criteria at most. The approach described in this thesis is not restricted by the number of performance criteria as is often the case in methods used by chemical engineers. Therefore, process design problems can be considered in their truly multiobjective character without unnecessary simplifications.

A process design tool based on IND-NIMBUS, an implementation of the interactive MOO method NIMBUS®, has been developed in this thesis. The IND-NIMBUS process design tool developed consists of three parts: a modelling tool, an optimizer and a graphical user interface. The modelling tool produces a numerical model of the process considered. The optimizer consists of NIMBUS® and some suitable single objective optimizer that finally produces new Pareto optimal solutions by solving the single objective subproblems produced by NIMBUS®. Different modelling tools and single objective optimizers can be used depending on the problem to be solved. The graphical user interface enables the designer to input his/her preferences during the interactive solution procedure as well as comparing the Pareto optimal solutions obtained with different types of visualizations. The preference information is used to guide the search of the most preferred solution by generating desirable Pareto optimal solutions.

Some preliminary ideas are also introduced to aid the designer in the decision making procedure. In this thesis, trade-off information is studied as a way of supporting the designer. The idea is to guide the designer towards the most preferred solution by showing him/her trade-off information during the interactive solution procedure. Trade-off information can help the designer in realizing what kind of compromises could be available. This information is also beneficial in convincing the designer that (s)he has found the best compromise solution available.

To test these ideas in practice, several industrial process design problems related, for example, to paper making and sugar industries were solved. Previously, these problems were considered in a simplified formulation by including only one or some of the important criteria. Here, these problems were considered in a totally new way with more than two important performance criteria included. The IND-NIMBUS process design tool was applied to these problems and all the problems were solved with the help of real designers who were experts in the specific areas of the applications considered. The results obtained were promising and the designers found the interactive solution procedure easy to understand and use. To summarize, we showed that interactive MOO can be succesfully applied to solving chemical process design problems in their true multiobjective character and our approach turned out to be efficient in practice.

Keywords: Chemical process design, multiobjective optimization, interactive methods, classification, NIMBUS® method, decision maker, scalarization

Author

Jussi Hakanen
Department of Mathematical Information Technology
University of Jyväskylä
Finland

Supervisors

Professor Kaisa Miettinen
Department of Business Technology
Helsinki School of Economics
Finland

Professor Marko M. Mäkelä
Department of Mathematics
University of Turku
Finland

Reviewers

Professor Hirotaka Nakayama
Department of Information Science and Systems Engineering
Konan University
Japan

Professor Stratos Pistikopoulos
Centre for Process Systems Engineering
Imperial College London
United Kingdom

Opponent

Professor Margaret M. Wiecek
Department of Mathematical Sciences
Clemson University
USA

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisors Prof. Kaisa Miettinen and Prof. Marko Mäkelä who have guided me through this research work and encouraged me during difficult moments. I am also grateful to Dr. Jussi Manninen, Dr. Sakari Kaijaluoto, Mr. Juha Hakala and Mr. Juha Leppävuori from the VTT Technical Research Center of Finland for providing me support in the field of chemical engineering and chemical process design which is not my expertise. I would also like to thank Prof. Lorenz T. Biegler and Mr. Yoshiaki Kawajiri from Carnegie Mellon University, USA, for making my visit there a success and cooperating with me in the resarch for multiobjective SMBs. For evaluating my thesis, I would like to thank Prof. Hirotaka Nakayama from Konan University, Japan, and Prof. Efstratios N. Pistikopoulos from the Imperial College, United Kingdom. I would also like to express my appreciation to the whole optimization group, especially Mr. Vesa Ojalehto for his technical support and Mr. Timo Aittokoski for comments about the thesis. Prof. Miettinen was also of valuable assistance with the english language.

This research was financially supported by the COMAS graduate school of the University of Jyväskylä, Tekes, the Finnish Funding Agency for Technology and Innovation (project *NIMBUS – Multiobjective Optimization in Product Development*), the Emil Aaltonen foundation, the Jenny and Antti Wihuri foundation and the Ellen and Artturi Nyyssönen foundation.

Finally, I would like to thank my wife Merja and my mother Helena for their love and support during this work. They have encouraged me all the time, especially at times when my PhD project has tested my belief. They both are the source of joy and happiness and the most important part of my life.

LIST OF FIGURES

CONTENTS

LIST OF INCLUDED ARTICLES

I J. Hakanen, K. Miettinen, and M.M. Mäkelä, An Application of Multiobjective Optimization to Process Simulation, *In CD-Rom Proceedings of ECCOMAS 2004, the 4th European Congress on Computational Methods in Applied Sciences and Engineering, Ed. by Neittaanmäki, P., Rossi, T., Korotov, S., Onate, E., Periaux, J., Knörzer, D., Volume II, Jyväskylä, Finland, 2004.*

II J. Hakanen, K. Miettinen, M.M. Mäkelä, and J. Manninen, On Interactive Multiobjective Optimization with NIMBUS in Chemical Process Design, *Journal of Multi-Criteria Decision Analysis* **13**, *125–134 (2005).*

III J. Hakanen, J. Hakala, and J. Manninen, An Integrated Multiobjective Design Tool for Process Design, *Applied Thermal Engineering* **26**, *1393–1399 (2006).*

IV J. Hakanen, Y. Kawajiri, K. Miettinen, and L.T. Biegler, Interactive Multi-Objective Optimization for Simulated Moving Bed Processes, *Control & Cybernetics. To appear.*

V J. Hakanen, Y. Kawajiri, K. Miettinen, and L.T. Biegler, Interactive Multiobjective Optimization of Superstructure SMB Processes, *Submitted to the Post conference volume of MOPGP 2006, the 7th International Conference on Multi-Objective Programming and Goal Programming, Tours, France.*

VI J. Hakanen and P. Eskelinen, Ideas of Using Trade-off Information in Supporting the Decision Maker in Reference Point Based Interactive Multiobjective Optimization, *Working Paper W-405, Helsinki School of Economics, 2006.*

1 INTRODUCTION

This thesis is devoted to using interactive multiobjective optimization in solving process design problems arising from chemical engineering involving multiple objectives. The main question for this research was how can chemical process design benefit from interactive multiobjective optimization? Chemical process design problems are challenging real-world optimization problems and they often include several conflicting performance criteria that need to be considered simultaneously. Optimization of these problems is not a straightforward task even in the case of a single objective function [8] and, therefore, answering the research question requires studying the challenges that chemical process design problems set on optimization.

In this thesis, we concentrate on processes in the chemical industry like, for example, paper making processes. Process simulation has become an important tool in chemical process design [8]. Previously, processes were planned only by making test runs with the process equipment or with scale models of the equipment. That is, however, very expensive and time consuming. With process simulation, the number of test runs required in planning can be reduced significantly.

It is very useful to be able to simulate the behaviour of a process. Rigorous mathematical models of many chemical processes have existed for a long time, but the development of computers and programming languages, such as *Fortran*, enabled the numerical simulation of the models in the late 1950's [97]. At first, only single process devices were simulated at a time due to the limited computer capacity. When the capacity increased, it became possible to make simulations about the whole process in a reasonable time. Recently, the continuing increase of the computer capacity has enabled the simulation of very complex processes or whole factories in few seconds by a microcomputer, which previously required a supercomputer and plenty of computation time.

Nowadays, parts of industrial processes or even whole processes can be modelled mathematically and the models are sufficiently accurate in order to use simulation in process design with computers. Usually, the models of industrial processes are quite complex and computationally very demanding. For some processes, one simulation run of the process can still take many hours or even many days requiring lots of computer capacity and sometimes parallel computing. After the process has been simulated, we have one description of the performance of the process with some fixed process parameters.

If we want to design a process that has the best possible performance, simulation is not enough. Mathematical models used in process simulation typically include several parameters that affect the efficiency of the process. Efficiency is measured in the sense of different objective functions or performance criteria. Therefore, finding the best combination of the parameters can lead to an improvement of the efficiency of the process. The best parameter values are found from a feasible set that is restricted by some constraints originating from the simulation model consisting of physical or chemical properties of the process or environmental, monetary, qualitative or quantitative restrictions to name a few [8, 46].

The performance of the process can be improved by applying optimization to the simulation model with respect to its parameters [8, 9, 46, 53] because simulation by trial and error does not necessarily give a good performance. Instead of simply simulating the process with some fixed parameter values, it is desirable to find such a combination of parameters that will lead to the best possible performance. This can be realized by optimizing some performance criterion subject to a set of constraints [7, 8, 9, 46, 53]. In optimization, the parameters of the mathematical models are called variables.

In real-world optimization problems, it is not always sufficient to consider only one performance criterion because there are usually several different aspects involved that produce many performance criteria. The various performance criteria describe the goodness of the process from different perspectives. In addition, they are usually conflicting which means that they attain their optima at different points and they can not be all optimized at the same time. For example, if we want to minimize costs and maximize product quality we need to use multiobjective optimization [14, 43, 64, 88, 89].

Multiobjective process design problems involve criteria that are from many different areas including, for example, economic aspects, environmental impact and process operability. In some of these areas, the mathematical modelling of the criteria is not so well developed so far that the models would describe the phenomena involved accurately enough in order to be used in optimization. In addition, performance criteria are often in different units, different scales or are different in nature. For these reasons, it is not easy to compare them directly, for example, by summing them all up, which would lead us to a problem of adding up apples and oranges. This means that we can not usually just directly combine all the objectives into the same objective function, for example, just add them all up, without changing the interrelationships of the objectives. The methods of multiobjective optimization (see, for example, [64]) help us to overcome these difficulties and present more sophisticated ways to convert multiple objectives into a single objective function, that is scalarizing the multiobjective optimization problem.

In multiobjective optimization, there is not necessarily a unique optimal solution like in classical single objective optimization, but a set of mathematically equivalent compromise solutions called *Pareto optimal solutions* can be identified.

A solution is Pareto optimal if no criterion can be improved without impairing at least some other criteria. Usually, we want to end up with one final solution and, therefore, we need some additional information that helps us to find the best Pareto optimal solution. A decision maker, DM, (or designer in process design) is a person who can provide this information and has valuable knowledge about the problem in question. The DM expresses preference information on what kind of solution is desirable while looking for the best compromise solution as the final solution.

Optimization of real-world industrial processes is often computationally demanding. Depending on the optimization method used, it may require a considerable number of evaluations of the process model during the optimization procedure, in order to obtain an optimal solution for a single objective function. On the other hand, as mentioned before real-world optimization problems typically have several conflicting objectives or performance criteria that need to be considered simultaneously, and it is important to have optimization tools that require only few process model simulations in order to obtain optimal or satisfactory solutions quickly.

A number of papers have been published describing the usage of multiobjective optimization in chemical process design including [2, 13, 18, 51, 54, 59, 85, 91, 110]. The most common approach has been to consider only two conflicting objective functions at a time. For some reason, solving the multiobjective optimization problem has often been understood as approximating the *Pareto optimal set*, that is, the set of all Pareto optimal solutions. In order to select the most preferred solution in that case, some kind of visualization of the Pareto optimal set is needed. The visualization becomes difficult or impossible for more than two objectives which naturally makes this kind of approach useful for only two objective functions. Probably, another reason for considering only two objectives has been that the designers have not been aware of the existence of methods developed for multiobjective optimization that can consider more than two objective functions and, therefore, have used only two. The biobjective optimization problem formulated by the designers has usually been solved by optimizing the weighted sum of the objectives or optimizing only one objective while considering the other as a constraint.

Both using weights or additional constraints are simple to implement and use but they have some severe drawbacks. For example, by optimizing the weighted sum of the objectives some of the Pareto optimal solutions that might be desired can not be found. On the other hand, if we optimize only one of the objectives it is not obvious which objective should be chosen to be optimised and the upper bounds for the objectives considered as inequality constraints are also difficult to define in order to find desired solutions. The cause for using these simple methods could have been that the designers have not necessarily known any better ones. In addition, the case of two objectives is a special case of multiobjective optimization and, usually, real-world problems have more than two objectives that need to be considered. For these reasons, we need to use more sophisticated methods of multiobjective optimization. We describe multiobjective optimization previously used in chemical process design more detailed in Chapter 3.

In this thesis, we use interactive multiobjective optimization to solve chemical process design problems involving more than two objective functions. In interactive multiobjective optimization, preference information is asked from the DM iteratively, thus, making the solution procedure interactive. Interactive methods enable the DM to adjust his/her preferences while (s)he obtains new information during the interactive procedure. Therefore, interactive methods act also as learning tools and provide the DM with information on interrelationships of conflicting criteria. In addition, interactive methods are computationally efficient because they only need to generate few Pareto optimal solutions. For these reasons, interactive multiobjective optimization methods are considered to be promising in solving real-world problems that can be computationally demanding [34, 64].

There have been some applications of interactive multiobjective optimization methods in chemical process design, for example, in the fields of carbochemical and paper industries and production planning [4, 28, 49, 93]. However, utilization of interactive methods has been infrequent and concentrated mainly on solving linear problems [4, 49]. In this research, we use the interactive NIMBUS® method [64, 67, 72] to solve nonlinear multiobjective chemical process design problems. We have chosen NIMBUS® because there are not many implementations of interactive multiobjective optimization methods available other than NIMBUS and it has been succesfully applied to many challenging multiobjective optimization problems including the optimal design of the paper machine headbox [36] and the optimal control problem in the continuous casting of steel [74]. There is an implementation operating on the Internet called *WWW-NIMBUS* [69, 72] (`http://nimbus.it.jyu.fi`) that is freely available for academic problems and a new implementation *IND-NIMBUS* [82] (`http://ind-nimbus.it.jyu.fi`) that can be used for solving industrial problems. Besides, NIMBUS® has been developed at the University of Jyväskylä and support is easily available as well as access to the original programming codes. Real-world processes are usually nonlinear by nature, and by using NIMBUS®, we do not have to be restricted only to linear problems. In addition, the preference information requested from the DM in the NIMBUS® method is easy to understand for the DM, thus, making the usage of NIMBUS® easy for the DM. More information about interactive methods in chemical process design can be found in Chapter 3.

Solving optimization problems related to chemical process design is not straightforward as mentioned before. There are a few more challenges that need to be taken into account that are not present, for example, in solving academic optimization problems. First of all, *exact gradient information* is not usually available which can cause convergence failures or difficulties for certain gradient based optimizers. A numerical model of the process is produced by a process simulator or some modelling tool, and they do not necessarily have analytical forms of the problem functions. Therefore, partial derivatives need to be numerically approximated using, for example, finite differences [3, 46]. In addition, function values can also be numerical approximations resulting from some iterative procedure and that can cause additional inaccuracy in approximated gradients [34]. Therefore, in some cases it can be necessary to use such optimizers that do not utilize gradient information.

Let us mention that one way to obtain accurate gradient information is to use *automatic differentiation* (AD, see, for example, [29, 38]). AD is a method to numerically evaluate the derivative of a function that is given by a computer program. It avoids the drawbacks of classical ways to compute derivatives, namely symbolic differentiation and finite difference approximations. Symbolic differentiation is time consuming while the drawbacks of using finite differences are accuracy and the selection of discretization. In AD, a new arithmetic is introduced that computes also the derivative of each arithmetic operation as well as the numerical value. Then, the derivative of the whole function is computed by utilizing the *chain rule of differentiation*. Applying AD in process simulation deserves further study, but it can be difficult in practical applications with large process simulation software packages because of the complexity of the software code.

Another challenge is the existence of so-called *non-physical solutions* which means that for some feasible combination of decision variables, the numerical model of the process leads to a non-physical state. By non-physical state we mean that for that particular combination of decision variables the physical model makes no sense. In these cases, simulator or modelling tool is not able to produce function values (or gradients) for these decision variables and the optimization method used must be able to handle such a situation. Finally, there is lots of data transmission between the optimization method and modelling tool, for example, problem dimensions, variable and function values as well as gradient information. Therefore, the interfaces between the modelling tool and the optimization methods must be designed carefully in order to have an efficient implementation [34]. The above-mentioned challengies related to accuracy of gradient information and non-physical solutions are considered in [34].

An industrial process design tool is a computer program or a combination of several computer programs that can be used to solve (chemical) process design problems. A process design tool consists of several different parts. First of all, we need a simulator or some modelling tool that produces a numerical model of the problem we are considering. Secondly, we need an optimizer which is able to solve optimization problems based on the numerical model. This optimizer should be able to consider all the different aspects of the problem, namely, all the different performance criteria included. The optimizer itself should consist of at least two parts: a multiobjective optimization method that handles conflicting performance criteria and an efficient single objective optimizer that finally produces new designs as the solutions of the scalarized multiobjective optimization problem produced by some multiobjective optimization method. Finally, we also need a graphical user interface that enables the usage of the tool and visualization and comparison of the Pareto optimal solutions obtained.

In this thesis, we describe a process design tool based on IND-NIMBUS that we have developed and applied to various chemical process design problems. To our knowledge, no such tools have earlier been described in the literature utilizing interactive multiobjective optimization for nonlinear problems. In the context of IND-NIMBUS, we have used different modelling tools and single objective optimizers depending on the problem in question. The tool has been found very useful in considering multiple performance criteria present in chemical process design problems.

The thesis consists of six papers [I]–[VI]. Papers [I]–[III] consider the usage of IND-NIMBUS in solving steady-state chemical process design problems that are modelled with the BALAS® process simulator, developed at the VTT Technical Research Center of Finland. They describe how the process simulator and IND-NIMBUS have been combined in order to be able to solve steady-state chemical process design problems utilizing interactive multiobjective optimization. These papers include different applications from paper making and power industries that are considered in a novel way, that is, with more than two conflicting objectives. Papers [IV, V] deal with the design of dynamic simulated moving bed processes that are used in separation of chemical products. The problems considered involve models with partial differential equations and they are for the first time studied with four different objective functions and they lead to large challenging optimization problems. The papers describe how to solve these types of optimization problems with efficient optimization tools within IND-NIMBUS concentrating especially on how to quickly produce new Pareto optimal solutions for large and computationally demanding problems. Note that all the industrial applications in these papers were considered in a novel way. In other words, these problems have not been previously solved with all these objectives function simultaneously. All the papers are aimed at both chemical engineering and multiobjective optimization communities.

In paper [VI], we propose some ideas of how to utilize trade-off information in supporting the DM during the interactive solution procedure. The ultimate goal is to aid the DM in finding the best compromise solution without taking too much of his/her time and, therefore, best utilize the benefits of the interactive solution procedure. Previously, trade-off information has mostly been used as parts of some multiobjective optimization methods [14, 32, 103, 111]. In addition, a lot of trade-off theory has been developed (see, for example, [14, 31, 40, 57, 58, 70, 87, 88]) but not so much practical and numerical experiences have been published. Therefore, the paper is focused on practical and numerical usability of trade-off information.

Interactive multiobjective optimization turned out to be very useful in providing chemical process design a way to solve these kind of problems sufficiently well. From the numerical applications considered, we could conclude that good and promising solutions could be obtained with this approach and the designers found the interactive solution procedure easy to use and understand. It is difficult to compare the results obtained with the previous ones because all the problems were considered with more objectives than before, that is, more realistically. In this way, the interdependencies of the conflicting objectives could be studied when the trade-offs between them became clearer to the designers.

In practice, there is a lot of work to be done in order to combine all the elements required in a good process design tool. For example, there are various algorithms and iterative procedures in both the optimizer and the modelling tool that need to be synchronized in order to obtain numerically reliable results. In addition, there are lots of interfaces between these different components that need to be constructed. In the process design tool that we have developed in this thesis, there are several components that we had to get working together, namely, IND-NIMBUS, the NIMBUS® method, the BALAS® process simulator, several single objective optimizers and the AMPL modelling language. Therefore, developng general interfaces that enable combining different parts easily and efficiently is an important issue in developing chemical process design tools and it also makes easier the usage of specialized optimizers and modelling tools.

This research gave us an understanding of what needs to be done in order to be able to efficiently consider industrial chemical process design problems as truly multiobjective ones. In a nutshell, it requires the usage of such interactive multiobjective optimization method that enables the designer to find the best compromise solution while generating only few Pareto optimal solutions by using efficient single objective optimizer that takes into account the special features of the problem considered. In addition, the method should have a graphical user interface that makes the usage of the method easy for the designer and offers also a possibility to compare the solutions obtained with the help of some visualizations. Other researchers can use these ideas and approaches to further develop this field because this thesis offers them some basis to start with.

The contents of this thesis is the following. First, in Chapter 2, we present basic ideas of multiobjective optimization and the methods that we have used in this research. Chapter 3 is devoted to chemical process design describing optimization previously done in this field as well as its challenges and possibilities to multiobjective optimization. In addition, construction of industrial process design tool based on IND-NIMBUS, that is, the tool we have developed, is described. Chapter 4 is devoted to industrial applications of interactive multiobjective process design from different areas of chemical engineering. In Chapter 5, we describe the author's conribution in this thesis and, finally, in Chapter 6, we give some concluding remarks and ideas for future research.

2 MULTIOBJECTIVE OPTIMIZATION

Optimization means finding the best solution to a problem within all the possible solutions. In other words, we must find the optimum (minimum or maximum) of one or more objective functions with respect to some constraints by varying the values of the variables. Traditionally, only one objective function is considered and then we are dealing with *mathematical programming* or, more exactly, *single objective optimization*.

Mathematical programming has been studied for a long time and there are lots of well developed methods for single objective optimization (see, for example, [5, 26, 41, 42]). Different methods have been developed for different types of optimization problems depending, for example, on the properties of objective and constraint functions or on the type of the variables. In this thesis, we deal with optimization problems with nonlinear functions and continuous variables.

In real-world optimization problems, there are often more than one function or criterion that we want to optimize simultaneously. In that case, the methods for single objective optimization are not enough, but we need to consider methods of *multiobjective optimization* [14, 64, 88, 89]. In this chapter, we present a summary of multiobjective optimization and we concentrate on interactive multiobjective optimization. In addition, we present the multiobjective optimization method we are using in this research, namely the interactive NIMBUS® method. We start by formulating a single objective optimization problem because multiobjective problems are usually scalarized, that is, converted into a single objective optimization problem and solutions of the multiobjective problem are obtained by solving it.

2.1 Multiobjective optimization problem

A single objective optimization problem can be formulated as follows:

$$\begin{aligned} &\text{minimize} && f(\boldsymbol{x}) \\ &\text{subject to} && \boldsymbol{x} \in S. \end{aligned} \tag{1}$$

Function $f : \mathbb{R}^n \to \mathbb{R}$ to be minimized is called an *objective function*. If the objective function f is to be maximized then it is equivalent to minimize the function $-f$. Minimization of the objective function is performed with respect to *decision variables* $\boldsymbol{x} \in \mathbb{R}^n$. The decision variables $\boldsymbol{x}$ form a *decision vector* belonging to the *decision space* $\mathbb{R}^n$. Usually, all the decisions in the decision space are not necessarily acceptable and we call the acceptable subset $\emptyset \neq S \subset \mathbb{R}^n$ the *feasible region*. The feasible region can be, for example, of the form

$$S = \{\boldsymbol{x} \in \mathbb{R}^n \mid g_i(\boldsymbol{x}) \leq 0,\ h_j(\boldsymbol{x}) = 0,\ \boldsymbol{x}^l \leq \boldsymbol{x} \leq \boldsymbol{x}^u\}, \tag{2}$$

where $g_i : \mathbb{R}^n \to \mathbb{R},\ i = 1, \ldots, m_g$, are *inequality constraints*, $h_j : \mathbb{R}^n \to \mathbb{R},\ j = 1, \ldots, m_h$, are *equality constraints* and $\boldsymbol{x}^l, \boldsymbol{x}^u \in \mathbb{R}^n$ are *lower* and *upper bounds* for the decision variables, respectively. In this thesis, we consider only optimization problems with continuous decision variables as mentioned before.

Solutions of problem (1) can be categorized in to local and global minima.

Definition 2.1.1
A solution $\boldsymbol{x}^* \in S$ *is a* local minimum *of problem (1) if there exists a neighborhood* $N(\boldsymbol{x}^*)$ *such that* $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x})$ *for all* $\boldsymbol{x} \in N(\boldsymbol{x}^*)$. *If* $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x})$ *for all* $\boldsymbol{x} \in S$, *then* $\boldsymbol{x}^*$ *is a* global minimum.

If the optimization problem has more than one objective function, it can be formulated as a multiobjective optimization problem

$$\begin{aligned} &\text{minimize} && \{f_1(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})\} \\ &\text{subject to} && \boldsymbol{x} \in S. \end{aligned} \tag{3}$$

Again, functions $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, k$, $k \geq 2$, to be minimized are called *objective functions* or *criteria*. In this thesis, we will use both the terms. The space $\mathbb{R}^k$, where the objective functions have their values, is called the *objective space* and the vectors in the objective space, $\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))^T$, are called *objective vectors*. The feasible region S can be defined in the same way as in (2).

In single objective optimization, the solution $\boldsymbol{x}^*$ of problem (1) gives the smallest value of the objective function in the feasible region. If we have more than one objective function, the definition of optimality must be redefined because individual objective functions usually attain their minima at different points $\boldsymbol{x} \in S$. The concept of optimality often used in multiobjective optimization is called *Pareto optimality*.

Definition 2.1.2
A decision vector $\boldsymbol{x}^* \in S$ *is called a* Pareto optimal *solution if there does not exist another decision vector* $\boldsymbol{x} \in S$ *such that* $f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{x}^*)$ *for all* $i = 1, \ldots, k$ *and* $f_j(\boldsymbol{x}) < f_j(\boldsymbol{x}^*)$ *for at least one* j. *Furthermore, an objective vector* $\boldsymbol{f}(\boldsymbol{x}^*)$ *is called Pareto optimal if the corresponding decision vector* $\boldsymbol{x}^*$ *is Pareto optimal.*

The above definition means that the solution $\boldsymbol{x}^*$ is Pareto optimal if no criterion can be improved without impairing some other criterion. All the Pareto optimal solutions form a *Pareto optimal set*. Sometimes, Pareto optimal solutions are also called *non-dominated* or *efficient* [88]. In this thesis, we assume that there exist Pareto optimal solutions (according to corollary 3.2.1 in [88]). An illustrative example of Pareto optimality is shown in Figure 1 where the Pareto optimal set is denoted with a bold line. Note, that all the Pareto optimal solutions are mathematically equivalent, in other words, all the feasible solutions of multiobjective optimization problems have only partial ordering.

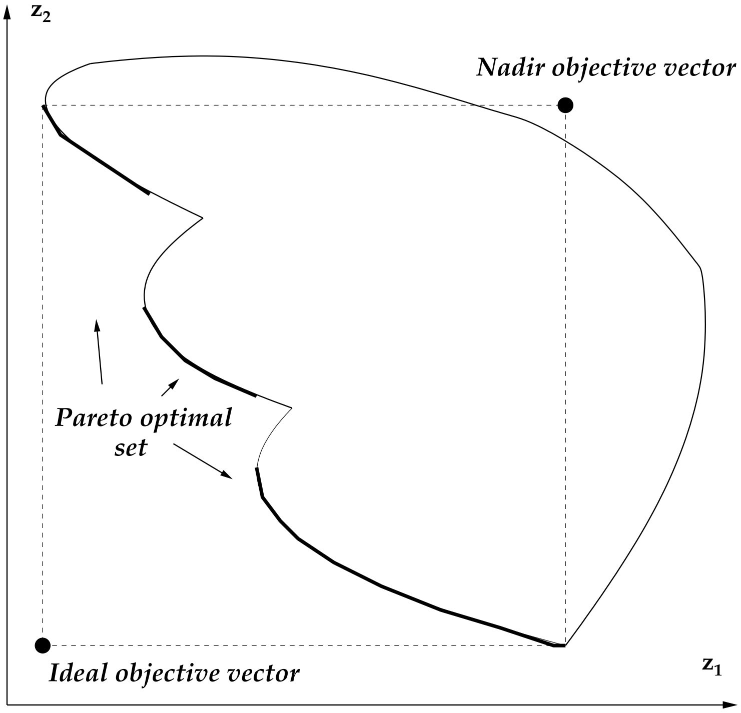


FIGURE 1 An example of a Pareto optimal set.

Definition 2.1.2 is for global Pareto optimality. In a similar way than we categorized global and local optima for the single objective optimization problem (1), local Pareto optimal solutions can be defined.

Definition 2.1.3
A decision vector $\boldsymbol{x}^* \in S$ *is* locally Pareto optimal *if there exists a neighborhood* $N(\boldsymbol{x}^*)$ *of* $\boldsymbol{x}^*$ *such that* $\boldsymbol{x}^*$ *is Pareto optimal in* $N(\boldsymbol{x}^*) \cap S$. *The objective vector* $\boldsymbol{f}(\boldsymbol{x}^*)$ *is locally Pareto optimal if the corresponding point* $\boldsymbol{x}^*$ *is locally Pareto optimal.*

There exist also other concepts of optimality like so-called *weak Pareto optimality*.

Definition 2.1.4
A decision vector $\boldsymbol{x}^* \in S$ *is called a* weakly Pareto optimal *solution if there does not exist another decision vector* $\boldsymbol{x} \in S$ *such that* $f_i(\boldsymbol{x}) < f_i(\boldsymbol{x}^*)$ *for all* $i = 1, \ldots, k$. *Furthermore, an objective vector* $\boldsymbol{f}(\boldsymbol{x}^*)$ *is called weakly Pareto optimal if the corresponding decision vector* $\boldsymbol{x}^*$ *is weakly Pareto optimal.*

Weak Pareto optimality means that there can be solutions where some of the objectives (but not all of them) might be improved without impairing any other one. Note that a Pareto optimal solution is always weakly Pareto optimal but a weakly Pareto optimal solution is not necessarily Pareto optimal. In other words, the Pareto optimal set is a subset of weakly Pareto optimal solutions. Weak Pareto optimality must be acknowledged because some multiobjective optimization methods can produce only weakly Pareto optimal solutions.

2.2 Some concepts in multiobjective optimization

To totally order mathematically equivalent Pareto optimal solutions in order to find the best compromise solution for the problem in question, we need some extra information. A *decision maker* is a person who is able to express preference information about the problem to be considered and, especially, about the different conflicting criteria. It is assumed that less is preferred to more to the DM when the objective functions are minimized. Assuming that the problem is correctly specified and the DM is rational, we can concentrate on the Pareto optimal solutions. In chemical process design, the DM is called a *designer*. In this thesis, we will use either the term decision maker or designer depending on the context. In any case, we mean the same person.

The concepts often used in multiobjective optimization problems are the *ideal objective vector* $\mathbf{z}^*$, *utopian objective vector* $\mathbf{z}^{**}$ and the *nadir objective vector* $\mathbf{z}^{nad}$ (see, for example, [64]). These vectors in the objective space are used to give ranges for objective function values in the Pareto optimal set. The ith component of the ideal objective vector z_i^* is the minimal value of the objective function f_i, that is, a minimum when f_i is minimized with respect to the set S. Note, that if the objective functions are conflicting, then the ideal objective vector is not feasible.

The utopian objective vector is slightly better than the ideal objective vector and, thus, also infeasible. However, it is used in some multiobjective optimization methods for computational reasons. The utopian objective vector for minimization problems is defined, for example, by $z_i^{**} = z_i^* - \varepsilon$, where z_i^{**} is the ith component of the utopian objective vector and $\varepsilon > 0$ is a small number. Both the ideal and the utopian objective vector form lower bounds for the Pareto optimal set.

The nadir objective vector gives upper bounds for the objective function values in the Pareto optimal set. These upper bounds can not be calculated exactly in general, and thus, are only approximations. A widely used method for approximating the nadir objective vector is to use the *payoff table* [64]. This table consists of values of all the objective functions calculated at the points where the values for the ideal objective vector were achieved. The worst value of the objective function f_i in this table is selected for the ith component of the nadir objective vector, that is, z_i^{nad}. However, this approximation can be very poor, either too low or too high. Note, that if $k = 2$, the payoff table gives an exact nadir objective vector. An example of ideal and nadir objective vectors is given in Figure 1.

A *reference point* $\bar{\mathbf{z}} = (\bar{z}_1, \ldots, \bar{z}_k)^T$ consists of aspiration levels $\bar{z}_i$ for each objective function f_i. The aspiration levels are given by the DM and they represent desired levels for each objective function.

2.3 Methods of multiobjective optimization

In this work, solving a multiobjective optimization problem means finding a Pareto optimal solution that best fulfils the requirements or hopes of the DM. Usually, it is not possible or reasonable to generate the whole Pareto optimal set, but it can be approximated by generating a suitable selection of Pareto optimal solutions. Pareto optimal solutions are usually produced by scalarizing the multiobjective optimization problem (3) into single objective optimization problem(s) according to the preferences of the DM and then solving it by utilizing the methods of single objective optimization. The scalarization depends on the multiobjective optimization method used.

Evolutionary multiobjective optimization (see, for example, [19]) uses evolutionary techniques in solving multiobjective optimization problems. In these methods, the original problem is not scalarized and there is a population of solutions that is manipulated. Operations having their origin in the evolution process of nature (selection, cross-over and mutation) are applied to members of the population in order to find better solutions. The idea is to approximate, that is, to cover the Pareto optimal set with the population. These methods do not utilize gradient information, but only the objective and constraint function values and, therefore, they work well for mathematically challenging problems where the computation of gradients is difficult or the gradients obtained could be inaccurate. Evolutionary multiobjective optimization methods are not considered in this thesis because they can not guarantee the Pareto optimality of the solutions obtained. In addition, they are best suited for only two objective functions because they often use lots of function evaluations and the number of function evaluations required increases significantly with the number of objectives. These methods can be very time consuming when applied to real-world problems with more than two objective functions.

Many different solution methods have been developed for multiobjective optimization (see, for example, [43, 64, 89]). Solution methods can be divided into four categories according to the role of the DM in the solution procedure [43, 64]. These categories are for methods that do not involve a DM, a posteriori methods, a priori methods and interactive methods. A posteriori and a priori methods take the preferences of the DM into account after and before the problem is solved, respectively, while interactive methods require interaction between the method and the DM. This division of the methods is not unique because some methods can be considered to belong to different categories depending on how they are used.

Next, the four categories are introduced very briefly and some examples of the methods are presented. The presentation is based on the book [64] where more information about different methods can be found.

2.3.1 No-preference methods

No-preference methods do not utilize the DM's expertise at all. Instead, some simple method is used to produce a Pareto optimal solution which is then accepted or not. These methods are used when there is no DM available. The benefit of these methods is that they are fast but the main drawback is that the solution obtained may be very bad for the problem in question because it does not take into account the DM's preferences. An example of no-preference methods is the *method of global criterion* [107, 109] where the distance to the ideal objective vector is minimized.

A special solution related to ideal and nadir objective vectors is the *neutral compomise solution* introduced in [102]. The neutral compromise solution is approximately in the middle of the Pareto optimal set. By "in the middle" we mean that the distance is measured by the *min-max metric*, that is, the neutral compromise solution is defined as the solution of problem

$$\begin{aligned} &\text{minimize} \quad \max_{i=1,\ldots,k} \left[\frac{f_i(\boldsymbol{x}) - z_i^{mid}}{z_i^{nad} - z_i^{**}} \right] + \rho \sum_{i=1}^{k} \frac{f_i(\boldsymbol{x}) - z_i^{mid}}{z_i^{nad} - z_i^{**}} \\ &\text{subject to} \quad \boldsymbol{x} \in S, \end{aligned} \tag{4}$$

where $z_i^{mid} = (z_i^{nad} + z_i^{**})/2$ is used as a reference point. The latter part of the objective function in (4) is used to guarantee Pareto optimality and not merely weak Pareto optimality of the solution and the parameter ρ must be strictly positive. See [64] for details. Note that the neutral compromise solution does not involve any preferences of the DM. Information about different neutral compromise solutions can be found in [102].

2.3.2 A posteriori methods

In *a posteriori methods*, the Pareto optimal set or a subset of Pareto optimal solutions is generated and presented to the DM. Then the DM selects the best solution representing her or his preferences. A posteriori methods are well suited for problems with two objectives because the solutions representing the Pareto optimal set can be easily visualized which makes the selection of the most preferred solution easier to the DM. The drawbacks of these methods are that the generation of the Pareto optimal set or a part of it is often quite time consuming and difficult. Furthermore, the DM has to select from a large number of candidates which may be cognitively demanding. (It is not obvious how to present k-dimensional objective vectors to the DM.) Widely used a posteriori methods are the *weighting method* [25, 108] where the weighted sum of the objective functions is optimized and the *ε-constraint method* [14, 33] where one of the objective functions is optimized while the rest of the objective functions are converted into inequality constraints by introducing bounds for them. Different Pareto optimal solutions are obtained by changing the weights and the upper bounds for new inequality constraints for the weighting method and the ε-constraint method, respectively.

2.3.3 A priori methods

A priori methods ask the preferences of the DM before the method is applied to the problem. Then, according to the preference information given by the DM, the method formulates a single objective optimization problem. This optimization problem is then solved, and the solution obtained is supposed to represent the preferences of the DM as well as possible. The quality of the solution depends on the problem in question and, also, on the method used. A benefit of a priori methods is that the preferences of the DM can be taken into account when Pareto optimal solutions are generated. Difficulties in these methods are that it may be difficult for the DM to express the preference information because (s)he does not know beforehand what is possible to achieve and, therefore, can be disappointed with the solution obtained. *Lexicographic ordering* [24] and *goal programming* [15, 16, 17] are examples of a posteriori methods. In lexicographic ordering, the DM defines a strict preference order for the objective functions that are then optimized in the same order. In goal programming, the DM specifies a desired aspiration level for each objective function and the idea is to minimize the deviation between the objective function value and the aspiration level.

2.3.4 Interactive methods

Interactive methods [64] require iterative co-operation between the method and the DM. Usually, at first some initial Pareto optimal solution is generated and presented to the DM. This can be, for example, the neutral compromise solution. Then, the DM expresses his or her opinion on which way the current solution should be improved or gives some other information required by the method used. Based on this information, Pareto optimal solution(s) are generated, and the solution(s) obtained are presented to the DM. This iterative solution procedure continues until the DM is satisfied with some solution or does not want to continue any more.

A benefit of interactive methods is that the DM can guide the solution procedure and is able to learn about the behaviour of the problem. (S)he can study the interrelationships of the objective functions and obtain a wider understanding of their effects on the whole problem. Therefore, as the DM gains more and more understanding of the behaviour of the problem, (s)he can adjust his or her preferences accordingly. Another benefit is that interactive methods are computationally efficient, because quite a small number of Pareto optimal solutions is usually needed to compute. Furthermore, the DM can concentrate only on those solutions that are of interest. On the other hand, the drawback of interactive methods is that they require the DM to take part in the solution procedure actively and, thus, the DM has to be willing to spend his or her time.

Different interactive methods differ in how the solution information is shown to the DM, how the DM expresses his or her preferences and how the problem is scalarized. Examples of ways to express preferences are the classification of the objective functions and the reference points [64, 99, 102]. This type of preference information is utilized, for example, in the interactive NIMBUS® method to be presented next.

2.4 NIMBUS® method

NIMBUS® is an interactive multiobjective optimization method for nonlinear problems [64, 67, 68, 72]. NIMBUS® has been used to solve computationally demanding multiobjective optimization problems, for example, the optimal design of the paper machine headbox [36], the optimal control problem in the continuous casting of steel [74], and a structural design problem [73]. There exists several versions of the NIMBUS® method and the synchronous version [72] is used in this research.

NIMBUS® is based on the classification of the objective functions into up to five classes [67]. The classification is made at the current Pareto optimal objective vector $\boldsymbol{f}(\boldsymbol{x}^c)$ which is shown to the DM. By classifying the objective functions the DM indicates how the current solution should be improved. In the classes the value of the objective function f_i

- should be improved as much as possible ($i \in I^{imp}$),
- should be improved until some specified aspiration level $\bar{z}_i$ ($i \in I^{asp}$),
- is satisfactory at the moment ($i \in I^{sat}$),
- can impair till some specified bound ε_i ($i \in I^{bound}$) and
- can change freely ($i \in I^{free}$).

Note that $I^{imp} \cup I^{asp} \cup I^{sat} \cup I^{bound} \cup I^{free} = \{1, \ldots, k\}$. In the sense of Pareto optimality the classification is acceptable if $I^{imp} \cup I^{asp} \neq \emptyset$ and $I^{bound} \cup I^{free} \neq \emptyset$, because one must let some objective function to impair in order to improve some other objective function. Aspiration levels $\bar{z}_j$, $j \in I^{asp}$, and upper bounds ε_i, $i \in I^{bound}$, are asked from the DM, if these classes are used.

According to the classification by the DM, the original multiobjective optimization problem is converted into several single objective optimization problems, so-called subproblems in the synchronous version of the NIMBUS® method [72]. Various subproblems are used because there is no unique way to use classification information in multiobjective optimization problems [71] and it is the DM who is the best expert to say which solution follows the preference information specified in the most satisfactory way. New Pareto optimal solution candidates are obtained by solving these subproblems. The subproblems can be solved by any suitable single objective optimizer.

There are four different subproblems used in the synchronous version of NIMBUS®, the standard subproblem formulated by the developers of the NIMBUS® method and three subproblems based on reference points. In all the subproblems, the latter part of the objective function is called the *augmentation term* and it guarantees that the solution obtained is Pareto optimal as mentioned before in the context of the neutral compromise solution (see [64] and [72]). The augmentation parameter $\rho > 0$ is a small coefficient. The standard (STD) subproblem is of the form

$$\begin{aligned} \text{minimize} \quad & \max_{i \in I^{imp},\, j \in I^{asp}} \left[\frac{f_i(\boldsymbol{x}) - z_i^*}{z_i^{nad} - z_i^{**}}, \frac{f_j(\boldsymbol{x}) - \bar{z}_j}{z_j^{nad} - z_j^{**}} \right] + \rho \sum_{i=1}^{k} \frac{f_i(\boldsymbol{x})}{z_i^{nad} - z_i^{**}} \\ \text{subject to} \quad & f_i(\boldsymbol{x}) \le f_i(\boldsymbol{x}^c) \text{ for all } i \in I^{imp} \cup I^{asp} \cup I^{sat}, \\ & f_i(\boldsymbol{x}) \le \varepsilon_i, \text{ for all } i \in I^{bound}, \\ & \boldsymbol{x} \in S. \end{aligned} \tag{5}$$

The three subproblems based on reference points are taken from the literature, and selected according to the comparison in [71]. These subproblems are based on the satisficing trade-off method [77, 79, 80, 81], the achievement (scalarizing) functions [99, 100, 101] used in the reference point method and the GUESS method [12]. These three additional subproblems are very similar in structure although it has been shown [71] that they typically generate different enough solutions. The formulations of these subproblems can be found in [72]. From the classification information provided by the DM, we can form a reference point $\hat{\boldsymbol{z}} \in \mathbb{R}^k$, where $\hat{z}_i = z_i^*$ for $i \in I^{imp}$, $\hat{z}_i = \bar{z}_i$ for $i \in I^{asp}$, $\hat{z}_i = f_i(\boldsymbol{x}^c)$ for $i \in I^{sat}$, $\hat{z}_i = \varepsilon_i$ for $i \in I^{bound}$, and $\hat{z}_i = z_i^{nad}$ for $i \in I^{free}$.

The number of subproblems used can be changed during the optimization procedure, that is, the DM can select the number of different solutions to be generated after each classification. For example, some problems can be computationally so demanding that the usage of all the subproblems is not practical.

In NIMBUS®, there is also a possibility to generate intermediate Pareto optimal solutions between any two Pareto optimal solutions. These intermediate solutions are generated by computing objective vectors at the decision variables that are equally divided between the selected two Pareto optimal solutions. These solutions are not necessarily Pareto optimal but they are always projected into the set of Pareto optimal solutions by applying some of the reference point based subproblems.

A simplified flowchart representing the NIMBUS® algorithm is presented in Figure 2. The optimization procedure starts with the computation of a neutral compromise solution (4) which is approximately in the middle of the Pareto optimal set as described before. The iteration procedure starts with the classification at the current solution. After the method has produced new solution candidates corresponding to the classification, the DM can generate intermediate solutions between them or select one of them as the basis of the next classification. Note that different subproblems do not necessarily produce different solutions. In that case, only the different ones are shown to the DM. If the DM is satisfied with the solution obtained, the optimization procedure stops. Otherwise, (s)he can make another classification. The promising solutions obtained during the solution procedure can be stored in a database, so that the DM can select also solutions previously obtained.

All the subproblems in the synchronous NIMBUS® method are of min-max type and, thus, are always nonsmooth independently of the properties of the functions in the original multiobjective optimization problem. To solve the subproblems we, therefore, need a nonsmooth optimizer, for example, the proximal bundle optimizer [63] or an optimizer that does not assume differentiability (see, for example, [27]). However, we can obtain smooth variants of the subproblems quite easily if the original functions in the problem are differentiable. Smooth variants enable us to select the single objective optimizer used from a wider class of smooth optimizers to meet the special requirements often present in real-world optimization problems. As an example, we present a smooth variant of the STD subproblem:

$$
\begin{array}{ll}
\underset{\boldsymbol{x},\delta}{\text{minimize}} & \delta + \rho \sum\limits_{i=1}^{k} \frac{f_i(\boldsymbol{x})}{z_i^{nad} - z_i^{**}} \\
 & \\
\text{subject to} & \frac{f_i(\boldsymbol{x}) - z_i^{*}}{z_i^{nad} - z_i^{**}} \leq \delta, \quad i \in I^{imp}, \\
 & \frac{f_j(\boldsymbol{x}) - \bar{z}_j}{z_j^{nad} - z_j^{**}} \leq \delta, \quad j \in I^{asp}, \\
 & f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{x}^c) \text{ for all } i \in I^{imp} \cup I^{asp} \cup I^{sat}, \\
 & f_i(\boldsymbol{x}) \leq \varepsilon_i, \text{ for all } i \in I^{bound}, \\
 & \boldsymbol{x} \in S, \quad \delta \in \mathbb{R}.
\end{array}
\tag{6}
$$

When compared to the original STD subproblem (5), we have introduced a new variable δ in the objective function. As a result, we have obtained a number of additional inequality constraints involving δ. Note, that we are now optimizing with respect to both $\boldsymbol{x}$ and δ.

2.5 IND-NIMBUS

IND-NIMBUS [82] is an implementation of the NIMBUS® method for solving (industrial) nonlinear multiobjective optimization problems (`http://ind-nimbus.it.jyu.fi/`). IND-NIMBUS provides a graphical user interface (GUI) for the designer to participate in the solution procedure and to compare and visualize the Pareto optimal solutions obtained. IND-NIMBUS can consider several conflicting objectives or performance criteria simultaneously and offers an interactive solution procedure based on the NIMBUS® method. Let us mention that IND-NIMBUS can handle objective functions to be either minimized or maximized. We have used IND-NIMBUS to solve all the chemical process design problems

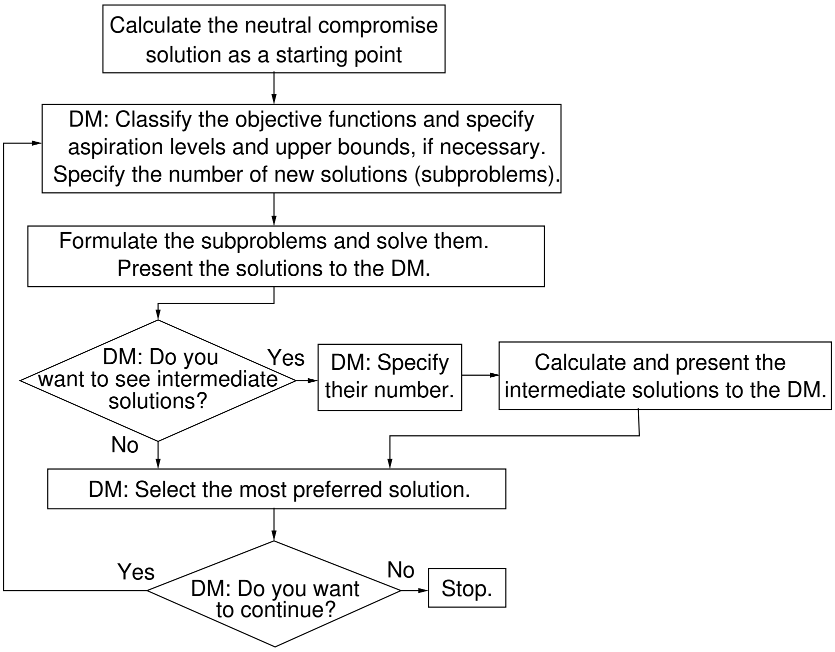

Calculate the neutral compromise solution as a starting point
DM: Classify the objective functions and specify aspiration levels and upper bounds, if necessary. Specify the number of new solutions (subproblems).
Formulate the subproblems and solve them. Present the solutions to the DM.
DM: Do you want to see intermediate solutions?
Yes
DM: Specify their number.
Calculate and present the intermediate solutions to the DM.
No
DM: Select the most preferred solution.
Yes
DM: Do you want to continue?
No
Stop.

FIGURE 2 A flowchart of the NIMBUS® algorithm.

considered in this thesis.

Our goal is that IND-NIMBUS could easily be connected to different simulators and modelling tools. Currently, IND-NIMBUS has been connected to the BALAS® process simulator [II, 34], virtual paper machine [60, 61], and the Numerrin modelling system [39]. In addition, we want to be able to use different single objective optimizers to solve the subproblems produced by the NIMBUS® method. So far, we have used the nonsmooth proximal bundle optimizer [63] (in [I, II, III]), a real-coded genetic algorithm with two variants of constraint handling [75] (in [35]) and a large-scale nonlinear optimization package IPOPT [95, 96] (in [IV, V]). The local proximal bundle optimizer is developed for nonsmooth optimization problems and it is based on approximating the subdifferential of the objective function by using subgradients [62, 63]. IPOPT is a local large-scale nonlinear optimization package based on a Newton-based interior point (barrier) algorithm with a filter line-search method and it is used to solve smooth single objective optimization problems. The ability to connect different optimizers to IND-NIMBUS enables the usage of specialized optimizers tailored for the needs of the problem in question.

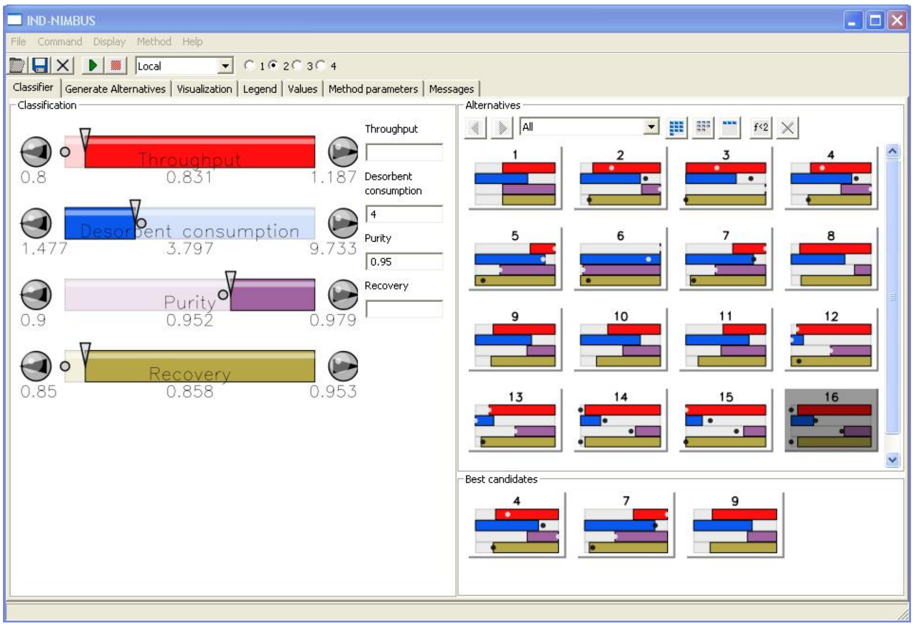
The GUI of IND-NIMBUS consists of tabs for different parts of the solution procedure. For example, there are different tabs for classification, generating intermediate solutions, visualization of the Pareto optimal solutions obtained, showing values of the objective functions and the decision variables of the solutions, changing parameters of the underlying optimizers as well as the NIMBUS® method and for the messages produced by the optimization methods. Screenshots of different tabs are shown in Figures 3–6. Next, we briefly describe four of the tabs, namely the classification, the intermediate solution generation, the visualization and the numerical values tabs. Above the tabs, there is a toolbar for controlling the solution procedure, for example, starting and stopping calculation and selecting the underlying single objective optimizers used.

The main tab is used for classification (Figure 3). This tab consists of three parts: the current Pareto optimal solution, in which the classification is made, is presented in the left side of the tab, the solutions obtained during the solution procedure so far are shown in the upper right corner of the tab and the best candidates for the final solution (selected by the designer) are presented in the lower right corner. Each objective function is represented by a colored bar. General idea is that the shorter the colored bar, the better is the corresponding objective function value. Note, that the bars for the objective functions to be minimized start from left while the bars for the objective functions to be maximized start from right. The values at the left and right sides of each bar represent the ranges of the Pareto optimal set. The designer needs to select the solution which (s)he wants to improve from the set of already obtained solutions. Then (s)he can perform the classification by clicking the bars representing the objective function values of the current solution in the upper left corner of the tab. The numerical fields in the right side of the bars can be used to give aspiration levels and bounds required. Clicking different parts of the bars corresponds to the classification presented in Section 2.4 so this is an implementation of the classification in NIMBUS®.

In the classification, the designer tries to improve the objective functions that (s)he believes are not satisfactory (larger or smaller values are preferred depending on whether the objective function is to be minimized or maximized). Note that due to the Pareto optimality the classification is acceptable if the designer is willing to sacrifice in some objective functions in order to gain in some other objective functions [72]. After the designer starts the calculation, new solution candidates will appear into the set of solutions obtained. The designer can put the solutions that seem to be good at the moment in the set *best candidates* with drag and drop. This set is used for helping the designer to keep track of the best solutions obtained so far. The number of new Pareto optimal solutions generated as well as the optimizer used can be adjusted from the toolbar.

The tab for generating intermediate solutions between any two chosen solutions is presented in Figure 4. This tab is divided into four parts: all the solutions obtained and the best candidates for the final solution are shown in the upper and the lower right corners of the tab, respectively, while the two chosen end solutions and the intermediate solutions produced are shown in the upper and the lower left corner of the tab, respectively. To generate intermediate solutions [72], the designer needs to drag and drop two solutions (end points) already obtained to the upper left corner of the tab, choose the number of intermediate solutions to be produced and start the calculation. When the intermediate solutions are produced, they will appear to the pool of the intermediate solutions in the lower left corner one by one.

The tab for visualizing the Pareto optimal solutions obtained is shown in Figure 5. The aim of visualization is to give support for the designer in comparing different Pareto optimal solutions and to help him/her obtain more information



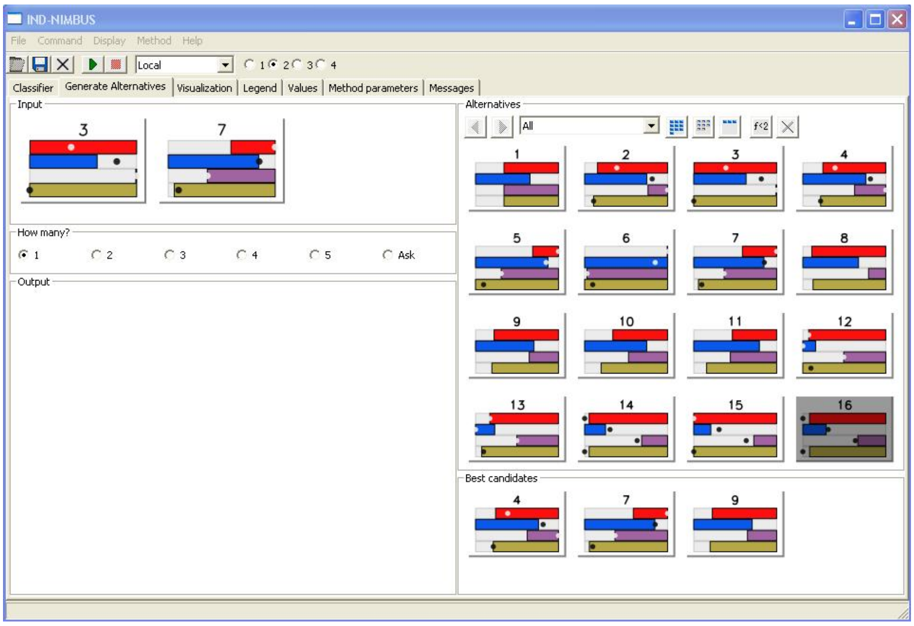
Throughput

Desorbent consumption 4

Purity 0.95

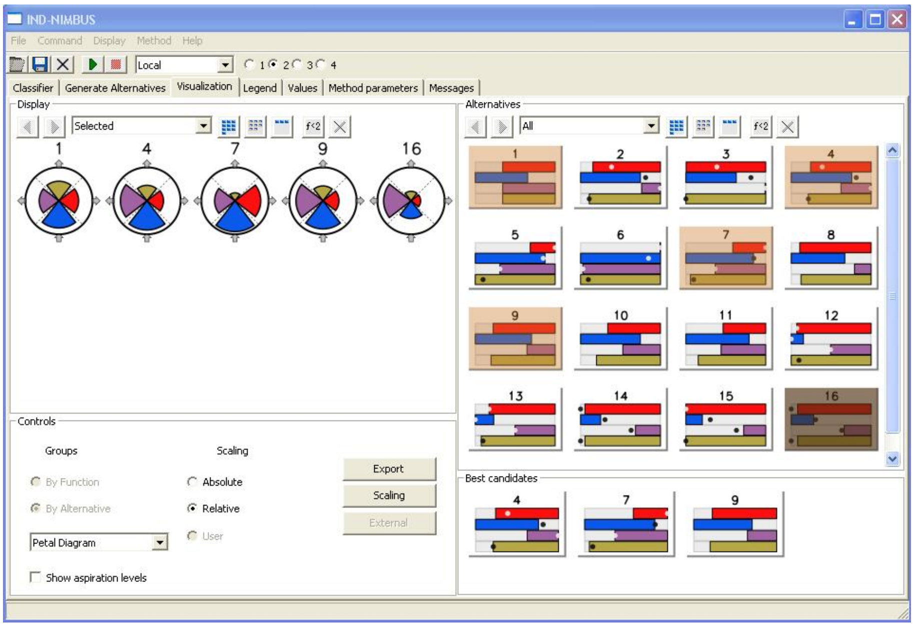
Recovery

FIGURE 3 A screenshot of IND-NIMBUS: classification tab.


IND-NIMBUS
File Command Display Method Help
Local
○ 1 ● 2 ○ 3 ○ 4
Classifier | Generate Alternatives | Visualization | Legend | Values | Method parameters | Messages
Input
3 7
Output
Alternatives
All
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Best candidates
4 7 9

How many? ● 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ Ask

FIGURE 4 A screenshot of IND-NIMBUS: intermediate solutions tab.



Groups

○ By Function
● By Alternative

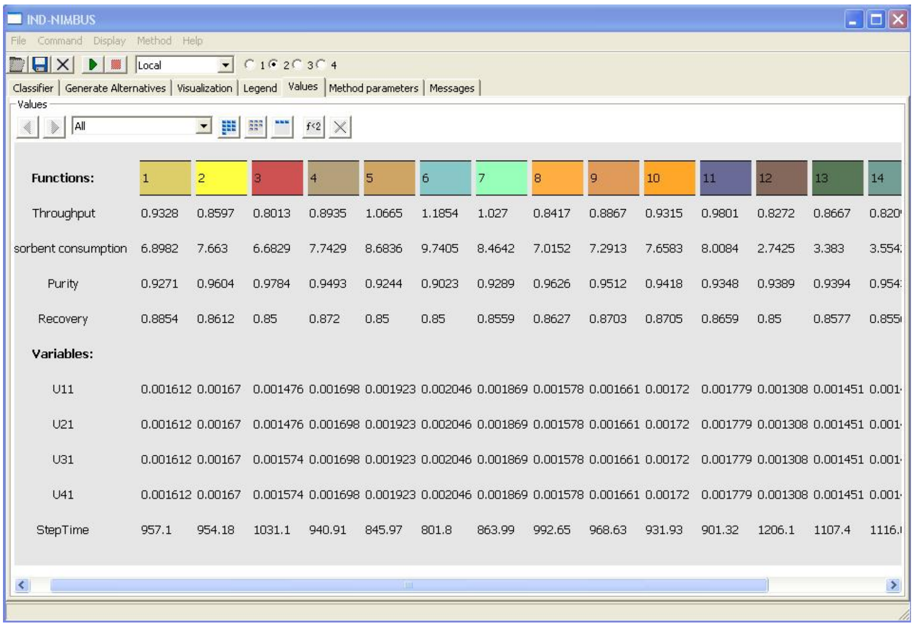
Scaling

○ Absolute
● Relative
○ User

Petal Diagram

□ Show aspiration levels

FIGURE 5 A screenshot of IND-NIMBUS: visualization tab.



Functions:	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Throughput	0.9328	0.8597	0.8013	0.8935	1.0665	1.1854	1.027	0.8417	0.8867	0.9315	0.9801	0.8272	0.8667	0.820
sorbent consumption	6.8982	7.663	6.6829	7.7429	8.6836	9.7405	8.4642	7.0152	7.2913	7.6583	8.0084	2.7425	3.383	3.554
Purity	0.9271	0.9604	0.9784	0.9493	0.9244	0.9023	0.9289	0.9626	0.9512	0.9418	0.9348	0.9389	0.9394	0.954
Recovery	0.8854	0.8612	0.85	0.872	0.85	0.85	0.8559	0.8627	0.8703	0.8705	0.8659	0.85	0.8577	0.855
Variables:														
U11	0.001612	0.00167	0.001476	0.001698	0.001923	0.002046	0.001869	0.001578	0.001661	0.00172	0.001779	0.001308	0.001451	0.001
U21	0.001612	0.00167	0.001476	0.001698	0.001923	0.002046	0.001869	0.001578	0.001661	0.00172	0.001779	0.001308	0.001451	0.001
U31	0.001612	0.00167	0.001574	0.001698	0.001923	0.002046	0.001869	0.001578	0.001661	0.00172	0.001779	0.001308	0.001451	0.001
U41	0.001612	0.00167	0.001574	0.001698	0.001923	0.002046	0.001869	0.001578	0.001661	0.00172	0.001779	0.001308	0.001451	0.001
StepTime	957.1	954.18	1031.1	940.91	845.97	801.8	863.99	992.65	968.63	931.93	901.32	1206.1	1107.4	1116.

FIGURE 6 A screenshot of IND-NIMBUS: the tab presenting the function and variable values.

on interrelationships of the objective functions. Again, this tab is divided into three parts: the left side of the tab is reserved for the visualization of the objective function values of the selected solutions while all the solutions and the best candidates are shown in the right side of the tab, respectively.

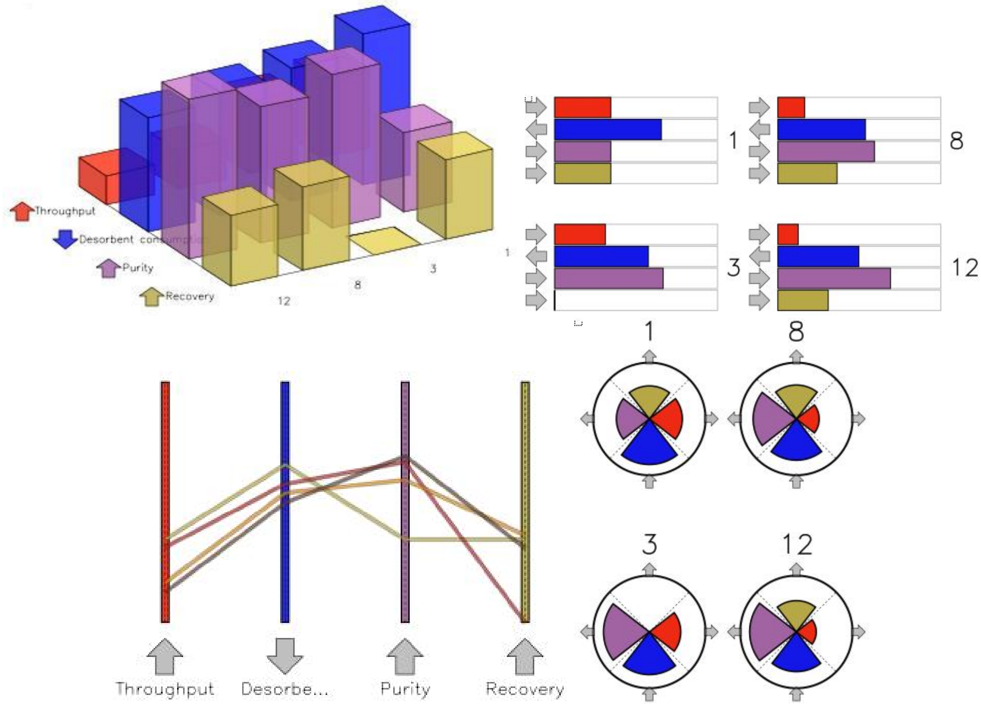


FIGURE 7 Different visualizations of solutions obtained by IND-NIMBUS.

There are several types of visualizations available and the designer can choose the one (s)he finds most suitable. The designer selects the solutions to be visualized. The solutions can be visualized with absolute or relative values of the objective functions. In some visualizations, the solutions can be grouped either by criteria or alternative, that is, one can view either the solutions obtained with the objective function values or the objective function values with all the solutions. Relative values mean here that they are scaled between the lower and the upper bounds of the Pareto optimal solution set [64]. Figure 7 shows examples of different types of visualizations that IND-NIMBUS can offer. The visualizations shown are *3d bars*, *barcharts*, *value paths*, and *petal diagrams*, respectively. Other visualizations offered by IND-NIMBUS are *whisker plot*, *multiway dot plot*, and *spider web*. More information about visualizations in IND-NIMBUS can be found in [65, 66].

The values tab shown in Figure 6 presents the values of the objective functions and the decision variables. Each Pareto optimal solution forms one column in the table. Note that the number of decimals shown can be adjusted.

2.6 Supporting the decision maker

Currently, IND-NIMBUS has a couple of ways to support the decision maker during the interactive solution procedure. First of all, the lower and the upper bounds of the Pareto optimal set are always shown to the DM in order to aid in classification and in specifying aspiration levels and bounds. Secondly, the DM can explore Pareto optimal solutions already obtained and compare them with the help of different visualizations. In addition to already available support, we are interested in supporting the DM in what kind of classifications (s)he should make in order to find the most preferred compromise solution without taking too much of his/her time. It is found, that in practice the DMs do not want to spent too much time in the solution procedure although they consider interactive solution procedure useful [55]. By generating new solution(s) quickly and reducing the number of iterations required, we can ease the DM's role in the solution procedure and best utilize good properties of the interactive solution procedure.

One possible support for the DM in selecting the next classification/reference point is trade-off information [14, 31]. The concept of trade-off is commonly used in the context of multiobjective optimization because Pareto optimal solutions are mathematically incomparable and one has to sacrifice in some criteria in order to gain in some other criteria and that is called trading-off. In this thesis, we study trade-off information and how it could be utilized in supporting the DM [VI]. We do not want to develop a new interactive method but to support the usage of already available methods. In addition, it is important that supporting information can be obtained without additional computational cost especially for real-world problems.

Trade-off information does not involve any preference information from the DM, but it describes interdependencies between different objective functions and how their values change with respect to others. Pareto optimal solutions for problem (3) can be produced, for example with the reference point method [100]. In this thesis, we study trade-off information in the context of reference point based scalarization. Reference points are closely related to classification. From the classification information provided by the DM a reference point can be obtained. In other words, if the DM makes a classification, for example, with IND-NIMBUS, we can formulate a reference point with the help of the classification information and then use a reference point based method to produce a Pareto optimal solution. Note that the synchronous NIMBUS® formulates three reference point based subproblems as mentioned before. Because of the above mentiond connection between classification and reference points and because the classification based scalarizing function (5) includes so much information, using a reference point based scalarization gives us more simple starting point for this research.

A general reference point problem which is used, for example, in the reference point methods of Wierzbicki [100] can be written in the differentiable form

$$
\begin{aligned}
\underset{\boldsymbol{x},\delta}{\text{minimize}} \quad & \delta \\
\text{subject to} \quad & w_i(f_i(\boldsymbol{x}) - \bar{z}_i) \leq \delta, \quad i = 1, \ldots, k, \\
& \boldsymbol{x} \in S, \quad \delta \in \mathbb{R},
\end{aligned}
\tag{7}
$$

where $w_i > 0, i = 1, \ldots, k$ are the fixed weighting coefficients used for scaling and $\bar{z}_j$, $j = 1, \ldots, k$ are the aspiration levels of the reference point $\bar{\boldsymbol{z}}$. The weighting coefficients form a *weighting vector* $\boldsymbol{w} = (w_1, \ldots, w_k)^T$. Different Pareto optimal solutions are obtained by changing the reference point. This type of reference point problem is used also in the synchronous NIMBUS®

A widely studied scalarization was presented by Pascoletti and Serafini in [83]. In their scalarization, optimality is defined through a convex cone and the scalarization itself contains several parameters, namely a point $\boldsymbol{p}$ and a direction $\boldsymbol{q}$. By changing the values of the parameters all the optimal solutions with respect to the convex cone used can be found. The scalarization is quite general involving abstract function spaces and it is widely studied in theory. The general reference point problem presented above is a special case of this scalarization. The convex cone defining Pareto optimality is the nonnegative orthant $\mathbb{R}^k_- = \{\boldsymbol{z} \in \mathbb{R}^k \mid z_i \leq 0,\ i = 1, \ldots, k\}$. In addition, by selecting $\boldsymbol{p} = \bar{\boldsymbol{z}}$ and $\boldsymbol{q} = \boldsymbol{w}$, we obtain the problem (7).

To illustrate the ideas in this thesis, we consider the reference point problem based on the achievement (scalarizing functions) that are presented, for example, in [100]. If a basic variation of achievement functions is used, the following problem can be formulated:

$$
\begin{aligned}
\underset{\boldsymbol{x},\delta}{\text{minimize}} \quad & \delta \\
\text{subject to} \quad & \tfrac{f_i(\boldsymbol{x}) - \bar{z}_i}{z_i^{nad} - z_i^{**}} \leq \delta, \quad i = 1, \ldots, k, \\
& \boldsymbol{x} \in S, \quad \delta \in \mathbb{R}.
\end{aligned}
\tag{$P(\bar{\boldsymbol{z}})$}
$$

The parameters in this problem are the aspiration levels $\bar{z}_i$, $i = 1, \ldots, k$. Problem $(P(\bar{\boldsymbol{z}}))$ is obtained from (7) by introducing weighting coefficients $1/(z_i^{nad} - z_i^{**})$ corresponding a basic variation of achievement functions. Let $\bar{\boldsymbol{z}}^*$ be a reference point related to the solution $\boldsymbol{x}^*$. The following theorem can be formulated for the solutions of problem $(P(\bar{\boldsymbol{z}}^*))$ [64]:

Theorem 2.6.1
Let $\bar{\boldsymbol{z}}^* \in \mathbb{R}^k$ be a given reference point. Then, a unique solution $\boldsymbol{x}^*$ of problem $(P(\bar{\boldsymbol{z}}^*))$ is Pareto optimal.

Note that if we use local optimizer to solve problem (7), we can obtain only local Pareto optimal solutions. Next, we present some concepts related to trade-offs and our presentation follows [31]. *Pairwise trade-off* describing the changes of two objective functions f_i and f_j in two points $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ is defined by

$$T_{ij}(\boldsymbol{x},\hat{\boldsymbol{x}}) = \frac{f_i(\boldsymbol{x}) - f_i(\hat{\boldsymbol{x}})}{f_j(\boldsymbol{x}) - f_j(\hat{\boldsymbol{x}})}, \quad \boldsymbol{x},\hat{\boldsymbol{x}} \in S,$$

where $f_j(\boldsymbol{x}) \neq f_j(\hat{\boldsymbol{x}})$. If $f_l(\boldsymbol{x}) = f_l(\hat{\boldsymbol{x}})$ for all $l \neq i,j$, we call T_{ij} *partial trade-off*. On the other hand, if $f_l(\boldsymbol{x}) \neq f_l(\hat{\boldsymbol{x}})$ for at least one $l \neq i,j$, then T_{ij} is called *total trade-off*.

With the help of pairwise trade-offs, *total trade-off rate* at the point $\boldsymbol{x} \in S$ to direction $\boldsymbol{d}$ can be defined by

$$t_{ij}(\boldsymbol{x},\boldsymbol{d}) = \lim_{\alpha \to 0} T_{ij}(\boldsymbol{x} + \alpha\boldsymbol{d},\boldsymbol{x}),$$

where we assume that $\boldsymbol{d}$ is a feasible direction, that is, there exists $\alpha_0 > 0$ such that $\boldsymbol{x} + \alpha\boldsymbol{d} \in S$ for all $0 \leq \alpha \leq \alpha_0$. If $\boldsymbol{d}$ is a feasible direction such that there exists $\bar{\alpha} > 0$ satisfying $f_l(\boldsymbol{x} + \alpha\boldsymbol{d}) = f_l(\boldsymbol{x})$ for all $l \neq i,j$ for all $0 \leq \alpha \leq \bar{\alpha}$, then the corresponding t_{ij} is called a *partial trade-off rate*.

Note that in continuously differentiable case the total and partial trade-off rates can alternatively be defined by (see [64])

$$t_{ij}(\boldsymbol{x},\boldsymbol{d}) = \frac{\nabla f_i(\boldsymbol{x})^T\boldsymbol{d}}{\nabla f_j(\boldsymbol{x})^T\boldsymbol{d}}$$

and

$$t_{ij}(\boldsymbol{x}) = \frac{\partial f_i(\boldsymbol{x})}{\partial f_j},$$

respectively, where $\boldsymbol{x} \in S$, $\boldsymbol{d}$ is a feasible direction and $t_{ij}(\boldsymbol{x})$ denotes a partial trade-off rate.

In this thesis, we consider only trade-off rates and by trade-off information we mean trade-off rates. In addition, we study trade-off rates only at the Pareto optimal points. Here, for simplicity, we consider only unconstrained problems, so the feasible region S is now supposed to be $\mathbb{R}^n$. We assume also that the objective functions f_i, $i = 1,\ldots,k$, are twice continuously differentiable, that is, their derivatives and second derivatives with respect to $\boldsymbol{x}$ are continuous functions.

Problem $(P(\bar{\boldsymbol{z}}))$ is a special case of the hyperplane scalarization method introduced by Sakawa and Yano [87, 106]. In the hyperplane scalarization, partial trade-off rates can be obtained with the help of the Karush–Kuhn–Tucker (KKT) multipliers [50, 56]. Using KKT multipliers is a convenient way to obtain trade-off information because many single objective optimizers produce the optimal KKT multipliers without any additional computational effort, which was one of our goals. In this thesis, we use partial trade-offs computed with the method of Sakawa and Yano. Following the presentation of Sakawa and Yano [87] and under some assumptions, partial trade-off rates for the local Pareto optimal solution $(\boldsymbol{x}^*,\delta^*)^T$ of problem $(P(\bar{\boldsymbol{z}}^*))$ can be calculated from the formula

$$t_{ij}(\boldsymbol{x}^*) = -\frac{\lambda_j^*}{\lambda_i^*} \quad \text{for all } i,j = 1,\ldots,k, \tag{8}$$

where $\boldsymbol{\lambda}^*$ is the vector of the optimal KKT multipliers related to the solution $(\boldsymbol{x}^*, \delta^*)^T$ and the corresponding reference point $\bar{\boldsymbol{z}}^*$. Note that trade-offs do not explicitly depend on the reference levels but implicitly through the KKT multipliers. The theory behind the result (8) is based on the sensitivity theorem (see, for example, [23]) which in turn relies heavily on the implicit function theorem. The assumptions required are the following:

(i) $(\boldsymbol{x}^*, \delta^*)^T$ is a regular point of the constraints of problem $(P(\bar{\boldsymbol{z}}^*))$, that is, the gradients of the active inequality constraints are linearly independent,

(ii) the second order sufficient KKT optimality conditions are satisfied at $(\boldsymbol{x}^*, \delta^*)^T$ (see, for example, [23]), and

(iii) there are no degenerate constraints, that is, if $f_i(\boldsymbol{x}) = (z_i^{nad} - z_i^{**})\delta + \bar{z}_i$, then $\lambda_i > 0$.

By selecting a suitable direction $\boldsymbol{d}^*$ in the differentiable case of total trade-off rate, we can obtain a connection to the KKT multipliers. It can be shown [VI] that for the aspiration levels $\bar{z}_i^*$, $i = 1, \ldots, k$, related to the solution $(\boldsymbol{x}^*, \delta^*)^T$ we get

$$t_{ij}(\boldsymbol{x}^*, \boldsymbol{d}^*) = \frac{\nabla f_j(\boldsymbol{x}^*)^T \boldsymbol{d}^*}{\nabla f_i(\boldsymbol{x}^*)^T \boldsymbol{d}^*} = -\frac{\lambda_j}{\lambda_i}, \tag{9}$$

where, $\boldsymbol{d}^* = (d_1^*, \ldots, d_k^*)^T$, $d_l^* = \partial x_l(\bar{\boldsymbol{z}}^*)/\partial \bar{z}_i$ and $\boldsymbol{x}(\bar{\boldsymbol{z}})$ is a function depending on the aspiration levels and defined in some neighborhood $N(\bar{\boldsymbol{z}}^*)$ such that $\boldsymbol{x}(\bar{\boldsymbol{z}}^*) = \boldsymbol{x}^*$.

In this thesis, trade-offs are used to support the DM during the interactive solution procedure as mentioned before. Previously, trade-offs were used as a part of some multiobjective optimization method (see, for example, [14, 32, 76, 81, 103, 111]). That is, trade-off information was a crucial part of the method. Our aim is to propose some ideas for utilizing trade-off information as a supporting tool for the DM and not to use this information in the solution method itself. In addition, although the theoretical background of trade-off information has been widely studied [11, 14, 31, 40, 47, 48, 57, 58, 70, 78, 87, 88, 104, 105], there have been only few numerical examples. Therefore, there is no good evidence of practical usefulness of trade-off information as a tool for decision support.

Trade-off information can be utilized in the following way. When the DM gives a reference point (or makes a classification) and obtains a new Pareto optimal solution, partial trade-off rates are also calculated in that same solution point and they are shown to the DM. From the trade-off information the DM can study what kind of reference point (s)he should give next. For example, if it seems that a large improvement can be obtained in some objective by making a small sacrifice in some other objective it might be desirable to use that kind of reference point next. Trade-off information is most useful in the latter stages of the solution procedure when the DM has found a promising region and starts to search the best compromise solution from there.

In order to ease the cognitive burden set on the DM, actual values of partial trade-off rates are not shown to the DM. Instead, our idea is to show the magnitude of change, that is, whether it is small, neutral or significant. We have introduced two ways of presenting trade-off information to the DM, namely, the *arrow matrix visualization* and the *compressed trade-off visualization*. First of all, a partial trade-off rate is computed for each pair of the objective functions at the current Pareto optimal solution. In the arrow matrix visualization, the trade-off information obtained is divided into three categories: small, neutral, and significant. These categories are represented by white, gray and black arrows, respectively, and the arrow matrix consisting of arrows denoting partial trade-off rates is shown to the DM. An example of an arrow matrix visualization is shown in Figure 8. For example, let us assume that the DM wants to improve the value of the second objective. By looking at the second row of the arrow matrix in Figure 8 one can see that improving the second objective function results only to a small impairment of the values of the other objective functions. Note that the diagonal is empty because the partial trade-off rate for pair *ii* is always equal to 1. If the DM is interested only in improving some specific objective function, then the corresponding row of the arrow matrix visualization can only be shown to the DM. This could ease the cognitive burden set on to the DM even more.

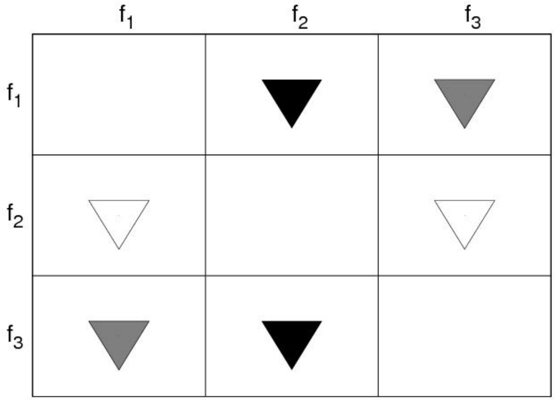


FIGURE 8 An example of the arrow matrix visualization.

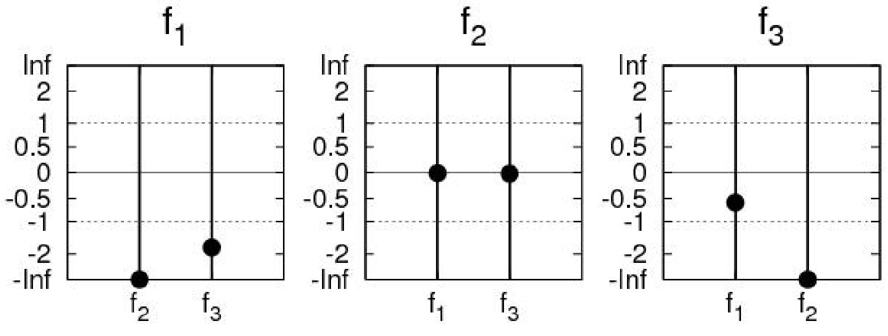


FIGURE 9 An example of the compressed trade-off visualization.

In compressed trade-off visualization, a *sigmoid function* $s : \mathbb{R} \to (-1, 1)$,

$$s(t) = \tanh\left(\frac{t}{2}\right) = \frac{e^{t/2} - e^{-t/2}}{e^{t/2} + e^{-t/2}}, \tag{10}$$

is used. The idea is to compress large trade-off rates because we only need to know whether it is small, neutral or significant. An example of the compressed trade-off visualization is shown in Figure 9 which represents the same trade-off data as Figure 8. As can be seen, the scale in vertical axis is compressed with the sigmoid function. The ranges for small, neutral and significant trade-off rates are defined to be $(-1/2, 0]$, $[-2, -1/2]$ and $(-\infty, -2)$, respectively [VI]. Different graphs in Figure 9 correspond to the rows of the arrow matrix visualization in Figure 8. For example, the graph in the middle of Figure 9 corresponding to the second row of the arrow matrix shows that if we want to improve the second objective the trade-off rates for the other objectives are small. In other words, the value of the first or the third objective will not change much. The compression effect to large trade-off rates is can be seen in Figure 9.

To test our ideas we generated 1000 reference points and studied what kind of advantage we could get by using trade-off information in selecting the next reference point [VI]. The results indicated that by utilizing trade-off information computed with the method of Sakawa and Yano we obtained such Pareto optimal solutions where we gained more than we lost which is desirable.

3 CHEMICAL PROCESS DESIGN

In this thesis, we are studying what kind of benefits interactive multiobjective optimization can offer to chemical process design [8, 97]. *Chemical process design* deals with designing efficient processes for the needs of various chemical industries. Chemical process industry is a broad field including, for example, paper making, sugar and pharmaceutical industries. Like in many real-world optimization problems, there are often several objectives or performance criteria present in chemical process design problems. These performance criteria need to be considered simultaneously and, usually, they are conflicting. Therefore, the methods of multiobjective optimization need to be used in order to solve them properly.

Chemical process design is based on process simulation [6, 46, 97] that deals with modelling of real-world processes. Chemical process simulation can be roughly divided into two groups, steady-state simulation and dynamic process simulation. The idea of *steady-state simulation* is to calculate the energy and the mass balances of the process, that is, to find such a state for the process that satisfies the physical conservation laws for the energy and the mass. If these laws are satisfied, the process is said to be in a *steady-state*. Time plays no role in steady-state simulation, while in *dynamic simulation*, the behaviour of the process is studied with respect to time. Note that a dynamic system can also reach a steady-state if there are no changes in the process anymore. In this thesis, steady-state simulation is used in papers [I]–[III] while papers [IV, V] deal with dynamic simulation models.

In this chapter, we consider optimization previously published related to chemical process design. Especially, we review the multiobjective optimization approaches used. In addition, we introduce different parts of the process design tool based on IND-NIMBUS that we have developed and used in this thesis.

3.1 Optimization in chemical process design

An important part of the process design is optimization. Process optimization has been used to improve the performance of processes since early 1970's [46]. During the last twenty years, optimization has also been used to determine the best possible process configuration. Simulation models have parameters that affect the behaviour and performance of the process. These parameters can be divided into two groups, equipment parameters related to process devices like heat transfer surface area and operational parameters like temperatures and pressures. If we want to improve the performance of the process by using only simulation tools, it can be achieved only with trial and error. In other words, the process is simulated with different values of the parameters and then the results are compared and the best combination is chosen as the final solution. That is very time consuming and the real optimum can only be achieved by chance. However, the best parameter values for some specific property of the process can be found by using optimization [7, 8, 9, 20, 30, 46, 53].

There are mainly two kinds of optimization problems in process design, namely the parameter optimization problems and process synthesis problems (see [8] and references therein). In *parameter optimization problems*, the performance of the process is optimized by finding the best values of process parameters with a fixed process structure. On the other hand, *process synthesis problems* try to find the best possible process structure from different configurations. They usually include both continuous and integer-valued variables because different connections included in a process structure are typically described by using binary variables, that is, variables having either the value 0 or 1. In this thesis, we consider parameter optimization problems. That enables us to use optimizers capable of handling only continuous variables. Note that parameter optimization and process synthesis problems can be based on either steady-state or dynamic models.

Traditionally, solving chemical process optimization problems has been realized by optimizing one criterion subject to some constraints. Commonly used optimization strategies have been sequential quadratic programming (SQP) -based methods that utilize the Karush–Kuhn–Tucker optimality conditions [50, 56] and optimize quadratic approximation of the objective function with respect to linear approximation of the constraints [5, 26]. SQP methods are best suited for smooth, small or medium-sized optimization problems and they need gradient information. Applications of SQP methods to chemical process optimization problems can be found, for example, in [1, 9, 10, 21, 44, 46, 53]. They have been applied to both steady-state [9, 44, 46, 53] and dynamic simulation models [1, 10, 21]. Recently, interior point optimization methods have been succesfully applied to large dynamic optimization problems [7, 52]. An excellent review of optimization in chemical process design can be found in [7] while future challenges are described in [30]. Note, that for some reason, multiobjective optimization is not considered in [30].

In real-life optimization problems, such as process optimization problems, there are usually several performance criteria that we want to optimize simultaneously. In addition, they are usually conflicting, that is, they all do not attain their optima at the same solution point. When process optimization problem is constructed, different criteria are usually forced into a single objective function because designers are not always aware that multiobjective optimization problems can be solved. It is not satisfactory to leave some of the objectives out of the problem or consider them as constraints just because designers usually can solve only problems with one or two criteria. Because usually only two criteria are considered, the designers think that solving a multiobjective optimization problem equals generating the whole Pareto optimal set. However, that works well only for problems with two criteria because with more than two criteria the number of Pareto optimal solutions required increases significantly and, also, visualization of the Pareto optimal set becomes difficult or impossible. Examples of commonly used methods are optimizing the weighted sum of the objective functions or selecting just one objective function to be optimised while considering others as constraints. These are the weighting and the ε-constraint methods mentioned in Section 2.3 and they are often used without knowing that they actually are methods of multiobjective optimization.

Both the weighting and the ε-constraint methods are easy to implement but have severe drawbacks. The weighting method does not work with non-convex optimization problems because it can not found the Pareto optimal solutions located in the non-convex part of the Pareto optimal set [64]. In addition, selecting the weights is not straight-forward because the objective function values of the solutions obtained do not necessarily have the same proportions as the weights used [79]. In the ε-constraint method, the problem is to select one of the objective functions (f_j) to be optimized. In addition, bounds ε_i, $i \neq j$, have to be assigned to the other objective functions that are considered as inequality constraints. Different Pareto optimal solutions can be obtained for different bounds. However, it is difficult to choose such bounds that will give desired solutions because the selected bounds are strict [64].

For the reasons mentioned above, in order to have the best possible solution, we have to use more sophisticated methods than single objective optimization and the simplest multiobjective optimization methods. To be more specific, we must use the methods of multiobjective optimization presented in Chapter 2.

Some multiobjective optimization methods have already been used in process optimization, see, for example, [2, 13, 18, 51, 54, 59, 91]. Usually, the methods used are very simple and easy to implement such as the weighting method or the ε-constraint method as mentioned before. That also enables the usage of already available single objective optimizers, because the formulation of the scalarized objective function is so simple. The multiobjective optimization problems commonly solved in process simulation have had only two objective functions and the aim has been to identify the *Pareto curve*, that is, the whole Pareto optimal set for two objectives [13, 59]. The two objectives used have usually described maximized profit and minimized environmental impact. For these kinds of problems, the Pareto curve can be easily visualized and it shows the interrelationships of the objectives. Therefore, the usage of simple methods is well argumented.

Ko and Moon [54] used the modified weighting method to solve cyclic thermal swing adsorption and cyclic rapid pressure swing adsorption processes with two objective functions in both cases. Note that their approach works only for two objectives. Lim et al. [59] applied three different methods to produce an approximation of the Pareto optimal set, namely the weighting method, goal programming and parameter space investigation [90]. They applied these methods to two cases (a simplified allyl chloride process and a methyl ethyl ketone process) which both had two objectives. Chakraborty and Linninger [13] used the ε-constraint method to design plant-wide waste management strategies for batch manufacturing sites. They considered two criteria, total cost and global polution index. Kawajiri and Biegler [51] applied also the ε-constraint method to solve two objective simulated moving bed (SMB) optimization problem.

Some applications of evolutionary multiobjective optimization have been also reported. Submarani et al. [91, 110] applied non-dominated sorting genetic algorithm (NSGA) to solve optimization problems arising from reactive SMB and varicol systems. They considered only two and three objective cases because they found it difficult to analyze the results obtained for more than three objectives. Roosen et al. [85] considered a combined cycle power system and used an evolutionary base strategy to solve a two objective optimization problem.

There have been also some theoretical consideration of multiobjective optimization methods in chemical engineering. In 1983, Clark and Westerberg [18] made a review of multiple criteria decision making methods to inform people involved in chemical engineering. They considered a two level optimization problem where one of the constraints was an optimization problem itself and showed that a general multiobjective optimization problem is a special case of the two level optimization problem.

As can be seen from the examples above, multiobjective optimization of chemical processes has usually consisted of applying very simple methods for two objective problems and the idea has been to generate a representation of the Pareto optimal set. However, there are usually more than two conflicting objectives present and, then, the visualization of the Pareto curve becomes much more difficult if not impossible and computational demand grows due to the larger number of Pareto optimal solutions required as already mentioned. In addition, the mutual dependence of the objectives becomes more difficult to understand for the DM. Interactive methods are well suited for problems having more than two objectives [64]. They are also computationally efficient, which will decrease the total time of solving the multiobjective optimization problem. This is quite important especially in problems where the evaluation of the objective or the constraint functions is time-consuming, which is the case in many real-life problems. For these reasons, we use interactive multiobjective optimization methods in this thesis.

The usage of interactive methods in chemical process optimization has been infrequent in the literature [4, 28, 49, 93], although interactive methods have been considered advantageous in multiobjective optimization. Umeda and Kuriyama [93] applied the interactive surrogate worth trade-off method [32] to a simple design of a toluene-steam dealkylation process where three objectives were considered, namely, investment cost, annual operating cost and reliability. Grauer et al. [28] made a review of the methods of multiobjective optimization and gave an overview of existing software. In addition, they discussed the design of a twin-screw extruder, the control of a film-hardening process and a production planning problem with three, two and three objective functions, respectively. Reference level and reference point approaches [98] were used. Barnikow et al. [4] applied the interactive multiobjective package IMPROVE to a strategic planning of carbochemical industry. IMPROVE is based on the reference point approach of Wierzbicki [100] and it is designed for linear multiobjective optimization. Linear multiobjective optimization was also considered by Kallio et al. [49] when they used the interactive mathematical programming package SESAME based on the reference point approach to solve a multiobjective trajectory optimization problem for a dynamic model simulating the development of Finnish paper industry.

In this thesis, we consider nonlinear, continuous multiobjective optimization problems arising from chemical process design as mentioned before. In addition, we deal only with deterministic optimization problems as opposed to stochastic optimization problems that involve uncertainties (see, for example, a review [84]). The main goal is to be able to treat chemical process design problems involving multiple criteria in such a way that there is no need for unnecessary simplifications and the real nature of the problems can be maintained. As the cited literature shows, there is a need for tools that can consider more than two performance criteria simultaneously without making unnecessary restrictions to the model only because of the lack of knowledge to solve such problems. To be more specific, the idea is to support the designer in obtaining desirable compromises between conflicting performance criteria present. In addition, interactive methods do not need to generate too many Pareto optimal solutions which can be computationally demanding for real-world problems as was previously noted. For these reasons, we use the process design tool based on IND-NIMBUS in this thesis. The classification-based interactive solution procedure offers a direct relationship between the classification information and the Pareto optimal solution obtained and, therefore, the effect of classification can be directly observed.

3.2 Constructing industrial process design tools

There are many different aspects that need to be considered when constructing an industrial process design tool. By *industrial process design tool* we mean a tool that can be used to solve chemical process design problems. A process design tool consists of at least three parts; a modelling tool, an optimizer and a graphical user interface. Figure 10 shows a schematic diagram of a process design tool and data transfer between different parts. First of all, the model of the process is formed with the modelling tool. During the interactive solution procedure, the GUI provides preferences of the designer to the optimizer that formulates and solves the optimization problem by utilizing function values and possible gradients coming from the modelling tool. Then, the optimizer returns new solutions to the GUI to be evaluated by the designer. In this thesis, we have used a process design tool based on IND-NIMBUS with different modelling tools and single objective optimizers depending on the process design problem considered. In the following, we call this tool the *IND-NIMBUS process design tool*.

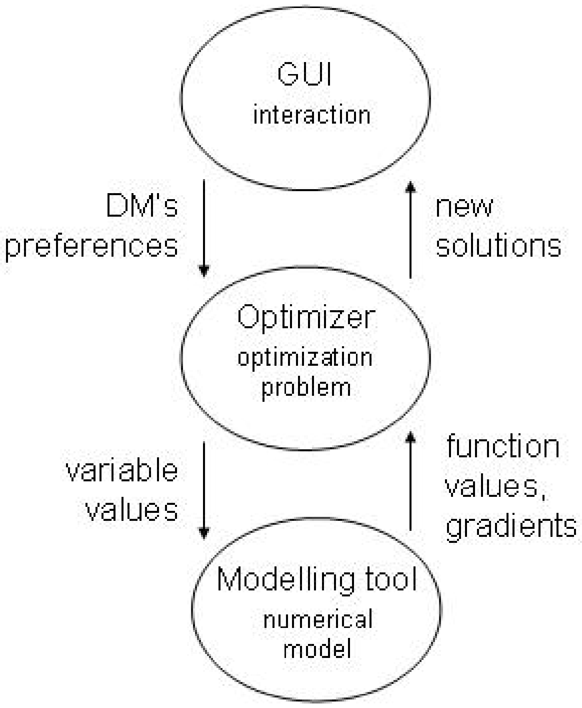
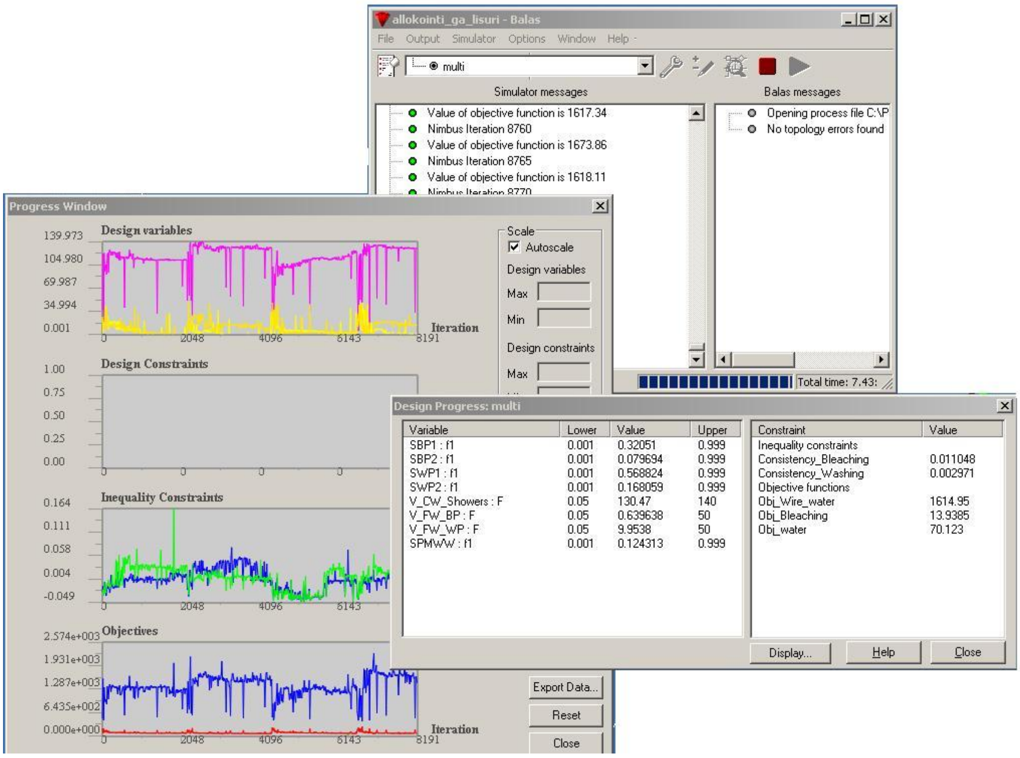


FIGURE 10 A schematic diagram of a process design tool.

In chemical process design, we need a process simulator or some modelling tool that generates a numerical model of the process considered. This numerical model is a basis for process simulation and, also, any optimization related to process design. The modelling tool provides the function values and, if necessary, also the gradients of the functions. Depending on the modelling tool, gradient information can be either exact or only an approximation. In addition, function values can be approximations if there are no analytic forms of the functions available. Sometimes, there can be a single objective optimizer implemented in the modelling tool that is specifically tailored to handle optimization problems produced by the modelling tool. An example of such a tool with a single objective optimizer is the BALAS® process simulator with an SQP optimizer (`http://balas.vtt.fi/`).

An optimizer is an important part of our process design tool. In solving design problems, we need an optimization tool that consists of a multiobjective optimization method that can efficiently handle all the conflicting criteria present in the problem. To our knowledge, there are no such tools available that utilize interactive multiobjective optimization and, therefore, this research has novel value. As mentioned earlier, multiobjective optimization methods usually scalarize the original multiobjective problem into a single objective optimization problem which produces a Pareto optimal solution for the problem. Therefore, it is also important to have an efficient single objective optimizer available to be used within the multiobjective optimization method. The single objective optimizer used should be chosen so that it takes into account special features present in the optimization problem.

In this thesis, we have considered interactive multiobjective optimization methods (namely the NIMBUS® method) because they have found to be useful in solving computationally demanding real-world optimization problems with more than two conflicting objectives [64]. As mentioned earlier, they need to generate only few Pareto optimal solutions and, also, enable the designer to learn more about the behaviour of the problem and the interrelationships of the conflicting criteria. We have also used different single objective optimizers within IND-NIMBUS, namely the proximal bundle optimizer in [I]–[III] and the interior point optimizer in [IV, V].



- Value of objective function is 1617.34
- Nimbus Iteration 8760
- Value of objective function is 1673.86
- Nimbus Iteration 8765
- Value of objective function is 1618.11
- Nimbus Iteration 8770

- Opening process file C:\P
- No topology errors found

Variable	Lower	Value	Upper
SBP1 : f1	0.001	0.32051	0.999
SBP2 : f1	0.001	0.079694	0.999
SWP1 : f1	0.001	0.568824	0.999
SWP2 : f1	0.001	0.168059	0.999
V_CW_Showers : F	0.05	130.47	140
V_FW_BP : F	0.05	0.639638	50
V_FW_WP : F	0.05	9.9538	50
SPMWW : f1	0.001	0.124313	0.999

Constraint	Value
Inequality constraints	
Consistency_Bleaching	0.011048
Consistency_Washing	0.002971
Objective functions	
Obj_Wire_water	1614.95
Obj_Bleaching	13.9385
Obj_water	70.123

FIGURE 11 A screenshot of the GUI of the BALAS® process simulator.

With the help of GUI, the designer participates in the design procedure. Especially in interactive design procedures, the GUI plays an important role because with the GUI the designer can input his/her preference information. IND-NIMBUS provides a good GUI for a process design tool because it can consider several conflicting performance criteria. In addition, the GUI can provide visualizations about different designs found and it translates the numerical information obtained from simulation and optimization into meaningful data for the designer. The modelling tools used can also have their own GUIs where the models can be adjusted and, possibly, the optimization problem considered can be formulated. For example, the BALAS® process simulator has a GUI which is shown in Figure 11.

In using interactive multiobjective optimization methods it is really important what kind of preference information the designer needs to provide and how it is requested. IND-NIMBUS has turned out to be really good in that respect because it does not require too much from the designer. All (s)he has to do is to decide how the current solution should be improved and classification is an easy way to do that. The only numerical values that the designer needs to provide are the aspiration levels and bounds if the classes I^{asp} and I^{bound} are used.

4 INDUSTRIAL APPLICATIONS

We have applied our interactive multiobjective optimization approach with the IND-NIMBUS process design tool to a number of real-world chemical process design problems. These problems arise from various areas of chemical industry, for example, from paper making and sugar industries as well as energy sector. All these industrial applications were considered in a novel way, that is, they had never before been solved as truly multiobjective problems with all these objective functions involved. In all the applications, we had more than two objective functions which has not been common in chemical process design as mentioned before.

All the problems were solved with the help of a real designer having knowledge about related areas. The IND-NIMBUS process design tool was used in all the applications and the underlying single objective optimizer as well as the modelling tool were varied between different applications. In this chapter, we present general forms of the chemical process design problems considered, briefly describe the industrial applications and also give some comments about the results obtained.

4.1 Chemical process optimization problems

In this thesis, there are two kinds of process design problems considered, namely *multiobjective steady-state process optimization problems* and *multiobjective dynamic process optimization problems*. In steady-state optimization problems, the process model consists of nonlinear equations while in dynamic optimization problems the process is modelled with partial differential equations (PDE). Next, we present general formulations of these problems.

A general multiobjective steady-state process optimization problem is of the form

$$
\begin{aligned}
\underset{\boldsymbol{x}}{\text{minimize}} \quad & \{f_1(\boldsymbol{\sigma}(\boldsymbol{x}),\boldsymbol{y}(\boldsymbol{x})),\ldots,f_k(\boldsymbol{\sigma}(\boldsymbol{x}),\boldsymbol{y}(\boldsymbol{x}))\} \\
\text{subject to} \quad & \boldsymbol{F}(\boldsymbol{\sigma}(\boldsymbol{x}),\boldsymbol{y}(\boldsymbol{x})) = 0, \\
& \boldsymbol{h}(\boldsymbol{\sigma}(\boldsymbol{x}),\boldsymbol{y}(\boldsymbol{x})) = 0, \\
& \boldsymbol{g}(\boldsymbol{\sigma}(\boldsymbol{x}),\boldsymbol{y}(\boldsymbol{x})) \leq 0, \\
& \boldsymbol{x}^l \leq \boldsymbol{x} \leq \boldsymbol{x}^u.
\end{aligned}
\tag{11}
$$

The objective functions f_i, $i = 1,\ldots,k$, depend on the decision variables $\boldsymbol{x} \in \mathbb{R}^n$ through the iteration variables $\boldsymbol{\sigma} \in \mathbb{R}^p$ and the design variables $\boldsymbol{y} \in \mathbb{R}^s$. The iteration variables are used to handle the feedbacks of the process while the design variables, typically describing some properties of the process devices, are used to fix some properties of the process. An example of a feedback is shown in Figure 12 where a simplified bleaching process reusing unused bleach is presented. If there are any feedbacks in the process, a steady-state needs to be computed iteratively and some variables need to be selected as iteration variables [8]. An example of design variables is that if we want to fix the temperature of some component in some particular part of the process, we have to select one variable as a design variable which can control temperature in that part. Note that the iteration and the design variables depend on the decision variables. The physical constraints of the problem are defined by the error functions $\boldsymbol{F} : \mathbb{R}^p \times \mathbb{R}^s \rightarrow \mathbb{R}^p$ for the iteration variables. The error functions are of the form

$$
\boldsymbol{F}(\boldsymbol{\sigma}(\boldsymbol{x}),\boldsymbol{y}(\boldsymbol{x})) = \boldsymbol{\sigma}(\boldsymbol{x}) - \tilde{\boldsymbol{\sigma}}(\boldsymbol{\sigma}(\boldsymbol{x}),\boldsymbol{y}(\boldsymbol{x})), \tag{12}
$$

where $\tilde{\boldsymbol{\sigma}}(\boldsymbol{\sigma}(\boldsymbol{x}),\boldsymbol{y}(\boldsymbol{x})) \in \mathbb{R}^p$ is a vector for the values of the iteration variables calculated from the process. Thus, the error functions (12) represent the error between the current values of the iteration variables $\boldsymbol{\sigma}$ and the new values of the iteration variables $\tilde{\boldsymbol{\sigma}}(\boldsymbol{\sigma}(\boldsymbol{x}),\boldsymbol{y}(\boldsymbol{x}))$ that are calculated from the process using the current values.

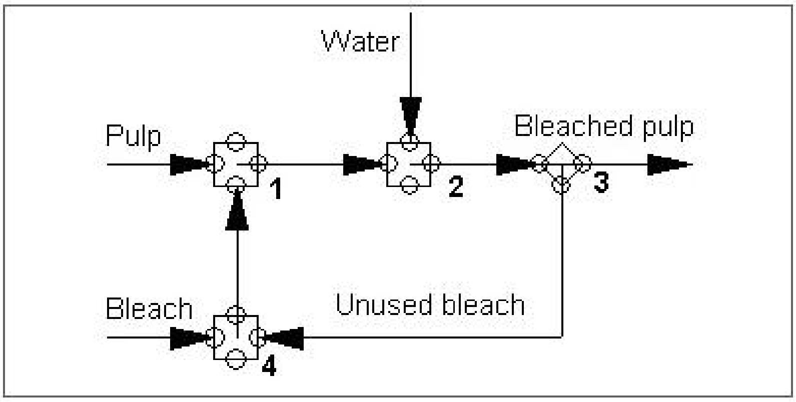


FIGURE 12 A simplified flowchart of a bleaching process which reuses unused bleach.

The design variables are used to fix some output values of the process according to the design constraints $\boldsymbol{h} : \mathbb{R}^p \times \mathbb{R}^s \to \mathbb{R}^s$. The design constraints are typically requirements to process outputs, for example, the temperature of the final product of the process must be fixed to some known value. Note that the number of design variables and design constraints has to be equal. There can also be inequality constraints $\boldsymbol{g} : \mathbb{R}^p \times \mathbb{R}^s \to \mathbb{R}^m$ that impose bounds for selected properties and simple lower and upper bounds, $\boldsymbol{x}^l$ and $\boldsymbol{x}^u$ respectively, for the decision variables.

A general multiobjective dynamic process optimization problem is of the form

$$\begin{aligned}
\underset{\boldsymbol{x}}{\text{minimize}} \quad & \{f_1(\boldsymbol{x}, \sigma(\boldsymbol{y}, t_f)), \ldots, f_k(\boldsymbol{x}, \sigma(\boldsymbol{y}, t_f))\} \\
\text{subject to} \quad & \boldsymbol{F}\left(\boldsymbol{x}, \frac{\partial\sigma(\boldsymbol{y},t)}{\partial y_1}, \ldots, \frac{\partial\sigma(\boldsymbol{y},t)}{\partial y_s}, \frac{\partial\sigma(\boldsymbol{y},t)}{\partial t}\right) = 0, \\
& \sigma(\boldsymbol{x}, \boldsymbol{y}, 0) = \sigma_0(\boldsymbol{y}), \quad \sigma(\boldsymbol{x}, \boldsymbol{0}, t) = \bar{\sigma}(t), \\
& \boldsymbol{h}(\boldsymbol{x}, \sigma(\boldsymbol{y}, t)) = 0, \\
& \boldsymbol{g}(\boldsymbol{x}, \sigma(\boldsymbol{y}, t)) \leq 0, \\
& \boldsymbol{x}^l \leq \boldsymbol{x} \leq \boldsymbol{x}^u.
\end{aligned} \tag{13}$$

Now, $\boldsymbol{f}$ is a vector-valued objective function at the final time t_f, $\sigma(\boldsymbol{y}, t) \in \mathbb{R}^p$ is the vector of the differential variables depending on the state variables $\boldsymbol{y} \in \mathbb{R}^s$ and time $t > 0$ while $\boldsymbol{x} \in \mathbb{R}^n$ is again the vector of the decision variables. The optimization problem is constrained with the PDE constraints $\boldsymbol{F} : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^l$, initial and boundary constraints and possible inequality and equality constraints $\boldsymbol{g}$ and $\boldsymbol{h}$, respectively. Vectors $\sigma_0(\boldsymbol{y})$ and $\bar{\sigma}(t)$ are the initial and the boundary values for σ, respectively.

There is at least two ways to solve problem (11). The simplest way is to consider the steady-state model as a black-box and use an optimizer with respect to the decision variables only. In other words, for every iteration, the optimizer gives decision variable values $\boldsymbol{x}$ to the modelling tool which computes corresponding steady-state and returns the values of the objective $\boldsymbol{f}$, and the inequality constraint functions $\boldsymbol{g}$ and possibly gradients, to the optimizer for calculation of the next values of the decision variables. A steady-state consists of such values of the iteration variables σ and the design variables $\boldsymbol{y}$ that the equality constraints of problem (11) are satisfied. Note that a steady-state is not necessarily unique. A steady-state can be found, for example, by using some Quasi-Newton method [46]. These methods are called *feasible-path methods* because the process is in a steady-state in every iteration. A benefit of this method is that the number of variables in the optimization problem is small, but on the other hand, the computation of a steady-state in every iteration of the optimization procedure is time consuming.

More efficient way to solve problem (11) is to apply, for example, an SQP optimizer directly to the problem. In other words, a steady-state is not required in every SQP iteration, but the process is guaranteed to be in a steady-state at the optimal solution. This type of methods are called *infeasible-path methods*. The number of variables in the optimization problem is significantly higher for these methods, because all the variables $\boldsymbol{x}$, $\boldsymbol{\sigma}$ and $\boldsymbol{y}$ are considered by the optimizer. However, infeasible-path methods are usually faster because no additional time is used in computing steady-states. Note that in infeasible-path methods, there is no distinction between the decision and the design variables because the process needs not to be in a steady-state for every iteration of the optimization method. To be more precise, a feasible-path optimizer operates only on the decision variables while an infeasible path optimizer operates on both the decision and the design variables and, therefore, they can be combined.

Problem (13) can be solved by discretizing the continuous variables in the problem. Either single-discretization, where only spatial variables are discretized, or full-disretization, where both the spatial and temporal variables are discretized, can be used. For example, the *Finite Element Method (FEM)* can be used in discretization (see, for example, [45]). In FEM, the computational domain is divided into small regions called *elements* and the solution in each element is then approximated by a function which is some combination of the values of the basis functions in the *nodes* of the element. The solution for the whole domain can be obtained from the solutions of individual elements. After the discretization, an optimizer is applied to the discretized problem.

4.2 Water allocation problem

In this section, we present a numerical example related to the paper making process considered in [I]. It is a water allocation problem for an integrated plant containing a thermomechanical pulping plant and a paper mill.

The water management in paper making is guided by the need to produce paper efficiently. The process requires a certain amount of fresh water to keep the disturbing contaminants on a level that is acceptable for both paper quality and machine runnability. In modern mills, the water consumption has been pressed down to 5-10 m^3 per a ton of paper by good water management [22], that is, matching water sources (e.g. filtrates) with water sinks (e.g. dilution duties) as illustrated in Figure 13. At the beginning of the paper making process, wood is processed into pulp. This processing includes refining, screening, washing and bleaching. Then, the pulp is led to the paper machine where it is processed into paper. During the paper making process, (fresh) water is needed for various diluting duties in the washers and screening.

The upper part of Figure 13 represents the thermomechanical pulping plant and the lower part represents the paper mill. The goal is to minimize the amount of fresh water taken into the process and also to minimize the amount of dissolved organic material in critical parts of the process by determining the right

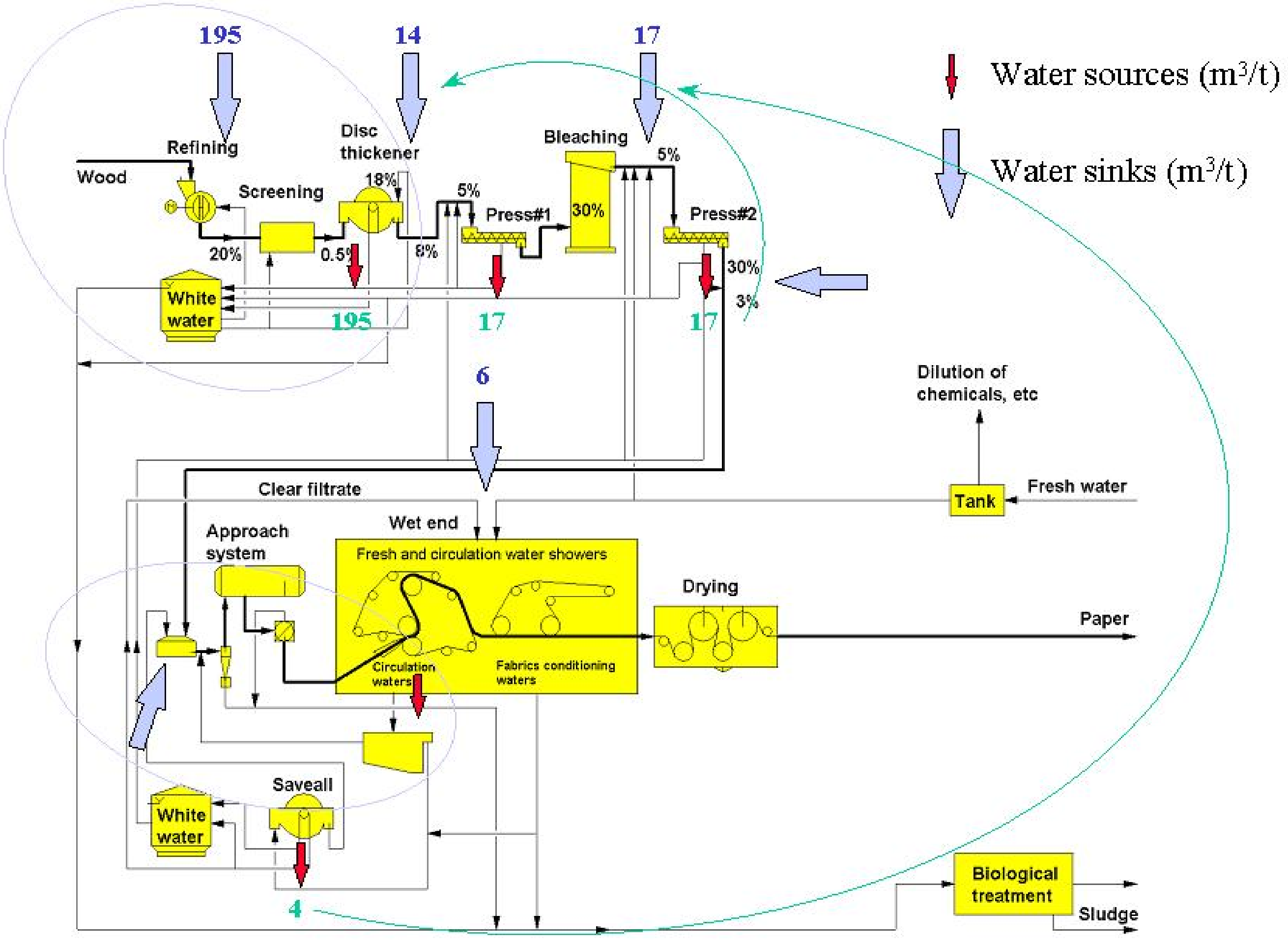


FIGURE 13 Water sinks and sources in an integrated plant.

recycling of water in the paper making process. These criteria are clearly conflicting because if the amount of fresh water is reduced, then more organic material is accumulated into the water in the process. On the other hand, if the amount of fresh water is increased, then more organic material exits the process in waste water and the concentration of the organic material decreases.

The optimization problem consists of three objective functions. The first objective function describes the concentration of dissolved organic material in the white water of the paper machine. White water is water that is removed from paper web in paper machine. This objective will have an impact on the use of chemicals and quality of paper produced in the paper machine. The second objective function describes the concentration of dissolved organic material in the pulp entering the bleaching process. This objective will influence the bleaching chemical use and pulp brightness. The third objective function describes the amount of fresh water taken into the paper making process. The problem has two inequality constraints that restrict the consistency of the pulp going to the bleaching press and the washing press, respectively.

There are altogether eight decision variables in the problem. The variables are of two types: splitters and valves. A splitter divides the incoming stream into two outlets. Its value, that may vary between 0 and 1, describes the portion of the flow going into the first outlet. Note, that the values of the splitters are continuous. A valve controls the amount of water taken into the process and the value of the flow through the valve is kilogramms per second.

The water allocation problem is a steady-state process optimization problem and it was modelled by using the BALAS® process simulator. The nonsmooth proximal bundle optimizer was used to solve the single objective subproblems produced by the NIMBUS® method. The water allocation problem is a multiobjective problem by its nature. Previously, this problem has been solved with the SQP optimizer of the BALAS® process simulator using the consumption of fresh water as the objective function and converting the other two objective functions into inequality constraints. In other words, the ε-constraint method was used. However, it can produce only one solution at a time corresponding to the selected upper bounds for these new inequality constraints. It can also be quite difficult to find correct upper bounds to find the most desirable solution without knowing the behaviour of the problem well enough and the roles of the objective functions and the constraints need not to be fixed. With the IND-NIMBUS process design tool, many different solutions can be generated according to the preferences of the designer and the study of the interrelationships of the different objective functions is more flexible.

The values of the consistencies of dissolved organic material in the white water of the paper machine and in the pulp entering the bleaching have no exact upper bounds which makes the usage of the ε-constraint method poorly justified. Therefore, the selection of the best solution is not self-evident but the role of the designer is emphasized and, thus, a need for an interactive solution method is obvious. In our case study [I], the designer wanted primarily to study the effect of the first two objective functions on the minimization of the fresh water consumption because our approach is more flexible for this purpose than the ε-constraint method which had been used previously. The solution procedure with the IND-NIMBUS process design tool provided a better understanding of the interrelationships of the objective functions when compared to the previous studies. The designer was able to rigorously study these relationships with various levels for dissolved organic materials and what kind of overall effect they have on the fresh water consumption. A detailed description of the interactive solution procedure and the results obtained are presented in [I].

4.3 Heat recovery system design

The case study presented in this section is a typical example of designing a heat recovery system for the process water system of a paper mill (see Figure 14) and it is considered in [II, III]. The process in question is a virtual fine paper mill, which operates in a climate typical to northern latitudes, where ambient temperature varies according to the season.

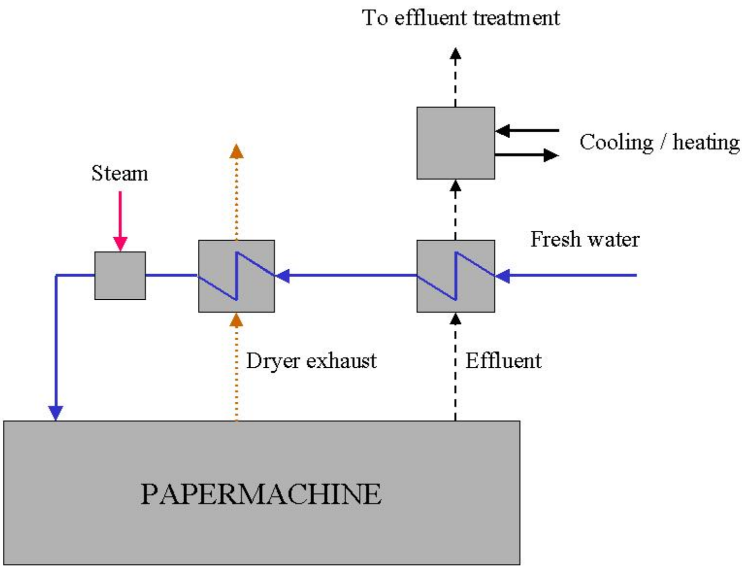


FIGURE 14 A simplified flowchart of the heat recovery for the process water system of a paper mill.

The aim is to organize the heat management of the process water system of a paper mill in the most efficient way. A special characteristic of this optimization problem is to consider also the effect of seasonal changes in the climate that affects heat management. For example, fresh water taken into the process is much colder in the winter than in the summer.

Fresh water taken into the paper making process needs to be heated to the process temperature (i.e. 60 ^{o}C). The heat sources available are the effluent from the process (at around 50 ^{o}C), which needs to be cooled down to around 37 ^{o}C to be suitable for an effluent treatment process, and dryer exhaust, which is moist air at the temperature 85 ^{o}C. In addition, steam can be used for the final heating of process water and effluent temperature can be controlled by external cooling or heating.

The design task is to determine the amount of heat recovered from the heat sources of the process to the heat sinks, that is, to estimate the size of heat exchangers and the amount of external energy needed for heating or cooling of the effluent coming from the paper machine. Both summer and winter scenarios are included in the same model by combining two parallel process models for summer and winter conditions that are solved simultaneously. Ambient air/water temperatures are 20/20 ^{o}C and -5/2 ^{o}C for summer and winter, respectively.

If the heat management was designed only according to winter conditions, the sizing of the heat recovery system would be too large resulting in large investment costs. On the other hand, if it was designed only for summer conditions, the energy consumption would be very high, because the heat recovery system would then be too small-sized.

The higher the degree of heat recovery, the less external energy is required to satisfy the needs of the process. On the other hand, the size of heat exchangers and hence investment costs rises with an increased degree of heat recovery. Seasonal changes add to the complexity of the problem, since a recovery system designed for winter conditions can be oversized for summer conditions and lead to a need for external cooling, for instance.

As already mentioned, the main trade-off here is between running costs, that is energy, and investment costs. Typically this trade-off is handled in single objective optimization by formulating an objective function, which consists of annualised energy and investment costs with estimated amortisation time and interest rate for the capital. The cooling or heating of effluent before treatment can be primarily either energy or investment cost. In case of heating or cooling with water, the running costs will dominate. However, in many cases, the use of water for cooling is not possible, and then a cooling facility is needed, which in the design phase is mainly an investment cost.

In this case, we have adopted four separate objective functions to be minimized: Steam needed for heating water for summer and winter conditions, estimation of area for heat exchangers (heat exchange from effluent and dryer exhaust in winter conditions, which represents the maximum values for the exchangers), and the amount of cooling or heating needed for the effluent. The first two objective functions tell us how much, on the average, we need to provide steam for the system and give also an estimation of the size of a steam distribution system needed for the mill. The third objective provides us the amount of heat exchange area needed. Once the area is known, we can estimate the investment cost more accurately from real vendor data rather than using a general cost correlation. The fourth objective, which is the amount of energy needed to regulate the temperature of effluent, is really an indication of the goodness of the design, since a value deviating from zero indicates either an additional investment in the form of a cooling tower or need for, for example, steam.

The three decision variables consisted of the area of the effluent heat exchanger and the approach temperatures of the dryer exhaust heat exchangers for both summer and winter operations. There were no additional equality constraints and, hence, there were three degrees of freedom, that is, three decision variables.

The heat recovery system design problem is also modelled as a steady-state process optimization problem with BALAS® and, again, the proximal bundle optimizer was used to produce Pareto optimal solutions through the subproblems. Traditionally, this type of a problem has been formulated as a single objective optimization problem hiding the interrelationships between the objectives. In addition, in the single objective optimization, monetary values have to be assigned a priori to energy and investments with a large degree of uncertainty in the correlations. Our approach eliminates these uncertainties from the model and leaves it to the designer to assess the costs and their uncertainties a posteriori when the required energy and material flows are much better defined. A detailed description of the interactive solution procedure and the results obtained are presented in [II, III].

4.4 Modified CGAM problem

The problem considered in this section is a modified version of the CGAM problem considering a cogeneration power plant and it is studied in [III]. The CGAM optimization problem was named after the first initials of C. Frangopoulos, G. Tsatsaronis, A. Valero and M. Spakovsky [94] who first introduced the problem in 1994. It was set up to be able to compare different thermoeconomic methodologies. This problem is widely used as a test problem for single objective optimization methods in process optimization.

The purpose of a cogeneration plant is to produce electricity by a gas turbine and steam by a heat recovery steam generator. A flowsheet of a simplified cogeneration plant is shown in Figure 15. First, air taken into the process is compressed in the air compressor. The compressed air is then heated and led to the combustion chamber where fuel is burnt to operate the gas turbine, which produces electricity. Next, gas from the turbine heats the combustion air which is led to the heat recovery steam generator where steam is produced.

The problem is based on a cogeneration plant capable of producing 30 MW of electricity and 14 kg/s of saturated steam at 20 bar. The plant consists of an air compressor, a gas turbine, an air preheater, a combustion chamber and a heat recovery steam generator (HRSG). The plant uses methane as fuel and air is atmospheric at the temperature 25^{o}C.

The aim in the original CGAM optimization problem was to minimize the costs including fuel costs and annualized investment costs. The problem was a

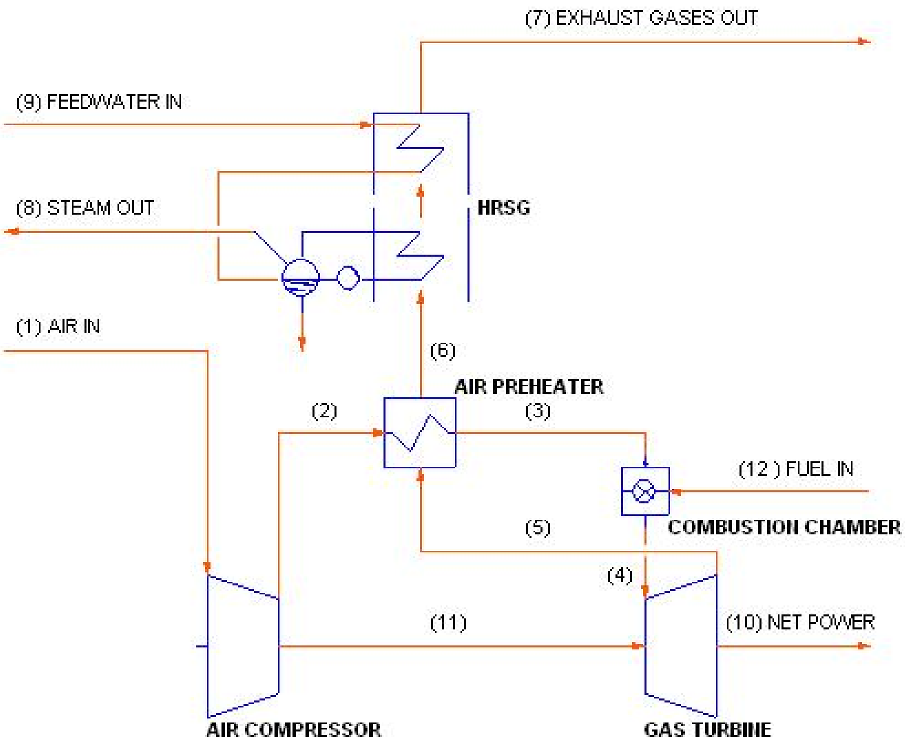


FIGURE 15 A flowsheet of the modified CGAM problem produced by the BALAS® software.

single objective optimization problem where the objective function was a sum of the costs. In 2002, Toffolo and Lazzaretto [92] introduced a bicriteria formulation of the CGAM problem where they took exergetic energy as another objective function.

The three criteria formulation of the original CGAM problem was presented in [III]. First of all, we have used a more accurate model than in the works of Valero et al. and Toffolo and Lazzaretto based on a procedure oriented simulator. Instead of requiring a 30 MW shaft power production, we have used the shaft power produced as an additional objective function to better control the process. In addition, the requirement of producing 14 kg/s of saturated steam is treated as the minimum requirement, that is, it has been converted into an inequality constraint.

The three objective functions used are the following. The first objective function describes the loss of energy and it is formulated to be minimized in the form 1 - efficiency. The efficiency of the plant is defined as the sum of shaft power in the air compressor (takes a negative value), shaft power in the gas turbine and the heat transfer rate of the HRSG divided by the heat transfer rate in the combustion chamber. The second objective function describes the total costs and it is the sum of annualized purchase costs of equipment and variable costs (fuel). The third objective function describes the shaft power for the assumed generator.

The objective functions used in this problem are conflicting. When the losses are reduced, the total costs should rise due to higher purchase costs. In addition, if the net shaft power produced is increased, then the investment and/or the fuel costs should rise.

The problem includes eight decision variables. The decision variables are the pressure after the air compressor, polytropic efficiency of the air compressor, polytropic efficiency of the gas turbine, the surface area of the air preheater, the temperature of gas after the combustion chamber, the air mass flow, the heat exchange surface area of the economizer and the boiler approach temperature.

The inequality constraints imposed limit the maximum temperature before the gas turbine, the maximum water temperature after the economizer, the exhaust gas temperature, the maximum isentropic efficiency of the compressor, the maximum isentropic efficiency of the turbine and the minimum steam production. Thus, there are six inequality constraints altogether. The efficiency limitations are due to technology available and the purchase cost formulations to grant the positive investment costs.

The modified CGAM problem is a steady-state process optimization problem and it was modelled by using BALAS® as in the previous applications. In addition, we used the proximal bundle optimizer as a single objective optimizer in the IND-NIMBUS process design tool. The CGAM problem is originally treated as a single objective optimization problem as mentioned before. As mentioned before, Toffolo and Lazzaretto used multiobjective evolutionary algorithms to find the Pareto optimal solution set of the bicriteria problem they introduced. However, with the IND-NIMBUS process design tool the problem can be modified more flexibly if necessary and there is no need to generate as many solutions as is needed when the whole Pareto optimal set is approximated. This means computational efficiency and results to much faster design procedure. The modified CGAM problem with three objective functions considered in this work illustrates well the flexibility of the new IND-NIMBUS process design tool. A detailed description of the interactive solution procedure and the results obtained are presented in [III].

4.5 Multiobjective Simulated Moving Bed Process

The application considered in this section is related to separation of chemical products and it is considered in [IV, V]. Efficient purification techniques are crucial in chemical process industries. In production of pharmaceuticals, it is often necessary to purify a product to nearly 100% for regulatory reasons. In food and sugar industries, removal of undesired components that degrade product quality is essential. Liquid chromatographic separation has been widely used for products with an extremely high boiling point, or thermally unstable products such as proteins. In liquid chromatographic processes, a small amount of feed mixture is supplied to an end of a column which is packed with adsorbent particles, and then pushed toward the other end with desorbent (water, organic solvent, or mixture of these).

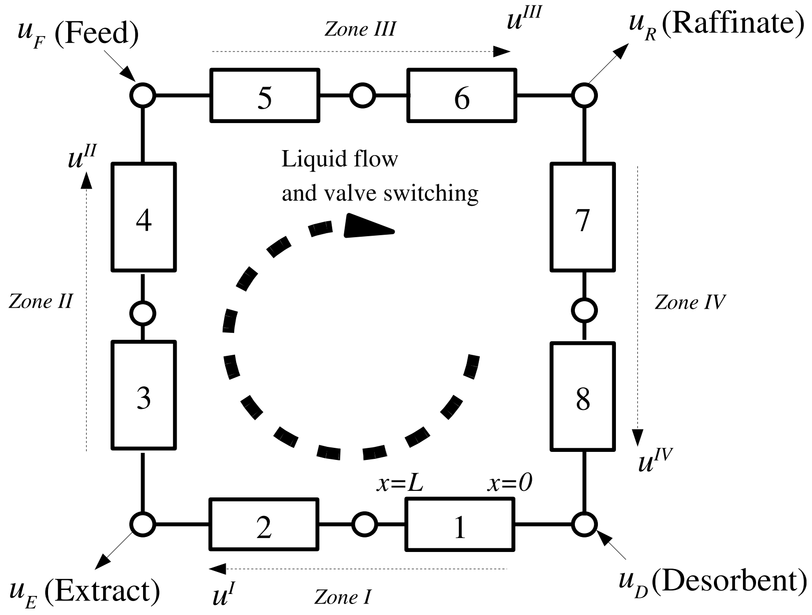
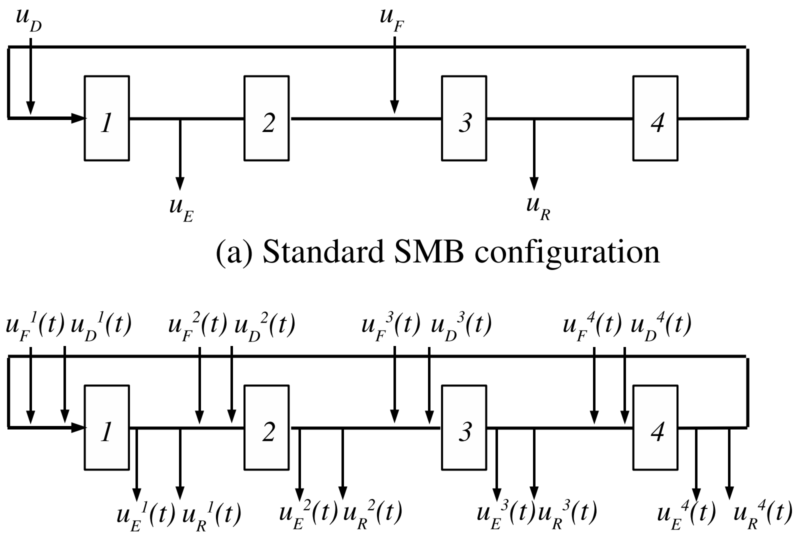


FIGURE 16 A schematic diagram of SMB process.

Simulated Moving Bed (SMB) process is a realization of a continuous and counter-current operation of liquid chromatographic separation process and it emerged from industry in the 1960's [86]. SMB's have been used in many important separations in pharmaceutical, petrochemical and sugar industries. An SMB unit consists of multiple columns which are packed with adsorbent particles. The columns are connected to each other making a circulation loop, as shown in Figure 16. In the SMB process of Figure 16, there are altogether eight columns. The feed mixture is inserted into the process in the upper left corner, while desorbent input is in the lower right corner. The two products raffinate and extract are collected in the upper right corner and lower left corner, respectively. Feed mixture and desorbent are supplied between columns continuously. At the same time, two products, raffinate and extract, are withdrawn from the loop also continuously. The two inlet and two outlet streams are switched in the direction of the liquid flow at a regular interval, or steptime. Because of the four inlet/outlet streams, there are four liquid velocity zones in the SMB loop as shown in Figure 16.

The SMB model consists of partial differential equations (PDE) for the concentrations of chemical components, restrictions for the connections between different columns and cyclic steady-state constraints [IV]. In [IV], a standard formulation of the SMB model is used while a superstructure formulation of SMBs introduced in [51] is used in [V]. The superstructure formulation is a more general way to represent SMBs and it can produce novel SMB operating schemes [51]. Figure 17 shows the differencies between the standard and the superstructure SMB configurations. The SMB problem considered in papers [IV, V] is for the separation of fructose and glucose. The values for the parameters in the SMB model are from [37, 52].



(a) Standard SMB configuration

(b) Superstructure

FIGURE 17 A schematic diagram of the standard and the superstructure SMB processes.

For the multiobjective optimization of SMB processes, we used four different objective functions for this problem: maximize throughput, minimize consumption of solvent in the desorbent stream, maximize product purity, and maximize recovery of the valuable component in the product. For the PDE model of SMB, full discretization was used, that is, both temporal and spatial variables were discretized leading to a huge system of algebraic equations. More precisely, the spatial domain is discretized via a central difference method, thus converting the PDEs to ODEs. The resulting ODEs are then discretized in the temporal domain using a Runge-Kutta discretization, Radau collocation on finite elements [52]. Therefore, the SMB optimization problems for both the standard and the superstructure formulations are large. The standard SMB optimization problem has 33 997 decision variables and 33 992 equality constraints while the superstructure SMB optimization problem has 34 102 decision variables and 34 017 equality constraints. Note that the degrees of freedom in the superstructure formulation is much higher (altogether 85) than in the standard SMB formulation (5 degrees of freedom).

The SMB is a dynamic process operating on periodic cycles which makes it a challenging optimization problem. To solve this problem we used the smooth variants of the subproblems of the NIMBUS® method and, therefore, we were able to use the IPOPT optimizer within IND-NIMBUS to produce new Pareto optimal solution candidates for this SMB problem. The IPOPT optimizer was chosen because it has been developed for solving large scale optimization problems. Two variants of SMB models were used: the standard SMB model was used in [IV] and the superstructure SMB model in [V]. Both SMB models were implemented with the AMPL modelling language. Because the SMB models were implemented in AMPL and IPOPT can be called straight from AMPL, we implemented the NIMBUS® method in AMPL. The AMPL implementation was then connected to IND-NIMBUS. We found that it is not straightforward to utilize a modelling language like AMPL to multiobjective optimization because it has an assumption that only one objective function can be considered at a time.

Considering four objective SMB design problem was a novel approach because, previously, there were only two or three objective functions considered as was shown in [IV, V]. This enabled full utilization of the properties of the SMB model without any unnecessary simplifications due to the lack of solution strategies. In addition, the designer obtained more thorough understanding of the interrelationships of different objectives considered and, thus, learned more about the behaviour of the problem. A detailed description of the interactive solution procedure and the results obtained are presented in [IV] for the standard model and in [V] for the superstructure model.

4.6 Comments

Finally, we give some general comments coming from the designers who solved the applications presented in this chapter and, also, some comments about the behaviour of the designers. A clear benefit of the new approach from the designer's viewpoint was that modelling and interactive multiobjective optimization are integrated into one tool and they did not have to use several separate tools to design these processes. The designers considered the results obtained with this approach more reliable because they could take into account different aspects of the design problem simultaneously. In addition, they saw that this kind of optimization is also a good way to test the reliability and robustness of the models.

On the other hand, the designers felt that this approach puts pressure on the modelling of different types of performance criteria. As mentioned before, certain criteria are difficult to model, which can make it difficult to include all the relevant aspects into the problem formulation. Furthermore, the designers felt that they could have needed more support during the interactive decision making procedure because it may be difficult to realize what is possible to achieve. In addition, some of the designers thought that it took too long to produce new Pareto optimal solutions which means that it is really important to have efficient single objective optimizers as mentioned before.

During the interactive solution procedures of the applications presented here, we noticed that the designers valued this kind of a solution procedure where they were able to participate in the procedure. Another benefit was that they could make simulations, optimize the process and visualize the results obtained with only one tool, that is, IND-NIMBUS. Although the designers liked the interactive solution procedure, it could be noticed that they did not want to spent too much time. This means that both the time needed to generate new solutions and the total duration of the interactive procedure should be as short as possible in order to keep the designers interested enough. To summarize, in addition to fast single objective optimizers we must provide the designers efficient tools for supporting the designer so that we could speed up the interactive solution procedure.

5 AUTHOR'S CONTRIBUTION

The ideas for this research originated from the project *NIMBUS – Multiobjective Optimization in Product Development* supported by Tekes, the Finnish Funding Agency for Technology and Innovation. The NIMBUS project was aimed to develop IND-NIMBUS and gain experiences in applying it to various industrial multiobjective optimization problems. Especially, the project meetings with Dr. Sakari Kaijaluoto and Dr. Jussi Manninen from the VTT Technical Research Center of Finland gave useful ideas. In addition to that, most of the research for the papers included in this thesis was made by the author collaborating closely with the author's advisors Prof. Kaisa Miettinen and Prof. Marko M. Mäkelä.

The ideas related to using IND-NIMBUS in chemical process design with the BALAS® simulator, that originated from the above mentioned project, were applied and further developed by the author in the papers [I]–[III] while more information and experiences were obtained. The author contributed on the implementation connecting the IND-NIMBUS process design tool to BALAS® that enabled the testing and further development of the ideas. The combination of the IND-NIMBUS process design tool and the BALAS® process simulator was used in all the papers [I]–[III]. These papers were mainly written by the author with helpful ideas, perspectives and comments from his advisors. The industrial applications considered in these papers were obtained from VTT. The water allocation problem considered in paper [I] was formulated by Kaijaluoto and Manninen. The heat recovery system design problem presented in paper [II] was modelled by Manninen and he also conributed in the description of the problem. The numerical example related to power plants in paper [III] was implemented by Mr. Juha Hakala from VTT who also contributed in the writing of the paper. The author participated in expressing the optimization problems when they were solved.

The papers [IV, V] were also mainly written by the author but the co-authors contributed in the writing process. The research was a co-operation with Mr. Yoshiaki Kawajiri and Prof. Lorenz T. Biegler from Carnegie Mellon University who also provided the models of the SMB processes. The author implemented the NIMBUS® method in the AMPL modelling language which enabled the usage of IND-NIMBUS with the IPOPT optimizer and the SMB models implemented in AMPL in both papers [IV, V]. The IPOPT optimizer was developed at the Carnegie Mellon University by Prof. Biegler's group. Prof. Miettinen's expertise in multiobjective optimization was utilized in these papers.

All the applications in the papers [I]–[V] were solved in a close co-operation with the author and designers who made the classifications and selected the preferred solutions at each stage of the interactive solutions procedure. The implementation of IND-NIMBUS was made by Mr. Vesa Ojalehto and Mr. Tommi Ronkainen. In addition, Mr. Juha Leppävuori from the VTT Technical Research Center of Finland was responsible for connecting the BALAS® simulator to the interface connecting BALAS® and IND-NIMBUS. Prof. Mäkelä implemented the proximal bundle optimizer including quadratic solvers by Prof. Krzysztof C. Kiwiel and Dr. Ladislav Lukšan.

The paper [VI] was a cooperation with Mr. Petri Eskelinen from Helsinki School of Economics. The test environment was constructed together by the authors of the paper and the paper itself was written jointly. The theoretic part of the paper was constructed by the author while all the tests were performed by Mr. Eskelinen. Prof. Miettinen and Prof. Mäkelä provided also useful comments and had a great influence in providing ideas in the beginning of this research.

6 CONCLUSIONS AND FUTURE WORK

In this thesis, we have studied interactive multiobjective optimization in solving multiobjective chemical process design problems. The aim has been to study what interactive multiobjective optimization can offer to chemical process design with multiple conflicting performance criteria.

We have developed and applied an IND-NIMBUS based process design tool to solve various real-world chemical process design problems arising, for example, from paper making and sugar industries. The IND-NIMBUS process design tool is an implementation of the interactive multiobjective optimization method NIMBUS® for industrial process design. All the problems were considered in a novel way with more than two conflicting objective functions and solved with the help of designers who had real knowledge about these applications. The solution procedure is interactive utilizing the experiences of the designer to guide the search towards desired compromise solutions of the multiobjective process design problem. In addition, we have proposed ideas of supporting the designer with trade-off information during the interactive solution procedure. Trade-off information is used to help the designer in decision making in order to direct the search so that it is possible to quickly find the best compromise solution for the problem in question.

With the help of this research, we have got an answer to the main research question by solving different industrial applications. We have shown that interactive multiobjective optimization is a useful approach to solve chemical process design problems with multiple performance criteria and it offers designers a practical and clear way to consider the conflicting criteria involved. In addition, with the help of the IND-NIMBUS process design tool, designers can learn more about the process design problems they are solving and better understand the behaviour of the phenomena present in the problem and the interrelationships of different criteria involved. Through our approach, we were able to obtain good results with the industrial applications considered.

To summarize, the main contribution of this thesis consists of three parts. First of all, we have shown that using interactive multiobjective optimization in solving multiobjective chemical process design problems is a useful way of considering these problems and no unnecessary simplifications or restrictions coming from the number of objectives need to be made. Secondly, we have studied the requirements of solving these problems in their true multiobjective character while recognizing the computational demand of real-world industrial problems. These requirements are taken into account in the IND-NIMBUS process design tool and in all of its individual parts. Finally, we have applied this approach in solving several real-world chemical process design problems that had never before been considered in this form with more than two objectives. We were able to obtain promising results and the designers found this kind of approach easy to understand and use.

In our opinion, interactive multiobjective optimization has potential to become widely used in the future of chemical process design because that approach has turned out to be useful during this research. Of course, we are always assuming that proper mathematical models for different performance criteria exist. This thesis shows that by utilizing interactive multiobjective optimization the chemical process design problems can be considered in their true multiobjective character without simplifying the problem. In this way, we can obtain realistic results and possibly novel designs without ruling out promising solutions by restricting the problem too much. It is worthwhile to note that the applications considered here were much more realistic because several criteria were considered at a time, but still it is not clear whether all the relevant criteria were considered. It is not so simple to guarantee that all the relevant criteria are considered and this issue is depending on the designer. It is possible that, in the future, the applications considered in this thesis will be solved with some additional performance criteria.

This thesis deals with possibilities and limitations of interactive approaches and, in our opinion, serves as a basis for future studies. Other researcher can use our research as a starting point when new process design tools are developed because we describe the different parts that in our opinion should be included and, also, discuss on what should be taken into account when these parts are developed. An important message of this thesis for the designers is that design problems with more than two performance criteria can truly be solved.

There are of course some challenges ahead. First of all, it is utmost important to get the designers be aware that these kind of techniques exist. If they do not know that, they can not of course use them. Secondly, there are only a few implementations of interactive multiobjective optimization methods available and we do not know any other such implementation than IND-NIMBUS that can be used directly in chemical process design. Availability of more implementations would enhance the usage of interactive methods in chemical process design and contribute to design of better processes. In addition, the dimensions and the complexities of the processes that can be modelled numerically are growing and this means that more efficient optimization techniques need to be developed that enable obtaining solutions in realistic times. In other words, generating new Pareto optimal solutions should not take so long that the designer does not want to wait anymore and, therefore, the interactive nature of the solution procedure would be lost.

We have also some more specific ideas for future work. Of course, we wish to test the IND-NIMBUS process design tool with more real-world applications coming from different types of chemical process design problems. Related to this, we plan to construct general interfaces to IND-NIMBUS in order to better combine it with different modelling tools and single objective optimizers. This would enable using tools that better meet the requirements of different types of problems.

Future research related to utilizing trade-off information to support the designer needs first more extensive tests before any further development of our ideas. So far, we have considered trade-off information only for unconstrained multiobjective optimization problems. In the future, it will be interesting to extend our studies of reliability of trade-off information for constrained multiobjective optimization problems. Constraints can have an affect in trade-off information because trade-off rates give local information and if we move a little bit from a solution, we can go out of the feasible region. In that case, trade-off information must be able to handle the constraints. In addition, an interesting topic is to study how to use sensitivity information in supporting the DM.

YHTEENVETO (FINNISH SUMMARY)

Tässä väitöskirjassa tutkin mahdollisuutta soveltaa interaktiivista monitavoiteoptimointia kemian prosessien suunnitteluun. Teollisten kemian prosessien suunnittelussa tulee ottaa samanaikaisesti huomioon useita prosessin toimintaan vaikuttavia tavoitteita kuten esimerkiksi taloudelliset näkökulmat, ympäristövaikutukset sekä prosessin toiminnallisuus. Nämä tavoitteet ovat usein ristiriitaisia jolloin niitä kaikkia ei voi saavuttaa yhtä aikaa. Tällöin täytyy löytää jonkinlainen kompromissi.

Tämän työn tarkoituksena on käyttää interaktiivista monitavoiteoptimointia auttamaan päätöksentekijää eli suunnittelijaa löytämään paras mahdollinen kompromissi ristiriitaisten tavoitteiden välille. Monitavoiteoptimoinnissa kompromisseja kutsutaan Pareto-optimaalisiksi ratkaisuiksi, mikä tarkoittaa sitä ettei mitään tavoitetta voida parantaa huonontamatta jotain muuta tavoitetta. Suunnittelija on henkilö, joka kykenee tarkasteltavan prosessin kannalta vertailemaan Pareto-optimaalisia ratkaisuja, jotka ovat matemaattisesti yhtä hyviä. Aikaisemmin kemiallisten prosessien suunnittelussa on yleensä otettu huomioon vain korkeintaan kaksi tavoitetta, mikä on ollut heikkoutena useissa kemian insinöörien aiemmin käyttämissä menetelmissä. Tämän työn lähestymistapa ei rajoita tarkasteltavien tavoitteiden lukumäärää. Tämän vuoksi prosessisuunnittelutehtäviä pystytään nyt tarkastelemaan entistä realistisemmin tekemättä tarpeettomia yksinkertaistuksia.

Tässä tutkimuksessa on kehitetty interaktiiviseen NIMBUS®-monitavoiteoptimointimenetelmään perustuva prosessisuunnittelutyökalu IND-NIMBUS. IND-NIMBUS koostuu kolmesta osasta: mallinnustyökalusta, optimoijasta ja graafisesta käyttöliittymästä. Mallinnustyökalu tuottaa tarkasteltavasta prosessista numeerisen mallin, johon koko suunnitteluprosessi perustuu. Optimoija rakentuu NIMBUS®-menetelmästä sekä soveltuvista yksitavoitteisen optimoinnin menetelmistä, jotka lopulta tuottavat uudet Pareto-optimaaliset ratkaisut ratkaisemalla NIMBUS®-menetelmän tuottamat aputehtävät. Erilaisia mallinnustyökaluja ja yksitavoitteisen optimoinnin menetelmiä voidaan käyttää riippuen tarkasteltavasta prosessista ja sen erikoisominaisuuksista. Graafinen käyttöliittymä puolestaan auttaa suunnittelijaa syöttämään kompromissiratkaisuihin liittyvän paremmuussuhdetiedon interaktiivisen ratkaisuprosessin aikana sekä mahdollistaa saatujen Pareto-optimaalisten ratkaisujen tulkitsemisen ja vertailun erilaisten visualisointien avulla. Paremmuussuhdetietoa käytetään ohjaamaan parhaan kompromissiratkaisun hakua tuottamalla ratkaisuja, jotka parhaiten seuraavat suunnittelijan toiveita.

Esittelen myös joitain alustavia ideoita, joilla suunnittelijaa voidaan tukea päätöksentekoprosessin aikana. Tässä työssä tutkitaan eri kompromissiratkaisuihin liittyvän vaihtosuhdetiedon käyttöä päätöksenteon tukena. Vaihtosuhdetieto kertoo miten tavoitteiden arvot muuttuvat liikuttaessa Pareto-optimaalisten ratkaisujen joukossa. Ideana on opastaa suunnittelijaa kohti parasta kompromissia näyttämällä hänelle vaihtosuhdetietoa interaktiivisen ratkaisuprosessin aikana. Vaihtosuhdetieto voi auttaa suunnittelijaa hahmottamaan millaisia kompromisseja voi olla mahdollista saavuttaa ja siten pyrkiä kohti sellaisia kompromisseja, joissa voidaan saavuttaa paljon vähillä uhrauksilla. Tämä tieto voi myös vakuuttaa suunnittelijan siitä, että hän on löytänyt parhaan mahdollisen kompromissin.

Yllämainittujen ideoiden testaamiseksi käytännössä tässä tutkimuksessa on ratkottu useita teollisen prosessisuunnittelun tehtäviä. Nämä suunnittelutehtävät liittyvät mm. paperinvalmistus- ja sokeriteollisuuteen ja niissä kaikissa huomioitiin enemmän kuin kaksi tärkeää tavoitetta. Aiemmin näitä tehtäviä on tarkasteltu huomioimalla vain yksi tai osa tärkeistä tavoitteista, joten tässä työssä niitä tarkasteltiin kokonaan uudella tavalla. Nämä tehtävät ratkaistiin käyttämällä IND-NIMBUS -prosessisuunnittelutyökalua sekä päätöksentekijöinä oikeita suunnittelijoita, joilla oli kokemusta jä näkemystä kyseisistä prosesseista. Saadut tulokset olivat lupaavia ja suunnittelijat pitivät interaktiivista ratkaisuprosessia helposti ymmärrettävänä ja helppokäyttöisenä. Tässä työssä pystyttiin näyttämään, että interaktiivista monitavoiteoptimointia voidaan onnistuneesti soveltaa aidosti monitavoitteisiin kemian prosessisuunnittelutehtäviin ja työssä esitetty lähestymistapa osoittautui tehokkaaksi myös käytännössä.

REFERENCES

[1] D. Alkaya, J. Vasantharajan, and L.T. Biegler. Generalization of a Tailored Approach for Process Optimization. *Industrial & Engineering Chemistry Research*, 39:1731–1742, 2000.

[2] J. Andersson. A Survey of Multiobjective Optimization in Engineering Design. Technical Report LiTH-IKP-R-1097, Department of Mechanical Engineering, Linköping University, 2000.

[3] K.E. Atkinson. *An Introduction to Numerical Analysis*. John Wiley, New York, 2nd edition, 1989.

[4] A. Barnikow, R. Gollmer, and M. Grauer. The Interactive Multiobjective Package IMPROVE with Application to Strategic Planning of Carbochemical Industry. 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 242–250. Springer, Berlin, 1986.

[5] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty. *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, New York, 2nd edition, 1993.

[6] L.T. Biegler. Chemical Process Simulation. *Chemical Engineering Progress*, October:50–61, 1989.

[7] L.T. Biegler and I.E. Grossmann. Retrospective on Optimization. *Computers & Chemical Engineering*, 28:1169–1192, 2004.

[8] L.T. Biegler, I.E. Grossmann, and A.W. Westerberg. *Systematic Methods of Chemical Process Design*. Prentice Hall, New Jersey, 1997.

[9] L.T. Biegler and R.R. Hughes. Infeasible Path Optimization with Sequential Modular Simulators. *AlChE Journal*, 28:994–1002, 1982.

[10] L.T. Biegler and A. Wächter. SQP SAND Strategies that Link to Existing Modeling Systems. In Ghattas, Heinkenschloss, Keyes, Biegler, and van Bloemen Waanders, editors, *Large-Scale PDE Constrained Optimization*, Lecture Notes in Computational Science and Engineering 30. Springer–Verlag, Berlin, 2003.

[11] E.E. Bischoff. A Posteriori Trade-Off Analysis in Reference Point Approaches. In M. Grauer and A.P. Wierzbicki, editors, *Interactive Decision Analysis*, Lecture Notes in Economics and Mathematical Systems 229, pages 139–145. Springer–Verlag, Berlin, 1984.

[12] J.T. Buchanan. A Naïve Approach for Solving MCDM Problems: the GUESS Method. *Journal of the Operational Research Society*, 48:202–206, 1997.

[13] A. Chakraborty and A.A. Linninger. Plant-Wide Waste Management. 1. Synthesis and Multiobjective Design. *Industrial & Engineering Chemistry Research*, 41:4591–4604, 2002.

[14] V. Chankong and Y.Y. Haimes. *Multiobjective Decision Making: Theory and Methodology*. Elsevier Science Publishing Co., Inc., New York, 1983.

[15] A. Charnes and W.W. Cooper. *Management Models and Industrial Applications of Linear Programming*, volume 1. John Wiley & Sons, New York, 1961.

[16] A. Charnes and W.W. Cooper. Goal Programming and Multiple Objective Optimizations. *European Journal of Operational Research*, 1:39–54, 1977.

[17] A. Charnes, W.W. Cooper, and R.O. Ferguson. Optimal Estimation of Executive Compensation by Linear Programming. *Management Science*, 1:138–151, 1955.

[18] P.A. Clark and A.W. Westerberg. Optimization for Design Problems Having More than One Objective. *Computers & Chemical Engineering*, 7(4):259–278, 1983.

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

[20] V. Dua and E.N. Pistikopoulos. Parametric Optimization in Process Systems Engineering. *Proceedings of the Indian National Science Academy, A*, 69:429–444, 2003.

[21] G. Dünnebier, J. Fricke, and K.-U. Klatt. Optimal Design and Operation of Simulated Moving Bed Chromatographic Reactors. *Industrial & Engineering Chemistry Research*, 39:2290–2304, 2000.

[22] K. Edelmann. Closed Water Circulations and Environmental Issues in Paper Production. In *Proceedings of PI'99, International Conference on Process Integration*, volume 1, pages 189–203, Copenhagen, Denmark, 1999.

[23] A.V. Fiacco. *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*. Academic Press, Inc., New York, 1983.

[24] P.C. Fishburn. Lexicographic Orders, Utilities and Decision Rules: A Survey. *Management Science*, 20:1442–1471, 1974.

[25] S. Gass and T. Saaty. The Computational Algorithm for the Parametric Objective Function. *Naval Research Logistics Quarterly*, 2:39–45, 1955.

[26] P.E. Gill, W. Murray, and M.H. Wright. *Practical Optimization*. Academic Press Inc., London, 1981.

[27] D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company, Inc., 1989.

[28] M. Grauer, A. Lewandowski, and A.P. Wierzbicki. Multiple-Objective Decision Analysis Applied to Chemical Engineering. *Computers & Chemical Engineering*, 8:285–293, 1984.

[29] A. Griewank and G.F. Corliss (editors). *Automatic Differentiation of Algorithms: Theory, Implementation and Application*. SIAM, Philadelphia, 1991.

[30] I.E. Grossmann and L.T. Biegler. Part II. Future Perspective on Optimization. *Computers & Chemical Engineering*, 28:1193–1218, 2004.

[31] Y.Y. Haimes and V. Chankong. Kuhn–Tucker Multipliers as Trade-Offs in Multiobjective Decision-Making Analysis. *Automatica*, 15:59–72, 1979.

[32] Y.Y. Haimes, W.A. Hall, and H.T. Freedman. *Multiobjective Optimization in Water Resources Systems*. Elsevier Scientific Publishing Company, Amsterdam, 1975.

[33] Y.Y. Haimes, L.S. Lasdon, and D.A. Wismer. On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization. *IEEE Transactions on Systems, Man and Cybernetics*, 1:296–297, 1971.

[34] J. Hakanen. *Challenges of Process Integration to Interactive Multiobjective Optimization*. Licentiate thesis, University of Jyväskylä, Jyväskylä, 2004.

[35] J. Hakanen, K. Miettinen, and M.M. Mäkelä. Using Genetic Algorithms in Multiobjective Process Optimization. In G. Bugeda, J.-A. Désidéri, J. Périaux, M. Schoenauer, and G. Winter, editors, *Proceedings of the EUROGEN 2003, International Conference on Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, Barcelona, Spain, 2003. CD-ROM.

[36] J. Hämäläinen, K. Miettinen, P. Tarvainen, and J. Toivanen. Interactive Solution Approach to a Multiobjective Optimization Problem in Paper Machine Headbox Design. *Journal of Optimization Theory and Applications*, 116(2):265–281, 2003.

[37] K. Hashimoto, S. Adachi, H. Noujima, and H. Maruyama. Models for the Separation of Glucose/Fructose Mixture Using a Simulated Moving Bed Adsorber. *Journal of Chemical Engineering of Japan*, 16(5):400–406, 1983.

[38] J. Haslinger and R.A.E Mäkinen. *Introduction to Shape Optimization: Theory, Approximation and Computation*. SIAM, Philadelphia, 2003.

[39] E. Heikkola, K. Miettinen, and P. Nieminen. Multiobjective Optimization of an Ultrasonic Transducer using NIMBUS. *Ultrasonics*, 44:368–380, 2006.

[40] M.I. Henig and J.T. Buchanan. Trade-Off Directions in Multiobjective Optimization. *Mathematical Programming*, 78:357–374, 1997.

[41] R. Horst and P.M. Pardalos, editors. *Handbook of Global Optimization*, volume 1. Kluwer Academic Publishers, Dordrecht, 1995.

[42] R. Horst and P.M. Pardalos, editors. *Handbook of Global Optimization*, volume 2, Noncovex optimization and its applications. Kluwer Academic Publishers, Dordrecht, 1995.

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

[44] G.C. Itle, A.G. Salinger, R.P. Pawlowski, J.N. Shadid, and L.T. Biegler. A Tailored Optimization Strategy for PDE-based Design: Application to a CVD Reactor. *Computers & Chemical Engineering*, 28:291–302, 2004.

[45] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press, Cambridge, 1988.

[46] S. Kaijaluoto. *Process Optimization by Flowsheet Simulation*. PhD thesis, University of Cambridge, Cambridge, U.K., 1984.

[47] I. Kaliszewski and W. Michalowski. Generation of Outcomes with Selectively Bounded Trade-offs. *Foundations of Computing and Decision Sciences*, 20:113–122, 1995.

[48] I. Kaliszewski and W. Michalowski. Efficient Solutions and Bounds on Tradeoffs. *Journal of Optimzation Theory and Applications*, 94:381–394, 1997.

[49] M. Kallio, A. Lewandowski, and W. Orchard-Hays. An Implementation of the Reference Point Approach for Multiobjective Optimization. Working paper WP-80-35, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1980.

[50] W. Karush. Minima of Functions of Several Variables with Inequalities as Side Conditions. Master's thesis, Department of Mathematics, University of Chicago, 1939.

[51] Y. Kawajiri and L.T. Biegler. A Nonlinear Programming Superstructure for Optimal Dynamic Operations of Simulated Moving Bed Processes. *Industrial & Engineering Chemistry Research*, 45:8503–8513, 2006.

[52] Y. Kawajiri and L.T. Biegler. Optimization Strategies for Simulated Moving Bed and Powerfeed Processes. *AIChE Journal*, 52(4):1343–1350, 2006.

[53] T.P. Kisala, R.A. Trevino-Lozano, J.F. Boston, and H.I. Britt. Sequential Modular and Simultaneous Modular Strategies for Process Flowsheet Optimization. *Computers & Chemical Engineering*, 11:567–579, 1987.

[54] D. Ko and I. Moon. Multiobjective Optimization of Cyclic Adsorption Processes. *Industrial & Engineering Chemistry Research*, 41(1):93–104, 2002.

[55] P. Korhonen, J. Moskowitz, and J. Wallenius. Choice Behaviour in Interactive Multiple-Criteria Decision Making. *Annals of Operations Research*, 23:161–179, 1990.

[56] H.W. Kuhn and A.W. Tucker. Nonlinear Programming. In J. Neyman, editor, *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, Berkeley and Los Angeles, 1951. University of California Press.

[57] H. Kuk, T. Tanino, and M. Tanaka. Trade-Off Analysis for Vector Optimization Problems via Scalarization. *Journal of Information & Optimization Sciences*, 18:75–87, 1997.

[58] G.M. Lee and H. Nakayama. Generalized Trade-Off Directions in Multiobjective Optimization Problems. *Applied Mathematical Letters*, 10:119–122, 1997.

[59] Y. Lim, P. Floquet, X. Joulia, and S.D. Kim. Multiobjective Optimization in Terms of Economics and Potential Environment Impact for Process Design and Analysis in a Chemical Process Simulator. *Industrial & Engineering Chemistry Research*, 38:4729–4741, 1999.

[60] E. Madetoja, K. Miettinen, and P. Tarvainen. Issue Related to the Computer Realization of a Multidisciplinary and Multiobjective Optimization System. *Engineering and Computing*, 22:33–46, 2006.

[61] E. Madetoja, E.-K. Rouhiainen, and P. Tarvainen. Making Higher Quality Paper - Case Study of Optimizing a Multidisciplinary System. In M. Papadrakakis, E. Onate, and B. Schrefler, editors, *CD-Rom Proceedings of the Coupled Problems 2005, International Conference on Computational Methods for Coupled Problems in Science and Engineering*, volume 1, Santorini, Creece, 2005.

[62] M.M. Mäkelä. Survey of Bundle Methods for Nonsmooth Optimization. *Optimization Methods and Software*, 17(1):1–29, 2002.

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

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

[65] K. Miettinen. Graphical Illustration of Pareto Optimal Solutions. In T. Tanino, T. Tanaka, and M. Inuiguchi, editors, *Multi-Objective Programming and Goal Programming: Theory and Applications*, pages 197–202, Berlin, Heidelberg, 2003. Springer-Verlag.

[66] K. Miettinen. Supporting Comparison of Alternatives in Multiple Criteria Decision Making with the Help of Visualizations. Reports of the Department of Mathematical Information Technology, Series B, Scientific Computing B 1/2003, University of Jyväskylä, Jyväskylä, 2003.

[67] K. Miettinen and M.M. Mäkelä. Interactive Bundle-Based Method for Nondifferentiable Multiobjective Optimization: NIMBUS. *Optimization*, 34:231–246, 1995.

[68] K. Miettinen and M.M. Mäkelä. Comparative Evaluation of Some Interactive Reference Point-Based Methods for Multi-Objective Optimisation. *Journal of the Operational Research Society*, 50:949–959, 1999.

[69] K. Miettinen and M.M. Mäkelä. Interactive Multiobjective Optimization System WWW-NIMBUS on the Internet. *Computers & Operations Research*, 27:709–723, 2000.

[70] K. Miettinen and M.M. Mäkelä. On Generalized Trade-Off Directions in Nonconvex Multiobjective Optimization. *Mathematical Programming*, 92:141–151, 2002.

[71] K. Miettinen and M.M. Mäkelä. On Scalarizing Functions in Multiobjective Optimization. *OR Spectrum*, 24:193–213, 2002.

[72] K. Miettinen and M.M. Mäkelä. Synchronous Approach in Interactive Multiobjective Optimization. *European Journal of Operations Research*, 170:909–922, 2006.

[73] K. Miettinen, M.M. Mäkelä, and R.A.E. Mäkinen. Interactive Multiobjective Optimization System NIMBUS Applied to Nonsmooth Structural Design Problems. In J. Dolezal and J. Fidler, editors, *System Modelling and Optimization, Proceedings of the 17th IFIP, Conference on System Modelling and Optimization*, pages 379–385, Prague, Czech Republic, 1996.

[74] K. Miettinen, M.M. Mäkelä, and T. Männikkö. Optimal Control of Continuous Casting by Nondifferentiable Multiobjective Optimization. *Computational Optimization and Applications*, 11:177–194, 1998.

[75] K. Miettinen, M.M. Mäkelä, and J. Toivanen. Numerical Comparison of Some Penalty-Based Constraint Handling Techniques in Genetic Algorithms. *Journal of Global Optimization*, 27:427–446, 2003.

[76] H. Nakayama. Nondominated Tradeoffs and Termination in Interactive Multiple Objective Linear Programming. In A.G. Lockett and G. Islei, editors, *Improving Decision Making in Organizations*, pages 410–423. Springer–Verlag, 1989.

[77] H. Nakayama. Sensitivity and Trade-Off Analysis in Multiobjective Programming. In A. Lewandowski and V. Volkovich, editors, *Methodology and Software for Interactive Decision Support*, pages 86–93. Springer–Verlag, 1989.

[78] H. Nakayama. Trade-off Analysis Using Parametric Optimization Techniques. *European Journal of Operational Research*, 60:87–98, 1992.

[79] H. Nakayama. Aspiration Level Approach to Interactive Multi-Objective Programming and Its Applications. In P.M. Pardalos, Y. Siskos, and C. Zopounidis, editors, *Advances in Multicriteria Analysis*, pages 147–174. Kluwer Academic Publishers, 1995.

[80] H. Nakayama and K. Furukawa. Satisficing Trade-Off Method with an Application to Multiobjective Structural Design. *Large Scale Systems*, 8:47–57, 1985.

[81] 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. Springer–Verlag, 1984.

[82] V. Ojalehto, K. Miettinen, and M.M. Mäkelä. Implementing IND-NIMBUS Software for Interactive Multiobjective Optimization. Reports of the Department of Mathematical Information Technology, Series B, Scientific Computing B 8/2005, University of Jyväskylä, Jyväskylä, 2005.

[83] A. Pascoletti and P. Serafini. Scalarizing Vector Optimization Problems. *Journal of Optimization Theory and Applications*, 42:499–524, 1984.

[84] E.N. Pistikopoulos. Uncertainty in Process Design and Operations. *Computers & Chemical Engineering*, 19:553–563, 1995.

[85] P. Roosen, S. Uhlenbruck, and K. Lucas. Pareto Optimization of a Combined Cycle Power System as a Decision Support Tool for Trading Off Investment vs. Operating Cost. *International Journal of Thermal Sciences*, 42:553–560, 2003.

[86] D.M. Ruthven and C.B. Ching. Counter-current and Simulated Counter-current Adsorption Separation Processes. *Chemical Engineering Science*, 44:1011–1038, 1989.

[87] M. Sakawa and H. Yano. Trade-off Rates in the Hyperplane Method for Multiobjective Optimization Problems. *European Journal of Operational Research*, 44:105–118, 1990.

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

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

[90] R.E. Steuer and M. Sun. The Parameter Space Investigation Method of Multiple Objective Nonlinear Programming: A Computation Investigation. *Operations Research*, 43:641–648, 1995.

[91] H. Submarani, K. Hidajat, and A. Ray. Optimization of Reactive SMB and Varicol Systems. *Computers & Chemical Engineering*, 27:1883–1901, 2003.

[92] A. Toffolo and A. Lazzaretto. Evolutionary Algorithms for Multi-Objective Energetic and Economic Optimization in Thermal System Design. *Energy*, 27:549–567, 2002.

[93] T. Umeda and T. Kuriyama. Interactive Solution to Multiple Criteria Problems in Chemical Process Design. *Computers & Chemical Engineering*, 4:157–165, 1980.

[94] A. Valero, A.M. Lozano, L. Serra, G. Tsatsaronis, J. Pisa, C. Frangopoulos, and M.R. Spakovsky. CGAM Problem: Definition and Conventional Solution. *Energy*, 19:279–286, 1994.

[95] A. Wächter and L.T. Biegler. Line Search Filter Methods for Nonlinear Programming: Motivation and Global Convergence. *SIAM Journal of Optimization*, 16:1–31, 2005.

[96] A. Wächter and L.T. Biegler. On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming. *Mathematical Programming*, 106:25–57, 2006.

[97] A.W. Westerberg, H.P. Hutchison, R.L. Motard, and P. Winter. *Process Flowsheeting*. Cambridge University Press, Cambridge, 1979.

[98] A.P. Wierzbicki. A Methodological Guide to Multiobjective Optimization. In K. Iracki, K. Malanowski, and S. Walukiewitz, editors, *Optimization Techniques, Part 1*, pages 99–123. Springer-Verlag, 1980.

[99] A.P. Wierzbicki. A Mathematical Basis for Satisficing Decision Making. In N.N. Morse, editor, *Organizations: Multiple Agents with Multiple Criteria*, pages 465–485. Springer-Verlag, 1981.

[100] A.P. Wierzbicki. A Mathematical Basis for Satisficing Decision Making. *Mathematical Modelling*, 3(25):391–405, 1982.

[101] A.P. Wierzbicki. On the Completeness and Constructiveness of Parametric Characterizations to Vector Optimization Problems. *OR Spektrum*, 8(2):73–87, 1986.

[102] A.P. Wierzbicki. Reference Point Approaches. In T. Gal, T.J. Stewart, and T. Hanne, editors, *Multicriteria Decision Making: Advances in MCDM Models, Algorithms, Theory, and Applications*, pages 9–1–9–39. Kluwer Academic Publishers, 1999.

[103] J.-B. Yang, C. Chen, and Z.-J. Zhang. The Interactive Step Trade-Off Method (ISTM) for Multiobjective Optimization. *IEEE Transactions on Systems, Man and Cybernetics*, 20:688–695, 1990.

[104] J.-B. Yang and D. Li. Normal Vector Identification and Interactive Trade-off Analysis Using Minimax Formulation in Multiobjective Optimization. *IEEE Transactions on Systems, Man and Cybernetics–Part A: Systems and Humans*, 32:305–319, 2002.

[105] H. Yano and M. Sakawa. Trade-off Rates in the Weighted Tchebycheff Norm Method. *Large Scale Systems*, 13:167–177, 1987.

[106] H. Yano and M. Sakawa. A Unified Approach for Characterizing Pareto Optimal Solutions of Multiobjective Optimization Problems: The Hyperplane Method. *European Journal of Operational Research*, 39:61–70, 1989.

[107] P.L. Yu. A Class of Solutions for Group Decision Problems. *Management Science*, 19(8):936–946, 1973.

[108] L. Zadeh. Optimality and Non-Scalar-Valued Performance Criteria. *IEEE Transactions on Automatic Control*, 8:59–60, 1963.

[109] M. Zeleny. Compromise Programming. In J.L. Cochrane and M. Zeleny, editors, *Multiple Criteria Decision Making*, pages 262–301. University of South Carolina Press, Columbia, South Carolina, 1973.

[110] Z. Zhang, K. Hidajat, and A.K. Ray. Multiobjective Optimization of SMB and Varicol Process for Chiral Separation. *AIChE Journal*, 48:2800–2816, 2002.

[111] S. Zionts and J. Wallenius. An Interactive Programming Method for Solving the Multicriteria Problem. *Management Science*, 22:652–663, 1976.