Department of Civil Engineering DTU

Frank Pedersen

A method for optimizing the performance of buildings

PHD THESIS

BYG • DTU

A method for optimizing the performance of buildings

Frank Pedersen

Ph.D. Thesis

Department of Civil Engineering
Technical University of Denmark

2006

A method for optimizing
the performance of buildings

Printed by Eurographic A/S, Copenhagen

Published by the Department of Civil Engineering

Technical University of Denmark

ISBN 87-7877-220-6

ISSN 1601-2917

Preface

This thesis is submitted to the Department of Civil Engineering at the Technical University of Denmark as a partial fulfillment of the requirements for the Danish Ph.d. degree.

The study described in the thesis has been carried out in the period from March 2002 to October 2006, and was financed by a scholarship from the Technical University of Denmark. The study was supervised by Professor Svend Svendsen from the Department of Civil Engineering, Associate Professor Benny Bøhm from the Department of Mechanical Engineering, and Associate Professor Hans Bruun Nielsen from Informatics and Mathematical Modelling.

Frank Pedersen

Frank Pedersen

Kongens Lyngby, October 31, 2006.

Acknowledgements

I want to thank my main supervisor Professor Svend Svendsen and my supervisors Associate Professor Benny Bøhm and Associate Professor Hans Bruun Nielsen for their engagement in this study, and for the many rewarding discussions during the study.

I also want to thank Professor Kaj Madsen at Informatics and Mathematical Modelling, Technical University of Denmark, and Professor John Bandler at McMaster University, Ontario, Canada, for sharing many inspiring ideas regarding the space mapping technique. Special thanks to John for the hospitality shown to me during my stays in Canada.

Ph.D. Jacob Søndergaard at Rambøll Denmark, as well as Ph.D. Ole Michael Jensen and M.Sc.Eng. Klaus Hansen at the Danish Building Research Institute have provided many helpful comments and suggestions that have significantly improved the thesis. I want to thank all three of them for their very helpful contributions.

I want to thank Ph.D. Alfred Heller at the Technical Knowledge Center of Denmark for getting me started at the Department of Civil Engineering, and for a very inspiring and fruitful collaboration during 2001 and 2002.

Finally, I want to thank my colleagues at the Department of Civil Engineering for many rewarding discussions, and for making the last three years a very pleasant experience. Special thanks to Ph.D. Peter Weitzmann and Ph.D. Toke Nielsen for their comments and suggestions for the thesis.

Abstract

This thesis describes a method for optimizing the performance of buildings. Design decisions made in early stages of the building design process have a significant impact on the performance of buildings, for instance, the performance with respect to the energy consumption, economical aspects, and the indoor environment. The method is intended for supporting design decisions for buildings, by combining methods for calculating the performance of buildings with numerical optimization methods. The method is able to find optimum values of decision variables representing different features of the building, such as its shape, the amount and type of windows used, and the amount of insulation used in the building envelope.

The parties who influence design decisions for buildings, such as building owners, building users, architects, consulting engineers, contractors, etc., often have different and to some extent conflicting requirements to buildings. For instance, the building owner may be more concerned about the cost of constructing the building, rather than the quality of the indoor climate, which is more likely to be a concern of the building user.

In order to support the different types of requirements made by decision-makers for buildings, an optimization problem is formulated, intended for representing a wide range of design decision problems for buildings. The problem formulation involves so-called performance measures, which can be calculated with simulation software for buildings. For instance, the annual amount of energy required by the building, the cost of constructing the building, and the annual number of hours where overheating occurs, can be used as performance measures.

The optimization problem enables the decision-makers to specify many different requirements to the decision variables, as well as to the performance of the building. Performance measures can for instance be required to assume their minimum or maximum value, they can be subjected to upper or lower bounds, or they can be required to assume certain values. The optimization problem makes it possible to optimize virtually any aspect of the building performance; however, the primary focus of this study is on energy consumption, economy, and indoor environment.

The performance measures regarding the energy and indoor environment are calculated using existing simulation software, with minor modifications. The cost of constructing the building is calculating using unit prices for construction jobs, which can be found in price catalogues. Simple algebraic expressions are used as models for these prices. The model parameters are found by using data-fitting.

In order to solve the optimization problem formulated earlier, a gradient-free sequential quadratic programming (SQP) filter algorithm is proposed. The algorithm does not require information about the first partial derivatives of the functions that define the optimization problem. This means that techniques such as using finite difference approximations can be avoided, which reduces the time needed for solving the optimization problem.

Furthermore, the algorithm uses so-called domain constraint functions in order to ensure that the input to the simulation software is feasible. Using this technique avoids performing time-consuming simulations for unrealistic design decisions.

The algorithm is evaluated by applying it to a set of test problems with known solutions. The results indicate that the algorithm converges fast and in a stable manner, as long as there are no active domain constraints. In this case, convergence is either deteriorated or prevented. This case is described in the thesis.

The proposed building optimization method uses the gradient-free SQP filter algorithm in order to solve the formulated optimization problem, which involves performance measures that are calculated using simulation software for buildings. The method is tested by applying it to a building design problem involving an office building. The results indicate that the method is able to find design decisions that satisfy all requirements to the decision variables and performance measures. Furthermore, the time needed by the algorithm for solving the optimization problem is acceptable.

There are still a number of unresolved issues regarding the building optimization method, which are suggested as further research in the field of building optimization methods.

Two papers are included in Appendix concerning so-called space mapping algorithms. These algorithms are relevant for developing fast and reliable building optimization methods.

Resumé

Denne afhandling beskriver en metode til optimering af bygningers ydelse. Design beslutninger foretaget i de tidlige stadier af bygningsdesignprocessen har en betydelig indflydelse på bygningers ydelse, f.eks. ydelse med hensyn til energiforbrug, økonomiske aspekter samt indeklima. Metoden har til hensigt at understøtte designbeslutninger for bygninger, ved at kombinere metoder til beregning af bygningers ydelse med numeriske optimeringsmetoder. Metoden er i stand til at finde optimale værdier for beslutningsvariabler, der repræsenterer forskellige egenskaber ved bygningen, f.eks. udformning, mængde og type af vinduer, samt isoleringsmængde anvendt i klimaskærmen.

De parter, der har indflydelse på designbeslutninger for bygninger, som f.eks. bygherrer, brugere, arkitekter, rådgivende ingeniører, entreprenører m.fl., har ofte forskellige og til en vis grad modstridende krav til bygningen. F.eks. kan bygherrer tænkes at være mere interesseret i anlægsomkostninger end indeklimaet, der formentlig er af større interesse for brugerne.

For at understøtte de forskellige typer af krav, der stilles af beslutningstagere for bygninger, formuleres et optimeringsproblem med det formål at repræsentere et stort antal designrelaterede beslutningsproblemer for bygninger. Problemformuleringen omfatter såkaldte ydelsesmål, der kan beregnes ved hjælp af simuleringssoftware for bygninger. F.eks. kan bygningens årlige energibehov, anlægsomkostninger, samt det årlige antal timer hvor overopvarmning forekommer, anvendes som ydelsesmål.

Optimeringsproblemet gør det muligt for beslutningstagere at specificere mange forskellige krav til beslutningsvariablerne, samt til bygningens ydelse. Ydelsesmål kan f.eks. kræves at antage deres største eller mindste værdi, de kan pålægges øvre eller nedre grænser, eller de kan kræves at antage angivne værdier. Optimeringsproblemet gør det muligt at optimere stort set hvilket som helst aspekt af bygningers ydelse, dog er der i dette studie primært fokus på ydelse med hensyn til energiforbrug, økonomi, samt indeklima.

Ydelsesmålene med hensyn til energi og indeklima beregnes ved hjælp af eksisterende simuleringssoftware, med mindre ændringer. Anlægsomkostningerne beregnes ved anvendelse af enhedspriser for byggearbejder, der kan findes i priskataloger. Simple algebraiske udtryk anvendes som modeller for disse priser. Modelparametrene findes ved hjælp af datafitting.

For at løse det formulerede optimeringsproblem, foreslås en gradientfri sekventiel kvadratisk programmerings (*sequential quadratic programming*, eller SQP) filter metode. Algoritmen kræver ikke kendskab til de første partielt afledte af de funktioner, der definerer optimer-

ingsproblemet. Dette betyder, at metoder såsom anvendelse af differenstilnærmelser kan undgås, hvilket reducerer den tid, det tager at løse optimeringsproblemet.

Algoritmen anvender desuden såkaldte *domain constraint functions* til at sikre, at inputtet til simuleringssoftwaren er realistisk. Ved anvendelse af denne teknik undgås tidskrævende simuleringer for urealistiske designbeslutninger.

Algoritmen evalueres ved at anvende den på testproblemer med kendte løsninger. Resultaterne indikerer, at algoritmen konvergerer hurtigt og stabilt, så længe der ikke er aktive *domain constraints*. I dette tilfælde er konvergens enten forringet eller forhindret. Dette tilfælde er beskrevet i afhandlingen.

Den foreslåede bygningsoptimeringsmetode anvender som nævnt den gradientfrie SQP filter metode til at løse det formulerede optimeringsproblem, hvori der indgår ydelsesmål beregnet ved hjælp af simuleringssoftware for bygninger. Metoden er testet ved at anvende den til løsning af et designbeslutningsproblem der omhandler en kontorbygning. Resultaterne indikerer, at metoden er i stand til at finde designbeslutninger, der opfylder alle krav til beslutningsvariable samt ydelsesmål. Desuden er algoritmens tidsforbrug til løsning af problemet tilfredsstillende.

Der er stadig en del uafklarede spørgsmål vedrørende bygningsoptimeringsmetoden, der er foreslået som mulige forskningsemner vedrørende bygningsoptimeringsmetoder.

To artikler er vedlagt i appendiks, der omhandler såkaldte *space mapping* algoritmer. Disse algoritmer er relevante i forbindelse med udvikling af hurtige og pålidelige bygningsoptimeringsmetoder.

Table of Contents

Chapter 1

Introduction

The main purpose of this thesis is to describe a method for optimizing the performance of buildings, and furthermore to improve the understanding of how numerical optimization methods can be used for supporting decision-making, with special focus on design decisions for buildings in the early stages of the design process.

1.1 Motivation

It is a well-established fact that it is easier and less costly to change design decisions for buildings in an early stage rather than later. Furthermore, changes made in early stages are believed to have a larger impact on the building performance than changes made later. It is therefore important to develop methods for supporting significant design decisions made in early stages. See for instance Poel [56] and Nielsen [49] for a more detailed description and discussion of the design process for buildings.

Supporting design decisions in early stages of the design process is addressed by using numerical optimization methods, which have been applied to virtually all fields of engineering. These methods can be used for suggesting decisions that are based on relevant decision criteria, such as energy performance, economy and the indoor environment, etc.

The parties who influence design decisions for buildings, such as building owners, building users, architects, consulting engineers, contractors, etc. (also referred to as *decision makers* in the following), often have different and to some extent conflicting requirements to buildings. For instance, the building owner may be more concerned about the budget for the building, rather than the indoor climate, which is more likely to be a concern of the building user. It is therefore important to develop methods that focus on design decisions in the early stages of the design process, and that are flexible. The methods must enable the decision maker to specify and modify requirements to buildings quickly and effortlessly.

1.2 Objective

The objective of this study is to develop and document a method for optimizing the performance of buildings (referred to as a *building optimization method* in the following), intended for supporting decisions in the early stages of the design process of buildings. The method combines numerical methods for calculating the performance of buildings with numerical optimization methods.

The method is intended for decision makers; however, the description of the method provided in this thesis is more suitable for developers of building optimization methods.

The following approach is used for developing and documenting the method:

1. A literature survey is made on how numerical optimization methods are being used for supporting design decisions for buildings.

2. A description is provided of how optimization methods can be used for supporting decision-making. An optimization problem is formulated, intended to be a mathematical model for a wide range of building design decision problems. The numerical methods used for estimating solutions to the problem must take the following concerns into account:

   (a) The partial derivatives of the functions defining the problem are (usually) not available.

   (b) The function values are not defined for all parameters used as arguments to the functions.

   (c) The time consumption needed for calculating the function values may be excessive.

3. Numerical methods are developed or adapted addressing the concerns formulated in the previous step. Sequential linear programming (SLP) methods and sequential quadratic programming (SQP) methods are considered for addressing 2(a) and 2(b), and space mapping (SM) methods are considered for addressing 2(c).

4. The numerical methods are tested on a set of test problems with known solutions in order to evaluate the convergence properties. Convergence theorems are not provided.

5. The resulting building optimization method is applied to mathematical models of the energy performance, the building economy, and the indoor environment.

6. The results obtained with the proposed building optimization method are discussed, in an attempt to evaluate the usability of the method from the point of view of a decision maker.

1.3 Outline of the thesis

The structure of the thesis follow the aforementioned approach, except that the details regarding the space mapping method are provided in the appendix, since more work is still required in order to fully integrate them with the building optimization method.

The outline of the thesis is:

Chapter 2 concerns the background for the study, including a literature survey, and an introduction to the notation and mathematical concepts used in the thesis.

Chapter 3 concerns a description of how optimization methods can be used for supporting decisions. An optimization problem is formulated, intended for representing a wide range of decision problems.

Chapter 4 concerns a description of the building optimization method.

Chapter 5 describes a gradient-free SQP filter method intended for solving the types of continuous optimization problems with continuous constraints described in Chapter 3.

Chapter 6 concerns numerical experiments for evaluating the building optimization method.

Chapter 7 summarizes the conclusions of the thesis.

Appendix A is the included paper [7], which concerns a space mapping interpolating surrogate method, developed for the purpose of optimizing time-consuming mathematical models of physical systems. The method is applied to a number of design optimization problems from microwave electronics.

Appendix B is the included paper [54], which concerns a space mapping method for enhancing the accuracy of simple mathematical models of building components, applied to a model of a thermally active building component.

Appendix C concerns the test problems used for testing the filter SQP method described in Chapter 5.

Appendix D provides default values for the constants parameters used by the building optimization method.

Appendix E provides the mathematical nomenclature used when describing optimization and numerical optimization methods.

Appendix F provides the nomenclature used when describing building physics and the economy of buildings.

1.4 Publications

As part of this study, contributions have been provided to the journal paper by Bandler et.al. [7], and the following conference papers: Bandler et.al. [3], Bandler et.al. [5], Bandler et.al. [6], Kragh et.al. [39] and Pedersen et.al. [54].

The papers by Bandler et.al. [7] and Pedersen et.al. [54] are included in Appendix A and B, respectively.

Chapter 2

Background

This chapter provides a description of the concepts that are relevant for developing and implementing methods for optimizing the performance of buildings. Aspects of optimization are addressed, as well as methods for assessing the performance of buildings. A literature survey of building optimization methods is furthermore provided. Finally, the delimitation of the study is provided.

2.1 Optimization

2.1.1 Continuous optimization

The definition of a continuous optimization problem is based on an objective function $f : \mathcal{D} \rightarrow \mathbb{R}$ and a set of constraint functions $c : \mathcal{D} \rightarrow \mathbb{R}^m$. The aim is to find a set of parameters $x^* \in \mathbb{R}^n$, where f obtains its smallest function value. The functions c are used for constraining the solution to a subset of $\mathbb{R}^n$. A distinction is made between inequality and equality constraints, that is, requirements to the solution in the form $c_i(x) \geq 0$ and $c_j(x) = 0$, respectively, for some indices i and j.

It is practical to represent the inequality and equality constraint functions by the vector-valued functions $c_\mathcal{I} : \mathcal{D} \rightarrow \mathbb{R}^{n_\mathcal{I}}$ and $c_\mathcal{E} : \mathcal{D} \rightarrow \mathbb{R}^{n_\mathcal{E}}$, respectively, where $\mathcal{I}$ and $\mathcal{E}$ are index sets. The number of inequality constraint functions is represented by $n_\mathcal{I}$, and the number of equality constraint functions by $n_\mathcal{E}$. It is assumed that $n_\mathcal{I} + n_\mathcal{E} = m$.

In general, for an index set $\mathcal{S}$ referring to a subset of the functions c, let $c_\mathcal{S} : \mathcal{D} \rightarrow \mathbb{R}^{n_\mathcal{S}}$ be a subset of the functions in c corresponding to the index values given in $\mathcal{S}$. Given the matrix $P_\mathcal{S} \in \mathbb{R}^{n_\mathcal{S} \times m}$:

$$(P_\mathcal{S})_{i,j} = \begin{cases} 1 & \text{if } \ \mathcal{S}_i = j \\ 0 & \text{otherwise} \end{cases} \quad i = 1, \ldots, n_\mathcal{S} \ \text{ and } \ j = 1, \ldots, m, \tag{2.1}$$

the functions $c_\mathcal{S}$ can be defined in the following way:

$$c_\mathcal{S}(x) = P_\mathcal{S} \cdot c(x). \tag{2.2}$$

It is assumed that $\mathcal{D} \subseteq \mathbb{R}^n$, i.e. the possibility that f and c are not defined for all $x \in \mathbb{R}^n$ is also considered. This may be necessary in order to ensure that the algorithms used for solving optimization problems, provide feasible input to for instance simulation software, if such software is used for calculating f and c.

It is assumed that the domain $\mathcal{D}$ for the objective and constraint functions can be defined in the following way:

$$\mathcal{D} = \{x \in \mathbb{R}^n : d(x) \geq 0\}\,, \tag{2.3}$$

where the functions $d : \mathbb{R}^n \rightarrow \mathbb{R}^{n_\mathcal{D}}$ are referred to as *domain constraint functions*. This concept is intended for representing functions that define the domain of objective and constraint functions for continuous, constrained optimization problems. Notice that with the formulation (2.3), equality domain constraints are not handled separately.

The term *domain constraint function* also occurs in other areas, such as image analysis, see for instance Ye et.al. [64]. In control theory, the terms *frequency domain constraint function* and *time domain constraint function* occur, see for instance Güvenç and Güvenç [30].

The optimization problems considered in this study have the following structure:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & c_\mathcal{I}(x) \geq 0 \\ & c_\mathcal{E}(x) = 0 \\ \text{with respect to} & x \in \mathcal{D}, \end{array} \tag{2.4}$$

For a thorough description of the general theory of optimization and numerical methods for solving (continuous) optimization problems, the reader is referred to Nocedal and Wright [51], and Conn et.al. [14]. Furthermore, Dennis and Schnabel [17] describe numerical methods for solving unconstrained optimization problems and nonlinear equations.

Numerical methods for solving the linear algebra subproblems, which are needed when implementing numerical optimization methods, are described by Golub and Van Loan [28].

The statements $c_\mathcal{I}(x) \geq 0$ and $c_\mathcal{E}(x) = 0$ in (2.4) are only true if they are true for all functions referred to by $\mathcal{I}$ and $\mathcal{E}$, respectively. The requirements to the functions that define (2.4) is that they are continuous and twice differentiable, within their respective domains.

The region $\mathcal{F} \subseteq \mathcal{D}$ where all constraints are satisfied, i.e.

$$\mathcal{F} = \{x \in \mathcal{D} : c_\mathcal{I}(x) \geq 0 \;\wedge\; c_\mathcal{E}(x) = 0\}\,, \tag{2.5}$$

is referred to as the feasible region. If $x \notin \mathcal{F}$, then x is referred to as an infeasible point. If $\mathcal{F} = \emptyset$, then the problem (2.4) is referred to as infeasible.

In Figure 2.1 (left) is shown an example of an unconstrained optimization problem, i.e. a problem with $\mathcal{I} = \emptyset$, $\mathcal{E} = \emptyset$ and $\mathcal{D} = \mathbb{R}^n$. In the same figure (right) is shown an example of a feasible optimization problem with equality, inequality and domain constraints, where $\mathcal{I} = \{1, 3\}$, $\mathcal{E} = \{2\}$, and $n_{\mathcal{D}} = 2$.

The following conventions are used when plotting optimization problems of the form (2.4):

The objective function is represented by a contour plot colored with gray tones, where light areas represent high function values, and dark areas represent low function values.

Inequality constraint functions: Gray regions represent parameter values where one or more inequality constraint function is negative.

Equality constraint functions are represented by dashed contour lines, where the function value is zero.

Domain constraint functions: White regions represent parameters outside the domain $\mathcal{D}$. Level curves for the objective and constraint functions are not shown in these regions, since they are not defined here.

The solution to an optimization problem is represented by the symbol $*$.

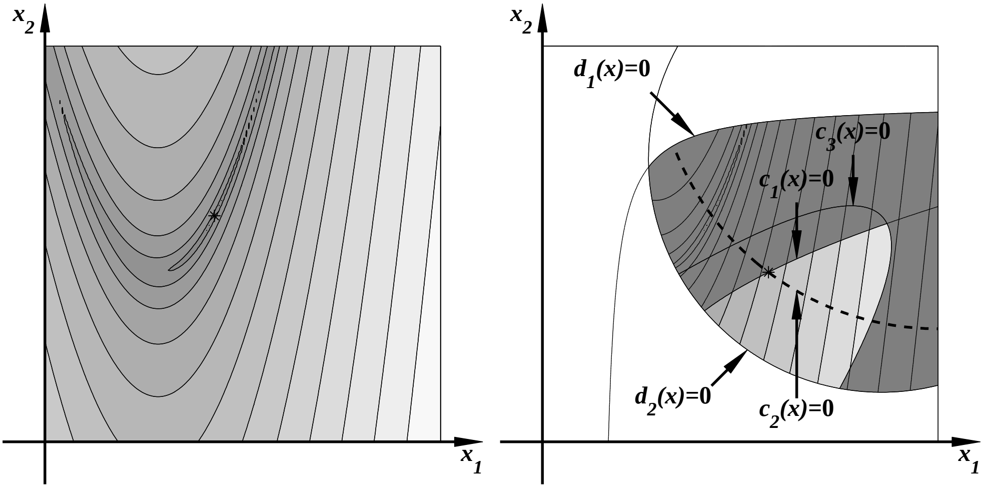


Figure 2.1: Left: An unconstrained optimization problem. Right: An optimization problem with equality, inequality and domain constraints.

2.1.2 Aspects of numerical optimization methods

Numerical optimization algorithms improve a solution estimate $x_k \in \mathbb{R}^n$ to (2.4) by calculating an increment, or step, $\Delta x_k \in \mathbb{R}^n$. If acceptable, Δx_k provides the iterate x_{k+1} for the next iteration:

$$x_{k+1} = x_k + \Delta x_k \tag{2.6}$$

The first iterate, x_0, (the starting point) is provided by the user.

Many numerical optimization algorithms calculate the step Δx_k by solving an approximated subproblem to (2.4), formed by using Taylor approximations to the functions that define (2.4). This results in either a linear optimization problem (that is, linear objective function and linear constraints):

$$\begin{array}{ll} \text{minimize} & a^\top \Delta x \\ \text{subject to} & A_\mathcal{I}\,\Delta x + b_\mathcal{I} \geq 0 \\ & A_\mathcal{E}\,\Delta x + b_\mathcal{E} = 0 \\ \text{with respect to} & \Delta x \in \mathbb{R}^n, \end{array} \tag{2.7}$$

or a quadratic optimization problem (that is, quadratic objective function and linear constraints):

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\Delta x^\top H \Delta x + a^\top \Delta x \\ \text{subject to} & A_\mathcal{I}\,\Delta x + b_\mathcal{I} \geq 0 \\ & A_\mathcal{E}\,\Delta x + b_\mathcal{E} = 0 \\ \text{with respect to} & \Delta x \in \mathbb{R}^n, \end{array} \tag{2.8}$$

Linear optimization problems are also referred to as linear programs (LP), and quadratic optimization problems are referred to as quadratic programs (QP). Hillier and Lieberman [33] describe methods for solving linear programs.

Algorithms that solve an optimization problem by generating a sequence of linear programs are referred to as sequential linear programming (SLP) algorithms, and algorithms that generate a sequence of quadratic programs are referred to as sequential quadratic programming (SQP) algorithms.

Global convergence of SLP and SQP algorithms can be ensured by establishing an upper limit ρ_k on Δx_k, such that $\|\Delta x_k\| \leq \rho_k$, where $\|\cdot\|$ is a suitable vector norm. Often $\|\cdot\|_\infty$ is used. The set of points

$$\mathcal{R}_k = \{\Delta x \in \mathbb{R}^n : \|\Delta x\| \leq \rho_k\} \tag{2.9}$$

is referred to as the *trust region.* The upper limit ρ_k is referred to as either a *move limit* or the *trust region radius.*

In order to evaluate the convergence properties of a numerical optimization algorithm, the rate of convergence is used. An algorithm is said to have linear convergence if the errors

$$e_k = x_k - x^*, \tag{2.10}$$

for two subsequent iterations are related in the following way:

$$\|e_{k+1}\| \leq \varepsilon \|e_k\| \quad \text{with} \quad 0 < \varepsilon < 1 \quad \text{and} \quad x_k \quad \text{close to} \quad x^*. \tag{2.11}$$

An algorithm is said to have quadratic convergence if the errors are related in the following way:

$$\|e_{k+1}\| \leq \varepsilon \|e_k\|^2 \quad \text{with} \quad 0 < \varepsilon < 1 \quad \text{and} \quad x_k \quad \text{close to} \quad x^*. \tag{2.12}$$

2.1.3 Multi-criteria optimization

Multi-criteria optimization concerns decision-making based on multiple criteria. This discipline is based on the fact that in general, it is not possible to find a decision that provides the optimum value for more than one decision criteria. For instance, it is unlikely that the most inexpensive building to construct is also the most energy efficient building.

If it is not possible to find optimum values for all decision criteria (which is often the case), then the decision-maker must accept a compromise between them. For instance, the most inexpensive building may require a large amount of energy, and the most energy efficient building may be quite expensive to construct. It may not be possible to find a design decision that is inexpensive and at the same time is energy efficient. The decision-maker must therefore accept a decision that is neither the most inexpensive nor the most energy efficient.

In this situation, the aim is to improve all decision criteria as much as possible. If a decision is found where it is not possible to improve one criterion without deteriorating one or more of the others, then that decision is denoted *Pareto efficient*. In other words, if it is possible to improve all decision criteria for a certain decision, then that decision is not Pareto efficient. This concept was first introduced by Pareto [52].

For any given decision problem, there usually exist a set of Pareto efficient decisions. This set is in the literature referred to as the *Pareto set*, the *Pareto frontier*, the *Pareto surface*, or the *efficient frontier*, among others. The main purpose of multi-criteria optimization is to provide methods for finding points belonging to the Pareto set, which can be used for making decisions based on multiple criteria.

The general theory of multi-criteria optimization, numerical methods for calculating the Pareto set, and examples of their applications to engineering, is described by Eschenauer [19]. Gembicki [26] describe a problem formulation called the *goal attainment problem*, which is widely used for calculating points in the Pareto set. Das and Dennis [16] provide an efficient method for calculating points in the Pareto set, which is called the *normal boundary intersection* (NBI) method.

2.1.4 Reliability analysis

Deterministic mathematical models of buildings are based on the assumption that all parameters that influence the performance of a building are known with full accuracy. In reality, however, there are uncertainties related to all parameters involved in models of buildings, such as climate parameters, prices for construction jobs, material properties such as concrete strength, etc.

It is therefore important to consider how to make decisions under uncertainties, which is addressed by reliability analysis. The main concern of reliability analysis is to find the probability of failure for a given design decision, and furthermore to find a design decision where this probability is below a certain level. The theory and methods of reliability analysis is described by Haldar and Mahadevan [32].

2.1.5 Sensitivity analysis

Sensitivity analysis for continuous optimization problems addresses the influence on the solution to an optimization problem, caused by changes in constant model parameters. This analysis can for instance be used for estimating the future development of optimum decisions, when constant parameters, such as climate parameters or prices, change over time. Requirements to the performance of buildings that are specified in building regulations, such as upper limits on the energy required by the building, are also represented by constant parameters. Sensitivity analysis may therefore be useful for estimating how optimum decisions change, if the building regulations are changed. Sensitivity analysis is described by Fiacco [21].

2.1.6 The space mapping technique

The space mapping technique was first introduced by Bandler et.al. [2]. A general introduction to the technique is provided by Bakr et.al. [9], and the many space mapping techniques developed over the years are reviewed by Bakr et.al. [8] and Bandler et.al. [4].

The space mapping technique is intended for optimization problems where either the objective or constraint functions, or both, are time-consuming or costly, and where the number of function evaluations required for solving the optimization problem therefore must be as small as possible.

The space mapping technique may therefore be useful when attempting to estimate optimum design decisions for buildings, based on accurate (and possibly time-consuming) mathematical models of the performance of buildings.

The space mapping technique requires the following two types of mathematical models to be available in order to solve an optimization problem:

The fine model, which is a detailed (and usually time-consuming or costly) mathematical model of the system to be optimized, and

The coarse model, which as a less detailed, but also less time-consuming and inexpensive model of the same system as the fine model.

The space mapping technique solves an optimization problem by using the coarse model to predict the location of the optimum for the fine model. In order to compensate for modeling errors in the coarse model, and to improve its prediction capabilities, the coarse model is modified by adjusting either the decision variables, constant model parameters, the output from the model, or a combination of these three types of parameters.

The various techniques developed for modifying the coarse model makes space mapping suitable not only for optimization, but also as a general tool for enhancing the accuracy of coarse models, and thereby providing mathematical models of systems or components, that are accurate and fast. A space mapping technique intended for enhancing the accuracy of a coarse model of a thermo-active building component is described in the paper included in Appendix B.

Initially, space mapping techniques modified the coarse model only by adjusting the decision variables. These techniques did not provide convergence to the fine model optimizer; however, they did converge to points close enough to the optimizer for practical purposes.

The first space mapping technique with provable convergence properties is provided by Madsen and Søndergaard [42]. This technique ensures convergence by performing a transition to a linear model of the fine model.

The interpolating surrogate technique, described by Bandler et.al. [7], aims at modifying the coarse model in such a way that the function value and the first partial derivatives match those of the fine model. The results obtained with this method indicate that it provides convergence to the fine model optimizer using only a very limited number of fine model function evaluations. However, a formal convergence theorem is not provided. The paper by Bandler et.al. [7] is included in Appendix A.

The interpolating surrogate technique has so far only been applied to unconstrained minimax optimization problems, that is, optimization problems in the form

$$\begin{array}{ll} \text{minimize} & \max\limits_{i=1,\ldots,m} \{f_i(x)\} \\ \text{with respect to} & x \in \mathbb{R}^n. \end{array} \tag{2.13}$$

In order to fully integrate the interpolating surrogate technique with the building optimization method described in this thesis, it must be able to solve optimization problems in the form (2.4). This means that the following concerns must be addressed:

1. The technique must be able to solve constrained optimization problems
2. It must be able to handle domain constraints
3. Finally, a gradient-free version of the technique is preferable.

These issues are not addressed in this study, but are recommended as further research.

2.2 The performance of buildings

There are many aspects of buildings that are relevant to take into account when assessing their performance, for instance:

1. The energy performance
2. The indoor environment
3. Economical aspects

In order to optimize the performance of buildings using numerical optimization methods, it is necessary to ensure that the performance can be expressed in terms of quantifiable measures. These measures are referred to as *performance measures* in this study. The background of the methods for calculating the performance measures for buildings is addressed in the following.

2.2.1 Energy and indoor environment

The energy performance of a building can be expressed in terms of, for instance, the annual amount of energy required for heating, cooling and ventilating the building, as well as energy for artificial lighting.

The performance with respect to the indoor environment can be expressed in terms of thermal comfort values, such as the predicted mean vote (PMV) or the predicted percentage of dissatisfied (PPD). The PMV and PPD values are proposed by Fanger [20]. The number of hours where overheating occurs can also be used as a measure for the quality of the indoor environment.

In order to calculate these performance measures, it is necessary to calculate solutions to the governing heat transfer equations for the building. The general theory of heat transfer, and numerical methods for solving heat transfer equations, is described by Patankar [53]. Hagentoft [31] furthermore describes aspects of lumped system analysis, which is commonly used when calculating the energy performance of buildings.

The method by Nielsen [50] calculates the energy performance of buildings, as well as the PMV and PPD values. This method is used for calculating the building performance with respect to energy and indoor environment in this study.

2.2.2 Economical aspects

The building performance with respect to economy can, for instance, be expressed as the cost of constructing, maintaining and operating the building.

These performance measures can be calculated using standard price catalogues for construction jobs. The V&S price catalogue [60] provides unit prices for construction jobs in Denmark. Furthermore, the V&S price catalogue [61] provides unit prices related to renovating and operating buildings in Denmark.

Calculating the cost of operating buildings furthermore requires energy prices, which, for instance, can be provided by national energy regulatory authorities.

2.3 Building optimization methods in the literature

Many different methods have been suggested over the years for optimizing the performance of buildings. The survey provided in this section concerns the most recent developments in this area.

Peippo et.al. [55] describe a method for finding the optimum technology mix for building projects. The method suggests parameters such as the shape of the building, the orientation, the amount of insulation and window areas, among others.

In order to find the optimum parameter values, the method uses a multivariate problem formulation, which includes the total annual cost for the building, as well as the total amount of energy required annually. The optimization problem is solved using cyclic coordinate search, as well as the direct search method proposed by Hooke and Jeeves [34].

The method described by Bouchlaghem [11] not only simulates the thermal performance of the building, but also applies numerical optimization techniques to determine the optimum design variables, which achieve the best thermal comfort conditions. The method takes into account design variables related to the buildings envelope and fabric, such as the plan aspect ratio, the orientation and the glazing ratio, among others.

The method is intended for finding the design decisions that provide the best thermal comfort level. Six different objective functions are investigated, which represent six different ways of quantifying the thermal comfort. Furthermore, the decision variables are subjected to linear constraints. The resulting constrained optimization problem is solved using a combination of the simplex method, described by Nelder and Mead [46], and the complex method described by Mitchell and Kaplan [45].

Caldas and Norford [25] describe a method for finding the width and height of windows that result in a building with the least amount of energy required for heating and artificial lighting. The optimization is based on results from detailed simulation software. The software automatically adjusts the amount of artificial lighting, such that the required illumination level is achieved. This results in an unconstrained optimization problem that is solved using a genetic algorithm. Genetic algorithms are described by Goldberg [27].

The method described by Jedrzejuk and Marks [35, 36, 37] decomposes the design problem into the following sub-problems: optimization of internal partitions, the shape of the building and finally coordination of the solutions. The shape of the building is represented by parameters such as wall lengths, number of storeys, ratios of window to wall areas, among others.

The method is based on a constrained multi-criteria formulation, which uses the construction costs, the seasonal demand for heating energy, and the pollution emitted by heat sources, as objective functions. The optimization problem is solved using a combination of analytical and numerical methods.

The method described by Nielsen and Svendsen [47] finds optimum decisions regarding the amount of insulation, the type of glazing, the window fraction of the external walls, among others. The method uses a constrained optimization formulation, where the life cycle cost of the building is used as objective function. Furthermore, the energy required by the building, and the number of hours where overheating occur, are subjected to upper limits. Finally, the daylight factor is subjected to a lower limit. The resulting optimization problem consists of discrete as well as continuous variables.

The optimization problem is solved using the simulated annealing method by Gonzalez-Monroy and Cordoba [29] for optimizing the discrete parameters, and the method by Hooke and Jeeves [34] for optimizing the continuous variables.

Wright et.al. [63] describe a method for optimizing the design and operation of a HVAC system. The decision variables include design parameters such as the coil width and height and the number of rows, as well as control parameters such as the supply air temperature, the air flow rate, and the on/off status of the system.

The method uses a multi-criteria formulation, using the operating cost of the system and the maximum thermal discomfort as objective functions. The method uses the so-called *simple genetic algorithm* described by Goldberg [27] for solving the optimization problem. The method provides a set of Pareto optimal points, which can be used for investigating the pay-off between the two objectives.

The method by Wang et.al. [62] is intended for green building design. It finds optimum decisions regarding the orientation, the plan aspect ratio, the window to wall ratio, among others. The method is based on a multi-criteria formulation, using the life cycle cost (LCC), and the life cycle environmental impact (LCEI) as objective functions. Furthermore, the continuous decision variables are subjected to box constraints, and the discrete variables to so-called selection constraints.

The optimization problem is solved using the multi-objective genetic algorithm by Fonseca and Flemming [24]. The method provides the Pareto set for the two objectives, which can be used for assessing the level of compromise between optimizing economical aspects of the building, and optimizing the environmental impact of the building.

The above mentioned studies consider the following decision variables (among others) to have a significant impact on the performance of buildings:

1. The shape of the building, expressed for instance in terms of the plan aspect ratio and the number of floors
2. The orientation of the building
3. The amount of insulation used in the building envelope
4. The window areas relative to the area of the external walls
5. The window type
6. The window shape
7. The design and operation of HVAC systems.

These variables are therefore suitable for optimization. The following performance measures (among others) are considered:

1. The amount of energy required for heating, cooling and ventilating the building, as well as energy for domestic hot water and energy for artificial lighting
2. The level of thermal comfort
3. The level of daylight utilization
4. The number of hours with overheating
5. The cost of constructing the building

6. The cost of operating the building
7. The life cycle cost of the building
8. The environmental impact of the building.

The considered studies use many different problem formulations in order to optimize the performance of buildings. There exist both single- and multi-criteria formulations, as well as unconstrained and constrained formulations. Furthermore, different performance measures are used differently in the studies. The solutions provided by the different studies are optimized either with respect to energy, economy, thermal comfort or environmental impact. This observation supports the idea that it is advantageous to develop flexible building optimization methods that enable the decision maker to optimize any aspect of the building performance.

2.4 Delimitation of the study

This study only concerns building optimization methods that support design decisions using a single-criterion formulation, with constraints. The study focuses on methods intended for the early stages of a design process. This means that methods based on detailed building models are not considered.

The design decisions considered in this study are the shape of the building, the amount of insulation used in the building envelope, and the type and relative area of the windows, compared with the area of the external walls. Decisions regarding the design and operation of HVAC systems are not considered.

The performance with respect to energy, indoor environment and economy are considered, but not the performance with respect to the environmental impact. Only non-residential buildings are considered.

Reliability and sensitivity analysis are not considered. Programming specific details, such as choosing programming language, developing graphical user interfaces, including computer-aided design modeling environments, are not considered. Furthermore, developing database management systems for managing the data needed for representing buildings is also not considered.

Chapter 3

A mathematical model of decisions

The purpose of this chapter is to provide a description of how optimization can be used for estimating optimum design decisions for systems such as buildings, governed by, for instance, partial or ordinary differential equation. A general description of design decision problems is provided, which aims at simplifying the process of translating such problems into optimization problems. Furthermore, an optimization problem is formulated, which represent a wide range of building design decision problems.

The governing equations for the considered system should ideally be solved analytically, but this is not possible in general. If analytical solutions are unavailable, the governing equations can be solved using numerical methods, which are often implemented in specialized simulation software. It is therefore necessary to consider ways in how to combine simulation software with optimization methods. An interface is described that addresses this issue.

3.1 Building design decision problems

Decisions made during the design process have consequences for the performance of the building. For instance, decisions regarding the shape of the building, the total window area and the amount of insulation material used in the building have consequences for the energy performance of the building, the economy of the building and the quality of the indoor environment.

The design decisions, as well as the performance of the building, can be subjected to requirements. For instance, the above mentioned decisions can be subjected to upper or lower limits, and they can be required to assume certain values. They can furthermore be required to be related to each other in certain ways. The window areas on the north- and south-facing façades can for instance be required to be equal.

Similar requirements can be applied to the performance of the building. Any measure representing the performance of the building, for instance the annual amount of energy required by the building, or the cost of constructing or operating the building, can be subjected to upper or lower bounds, they can be required to assume certain values, or they can be required to be related to each other in certain ways.

In addition, the performance can be required to be optimized, meaning that a measure representing the performance of the building can be required to assume its minimum or maximum value. For instance, the decision maker may wish to estimate the set of design decisions that provide the building with the least amount of energy required for heating and cooling, or the building with the smallest construction cost.

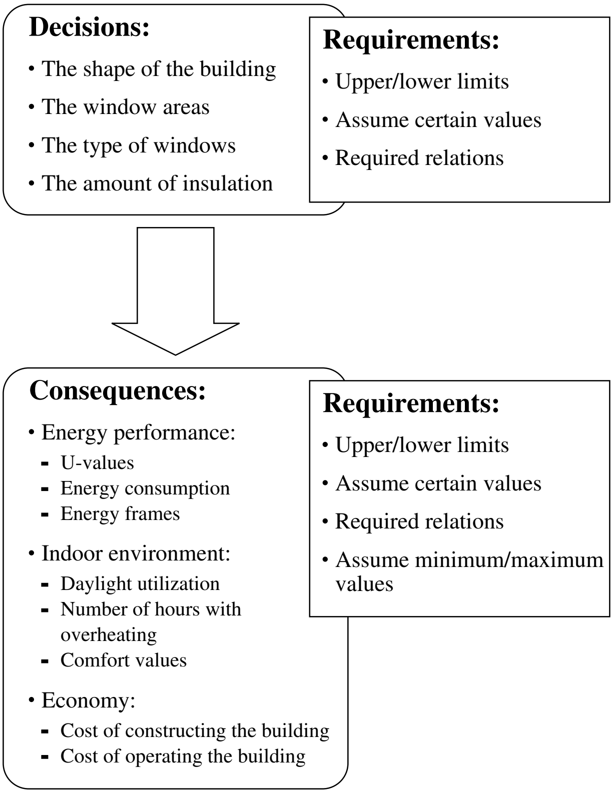


Figure 3.1: A conceptual illustration of a decision problem.

A conceptual illustration of a decision problem is shown in Figure 3.1. The figure furthermore provides examples of decisions, consequences and requirements that are relevant for design decisions for buildings. These concepts are defined in the following.

Decisions refers to the set of variables that the decision maker wishes to determine optimum values for. It is assumed that the decision maker has full control over them. Significant decision variables for buildings include (but is not limited to) the amount of insulation used in various building components, the area and type of windows used, the overall shape of the building, expressed in terms of, for instance, the width to length ratio, and the number of floors.

Consequences refers to quantifiable parameters that depend on the decision variables, and that can be used as measures for the performance of the system. Consequences of decisions are therefore also referred to as *performance measures* in the following. This concept is similar to the concept *utility function* used in operations research.

There are many performance measures that are relevant for buildings, for instance, the energy consumption of the building, the cost of constructing the building, the quality of the indoor environment, structural properties, the environmental impact, etc.

Requirements to decision variables and performance measures can be expressed in many different ways. For instance, they can be required to assume their maximum or minimum value, they can be subjected to upper or lower bounds, they can be required to assume specific values, or they can be required to be related to each other in certain ways. The performance measures can furthermore be required to assume their maximum or minimum value.

The following types of requirements are considered:

Optimality requirements: When an individual performance measure, or a linear combination hereof, is required to assume its maximum or minimum value.

Inequality requirements: When an individual decision variable or performance measure, or a linear combination hereof, are subjected to upper or lower bounds.

Equality requirements: When an individual decision variable or performance measure, or a linear combination hereof, is required to assume a specific value.

Feasibility: The decision variables are required to be feasible, meaning that it must be possible to assess the consequences of them. This requirement is relevant when combining simulation software with optimization methods, since it for instance can be used for preventing the optimization algorithm from performing simulations using input that has no physical meaning.

Optimality requirements can in theory also be applied to decision variables, but this possibility does not seem to be of any practical use. Furthermore, the considered inequality and equality requirements involve either decision variables or performance measures, but not both.

Estimating the consequences of design decisions for buildings often involve a large number of constant parameters, such as the location and orientation of the building, climate parameters, prices for construction jobs, and physical properties of building components and materials. The constant parameters are not included in this formulation of decision problems.

3.1.1 An example

Assume that a decision maker is required to estimate the amount of insulation and the window area for two façades of a building, such that the following requirements are satisfied:

1. The construction cost must be as small as possible.
2. The total amount of energy required for maintaining a satisfactory indoor environment must be below a certain level.
3. The utilization of natural light (expressed in terms of the daylight factor) must assume a required level.
4. The window area must be the same for both façades.

In this case, the decision variables are:

1. The amount of insulation
2. The window area for the first façade
3. The window area for the second façade.

The performance measures that are relevant for this decision problem are:

1. The total expenses used for constructing the building
2. The total amount of energy required for maintaining a satisfactory indoor environment
3. The daylight factor.

The first requirement to the building can be interpreted as an optimality requirement, the second one as an inequality requirement, and the third one as an equality requirement. The last requirement specifies a required relation between two decision variables.

Furthermore, if an optimization algorithm is used for estimating a design decision that solves the decision problem, then the following feasibility requirements ensure that the algorithm never attempts to estimate consequences of decisions that can not be carried out physically:

1. The amount of insulation must be positive
2. The window areas of the two façades must be positive and below the areas of the corresponding façades.

The next section concerns the details of how decision problems can be expressed as optimization problems.

3.2 Mathematical interpretation

In order to provide a mathematical interpretation of decision problems, it is assumed that decisions, consequences and requirements are quantifiable, and that the requirements can be expressed as linear functions of either decision variables or performance measures.

The consequences of a decision are represented by the vector-valued function $q : \mathcal{D} \to \mathbb{R}^{n_q}$, where $\mathcal{D} \subseteq \mathbb{R}^n$. The consequences depend on the decision variables, denoted $x \in \mathcal{D}$. The consequences, or performance measures, are obtained by calculating the function values $q(x)$.

Furthermore, the domain $\mathcal{D}$ of the performance measures is expressed in terms of domain constraint functions:

$$\mathcal{D} = \{x \in \mathbb{R}^n : d(x) \geq 0\}\,. \tag{3.1}$$

In the following it is described how requirements to decisions and consequences can be specified in terms of decision variables x and performance measures $q(x)$.

3.2.1 Requirements to decisions

In order to ensure that the decision variables are feasible, they are required to belong to the domain of the performance measures:

$$d(x) \geq 0. \tag{3.2}$$

The following linear inequality and equality requirements are applied to the decision variables:

$$A_{\hat{\mathcal{I}}} \cdot x \geq b_{\hat{\mathcal{I}}}, \tag{3.3}$$

and

$$A_{\hat{\mathcal{E}}} \cdot x = b_{\hat{\mathcal{E}}}, \tag{3.4}$$

where $A_{\hat{\mathcal{I}}} \in \mathbb{R}^{n_{\hat{\mathcal{I}}} \times n_q}$, $b_{\hat{\mathcal{I}}} \in \mathbb{R}^{n_{\hat{\mathcal{I}}}}$, $A_{\hat{\mathcal{E}}} \in \mathbb{R}^{n_{\hat{\mathcal{E}}} \times n_q}$ and $b_{\hat{\mathcal{E}}} \in \mathbb{R}^{n_{\hat{\mathcal{E}}}}$. The parameters $n_{\hat{\mathcal{I}}}$ and $n_{\hat{\mathcal{E}}}$ are the numbers of inequality and equality requirements for the decision variables, respectively.

The relation (3.3) can for instance be used for specifying upper or lower bounds on the decision variables. The relation (3.4) can be used for specifying required linear relations between the decision variables, for instance that some decision variables must be identical. It can furthermore be used for specifying required values for decision variables.

3.2.2 Requirements to consequences

The optimality requirements can be specified using the following expression as objective function for an optimization problem:

$$a_{\mathcal{O}}{}^{\top} q(x), \tag{3.5}$$

where $a_{\mathcal{O}} \in \mathbb{R}^{n_q}$.

The inequality and equality requirements can be specified using the relations

$$A_{\mathcal{I}} \cdot q(x) \geq b_{\mathcal{I}}, \tag{3.6}$$

and

$$A_{\mathcal{E}} \cdot q(x) = b_{\mathcal{E}}, \tag{3.7}$$

as inequality and equality constraints, respectively, for an optimization problem, where $A_{\mathcal{I}} \in \mathbb{R}^{n_{\mathcal{I}} \times n_q}$, $b_{\mathcal{I}} \in \mathbb{R}^{n_{\mathcal{I}}}$, $A_{\mathcal{E}} \in \mathbb{R}^{n_{\mathcal{E}} \times n_q}$ and $b_{\mathcal{E}} \in \mathbb{R}^{n_{\mathcal{E}}}$. The parameters $n_{\mathcal{I}}$ and $n_{\mathcal{E}}$ are the numbers of inequality and equality requirements for the performance measures, respectively.

3.2.3 An optimization problem for modeling decisions

The requirements to decision variables and performance measures can be expressed as the following optimization problem:

$$\begin{array}{lll} \text{minimize} & {a_{\mathcal{O}}}^\top q(x) & \left.\begin{array}{l} \\ \\ \\ \end{array}\right\} \text{requirements to consequences} \\ \text{subject to} & A_{\mathcal{I}} \cdot q(x) \geq b_{\mathcal{I}} & \\ & A_{\mathcal{E}} \cdot q(x) = b_{\mathcal{E}} & \\ & A_{\hat{\mathcal{I}}} \cdot x \geq b_{\hat{\mathcal{I}}} & \left.\begin{array}{l} \\ \\ \\ \end{array}\right\} \text{requirements to decisions} \\ & A_{\hat{\mathcal{E}}} \cdot x = b_{\hat{\mathcal{E}}} & \\ & d(x) \geq 0 & \\ \text{with respect to} & x \in \mathbb{R}^n & \end{array} \tag{3.8}$$

This formulation allows the decision maker to optimize any performance measure, or linear combinations hereof, and to specify linear equality or inequality requirements to decision variables, and to performance measures.

It is possible to specify requirements that render the optimization problem (3.8) infeasible. In this case, the decision maker must manually remove or relax the requirements until the feasible region of (3.8) becomes non-empty. Developing methods for addressing this issue is suggested as a possible topic for further research.

3.2.4 The example revisited

The decision variables for the problem described in Section 3.1.1 can be arranged as the following vector:

$$x = \begin{bmatrix} \text{The amount of insulation} \\ \text{The window area for the first façade} \\ \text{The window area for the second façade} \end{bmatrix} \tag{3.9}$$

and the performance measures can be arranged as:

$$q(x) = \begin{bmatrix} \text{The total expenses used for constructing the building} \\ \text{The total amount of energy required} \\ \text{The daylight factor} \end{bmatrix} \tag{3.10}$$

Requiring that the construction cost must be minimal can be expressed as

$$a_{\mathcal{O}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \tag{3.11}$$

since the objective function (3.5) in this case becomes:

$$a_{\mathcal{O}}{}^{\top} q(x) = q_1(x), \tag{3.12}$$

where $q_1(x)$ is the construction cost, according to the definition (3.10).

Subjecting the total amount of energy to an upper limit E_{max} can be done using

$$A_{\mathcal{I}} = [\ 0,\ -1,\ 0\ ] \quad \text{and} \quad b_{\mathcal{I}} = -E_{max}, \tag{3.13}$$

since (3.6) in this case becomes:

$$A_{\mathcal{I}} \cdot q(x) \geq b_{\mathcal{I}} \quad \Leftrightarrow \quad -q_2(x) \geq -E_{max} \quad \Leftrightarrow \quad q_2(x) \leq E_{max}, \tag{3.14}$$

where $q_2(x)$ is the total amount of energy required by the building.

Requiring that the daylight factor assumes a specified level DF^* can be expressed as

$$A_{\mathcal{E}} = [\ 0,\ 0,\ 1\ ] \quad \text{and} \quad b_{\mathcal{E}} = DF^*, \tag{3.15}$$

since (3.7) then becomes:

$$A_{\mathcal{E}} \cdot q(x) = b_{\mathcal{E}} \quad \Leftrightarrow \quad q_3(x) = DF^*. \tag{3.16}$$

where $q_3(x)$ is the daylight factor.

There are no inequality requirements to the decision variables, hence

$$A_{\hat{\mathcal{I}}} = \emptyset \quad \text{and} \quad b_{\hat{\mathcal{I}}} = \emptyset. \tag{3.17}$$

Requiring that the window areas for the two façades are equal can be expressed as

$$A_{\hat{\mathcal{E}}} = [\ 0,\ 1,\ -1\ ] \quad \text{and} \quad b_{\hat{\mathcal{E}}} = 0, \tag{3.18}$$

since (3.4) then becomes:

$$A_{\hat{\mathcal{E}}} \cdot x = b_{\hat{\mathcal{E}}} \quad \Leftrightarrow \quad x_2 - x_3 = 0 \quad \Leftrightarrow \quad x_2 = x_3. \tag{3.19}$$

The feasibility requirements can be implemented as the following domain constraint function:

$$d(x) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ A_f^{(1)} - x_2 \\ A_f^{(2)} - x_3 \end{bmatrix} \tag{3.20}$$

where $A_f^{(1)}$ and $A_f^{(2)}$ are the two façade areas. The requirement (3.2) is equivalent with:

$$d(x) \geq 0 \quad \Leftrightarrow \quad \begin{aligned} x_1 &\geq 0 \\ x_2 &\geq 0 \\ x_3 &\geq 0 \\ A_f^{(1)} - x_2 &\geq 0 \\ A_f^{(2)} - x_3 &\geq 0 \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} x_1 &\geq 0 \\ x_2 &\geq 0 \\ x_3 &\geq 0 \\ A_f^{(1)} &\geq x_2 \\ A_f^{(2)} &\geq x_3 \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} x_1 &\geq 0 \\ 0 &\leq x_2 \leq A_f^{(1)} \\ 0 &\leq x_3 \leq A_f^{(2)} \end{aligned} \tag{3.21}$$

This means, as expected, that the amount of insulation x_1 is required to be positive, and that the window areas x_2 and x_3 are required to be positive and below the façade areas $A_f^{(1)}$ and $A_f^{(2)}$, respectively.

3.3 Interfacing with simulation software

The consequences of decisions are found by solving the governing equations for the considered system. When analytical solutions are not available, they can be estimated using numerical methods implemented in simulation software.

This means that an interface is required between the optimization and simulation software. In the following a structure for a simulation software interface is described. The structure is described in general terms, and programming specific details are omitted.

The purpose of the interface is to implement the expression $q(x)$, which can be achieved using the following three steps:

1. Prepare the input for the simulation software, based on the decision variables x and constant parameters.

2. Perform the simulation.

3. Calculate the performance measures $q(x)$, based on the output from the simulation software.

These steps can be interpreted as the following composed mapping:

$$q = q^{(o)} \circ q^{(s)} \circ q^{(i)}. \tag{3.22}$$

The functions $q^{(o)}$, $q^{(s)}$ and $q^{(i)}$ are described in the following, and the structure of the simulation software interface is illustrated in Figure 3.2.

The first step is necessary in situations where the decision variables are not identical with the input required by the simulation software. If, for instance, parameters such as the width to length ratio of the building are used as decision variables, but the simulation software requires the actual width and length of the building as input, then a translation is required from the decision variables to the required input. This translation can be interpreted as a function $q^{(i)} : \mathcal{D} \to \mathbb{R}^{n_i}$.

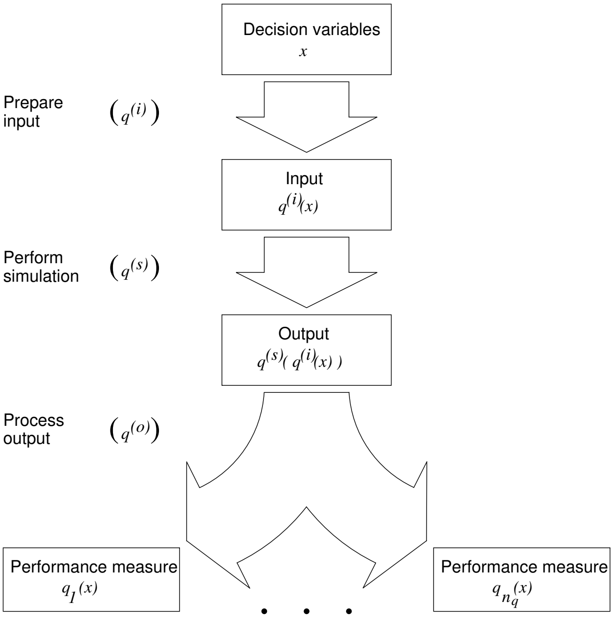


Figure 3.2: The general structure of an interface to the simulation software. The boxes represent variables and parameters, and the arrows represent software routines.

The next step consists of invoking the simulation software. This can be interpreted as a function $q^{(s)} : \mathbb{R}^{n_i} \rightarrow \mathbb{R}^{n_o}$ that maps the input to the simulation software, to the output.

Simulation software often generates a large amount of output, which in itself is not necessarily suitable for comparing different decisions, and therefore is not suitable for decision making. The purpose of the last step is therefore to process the output from the simulation software, in order to reduce it to a manageable set of performance measures. This can be interpreted as a function $q^{(o)} : \mathbb{R}^{n_o} \rightarrow \mathbb{R}^{n_r}$, that maps the output to the performance measures.

3.4 Final remarks

The mathematical model of decisions enables the decision maker to specify requirements to decisions and consequences in a variety of ways. It enables the decision maker to optimize any performance measure, and to specify linear relations between decision variables, and between performance measures.

The terminology used for describing decision problems consists of three basic concepts: Decisions, Consequences and Requirements. The intention is to make it easier to model decision problems as optimization problems, and thereby to enable all parties involved in developing and implementing building performance optimization methods, to communicate their concerns effectively.

If numerical methods are required for solving the governing equations for the considered system, then the numerical optimization method needs to address the following concerns:

1. The partial derivatives of the performance measures are (usually) not available. This makes it impractical to use gradient-based optimization methods, which require this information to be available. In this case, the partial derivatives are usually approximated using finite difference approximations, which increases the time used for solving the optimization problem.
2. The input to the simulation software must be feasible. The optimization method must be prevented from calculating the performance measures for infeasible decisions variables, since the simulation software may become unstable in this situation.
3. The time used for evaluating the performance measures may be excessive and/or costly. In this situation, the optimization method should attempt to calculate the performance measures as few times as possible, in order to reduce the time used for solving the optimization problem.

It has not been possible during this study to develop an optimization algorithm that addresses all three concerns. However, the gradient-free SQP filter algorithm described in Chapter 5 addresses the first two concerns, and the space mapping interpolating surrogate algorithm described in the paper included in Appendix A addresses the last one.

Furthermore, the space mapping modeling technique described in the paper included in Appendix B, may also be useful for reducing the amount of time required for calculating the performance of a building.

Chapter 4

A method for optimizing the performance of buildings

This chapter concerns a method intended for suggesting optimum design decisions in the early stages of the design process for buildings. The method is able to suggest decisions regarding the geometry of the building, the amount of insulation used in various building components, as well as the type of windows.

The suggestions made by the method are based on performance measures representing the energy performance of the building, the economy of the building, and the indoor environment of the building. The formulation (3.8) is used for estimating the design decisions.

4.1 Introduction

The building optimization method estimates optimum design decisions by optimizing the performance of a building with a simplified geometry. This section describes the simplified building, the design decisions and performance measures supported by the method, and the constant parameters. Furthermore, details regarding the requirements to the decision variables and performance measures are addressed.

In order to combine performance calculations with optimization methods, an interface with a structure as described in Section 3.3 is used. The interface is illustrated in Figure 4.1.

The methods used for calculating the performance of the simplified building are described in Section 4.2, and the details of preparing the input to and processing the output from these methods are described in Sections 4.3 and 4.4, respectively. The domain constraints used for ensuring that the input to the performance calculation methods is feasible, are described in Section 4.5. Concluding remarks are provided in Section 4.6.

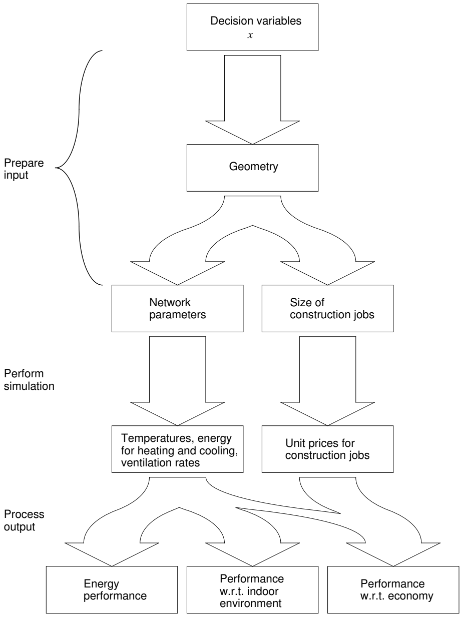

Decision variables
x
Prepare input
Geometry
Network parameters
Size of construction jobs
Perform simulation
Temperatures, energy for heating and cooling, ventilation rates
Unit prices for construction jobs
Process output
Energy performance
Performance w.r.t. indoor environment
Performance w.r.t. economy

Figure 4.1: A conceptual illustration of the interface used for calculating the performance measures.

4.1.1 A simplified building model

The consequences of design decisions are estimated using a building with a simplified geometry, as shown in Figure 4.2.

The building has a rectangular layout, as shown in Figure 4.3. All floors are identical, and each floor has window "bands" on two of the four external walls. The staircase tower is omitted, and only a single internal wall is included, which divide the building into two thermal zones, representing the front and back of the building. The performance of each of the two thermal zones is calculated separately.

The building has an annular foundation, which is included when calculating the building economy, but not when calculating the energy performance. The foundation is not shown in Figure 4.2.

The orientation of the building is defined as the counter-clockwise angle from due south to the main axis of the building.

The thicknesses of the insulated and uninsulated parts of the building are shown in the layout in Figure 4.3 and the cross-section A-A shown in Figure 4.4.

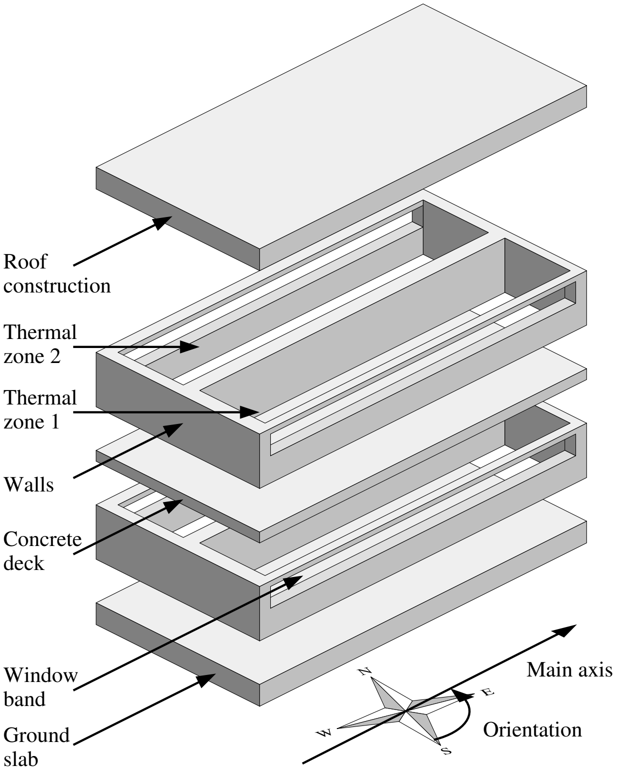


Figure 4.2: The building geometry used by the building optimization method for estimating the consequences of design decisions. This example only consists of two floors, but the method supports any number of floors.

4.1.2 Decision variables

The aim of the building optimization method is to make suggestions for the following design decisions:

1. The overall shape of the building.
2. The window fraction of the façade areas.
3. The window types.
4. The amount of insulation in ground slab, external walls and roof construction.

The shape of the building is expressed in terms of the width to length ratio ϱ:

$$\varrho = \frac{w_{\text{ext}}}{l_{\text{ext}}}. \tag{4.1}$$

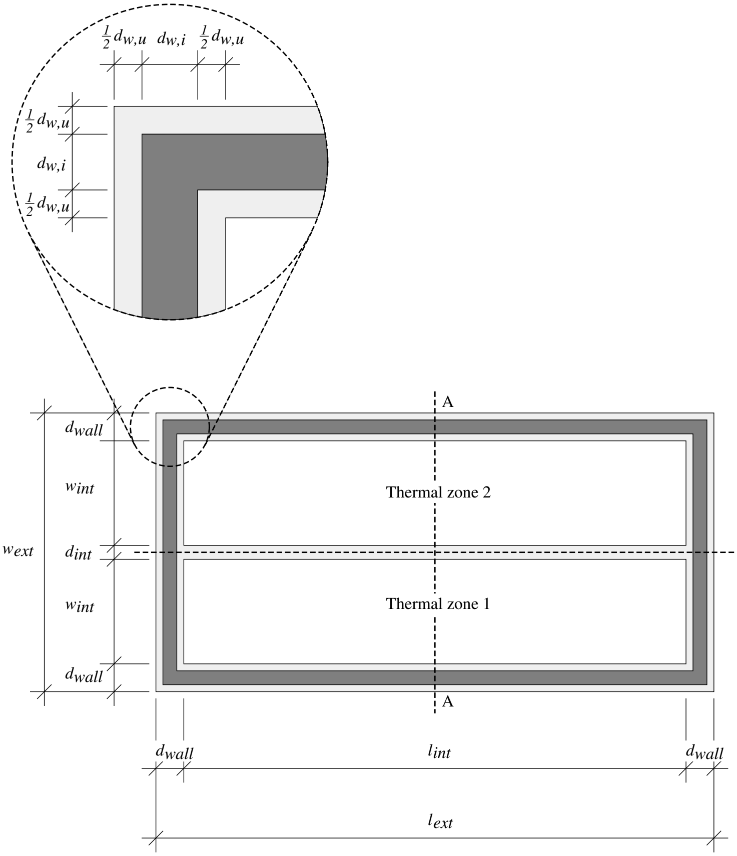


Figure 4.3: The layout of the building. Dark gray regions represent the insulated parts of the building, and light gray regions represent the uninsulated parts. The horizontal dashed line divides the building into two thermal zones.

The shape of the building is furthermore represented by the number of floors, N.

The window fraction of the façades for the two thermal zones are represented by the parameters $\sigma^{(1)}$ and $\sigma^{(2)}$, which are defined as the ratios between the window areas and the façade areas:

$$\sigma^{(1)} = \frac{A^{(1)}_{\text{win}}}{h_{\text{ext}}\, l_{\text{int}}}, \quad \text{and} \quad \sigma^{(2)} = \frac{A^{(2)}_{\text{win}}}{h_{\text{ext}}\, l_{\text{int}}}. \tag{4.2}$$

The window areas $A^{(1)}_{\text{win}}$ and $A^{(2)}_{\text{win}}$ for the two thermal zones, and the façade area $h_{\text{ext}}\, l_{\text{int}}$ apply to one floor.

In order to determine which window to use, the decision maker must provide a database consisting of, say, n_{win} windows. The decision maker selects a window by assigning weights

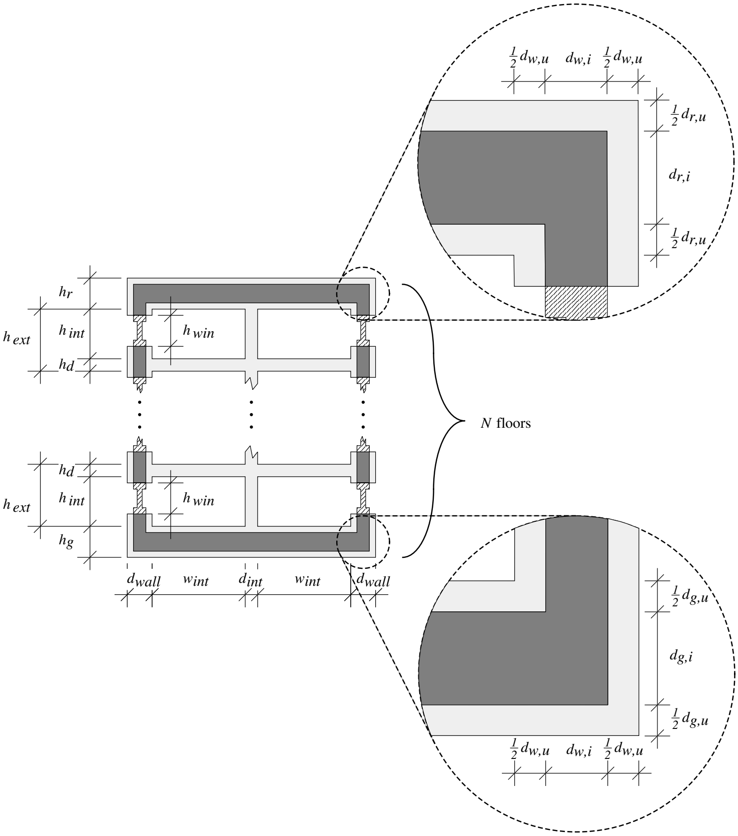


Figure 4.4: The cross-section A-A, defined in Figure 4.3.

$0 \leq \alpha_i \leq 1$ to the windows, where $i = 1, \ldots, n_{\text{win}}$. The weights $\alpha_j = 1$ and $\alpha_i = 0$ for $i \neq j$ specifies that window j is selected.

Two sets of weights, $\alpha^{(1)} \in \mathbb{R}^{n_{\text{win}}}$ and $\alpha^{(2)} \in \mathbb{R}^{n_{\text{win}}}$, represent the windows selected for the two thermal zones. These weights are used for calculating the window properties used as part of the input to the performance calculation methods, by forming a weighted sum of the properties of the windows stored in the database.

The amount of insulation used in the ground slab, the external walls and the roof construction, is represented by the parameters $d_{g,i}$, $d_{w,i}$ and $d_{r,i}$, respectively.

The full set of decision variables $x \in \mathbb{R}^{n_d}$ supported by the building optimization method is:

$$x = \begin{bmatrix} \varrho \\ N \\ \sigma^{(1)} \\ \sigma^{(2)} \\ d_{g,i} \\ d_{w,i} \\ d_{r,i} \\ \alpha^{(1)} \\ \alpha^{(2)} \end{bmatrix} \tag{4.3}$$

The size of the vector is

$$n_d = 2\, n_{\text{win}} + 7. \tag{4.4}$$

4.1.3 Constant parameters

Evaluating the consequences of design decisions for the simplified building involves a number of constant parameters, such as:

1. Various geometrical parameters, such as the thickness of the uninsulated parts of the ground slab, the external walls and the roof construction, as well as the thickness of the internal wall. The total heated floor area of the building is also not altered.
2. The position and orientation of the building.
3. Settings for the HVAC systems.
4. Physical properties, such as the thermal conductivity of the materials used in the building, as well as the physical properties of the windows.
5. Economical parameters, used for calculating the cost of constructing the building.
6. Climate parameters, such as external air temperature, air velocity, air pressure, relative humidity, etc. These parameters can be found in test reference year (TRY) or design reference year (DRY) data sets.

A detailed list of the default values of the constant parameters is provided in Appendix D. These constants are used when evaluating the building optimization method in Chapter 6, unless specified otherwise.

4.1.4 Performance measures

The energy performance of the building is represented by the following performance measures:

1. The annual amount of energy required by the building for heating, cooling, ventilation, domestic hot water and artificial lighting (Q_{tot})

2. Energy frame calculation required by the Danish building regulations (EF_3)

3. Energy frame calculation for acquiring the low energy class 2 label (EF_2)

4. Energy frame calculation for acquiring the low energy class 1 label (EF_1)

5. Heat loss through the building envelope, excluding windows and doors (Q_{be})

6. The thermal transmittance for the ground slab (U_g)

7. The thermal transmittance for the external walls (U_{wall})

8. The thermal transmittance for the roof construction (U_r)

9. The thermal transmittance for the windows used in zone 1 ($U_{win}^{(1)}$)

10. The thermal transmittance for the windows used in zone 2 ($U_{win}^{(2)}$).

The indoor environment of the building is represented by the following performance measures:

11. Annual number of overheating hours for zone 1 ($OH^{(1)}$)

12. Annual number of overheating hours for zone 2 ($OH^{(2)}$)

13. The ratio between the depth of the room and the window height for zone 1 ($DH^{(1)}$)

14. The ratio between the depth of the room and the window height for zone 2 ($DH^{(2)}$).

The economy of the building is represented by the following performance measures:

15. The cost of constructing the building (C_{con})

16. The annual cost of operating the building (C_{op})

The performance measures are represented by the function $q : \mathcal{D} \to \mathbb{R}^{n_q}$, where the number of performance measures is $n_q = 16$. The full set of performance measures supported by the building optimization method is:

$$q(x) = \begin{bmatrix} Q_{tot} \\ EF_3 \\ EF_2 \\ EF_1 \\ BE \\ U_g \\ U_{wall} \\ U_r \\ U_{win}^{(1)} \\ U_{win}^{(2)} \\ OH^{(1)} \\ OH^{(2)} \\ DH^{(1)} \\ DH^{(2)} \\ C_{con} \\ C_{op} \end{bmatrix} \tag{4.5}$$

where the arguments to the performance measures are omitted. The methods used for calculating the performance measures are described in Section 4.2.

4.1.5 Requirements

The building optimization method suggests design decisions by estimating numerical solutions to (3.8). This formulation enables the decision maker to specify optimality requirements to performance measures, and to specify inequality and equality requirements to performance measures as well as decision variables.

However, there are requirements related to discrete decisions, such as selecting a window from a window database, which always must be satisfied, regardless of the requirements specified by the decision maker. These are described in the following. Furthermore, how to incorporate the requirements to the energy performance of buildings into the decision-making is also addressed.

Discrete decisions

As described in Section 4.1.2, window j is selected by using the weights $\alpha_j = 1$ and $\alpha_i = 0$ for $i \neq j$. The weights are therefore always either 0 or 1, that is, $\alpha_i \in \{0, 1\}$ for $i = 1, \dots, n_{win}$.

However, the problem (2.4) requires that the variables are continuous, therefore the weights are allowed to assume any value between 0 and 1, that is:

$$0 \leq \alpha^{(1)} \leq 1, \tag{4.6}$$

$$0 \leq \alpha^{(2)} \leq 1. \tag{4.7}$$

There is no guarantee that the optimum weights found by solving (2.4) are integers; however, practical experience with the method indicates that this is often the case.

The requirements (4.6) and (4.7) are used as permanent parts of $A_{\hat{\mathcal{I}}}$ and $b_{\hat{\mathcal{I}}}$, which are used for specifying inequality requirements to the decision variables.

Furthermore, in order to ensure that exactly one window is selected for each of the two thermal zones, the sums of the weights must be one:

$$\sum_{i=1}^{n_{\text{win}}} \alpha_i^{(1)} = 1, \tag{4.8}$$

$$\sum_{i=1}^{n_{\text{win}}} \alpha_i^{(2)} = 1. \tag{4.9}$$

These requirements are used as permanent parts of $A_{\hat{\mathcal{E}}}$ and $b_{\hat{\mathcal{E}}}$, which are used for specifying equality requirements to the decision variables.

Building regulations

The Danish building regulations [13] specify upper limits on the annual amount of energy required by the building, as well as upper limits on the (linear) thermal transmittance for various parts of the building envelope.

These regulations implement the EU Directive [18] on the energy performance of buildings. The approach described in the following, intended for incorporating energy requirements to buildings into the decision-making, may therefore be useful for other countries with similar regulations.

However, the reader should notice that the approach described in the following does not guarantee that the design decision suggested by the building optimization method comply with the Danish building regulations. This is because the method does not use the energy performance calculation method required by the building regulations. The required method is implemented in the program BE06 described by Aggerholm and Grau [1]. Including this method into the building optimization method is a possible topic for further research.

The Danish requirements to the energy performance of non-residential buildings are described in terms of the total annual energy Q_{tot} delivered to the building for heating, cooling, ventilation, producing domestic hot water, and for artificial lighting. The following energy frame requirement (referred to as EF_3) must be satisfied for new non-residential buildings:

$$\frac{Q_{tot}}{A_{tot}} \leq 95 \frac{\text{kWh}}{\text{m}^2} + \frac{2200\,\text{kWh}}{A_{tot}}, \tag{4.10}$$

where A_{tot} is the total heated floor area, including internal and external walls.

The Danish building regulations also include energy labels, intended for motivating building owners to reduce the energy required by buildings even further. The following energy frame (referred to as EF_2) requirement must be satisfied in order to achieve the low energy class 2 label:

$$\frac{Q_{tot}}{A_{tot}} \leq 50\frac{\mathrm{kWh}}{\mathrm{m}^2} + \frac{1600\,\mathrm{kWh}}{A_{tot}}, \tag{4.11}$$

and the energy frame requirement (referred to as EF_1) for achieving the low energy class 1 label is:

$$\frac{Q_{tot}}{A_{tot}} \leq 35\frac{\mathrm{kWh}}{\mathrm{m}^2} + \frac{1100\,\mathrm{kWh}}{A_{tot}}. \tag{4.12}$$

The energy frame requirements can be formulated in the following way:

$$\begin{align} 95\frac{\mathrm{kWh}}{\mathrm{m}^2}A_{tot} + 2200\,\mathrm{kWh} - Q_{tot} &\geq 0 \tag{4.13}\\ 50\frac{\mathrm{kWh}}{\mathrm{m}^2}A_{tot} + 1600\,\mathrm{kWh} - Q_{tot} &\geq 0 \tag{4.14}\\ 35\frac{\mathrm{kWh}}{\mathrm{m}^2}A_{tot} + 1100\,\mathrm{kWh} - Q_{tot} &\geq 0, \tag{4.15} \end{align}$$

where the right hand sides of these expressions are used as performance measures:

$$\begin{align} EF_3 &= 95\frac{\mathrm{kWh}}{\mathrm{m}^2}A_{tot} + 2200\,\mathrm{kWh} - Q_{tot} \tag{4.16}\\ EF_2 &= 50\frac{\mathrm{kWh}}{\mathrm{m}^2}A_{tot} + 1600\,\mathrm{kWh} - Q_{tot} \tag{4.17}\\ EF_1 &= 35\frac{\mathrm{kWh}}{\mathrm{m}^2}A_{tot} + 1100\,\mathrm{kWh} - Q_{tot} \tag{4.18}\\ & \tag{4.19} \end{align}$$

The energy frame related to each of these performance measures is satisfied when a design decision is found where the performance measure is positive. The energy frames can thus be addressed by requiring that one or more of the performance measures EF_3, EF_2 and EF_1 must be positive.

The Danish building regulations specify (among others) the following upper limits on the thermal transmittance for the components used in the building envelope:

1. The thermal transmittance of the ground slab must be less than 0.30 $\mathrm{W/m^2K}$.
2. The thermal transmittance of the external walls must be less than 0.40 $\mathrm{W/m^2K}$.
3. The thermal transmittance of the roof construction must be less than 0.25 $\mathrm{W/m^2K}$.
4. The thermal transmittance of windows and doors must be less than 2.30 $\mathrm{W/m^2K}$.

The U-value requirements can be addressed by specifying upper limits on the performance measures U_g, U_{wall}, U_r, $U_{win}^{(1)}$ and $U_{win}^{(2)}$.

The Danish building regulations furthermore establish an upper bound on the heat loss through the building envelope, not including windows and doors. The heat loss Q_e per m^2 of the building envelope is required to be less than 6 W, that is:

$$\frac{Q_e}{A_e} \leq 6\,\frac{\mathrm{W}}{\mathrm{m}^2} \quad \Leftrightarrow \quad 6\,\frac{\mathrm{W}}{\mathrm{m}^2} \cdot A_e - Q_e \geq 0, \tag{4.20}$$

where A_e is the area of the building envelope, not including windows and doors. The left hand side of (4.20) is used as a performance measure:

$$BE = 6\,\frac{\mathrm{W}}{\mathrm{m}^2} \cdot A_e - Q_e. \tag{4.21}$$

The Danish building regulations also specify upper limits on the linear thermal transmittances for the interaction between various parts of the building envelope. However, calculating the linear thermal transmittance for the interaction between two building components requires numerical methods for solving partial differential equations. This could make the performance calculations more time-consuming, and thereby increase the time needed for solving the optimization problem (3.8).

All linear thermal transmittances are therefore assumed to be constant, and close to or equal to the upper limits specified by the building regulations, in order to provide a conservative estimate of the design decisions. Developing methods for performing fast and reliable calculations for the linear thermal transmittance is a possible topic for further research. Methods similar to the one described in the paper included in Appendix B might be useful for this purpose; however, this has not been investigated as part of this study.

4.2 The performance calculations

4.2.1 Energy and indoor environment

The energy performance and the performance regarding the indoor environment of the simplified building are calculated using the method proposed by Nielsen [50]. The method is implemented in the Matlab [44] program BuildingCalc, which is described by Nielsen [48]. This method is partially described in the following, as well as how it is used for calculating the performance of the simplified building.

It is assumed that all rooms of the simplified building are heated to the same temperature, and that the thermal interaction between them therefore can be disregarded. The internal wall is assumed to be adiabatic; therefore the interactions between the thermal zones are also omitted.

The energy performance of each of the two zones is estimated using the thermal networks shown in Figure 4.5. Notice that the interactions between the floors are omitted. Each floor of the building is represented by a network consisting of the following four nodes:

1. A node representing the external environment, with temperature T_{ext}.
2. A node representing the internal air, with temperature T_a.
3. A node representing the internal surfaces, with temperature T_s.
4. A node representing the thermal mass, with temperature T_w.

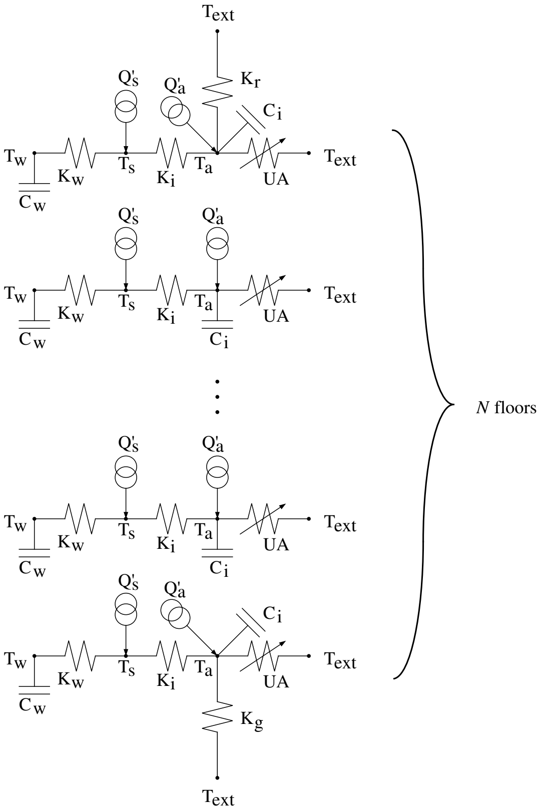


Figure 4.5: The thermal networks used for estimating the energy performance for each of the two zones.

The networks consists of the thermal conductance between the thermal mass and the surface K_w, the conductance between the surface and the internal air K_i, and the conductance between the internal air and the external environment UA, which is used for calculating the heat loss through the external walls and windows. The heat loss through the roof construction and the ground slab are calculated using the conductances K_r and K_g, respectively. The effective heat capacity of the constructions is represented by C_w, and the heat capacity of the internal air and property contents is represented by C_i.

The heat sources Q'_s and Q'_a are given by:

$$Q'_s = S\, w_s\, Q'_{sun}, \tag{4.22}$$

and

$$Q'_a = S\, w_a\, Q'_{sun} + Q'_l + Q'_h - Q'_c, \tag{4.23}$$

where Q'_{sun} is the transmitted solar energy, and S is the shading factor for the shading device, which is assumed to be variable. The factors w_s and w_a are the fractions of the solar energy absorbed by the internal surfaces and the internal air, respectively. The contribution from internal loads is represented by Q'_l, and the contributions from the heating and cooling systems are represented by Q'_h and Q'_c, respectively.

The only difference between the networks used for representing the thermal zones is that the first floor and the top floor have additional heat losses due to the heat losses through the roof construction and the ground slab. The networks representing the intermediate floors are all identical.

In order to simplify the calculations, the additional heat losses are distributed equally among all floors. All networks thereby become identical with the network proposed by Nielsen [50]. The thermal conductance between the internal and external air is denoted $\widehat{UA}$, which is given by:

$$\widehat{UA} = UA + \frac{K_r}{N} + b_g \cdot \frac{K_g}{N}. \tag{4.24}$$

This means that the energy performance of the building can be estimated using only one simulation for each thermal zone. The network used for estimating the energy performance of a single floor for each of the two thermal zones is shown in Figure 4.6.

The factor b_g in (4.24) is a temperature factor that compensates for smaller temperature differences for some building components, where

1. the external temperature of the component is not the same as the external air temperature, or
2. the internal temperature of the component is not the same as the internal air temperature.

Given the network parameters, information about the internal loads, settings for the HVAC systems, active solar shading devices and variable insulation system, as well as weather data and information about the location of the building, the method by Nielsen [50] provides, among others, hourly values of the required energy for heating and cooling the building, the internal air temperature, as well as ventilations rates for natural and mechanical ventilation systems.

The method furthermore provides evaluations of the thermal indoor environment, expressed in terms of the number of hours with overheating, based on a user-defined maximum allowed indoor air temperature. Finally, the method provides hourly values of the predicted mean vote (PMV), and the predicted percentage of dissatisfied (PPD).

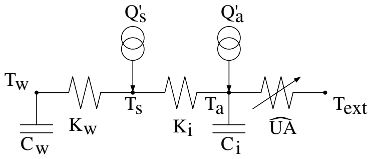


Figure 4.6: The thermal network resulting from distributing the additional heat losses equally among the floors. The network is used for estimating the performance of a single floor for each of the two thermal zones.

The current version of the building optimization method does not use the PMV and PPD values as performance measures. The number of hours with overheating is calculated using linear interpolation between hourly values of the indoor air temperature. This method is described in Section 4.4.2.

Furthermore, active solar shading devices are not used, and the windows have no overhang.

Parameter	Unit	Description
$T_a \in \mathbb{R}^{8760}$	°C	Vector with hourly values of the internal air temperature.
$Q'_h \in \mathbb{R}^{8760}$	W	Vector with hourly values of the required power for heating the building.
$Q'_c \in \mathbb{R}^{8760}$	W	Vector with hourly values of the required power for cooling the building.
$V' \in \mathbb{R}^{8760}$	m^3/s	Vector with hourly values of the mechanical ventilation rate.

Table 4.1: The output from the method by Nielsen [50], which is used for calculating the performance measures.

The output from the method by Nielsen [50], which is used for calculating the performance measures, is shown in Table 4.1.

Section 4.3.3 concerns calculations of the network parameters, and other parameters required by the performance calculation method, based on decision variables and constant parameters.

4.2.2 Economy

The performance regarding the economy of the building is calculated using simple mathematical models of data from standard price catalogues. The price calculations performed by the building optimization method are based on data from the V&S price catalogue [60], which concerns Danish gross unit prices for individual construction jobs[1].

[1] The English translations of the construction jobs are provided by the author of the thesis.

The unit prices are represented using three different mathematical models, which are described in the following and illustrated on examples from the price catalogue.

The unit price for a construction job depends on the number of purchased units, for instance, the unit price of forming depends on the total area of the required concrete forms. The unit price for some construction jobs also depend on secondary parameters, for instance, the unit price of pouring concrete not only depend on the amount of concrete, but also on the required strength of the concrete.

Table 4.2 provides three examples of unit prices for different construction jobs from the V&S price catalogue [60]. The first job does not involve a secondary parameter, and the other two jobs do.

The following model is used for representing unit prices for construction jobs that do not involve a secondary parameter:

$$p_1(u, \beta) = \beta_1 \, \exp(\beta_2 \, u) + \beta_3, \tag{4.25}$$

where $u \in \mathbb{R}$ is the number of units, and where $\beta \in \mathbb{R}^3$ is a vector consisting of model parameters.

The following two models are used for representing unit prices for construction jobs involving a secondary parameter:

$$\begin{aligned} p_2(u, s, \beta) &= \beta_1 \, s + \beta_2 \, \exp(\beta_3 \, u) + \beta_4 & (4.26)\\ p_3(u, s, \beta) &= \beta_1 \, \exp(\beta_2 \, s) + \beta_3 \, \exp(\beta_4 \, u) + \beta_5, & (4.27)\end{aligned}$$

where $s \in \mathbb{R}$ is the secondary parameter.

The model parameters $\beta \in \mathbb{R}^{n_\beta}$, with $n_\beta = 3$ for (4.25), $n_\beta = 4$ for (4.26), and $n_\beta = 5$ for (4.27), are calculated by estimating solutions to the following system of non-linear equations:

$$p_k(u_i, s_j, \beta) = \hat{p}_{ji}, \quad i = 1, \ldots, n_u, \quad j = 1, \ldots, n_s, \tag{4.28}$$

with $k \in \{1, 2, 3\}$. For $k = 1$, the argument s_j is omitted. The parameter $\hat{p}_{ji}$ is the catalogue price related to the number of purchased units u_i and the secondary parameter s_j. Furthermore, n_u and n_s are the numbers of columns and rows, respectively, used for organizing the unit prices for the considered construction job in the price catalogue.

Description	Gross unit price in DKR		
Forming (job no. 04.10.01,01):			
Units	$1000\,\mathrm{m}^2$	$5000\,\mathrm{m}^2$	$10000\,\mathrm{m}^2$
	185	163	155
Pouring concrete (job no. 04.10.31,01-04):			
Units	$10\,\mathrm{m}^3$	$50\,\mathrm{m}^3$	$100\,\mathrm{m}^3$
16 MPa	1590	1420	1360
20 MPa	1610	1450	1390
25 MPa	1630	1470	1410
30 MPa	1660	1490	1430
Delivering and installing uncoated, air-filled, double-glazed windows (job no. 04.35.05):			
Units	$5\,\mathrm{m}^2$	$20\,\mathrm{m}^2$	$50\,\mathrm{m}^2$
$0.4 - 0.5\,\mathrm{m}^2$	1260	1150	1090
$0.5 - 1.0\,\mathrm{m}^2$	949	863	816
$1.0 - 1.5\,\mathrm{m}^2$	826	750	708
$1.5 - 2.0\,\mathrm{m}^2$	802	728	686
$2.0 - 2.5\,\mathrm{m}^2$	771	699	659
$2.5 - 3.0\,\mathrm{m}^2$	794	721	680

Table 4.2: Three examples of unit prices for construction jobs. The secondary parameter for the unit price of pouring concrete is the strength of the concrete. The secondary parameter for the unit price of delivering and installing double-glazed windows is the area of the individual windows.

For $k = 1$, the system of equations (4.28) is determined, and for $k \in \{2, 3\}$, it is overdetermined. In all three cases, the model parameters are calculated as least squares solutions to (4.28). This requires the residual functions $\Delta p_k : \mathbb{R}^{n_\beta} \to \mathbb{R}^{n_u n_s}$, which are defined as:

$$\Delta p_{kl}(\beta) = p_k(u_i, s_j, \beta) - \hat{p}_{j,i}, \quad i = 1, \ldots, n_u, \quad j = 1, \ldots, n_s, \tag{4.29}$$

where $k \in \{1, 2, 3\}$, and where $l = n_s\,(i-1) + j$. The residuals (4.29) are thus calculated column-wise for the catalogue prices.

The model parameters β for the models (4.25), (4.26) and (4.27) are calculated by solving the following least squares data-fitting problems:

$$\begin{array}{ll} \text{minimize} & \|\Delta p_k(\beta)\|_2 \\ \text{with respect to} & \beta \in \mathbb{R}^{n_\beta}, \end{array} \tag{4.30}$$

for $k \in \{1, 2, 3\}$.

Table 4.3 shows model parameters for the construction jobs shown in Table 4.2. The model parameters are calculated by solving the data-fitting problems (4.30).

Job	Forming	Pouring concrete	Installing windows
No	04.10.01,01	04.10.31,01-04	04.35.05
Model	(4.25)	(4.26)	(4.27)
β_1	4.317435e+1	4.868319e+0	2.120962e+3
β_2	−2.807892e−4	3.238082e+2	−3.498643e+0
β_3	1.523953e+2	−2.807876e−2	1.845962e+2
β_4	–	1.267209e+3	−6.164742e−2
β_5	–	–	6.604592e+2

Table 4.3: Model parameters for the construction jobs shown in Table 4.2.

Figure 4.7 shows the catalogue prices for forming, together with the model (4.25). The model interpolates the data, since the system of non-linear equations (4.28) in this case is determined.

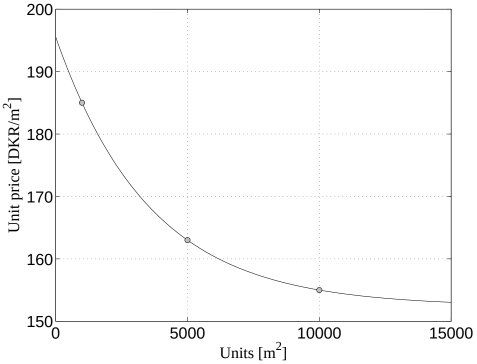


Figure 4.7: A plot of the unit prices for forming (job no. 04.10.01,01). The circles represent catalogue prices, and the solid line represent the mathematical model (4.25), using the model parameters from Table 4.3.

Figure 4.8 shows the catalogue prices for pouring concrete, together with the model (4.26), and Figure 4.9 shows the catalogue prices for delivering and installing double-glazed windows, together with the model (4.27). In both cases, the equations (4.28) are overdetermined, which means that the models do not interpolate the data, and the models (4.26) and (4.27) therefore have modeling errors.

The relative modeling errors for the catalogue price $\hat{p}_{j,i}$ is given by:

$$\frac{|p_k(u_i, s_j, \beta) - \hat{p}_{j,i}|}{\hat{p}_{j,i}}, \tag{4.31}$$

for $k \in \{2, 3\}$. The average relative modeling error for the unit prices of pouring concrete, and for delivering and installing windows, are in both cases below 2%.

The output from the performance calculations regarding the economy of the simplified building can be arranged in a vector

$$P \in \mathbb{R}^{n_{jobs}}, \tag{4.32}$$

containing the unit prices for the n_{jobs} construction jobs needed for constructing the building.

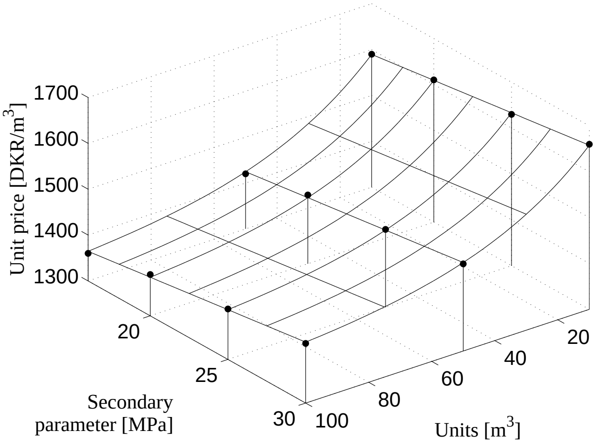


Figure 4.8: A plot of the unit prices for pouring concrete (job no. 04.10.31,01-04). The circles represent catalogue prices, and the grid represent the mathematical model (4.26), using the model parameters from Table 4.3. The average relative modeling error is 0.21%, and the maximum relative modeling error is 0.42%.

4.3 Preparing the input

This section concerns the details of calculating the input to the performance calculation methods in terms of the decision variables (4.3) and the constant parameters.

The input relies on the geometry of the building, such as areas and volumes of various building components. Based on this information, the network parameters required by the performance calculation method described by Nielsen [50] can be calculated. This method also requires the properties of the windows of the two thermal zones.

The geometry of the building is furthermore used for calculating the size of the jobs required for constructing the building.

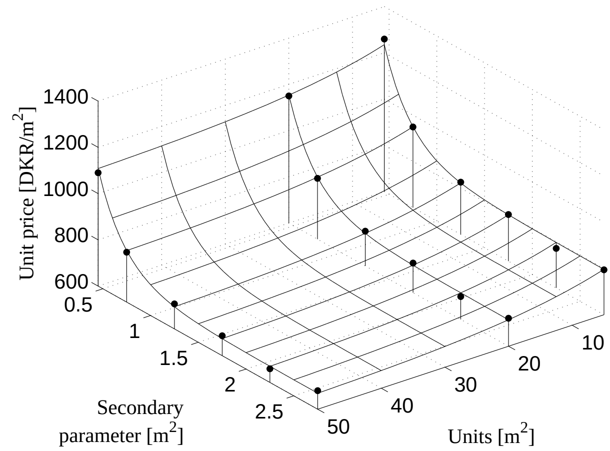


Figure 4.9: A plot of the unit prices for delivering and installing double-glazed windows (job no. 04.35.05). The circles represent catalogue prices, and the grid represent the mathematical model (4.27), using the model parameters from Table 4.3. The mid-points of the intervals shown in Table 4.2 are used as secondary parameters. The average relative modeling error is 1.23%, and the maximum relative modeling error is 3.36%.

4.3.1 Geometry

First the length and width of the building are addressed. From Figure 4.3, the following relations can be derived:

$$\begin{aligned} w_{ext} &= 2\,w_{int} + 2\,d_{wall} + d_{int} && (4.33)\\ l_{ext} &= l_{int} + 2\,d_{wall} && (4.34)\\ d_{wall} &= d_{w,i} + d_{w,u} && (4.35)\end{aligned}$$

The internal floor area A_{int} is related to the geometry of the building in the following way:

$$A_{int} = 2\,N\,l_{int}\,w_{int}. \qquad (4.36)$$

The parameters w_{ext}, w_{int}, l_{ext} and l_{int} are unknown. The parameters $d_{w,i}$ and N are decision variables, and $d_{w,u}$, d_{int} and A_{int} are constants.

Combining (4.1) with (4.33) and (4.34) gives:

$$\begin{aligned} \varrho &= \frac{2\,w_{int} + 2\,d_{wall} + d_{int}}{l_{int} + 2\,d_{wall}} \quad \Leftrightarrow && (4.37)\\ l_{int} &= \frac{1}{\varrho}\left(2\,w_{int} + 2\,d_{wall} + d_{int}\right) - 2\,d_{wall}. && (4.38)\end{aligned}$$

The parameter ϱ is a decision variable. Combining (4.38) with (4.36) gives:

$$\begin{aligned} A_{int} &= 2\,N\,w_{int}\,\left(\frac{1}{\varrho}\,(2\,w_{int}+2\,d_{wall}+d_{int})-2\,d_{wall}\right) \quad \Leftrightarrow && (4.39)\\ A_{int}\,\varrho &= 4\,N\,w_{int}^2+4\,N\,d_{wall}\,w_{int}+2\,N\,d_{int}\,w_{int}-4\,N\,\varrho\,d_{wall}\,w_{int} \quad \Leftrightarrow && (4.40)\\ \frac{A_{int}\,\varrho}{4N} &= w_{int}^2+\left((1-\varrho)\,d_{wall}+\frac{1}{2}\,d_{int}\right)\,w_{int}, && (4.41) \end{aligned}$$

which can be arranged as the following quadratic equation:

$$w_{int}^2+\left((1-\varrho)\,d_{wall}+\frac{1}{2}\,d_{int}\right)\,w_{int}-\frac{A_{int}\,\varrho}{4N}=0. \tag{4.42}$$

The solution can be expressed in terms of

$$B=(1-\varrho)\,d_{wall}+\frac{1}{2}\,d_{int}, \tag{4.43}$$

and

$$C=-\frac{A_{int}\,\varrho}{4N}. \tag{4.44}$$

The discriminant D for the equation (4.42) is:

$$D=B^2-4C=B^2+\frac{A_{int}\,\varrho}{N}>B^2, \tag{4.45}$$

which implies that

$$\sqrt{D}>|B|. \tag{4.46}$$

This means that (4.42) has a positive and a negative root. Only the positive root is relevant, which is given by:

$$w_{int}=\frac{1}{2}\left(-B+\sqrt{D}\right). \tag{4.47}$$

The parameter w_{int} is thus by (4.47) expressed in terms of known parameters. The parameter l_{int} can then be calculated using (4.38), and the parameters w_{ext} and l_{ext} can be calculated using (4.33) and (4.34), respectively.

Next, the window areas for the two thermal zones are considered. The relation (4.2) gives the following:

$$\begin{aligned} A_{win}^{(1)} &= \sigma^{(1)}\,h_{ext}\,l_{int} && (4.48)\\ A_{win}^{(2)} &= \sigma^{(2)}\,h_{ext}\,l_{int}, && (4.49) \end{aligned}$$

where $\sigma^{(1)}$ and $\sigma^{(2)}$ are decision variables, and where h_{ext} is a constant.

Since window bands are used for the simplified building, the window areas can be expressed as:

$$\begin{aligned} A_{win}^{(1)} &= h_{win}^{(1)}\, l_{int} && (4.50)\\ A_{win}^{(2)} &= h_{win}^{(2)}\, l_{int}, && (4.51) \end{aligned}$$

where $h_{win}^{(1)}$ and $h_{win}^{(2)}$ are the window heights for the two thermal zones. Combining these expressions with (4.48) and (4.49) gives the following expressions for the window heights:

$$\begin{aligned} h_{win}^{(1)} &= \sigma^{(1)}\, h_{ext} && (4.52)\\ h_{win}^{(2)} &= \sigma^{(2)}\, h_{ext} && (4.53) \end{aligned}$$

The circumferences $O_{win}^{(1)}$ and $O_{win}^{(2)}$ of the windows are given by the following expressions:

$$\begin{aligned} O_{win}^{(1)} &= 2(h_{win}^{(1)} + l_{int}) && (4.54)\\ O_{win}^{(2)} &= 2(h_{win}^{(2)} + l_{int}), && (4.55) \end{aligned}$$

The areas of the external walls for the two thermal zones are given by:

$$\begin{aligned} A_{wall}^{(1)} &= h_{ext}\,(l_{ext} + w_{ext}) - A_{win}^{(1)} = h_{ext}\,\left(l_{ext} + w_{ext} - \sigma^{(1)}\, l_{int}\right) && (4.56)\\ A_{wall}^{(2)} &= h_{ext}\,(l_{ext} + w_{ext}) - A_{win}^{(2)} = h_{ext}\,\left(l_{ext} + w_{ext} - \sigma^{(2)}\, l_{int}\right). && (4.57) \end{aligned}$$

These areas apply to a single floor.

The external area A_{ext} of a single floor of the building is:

$$A_{ext} = w_{ext}\, l_{ext}. \qquad (4.58)$$

The total heated floor area A_{tot} includes internal and external walls, and is given by:

$$A_{tot} = N \cdot A_{ext}. \qquad (4.59)$$

The internal surface areas $A_s^{(1)}$ and $A_s^{(2)}$ for the two thermal zones, which are required for conducting the performance calculations, are the sums of areas that contribute to the thermal capacity, such as internal walls, ceiling and floor areas, but not windows, and are for the simplified building given by:

$$\begin{aligned} A_s^{(1)} &= 2\, l_{int}\, w_{int} + 2\, l_{int}\, h_{int} + 2\, w_{int}\, h_{int} - A_{win}^{(1)} && (4.60)\\ A_s^{(2)} &= 2\, l_{int}\, w_{int} + 2\, l_{int}\, h_{int} + 2\, w_{int}\, h_{int} - A_{win}^{(2)} && (4.61) \end{aligned}$$

The internal floor height h_{int} is given by:

$$h_{int} = h_{ext} - h_{deck}, \qquad (4.62)$$

where h_{deck} is the height of the concrete decks.

The air volume for each thermal zone on every floor is given by:

$$V_{int} = l_{int}\, w_{int}\, h_{int}. \tag{4.63}$$

The external circumference O_{ext} of the building is given by:

$$O_{ext} = 2(w_{ext} + l_{ext}). \tag{4.64}$$

4.3.2 Window properties

The window properties used as input to the performance calculation method are calculated as a weighted sum of the properties of the n_{win} windows stored in the user-supplied window database. An example of a window database is provided in Table D.11 in Appendix D. The glazing category is calculated using the method proposed by Karlsson and Roos [38].

The window database can be arranged as a matrix $W \in \mathbb{R}^{9\times n_{win}}$, with the window properties stored column-wise. The weighted sums $w^{(1)} \in \mathbb{R}^9$ and $w^{(2)} \in \mathbb{R}^9$ of the window properties for the two thermal zones can be calculated as the products

$$w^{(1)} = W\alpha^{(1)}, \tag{4.65}$$

and

$$w^{(2)} = W\alpha^{(2)}. \tag{4.66}$$

The weighted sums $w^{(1)}$ and $w^{(2)}$ are used as part of the input to the performance calculation method.

4.3.3 Network parameters

Calculating the energy and indoor environment performance of the building is done by calculating the performance for each of the two thermal zones, which in turn requires the network parameters for the two zones to be available.

The thermal conductances $\widehat{UA}^{(1)}$ and $\widehat{UA}^{(2)}$ for the two thermal zones are given by:

$$\widehat{UA}^{(1)} = UA^{(1)} + \frac{K_r}{N} + b_g \cdot \frac{K_g}{N} \tag{4.67}$$

$$\widehat{UA}^{(2)} = UA^{(2)} + \frac{K_r}{N} + b_g \cdot \frac{K_g}{N}. \tag{4.68}$$

The thermal conductances $UA^{(1)}$ and $UA^{(2)}$ are given by:

$$UA^{(1)} = U_{wall}\, A^{(1)}_{wall} + U^{(1)}_{win}\, A^{(1)}_{win} + O^{(1)}_{win}\, \Psi_{ww} + \frac{1}{2} O_{ext}\, \Psi_{fw} \tag{4.69}$$

$$UA^{(2)} = U_{wall}\, A^{(2)}_{wall} + U^{(2)}_{win}\, A^{(2)}_{win} + O^{(2)}_{win}\, \Psi_{ww} + \frac{1}{2} O_{ext}\, \Psi_{fw}, \tag{4.70}$$

where the areas of the external walls for the two zones are given by (4.56) and (4.57), respectively, and where the areas of the windows are given by (4.48) and (4.49), respectively. Furthermore, the window circumferences are given by (4.54) and (4.55), respectively. The constant parameter Ψ_{ww} denotes the linear thermal transmittance caused by the thermal interaction between the windows and the external walls.

Finally, the constant parameter Ψ_{fw} is the linear thermal transmittance representing the thermal interaction between the external walls and the foundation. Only half the length of the foundation is included in in each thermal zone.

The thermal transmittance of the external walls is assumed to be identical for the two thermal zones, and is given by:

$$U_{wall} = \frac{1}{R_{int} + \frac{d_{w,u}}{\lambda_{w,u}} + \frac{d_{w,i}}{\lambda_{w,i}} + R_{ext}}. \tag{4.71}$$

The terms R_{int} and R_{ext} are the internal and external surface resistances, respectively. The terms $\lambda_{w,u}$ and $\lambda_{w,i}$ are the thermal conductivities of the uninsulated and insulated parts of the external wall, and the terms $d_{w,u}$ and $d_{w,i}$ are the thicknesses of the uninsulated and insulated parts. The term $d_{w,i}$ is a decision variable, and the rest are constants.

The total thermal transmittances, $U_{win}^{(1)}$ and $U_{win}^{(2)}$, of the windows for the two zones are parts of the weighted window properties $w^{(1)}$ and $w^{(2)}$:

$$U_{win}^{(1)} = w_3^{(1)} \tag{4.72}$$

$$U_{win}^{(2)} = w_3^{(2)} \tag{4.73}$$

The thermal conductance K_r used for calculating the heat loss through the roof construction is given by:

$$K_r = \frac{1}{2}\, U_r\, A_{ext}, \tag{4.74}$$

where the external floor area A_{ext} is given by (4.58). Only half the area is included, since (4.74) only applies to one thermal zone.

The thermal transmittance of the roof construction is given by:

$$U_r = \frac{1}{R_{int} + \frac{d_{r,u}}{\lambda_{r,u}} + \frac{d_{r,i}}{\lambda_{r,i}} + R_{ext}}, \tag{4.75}$$

where $\lambda_{r,u}$ and $\lambda_{r,i}$ are the thermal conductivities of the uninsulated and insulated parts of the roof construction, and the terms $d_{r,u}$ and $d_{r,i}$ are the thicknesses of these parts. The term $d_{r,i}$ is a decision variable, and the rest are constants.

The thermal conductance K_g used for calculating the heat loss through the ground slab can in a similar way be expressed as:

$$K_g = \frac{1}{2}\,(U_g\, A_{ext} + O_{ext}\, \Psi_{fw}), \tag{4.76}$$

where Ψ_{fw} denotes the linear transmittance caused by the thermal interaction between foundation and the external wall, which is a constant.

The thermal transmittance of the ground slab is given by:

$$U_g = \frac{1}{R_{int} + \frac{d_{g,u}}{\lambda_{g,u}} + \frac{d_{g,i}}{\lambda_{g,i}} + R_{ext}}. \tag{4.77}$$

The terms $\lambda_{g,u}$ and $\lambda_{g,i}$ are the thermal conductivities of the uninsulated and insulated parts of the ground slab, and the terms $d_{g,u}$ and $d_{g,i}$ are the thicknesses of these parts. The term $d_{g,i}$ is a decision variable, and the rest are constants.

The conductances $\widehat{UA}^{(1)}$ and $\widehat{UA}^{(2)}$ are thereby expressed in terms of decision variables and constants.

The internal thermal capacity C_i accounts for the internal air, as well as the household goods, which are assumed to be heated to the same temperature as the internal air. C_i does not account for constructions. The thermal capacity C_i is calculated by assuming that the total thermal capacity $C_{i,tot}$ of the building contents is a constant. The thermal capacity C_i is calculated by distributing $C_{i,tot}$ equally among all floors of the building, and among the two thermal zones. The thermal capacity C_i furthermore includes the thermal capacity of the air in each zone, which thereby becomes:

$$C_i = \frac{C_{i,tot}}{2N} + V_{int} \cdot \varrho_{air} \cdot c_{air}, \tag{4.78}$$

where the internal air volume V_{int} is given by (4.63), and where the constants ϱ_{air} and c_{air} are the density and specific thermal capacity of air, respectively.

The remaining network parameters are calculated by the performance calculation method by Nielsen [50], based on input provided by the user. The methods for calculating them are briefly described in the following.

The conductances $K_i^{(1)}$ and $K_i^{(2)}$ between the internal air and the internal surfaces, are calculated using the internal surface resistance R_{int}, and the surface areas $A_s^{(1)}$ and $A_s^{(2)}$.

The thermal capacities $C_w^{(1)}$ and $C_w^{(2)}$ of the constructions are calculated using the surface areas $A_s^{(1)}$ and $A_s^{(2)}$, which are the surfaces that contribute to the thermal capacity, and the specific heat capacity c_w for the constructions of the building.

The conductances $K_w^{(1)}$ and $K_w^{(2)}$ between the internal surface and the thermal mass are calculated using the surface areas $A_s^{(1)}$ and $A_s^{(2)}$, and an equivalent thermal resistance R_{eq}, which is a constant.

Finally, the parameters needed for calculating the heat sources Q'_s and Q'_a are addressed, which are also calculated by the performance calculation method, also based on input from the user.

The solar transmittance Q'_{sun} is calculated for each window of the building, based on information about the location of the building, the orientation and tilt of the windows, and various information about the surroundings of the building, such as ground albedo, various parameters used for defining the sky model, etc.

The shading factor S is calculated such that $S \in [S_{min}; 1]$, where S_{min} is a user-provided minimum allowed value for the shading factor. $S_{min} = 1$ means the solar shading is not used. The value of S is controlled by the set point for the cooling system, and hence does not depend on daylight or glare.

The contributions Q'_h and Q'_c to the heat source Q'_a from the heating and cooling systems depend on the internal air temperature, and the set points for the systems.

The calculation of the internal load Q'_l is based on information regarding the total internal load by users and equipment, as well as specifications for when the building is used.

The fractions w_s and w_a are constants, provided by the user.

4.3.4 The size of the construction jobs

This section concerns the size of the jobs required for constructing the building. These parameters are used for calculating unit prices for the construction jobs, which in turn are used for calculating the construction cost of the building.

Foundation

An annular foundation as shown in Figure 4.10 is used in order to calculate the construction costs for the foundation.

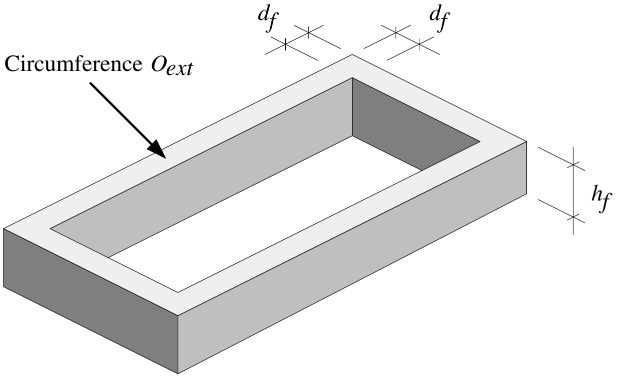


Figure 4.10: An annular foundation.

The following jobs are considered when calculating the construction costs:

Job no. 03.05.10,03-06: excavation for foundation

Job no. 03.15.04,01: bending and placing reinforcement

Job no. 03.15.07,01-04: concrete casting.

90 cm of soil is assumed to be removed when excavating for the foundation. The unit price depends on the length of the foundation, which is assumed to be the same as the external circumference O_{ext} of the building. It furthermore depends on the width d_f of the foundation, which is used as secondary parameter. The circumference O_{ext} is given by (4.64), and the width d_f is a constant.

The unit price of bending and placing reinforcement only depends on the total weight $m_{f,r}$ of the reinforcement mesh.

It is assumed that a total of $n_{f,m}$ reinforcement meshes are used for reinforcing the foundation, and that the total area of each reinforcement mesh is equal to the external surface area A_f of the foundation, which is given by:

$$A_f = O_{ext} \cdot h_f, \tag{4.79}$$

where h_f is the height of the foundation.

The reinforcement rods are assumed to be separated by a distance of $d_{f,m}$, as shown in Figure 4.11. The total length of the reinforcement rods used in an area of size $d_{f,m} \times d_{f,m}$ is $2 \cdot n_{f,m} \cdot d_{f,m}$. The volume of this amount of reinforcement is:

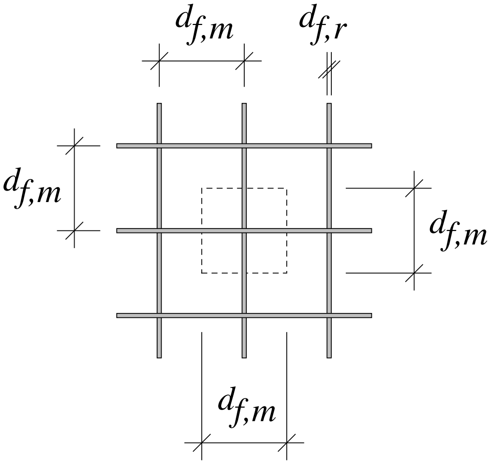


Figure 4.11: The reinforcement mesh used in the foundation.

$$V_r = \frac{\pi}{4} \cdot 2 \cdot n_{f,m} \cdot d_{f,m} \cdot d_{f,r}^2 = \frac{\pi}{2} \cdot n_{f,m} \cdot d_{f,m} \cdot d_{f,r}^2, \tag{4.80}$$

where $d_{f,r}$ is the diameter of the reinforcement rods. This amount of reinforcement applies to an area of size $d_{f,m} \times d_{f,m}$. The total volume for the reinforcement used in the foundation thus becomes:

$$V_{r,tot} = \frac{A_f}{d_{f,m}^2} \cdot V_r = \frac{\pi}{2d_{f,m}} \cdot A_f \cdot n_{f,m} \cdot d_{f,r}^2, \tag{4.81}$$

and the total weight is:

$$m_{f,r} = V_{r,tot} \cdot \varrho_r = \frac{\pi}{2d_{f,m}} \cdot \varrho_r \cdot A_f \cdot n_{f,m} \cdot d_{f,r}^2, \tag{4.82}$$

where ϱ_r is the density of the reinforcement rods, which is a constant.

These considerations do not include details regarding for instance corners.

The unit price of concrete casting depends on the volume of the concrete and the compressive strength τ_f of the concrete used in the foundation, which is used as a secondary parameter. It is assumed that the required amount of concrete is equal to the volume V_f of the foundation, which is given by:

$$V_f = O_{ext} \cdot h_f \cdot d_f - 4 \cdot h_f \cdot d_f^2 = h_f \cdot d_f \cdot (O_{ext} - 4 \cdot d_f) \tag{4.83}$$

The parameters and the price models used when calculating the cost of constructing the foundation are summarized in Table 4.4.

Job no.	Units	Sec. par.	Model
03.05.10,03-06	O_{ext}	d_f	(4.26)
03.15.04,01	$m_{f,r}$	–	(4.25)
03.15.07,01-04	V_f	τ_f	(4.26)

Table 4.4: Parameters and price models used for calculating the cost of constructing the foundation.

Ground slab

In order to calculate the cost of constructing the ground slab, it is assumed that it consists of the following layers:

1. A capillary-breaking layer
2. A layer of concrete used as wearing surface
3. A layer of insulation material
4. A layer of reinforced concrete.

This involves the following construction jobs:

Job no. 03.15.26: laying capillary breaking layer

Job no. 03.15.05,05-09: casting the wearing surface

Job no. 03.15.44,01-04: insulating the ground slab (mineral wool)

Job no. 03.15.51,01-03: reinforcing and casting the ground slab.

The unit prices for all jobs, except for casting the wearing surface, depend on the external floor area of the building, A_{ext}, as well as secondary parameters, which are described in the following.

The unit price of laying the capillary breaking layer depends on the thickness d_{cb} of the layer. Installing the insulation material in the ground slab depends on the thickness $d_{g,i}$. Reinforcing and casting the ground slab depends on the compression strength τ_{gs} of the concrete used in the ground slab.

The unit price of casting the wearing surface depends on the volume V_{ws} of the concrete and the compression strength τ_{ws} of the concrete used as wearing surface. The volume V_{ws} is given by:

$$V_{ws} = A_{ext} \cdot d_{ws}, \tag{4.84}$$

where d_{ws} is the thickness of the wearing surface.

The parameters and the price models used when calculating the cost of constructing the ground slab are summarized in Table 4.5.

Job no.	Units	Sec. par.	Model
03.15.26	A_{ext}	d_{cb}	(4.26)
03.15.05,05-09	V_{ws}	τ_{ws}	(4.26)
03.15.44,01-04	A_{ext}	$d_{g,i}$	(4.26)
03.15.51,01-03	A_{ext}	τ_{gs}	(4.26)

Table 4.5: Parameters and price models used for calculating the cost of constructing the ground slab.

External walls

In order to calculate the prices for construction the external walls, it is assumed that the external walls consists of the following layers:

1. An internal concrete wall
2. A layer of insulation material
3. An external brick wall.

This involves the following construction jobs:

Job no. 04.10.01,01: forming

Job no. 04.10.27,01: bending and placing reinforcement

Job no. 04.10.31,01-04: concrete casting

Job no. 04.15.30,01: constructing external brick wall

Job no. 04.15.05,01-05: insulating cavity wall (mineral wool).

The total area $A_{wall,tot}$ of the external walls of the building is given by:

$$A_{wall,tot} = N \cdot (A_{wall}^{(1)} + A_{wall}^{(2)}) \tag{4.85}$$

The unit price of forming only depends on the total area of the concrete forms, which is assumed to be twice the total area of the external walls. This consideration means that the area of the internal and external parts of the concrete forms are included in the total area of the concrete forms, but not details regarding for instance corners.

The unit price of bending and placing reinforcement only depends on the total weight $m_{wall,r}$ of the reinforcement rods. The weight $m_{wall,r}$ can be calculated using an expression similar to (4.82):

$$m_{wall,r} = \frac{\pi}{2d_{wall,m}} \cdot \varrho_r \cdot A_{wall,tot} \cdot n_{wall,m} \cdot d_{wall,r}^2, \tag{4.86}$$

where $d_{wall,m}$ is the width of the mesh cells used in the reinforcement meshes in the external walls. Furthermore, $n_{wall,m}$ is the number of meshes, and $d_{wall,r}$ is the diameter of the reinforcement rods.

The unit price of concrete casting depends on the total volume V_{wall} of the external walls, and the compression strength τ_{wall} of the concrete used in the external walls, which is used as a secondary parameter. The volume V_{wall} is given by the expression:

$$V_{wall} = A_{wall,tot} - 4 \cdot N \cdot h_{ext} \cdot d_{wall}^2. \tag{4.87}$$

The unit price of bricking up the external brick wall only depends on $A_{wall,tot}$.

The unit price of insulating the cavity of the external wall depends on $A_{wall,tot}$, and the thickness $d_{w,i}$ of the insulated part of the external wall, which is used as a secondary parameter.

The parameters and the price models used when calculating the cost of constructing the external walls are summarized in Table 4.6.

Job no.	Units	Sec. par.	Model
04.10.01,01	$2\,A_{wall,tot}$	–	(4.25)
04.10.27,01	$m_{wall,r}$	–	(4.25)
04.10.31,01-04	V_{wall}	τ_{wall}	(4.26)
04.15.30,01	$A_{wall,tot}$	–	(4.25)
04.15.05,01-05	$A_{wall,tot}$	$d_{w,i}$	(4.26)

Table 4.6: Parameters and price models used for calculating the cost of constructing the external walls.

Concrete decks

The following jobs are considered when calculating the cost of constructing the concrete decks:

Job no. 04.10.01,02: forming

Job no. 04.10.27,01: bending and placing reinforcement

Job no. 04.10.31,08-11: concrete casting.

The areas of the concrete decks are assumed to be the same as the area A_{ext} of the layout of the building.

The total area A_{deck} of the concrete decks is given by:

$$A_{deck} = (N - 1) \cdot A_{ext}, \tag{4.88}$$

since the number of concrete decks to construct is one less than there are floors.

The unit price of forming only depends on the area A_{deck}.

The unit price of bending and placing reinforcement only depends on the total weight $m_{deck,r}$ of the reinforcement rods. The weight $m_{deck,r}$ can be calculated using an expression similar to (4.82):

$$m_{deck,r} = \frac{\pi}{2 \cdot d_{deck,m}} \cdot \varrho_r \cdot A_{deck} \cdot n_{deck,m} \cdot d_{deck,r}^2, \tag{4.89}$$

where $d_{deck,m}$ is the width of the mesh cells used in the reinforcement meshes in the concrete decks. Furthermore, $n_{deck,m}$ is the number of meshes, and $d_{deck,r}$ is the diameter of the reinforcement rods.

The unit price of concrete casting depends on the total volume V_{deck} of the concrete decks, and the compression strength τ_{deck} of the concrete used in the decks, which is used as a secondary parameter. The volume V_{deck} is given by the expression:

$$V_{deck} = A_{deck} \cdot h_{deck}, \tag{4.90}$$

where h_{deck} is the height of the concrete decks, which is assumed to be identical for all decks.

The parameters and the price models used when calculating the cost of constructing the concrete decks are summarized in Table 4.7.

Job no.	Units	Sec. par.	Model
04.10.01,02	A_{deck}	–	(4.25)
04.10.27,01	$m_{deck,r}$	–	(4.25)
04.10.31,08-11	V_{deck}	τ_{deck}	(4.26)

Table 4.7: Parameters and price models used for calculating the cost of constructing the concrete decks.

Roof construction

A flat roof is used in order to calculate the cost of constructing the roof, with the following layers (from top to bottom):

1. Roofing foil
2. Insulation material
3. Roofing paper
4. Vapor seal
5. Structural (a concrete deck).

This involves the following construction jobs:

Job no. 04.10.01,02: forming

Job no. 04.10.27,01: bending and placing reinforcement

Job no. 04.10.31,08-11: concrete casting

Job no. 04.25.17: laying roofing paper including vapor seal

Job no. 04.25.31: laying insulating material and roofing foil.

The unit price of forming and installing roofing paper only depends on the area of the roof construction, which is assumed to be A_{ext}.

The unit price of bending and placing reinforcement only depends on the weight of the reinforcement rods. The weight $m_{roof,r}$ can be calculated using an expression similar to (4.82):

$$m_{roof,r} = \frac{\pi}{2 \cdot d_{roof,m}} \cdot \varrho_r \cdot A_{ext} \cdot n_{roof,m} \cdot d_{roof,r}^2, \tag{4.91}$$

where $d_{roof,m}$ is the width of the mesh cells used in the reinforcement meshes in the roof construction. Furthermore, $n_{roof,m}$ is the number of meshes, and $d_{roof,r}$ is the diameter of the reinforcement rods.

The unit price of concrete casting depends on the total volume V_{roof} of the concrete deck used in the roof construction, which is given by:

$$V_{roof} = A_{ext} \cdot h_{roof}, \tag{4.92}$$

where h_{roof} is the height of the concrete deck. The unit price furthermore depends on the compression strength τ_{roof} of the concrete used in the deck, which is used as a secondary parameter.

The unit price of laying roofing paper and vapor seal depends on A_{ext}.

The unit price of laying insulating material and roofing foil depends on A_{ext}, and the thickness $d_{r,i}$ of the insulation material, which is used as a secondary parameter.

The parameters and the price models used when calculating the cost of constructing the roof are summarized in Table 4.8.

Job no.	Units	Sec. par.	Model
04.10.01,02	A_{ext}	–	(4.25)
04.10.27,01	$m_{roof,r}$	–	(4.25)
04.10.31,08-11	V_{roof}	τ_{roof}	(4.26)
04.25.17	A_{ext}	–	(4.25)
04.25.31	A_{ext}	$d_{r,i}$	(4.26)

Table 4.8: Parameters and price models used for calculating the cost of constructing the roof.

4.4 Processing the output

This section concerns the methods for calculating the performance measures, based on the output from the performance calculation methods.

4.4.1 Energy related performance measures

The Danish building regulations [13] include the following contributions to the annual amount of energy Q_{tot} delivered to non-residential buildings: energy for heating and cooling the building, ventilation, producing domestic hot water, and artificial lighting.

The effect of internal loads and solar gains must be included. Furthermore, since buildings use many types of energy sources, they are weighted with primary energy factors. Electricity is weighted with 2.5, and energy sources such as oil, gas and distributed heating are weighted with 1.

Only energy delivered by the district heating system, Q_{dh}, and energy delivered by the power grid, Q_{el}, are considered, therefore Q_{tot} is given by:

$$Q_{tot} = Q_{dh} + 2.5 \cdot Q_{el}. \tag{4.93}$$

The energy Q_{dh} is used for heating the building and for producing domestic hot water:

$$Q_{dh} = Q_h + Q_w, \tag{4.94}$$

where Q_h is the energy for heating, and Q_w is the energy for producing domestic hot water.

The electric energy is used for cooling the building, for ventilation, and for providing artificial lighting:

$$Q_{el} = Q_{c,el} + Q_v + Q_l, \tag{4.95}$$

where $Q_{c,el}$ is electric energy for cooling, Q_v is the electric energy for ventilating, and Q_l is the electric for artificial lighting.

The annual amount of energy Q_h, that must be delivered for heating the building, and the annual amount of energy Q_c that must be removed for cooling the building, are given by:

$$Q_h = \frac{1}{1000}\sum_{i=1}^{8760}(Q_h')_i \cdot \Delta t \tag{4.96}$$

$$Q_c = \frac{1}{1000}\sum_{i=1}^{8760}(Q_c')_i \cdot \Delta t, \tag{4.97}$$

where $\Delta t = 1\,\mathrm{h}$ is the sample rate. The effect of internal loads and solar gains are included in Q_h and Q_c.

The energy efficiency of the district heating unit is not considered. The energy efficiency of the cooling system must be considered in order to find the amount of electric energy $Q_{c,el}$ it requires. Q_c and $Q_{c,el}$ are related through the coefficient of performance (COP) for the cooling system. The COP-value η_c for the cooling system is given by:

$$\eta_c = \frac{Q_c}{Q_{c,el}} \quad \Leftrightarrow \quad Q_{c,el} = \frac{Q_c}{\eta_c}. \tag{4.98}$$

The energy Q_w required for producing domestic hot water is calculated by assuming that the total volume of the hot water is proportional with the heated floor area A_{tot}. For non-residential buildings, the Danish building regulations [13] assume that 100 liters of hot water is required annually for every m^2 of heated floor area.

Therefore, the annual volume V_w of hot water required is

$$V_w = A_{tot} \cdot 100\,\mathrm{l/m}^2, \tag{4.99}$$

and the mass m_w of the water is

$$m_w = \varrho_w \cdot A_{tot} \cdot 100\,\mathrm{l/m}^2, \tag{4.100}$$

where ϱ_w is the density of water. The annual amount of energy Q_w required for heating the domestic hot water is:

$$Q_w = C_w \cdot \Delta T_w = m_w \cdot c_w \cdot \Delta T_w = \varrho_w \cdot A_{tot} \cdot c_w \cdot \Delta T_w \cdot 100\,\mathrm{l/m}^2, \tag{4.101}$$

where C_w and c_w are the heat capacity and the specific heat capacity, respectively, for the hot water, and where ΔT_w is the temperature difference required for heating the water.

The specific fan power (SFP-value) ε_v for the ventilation fan is defined as the ratio between the electrical power Q_v' required by the fan and the volume air flow V':

$$\varepsilon_v = \frac{Q_v'}{V'} \quad \Leftrightarrow \quad Q_v' = V' \cdot \varepsilon_v. \tag{4.102}$$

The annual amount of electric energy Q_v for ventilating the building is thus given by:

$$Q_v = \frac{1}{1000}\sum_{i=1}^{8760}(Q_v')_i \cdot \Delta t. \tag{4.103}$$

The annual amount of energy Q_l required for artificial lighting is calculated in a similar way as by Nielsen [47], except that the average daylight factor DF_{avg} is assumed to be constant, and thereby independent of the window area. A small value is used in order to provide a conservative estimate of Q_l.

Nielsen propose that 2 W of power is used per m^2 of internal floor area when the average internal illumination level I_{avg} is between 100 and 500 lux, and that 5 W of power is used per m^2 when I_{avg} is below 100 lux, that is:

$$Q'_l = \begin{cases} 0\,\mathrm{W} & \text{for} \quad I_{avg} \geq 500\,\mathrm{lux} \\ 2\,\mathrm{W/m^2} \cdot A_{int} & \text{for} \quad 100\,\mathrm{lux} \leq I_{avg} < 500\,\mathrm{lux} \\ 5\,\mathrm{W/m^2} \cdot A_{int} & \text{for} \quad I_{avg} < 100\,\mathrm{lux} \end{cases}, \tag{4.104}$$

where Q'_l is the power required for lighting. I_{avg} is related to DF_{avg} in the following way:

$$I_{avg} = DF_{avg} \cdot I_h, \tag{4.105}$$

where I_h is the global illuminance, which can be found in DRY climate data sets. It is assumed that electric lighting is only used when the building is used. Given the DRY data, the daylight factor, and the periods where the building is used, the following parameters can be calculated:

$$\varphi_1 \quad : \quad \text{annual number of hours where} \quad 100\,\mathrm{lux} \leq I_{avg} < 500\,\mathrm{lux} \tag{4.106}$$

$$\varphi_2 \quad : \quad \text{annual number of hours where} \quad I_{avg} < 100\,\mathrm{lux} \tag{4.107}$$

The parameters φ_1 and φ_2 only include periods where the building is used.

The following expression is used in order to calculate the annual amount of energy Q_l that is required for lighting:

$$Q_l = \frac{A_{int}}{1000}\left(2\,\mathrm{W/m^2} \cdot \varphi_1 + 5\,\mathrm{W/m^2} \cdot \varphi_2\right) \tag{4.108}$$

The annual amount of energy Q_{tot} required by the building is thus expressed in terms of the output from the performance calculation methods, as well as constants.

Given Q_{tot}, the performance measures EF_3, EF_2 and EF_1 can be calculated using the expressions (4.16), (4.17) and (4.18).

The performance measure Q_{be} is calculated using the expression (4.21), which requires the heat loss Q'_e and the area A_e of the building envelope. The heat loss Q'_e through the building envelope, not including windows and doors, is given by:

$$Q'_e = \left(U_{wall} \cdot (A^{(1)}_{wall} + A^{(2)}_{wall}) + U_r \cdot A_{ext} + U_g \cdot A_{ext} + \Psi_{fw} \cdot O_{ext}\right) \cdot \Delta T_{hl}, \tag{4.109}$$

where ΔT_{hl} is the design temperature difference. The area of the building envelope A_e, not including windows and doors, is given by:

$$A_e = A^{(1)}_{wall} + A^{(2)}_{wall} + 2 \cdot A_{ext}. \tag{4.110}$$

The expressions required for calculating U_g, U_{wall}, U_r, $U^{(1)}_{win}$ and $U^{(2)}_{win}$, are provided in Section 4.3.3. The expressions required for calculating $A^{(1)}_{wall}$, $A^{(2)}_{wall}$ and A_{ext} are provided in Section 4.3.1.

4.4.2 Performance measures for the indoor environment

The annual number of hours with overheating is calculated by making linear interpolation between the internal air temperatures T_a for the time steps t_i, $i = 1, \ldots, 8760$, as illustrated in Figure 4.12. Overheating occur when T_a exceeds the maximum allowed internal air temperature T_{max}.

Only time periods where the building is occupied contribute to the annual number of hours with overheating.

Rounding errors occur when calculating T_a. This must be taken into consideration when calculating time periods with overheating, since T_{max} is often used as set point for the cooling system. The effect of rounding errors is that when the cooling system is active, T_a is not exactly equal to T_{max}, as indicated in Figure 4.12.

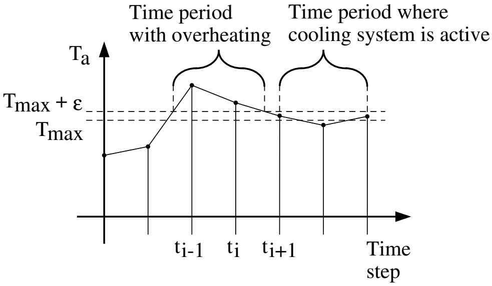

Time period with overheating
Time period where cooling system is active
Ta
Tmax + ε
Tmax
ti-1
ti
ti+1
Time step

Figure 4.12: An illustration of the method used for calculating the annual number of hours with overheating.

The time periods where overheating occur is therefore not calculated by comparing T_a with T_{max}, but by comparing T_a with $\hat{T}_{max} = T_{max} + \varepsilon$, where ε is a tolerance level. Using $\varepsilon \simeq 10^{-3}$K usually provides satisfactory results.

The contribution Δt_i to the annual number of hours with overheating, from the time period ranging from t_i to t_{i+1}, is calculated using the following linear interpolation:

$$\hat{T}_a(\Delta t) = T_i \,(1 - \Delta t) + T_{i+1}\, \Delta t. \tag{4.111}$$

The time Δt^*, where the interpolated internal air temperature becomes equal to $\hat{T}_{max}$, is given by:

$$T_i\,(1 - \Delta t^*) + T_{i+1}\,\Delta t^* = \hat{T}_{max} \quad \Leftrightarrow \quad \Delta t^* = \frac{\hat{T}_{max} - T_i}{T_{i+1} - T_i}. \tag{4.112}$$

The following scheme is used for calculating Δt_i:

$$\Delta t_i = \begin{cases} 0 & \text{if the building is empty} \\ 0 & \text{if} \;\; T_i < \hat{T}_{max} \;\; \text{and} \;\; T_{i+1} < \hat{T}_{max} \\ 1 - \Delta t^* & \text{if} \;\; T_i < \hat{T}_{max} \;\; \text{and} \;\; T_{i+1} \geq \hat{T}_{max} \\ \Delta t^* & \text{if} \;\; T_i \geq \hat{T}_{max} \;\; \text{and} \;\; T_{i+1} < \hat{T}_{max} \\ 1 & \text{if} \;\; T_i \geq \hat{T}_{max} \;\; \text{and} \;\; T_{i+1} \geq \hat{T}_{max} \end{cases}, \quad \text{for} \;\; i = 1, \ldots, 8759. \tag{4.113}$$

This approach is applied to the simulation results from both thermal zones, which provides the contributions $\Delta t_i^{(1)}$ and $\Delta t_i^{(2)}$, $i = 1, \ldots, 8759$. The annual number of hours with overheating for the two thermal zones, denoted $OH^{(1)}$ and $OH^{(2)}$, respectively, can thus be calculated using the following expressions:

$$OH^{(1)} = \sum_{i=1}^{8759} \Delta t_i^{(1)} \tag{4.114}$$

$$OH^{(2)} = \sum_{i=1}^{8759} \Delta t_i^{(2)} \tag{4.115}$$

The performance measures $DH^{(1)}$ and $DH^{(2)}$, which are used as very simple measures for the utilization level of natural light, are given by:

$$DH^{(1)} = \frac{w_{int}}{h_{win}^{(1)}} \tag{4.116}$$

$$DH^{(2)} = \frac{w_{int}}{h_{win}^{(2)}}, \tag{4.117}$$

where w_{int} is given by the expression (4.47), and where $h_{win}^{(1)}$ and $h_{win}^{(2)}$ are given by (4.52) and (4.53), respectively.

4.4.3 Performance measures for the economy

Given the unit prices $P \in \mathbb{R}^{n_{jobs}}$ for the construction jobs, the construction costs C_{con} for the building is given by the following expression:

$$C_{con} = \sum_{i=1}^{n_{jobs}} u_i \cdot P_i, \tag{4.118}$$

where u_i is the number of units required for construction job i. The number of units for the required construction jobs are provided in Tables 4.4, 4.5, 4.6, 4.7 and 4.8.

Only the cost of energy is considered when calculating the annual cost C_{op} of operating the building. C_{op} is therefore given by:

$$C_{op} = p_{el} \cdot Q_{el} + p_{dh} \cdot Q_{dh}, \tag{4.119}$$

where p_{el} and p_{dh} are energy prices for electrical energy and energy supplied by the district heating system, respectively.

4.5 Domain constraints

The domain constraints are used for ensuring that the input to the performance calculation methods is feasible. The input is assumed to be feasible, if the decisions suggested by the optimization algorithm can be carried out physically. All continuous decision variables are therefore subjected to lower bounds, in order to prevent them to become negative or below reasonable values. The window fractions and the width to length ratio are furthermore subjected to upper bounds, in order to prevent these parameters to become unrealistic. Finally, the physical window properties are required to be positive. The economical window properties, that is, the price model parameters, are not subjected to any requirements.

The following bounds are used for the continuous decision variables:

$$\begin{array}{rcccl} 0.1 & \leq & \varrho & \leq & 10 \\ & & N & \geq & 0.95 \\ 0.1 & \leq & \sigma^{(1)} & \leq & 0.9 \\ 0.1 & \leq & \sigma^{(2)} & \leq & 0.9 \\ & & d_{w,i} & \geq & 0.01 \\ & & d_{r,i} & \geq & 0.01 \\ & & d_{g,i} & \geq & 0.01 \end{array} \tag{4.120}$$

The entries of $w^{(1)}$ and $w^{(2)}$ that represent physical window properties are required to be positive:

$$w_i^{(1)} \geq 0 \quad \text{for} \quad i = 1, \ldots, 4 \tag{4.121}$$

$$w_i^{(2)} \geq 0 \quad \text{for} \quad i = 1, \ldots, 4 \tag{4.122}$$

There are no requirements to the remaining entries of $w^{(1)}$ and $w^{(2)}$, which represent price model parameters.

These considerations provide the following domain constraint function:

$$d(x) = \begin{bmatrix} \varrho & - & 0.1 \\ 10 & - & \varrho \\ N & - & 0.95 \\ \sigma^{(1)} & - & 0.1 \\ 0.9 & - & \sigma^{(1)} \\ \sigma^{(2)} & - & 0.1 \\ 0.9 & - & \sigma^{(2)} \\ d_{w,i} & - & 0.01 \\ d_{r,i} & - & 0.01 \\ d_{g,i} & - & 0.01 \\ w_1^{(1)} & & \\ \vdots & & \\ w_4^{(1)} & & \\ w_1^{(2)} & & \\ \vdots & & \\ w_4^{(2)} & & \end{bmatrix} \tag{4.123}$$

4.6 Final remarks

Implementing the proposed building optimization method requires numerical optimization methods for estimating solutions to (3.8). An optimization method intended for this purpose is described in Chapter 5.

The building optimization method can also be combined with detailed, time consuming methods for calculating the performance measures, which are likely to use an excessive amount of time. In this case, the optimization method described in Appendix A may be relevant for estimating solutions to (3.8).

Chapter 5

A gradient-free SQP filter algorithm

This chapter concerns a sequential quadratic programming (SQP) algorithm for solving optimization problems in the form (2.4), for instance the optimization problem (3.8). The algorithm is based on the sequential linear programming (SLP) filter algorithm described by Fletcher [23], but with a number of modifications for addressing the requirements described in Chapter 3.

5.1 Introduction

The first modification is that instead of restricting the step length using upper and lower bounds on the parameters, also known as box constraints, a quadratic damping term is added to the objective function. This implies that the trust region subproblems become quadratic programs rather than linear programs, hence the name. The algorithm uses an approach that combines a trust region methodology with a damped Newton approach. Damped Newton algorithms include the one proposed by Levenberg [40] and later by Marquardt [43].

The second modification is to ensure that the algorithm is able to handle domain constraints. The algorithm is not allowed to calculate the objective function, or the inequality or equality constraint functions outside their domain. The algorithm calculates the domain constraint functions in all iterations, and if it encounters a point outside the domain, it makes a step towards the domain, using an approach that is similar to the restoration step made by Fletchers algorithm. The steps towards the domain are therefore referred to as domain restoration steps.

The third modification is to make the algorithm gradient-free by approximating the first partial derivatives of the objective function, and the inequality, equality and domain constraint functions, using the Broyden rank one updating formula [12].

The following three algorithms are described in this chapter:

SLPF: A variant of Fletchers SLP filter algorithm, where the trust region radius is updated in the same way as the next algorithm, and that is able to handle domain constraints.

SQPF: The SQP filter algorithm, which uses a quadratic damping term for restricting the step length.

GFSQPF: A gradient-free version of the SQP filter algorithm, which uses Broyden updated approximations of the first partial derivatives.

The only difference between SLPF and SQPF is that SLPF uses box constraints, where SQPF uses a quadratic damping term. These two algorithms are compared in order to determine how much the performance is influenced by using the quadratic damping term.

The only difference between SQPF and GFSQPF is that SQPF requires that the first partial derivatives, or Jacobian matrices, for the functions defining the optimization problem, are available to the algorithm, where GFSQPF approximates the Jacobians using finite difference approximations in the first iteration, which are updated in subsequent iterations using Broydens rank one formula. These two algorithms are compared in order to determine how much the performance is influenced by using Broyden updated approximations rather than exact gradient information.

The SQPF algorithm has more in common with Fletchers algorithm than with "genuine" SQP algorithms, such as the ones described by Conn et.al. [14], Chapter 15. This is because the only purpose of the quadratic damping term is to restrict the step length. No attempts are made to estimate second order derivatives of the Lagrange function of the problem.

The details regarding the gradient-based algorithms SLPF and SQPF are described in subsequent sections, and the algorithms are summarized in Section 5.7. The details regarding the gradient-free algorithm GFSQPF are described in Section 5.8, and the algorithm is summarized in Section 5.9. The numerical experiments are described in Chapter 6.

5.2 The trust region subproblems

This section concerns the trust region subproblems used by the SLPF and SQPF algorithms for estimating a step towards the solution to (2.4). It is assumed that the first partial derivatives to f, $c_{\mathcal{I}}$, $c_{\mathcal{E}}$ and d exist and are known to the algorithm. The assumption that the derivatives are known is abandoned in later sections, however, the existence of the derivatives is assumed throughout the thesis.

If the iterate x_k belongs to the domain $\mathcal{D}$, i.e. if $d(x_k) \geq 0$, then the function values $f(x_k)$, $c_{\mathcal{I}}(x_k)$ and $c_{\mathcal{E}}(x_k)$, and also the first partial derivatives $\nabla f(x_k)$, $J_{c_{\mathcal{I}}}(x_k)$ and $J_{c_{\mathcal{E}}}(x_k)$ are available to the algorithm. Furthermore, the function value $d(x_k)$ and first partial derivatives $J_d(x_k)$ are available in all iterations.

These quantities are used for making the following Taylor approximations:

$$\begin{aligned} f(x_k + \Delta x) &\simeq f_k + \nabla f_k^\top \Delta x && (5.1)\\ c_\mathcal{I}(x_k + \Delta x) &\simeq c_{\mathcal{I},k} + J_{c_\mathcal{I},k}\, \Delta x && (5.2)\\ c_\mathcal{E}(x_k + \Delta x) &\simeq c_{\mathcal{E},k} + J_{c_\mathcal{E},k}\, \Delta x && (5.3)\\ d(x_k + \Delta x) &\simeq d_k + J_{d,k}\, \Delta x, && (5.4) \end{aligned}$$

Where the notation $f_k = f(x_k)$ is used, and similarly for the other parameters.

In order to improve the iterate, the SLPF algorithm solves an LP subproblem, which is formed by substituting the approximations (5.1) to (5.3) into (2.4), and augmenting the resulting problem with upper bounds on the allowed step length:

$$\text{LP}(x_k, \rho_k) \left\{ \begin{array}{ll} \text{minimize} & \nabla f_k^\top \Delta x \\ \text{subject to} & c_{\mathcal{I},k} + J_{c_\mathcal{I},k}\, \Delta x \geq 0 \\ & d_k + J_{d,k}\, \Delta x \geq 0 \\ & c_{\mathcal{E},k} + J_{c_\mathcal{E},k}\, \Delta x = 0 \\ & \|\Delta x\|_\infty \leq \rho_k \\ \text{with respect to} & \Delta x \in \mathbb{R}^n, \end{array} \right. \qquad (5.5)$$

where the constant term in the objective function is omitted, since it does not influence the solution. The box constraints $\|\Delta x\|_\infty \leq \rho_k$ can be translated into the following linear constraints:

$$\Delta x \geq -\rho_k \quad \text{and} \quad \Delta x \leq \rho_k \qquad (5.6)$$

Instead of restricting the step length with box constraints, they can be restricted by adding a quadratic damping term to the objective function. This implies the following QP subproblem:

$$\text{QP}(x_k, \mu_k) \left\{ \begin{array}{ll} \text{minimize} & \nabla f_k^\top \Delta x + \frac{1}{2}\mu_k\, \Delta x^\top \Delta x \\ \text{subject to} & c_{\mathcal{I},k} + J_{c_\mathcal{I},k}\, \Delta x \geq 0 \\ & d_k + J_{d,k}\, \Delta x \geq 0 \\ & c_{\mathcal{E},k} + J_{c_\mathcal{E},k}\, \Delta x = 0 \\ \text{with respect to} & \Delta x \in \mathbb{R}^n, \end{array} \right. \qquad (5.7)$$

where the constant term in the objective function again is omitted. This approach is used by the SQPF and GFSQPF algorithms.

The damping parameter μ_k is calculated in the following way:

$$\mu_k = \frac{\|\nabla f_k\|_2}{\rho_k} \qquad (5.8)$$

This ensures that $\|\Delta x_k\|_2 = \rho_k$, if (5.7) is unconstrained.

5.3 Regular restoration steps

The purpose of the regular restoration steps is to make a step towards the feasible region when the LP or QP subproblem becomes incompatible. The approach used in this thesis is to estimate a step that minimizes the largest constraint violation, including the domain constraints, i.e. a step intended for the following problem:

$$\begin{array}{ll} \text{minimize} & \max\{v(x)\} \\ \text{with respect to} & x \in \mathcal{D}, \end{array} \tag{5.9}$$

where

$$v(x) = \begin{bmatrix} -c_{\mathcal{I}}(x) \\ c_{\mathcal{E}}(x) \\ -c_{\mathcal{E}}(x) \\ -d(x) \end{bmatrix} \tag{5.10}$$

In order to formulate LP or QP subproblems to (5.9), the following Taylor approximation is used:

$$v(x_k + \Delta x) \simeq v_k + J_{v,k}\,\Delta x, \tag{5.11}$$

where $v_k = v(x_k)$ and $J_{v,k} = J_v(x_k)$. First we consider the LP subproblem.

Substituting (5.11) into (5.9) and augmenting the problem with box constraints gives the following approximated problem:

$$\begin{array}{ll} \text{minimize} & \max\{v_k + J_{v,k}\,\Delta x\} \\ & \|\Delta x\|_\infty \leq \rho_k \\ \text{with respect to} & \Delta x \in \mathbb{R}^n, \end{array} \tag{5.12}$$

By introducing the auxiliary parameter

$$\Delta\hat{x} = \max\{v_k + J_{v,k}\,\Delta x\}, \tag{5.13}$$

and using the relations

$$\Delta\hat{x} \geq v_k + J_{v,k}\,\Delta x, \tag{5.14}$$

then (5.12) can be rearranged as the following LP problem:

$$\text{RRLP}(x_k, \rho_k) \left\{ \begin{array}{ll} \text{minimize} & \Delta\hat{x} \\ \text{subject to} & \Delta\hat{x} \geq v_k + J_{v,k}\,\Delta x \\ & \|\Delta x\|_\infty \leq \rho_k \\ \text{with respect to} & \Delta x \in \mathbb{R}^n \text{ and } \Delta\hat{x} \in \mathbb{R} \end{array} \right. \tag{5.15}$$

Next we consider the QP subproblem. Instead of using box constraints, a quadratic damping term is added to the objective function of the approximated problem:

$$\begin{array}{ll} \text{minimize} & \max\{v_k + J_{v,k}\,\Delta x\} + \tfrac{1}{2}\mu_k\,\Delta x^\top \Delta x \\ \text{with respect to} & \Delta x \in \mathbb{R}^n, \end{array} \tag{5.16}$$

Substituting (5.13) into (5.16) and using the relations (5.14) gives the following QP problem:

$$\text{RRQP}(x_k, \mu_k) \begin{cases} \text{minimize} & \Delta\hat{x} + \frac{1}{2}\mu_k\, \Delta x^\top \Delta x \\ \text{subject to} & \Delta\hat{x} \geq v_k + J_{v,k}\, \Delta x \\ \text{with respect to} & \Delta x \in \mathbb{R}^n \text{ and } \Delta\hat{x} \in \mathbb{R} \end{cases} \tag{5.17}$$

Before solving (5.17), the damping parameter μ_k must be calculated. In the following, an approach is described that seems to provide a step that is at most of length ρ_k. The aim is to estimate a relation between μ_k and ρ_k, which is achieved by solving a simplified version of (5.17).

The set of active constraints for (5.17) can be expressed as:

$$\mathcal{A} = \{i \in \mathbb{N} : v_i(x_k) = \max\{v(x_k)\}\}\,. \tag{5.18}$$

The number of active constraints is denoted $n_\mathcal{A}$. The simplified version of (5.17) is formed by only including the active constraints:

$$\text{RRQP}_\mathcal{A}(x_k, \mu) \begin{cases} \text{minimize} & \Delta\hat{x} + \frac{1}{2}\mu\, \Delta x^\top \Delta x \\ \text{subject to} & \Delta\hat{x} \geq v_\mathcal{A} + J_{v,\mathcal{A}}\, \Delta x \\ \text{with respect to} & \Delta x \in \mathbb{R}^n \text{ and } \Delta\hat{x} \in \mathbb{R} \end{cases} \tag{5.19}$$

where $v_\mathcal{A} \in \mathbb{R}^{n_\mathcal{A}}$ denotes the subset of v_k consisting of active constraints, and where $J_{v,\mathcal{A}}$ are the corresponding first partial derivatives.

Since $v_\mathcal{A}$ only consists of active constraints, it can be expressed as:

$$v_\mathcal{A} = v_{max}\, u_\mathcal{A} \tag{5.20}$$

where

$$v_{max} = \max\{v_k\}\,, \tag{5.21}$$

and where

$$u_\mathcal{A} = [1, \ldots, 1]^\top \in \mathbb{R}^{n_\mathcal{A}}. \tag{5.22}$$

The Lagrange function for (5.19) is:

$$\mathcal{L}(\Delta\hat{x}, \Delta x, \lambda) = \Delta\hat{x} + \frac{1}{2}\mu\, \Delta x^\top \Delta x - (\Delta\hat{x}\, u_\mathcal{A} - J_{v,\mathcal{A}}\Delta x - v_\mathcal{A})^\top \lambda, \tag{5.23}$$

where $\lambda \in \mathbb{R}^{n_\mathcal{A}}$ are the Lagrange multipliers. The first order optimality conditions for (5.19) are:

$$\begin{aligned} \nabla_{\Delta\hat{x}}\mathcal{L}(\Delta\hat{x}, \Delta x, \lambda) &= 0 && (5.24)\\ \nabla_{\Delta x}\mathcal{L}(\Delta\hat{x}, \Delta x, \lambda) &= 0 && (5.25)\\ \nabla_{\lambda}\mathcal{L}(\Delta\hat{x}, \Delta x, \lambda) &= 0 && (5.26)\\ \lambda &\geq 0, && (5.27) \end{aligned}$$

where the first partial derivatives of $\mathcal{L}$ with respect to $\Delta\hat{x}$, Δx and λ are given by:

$$\begin{aligned}
\nabla_{\Delta\hat{x}}\mathcal{L}(\Delta\hat{x},\Delta x,\lambda) &= 1-u_{\mathcal{A}}^{\top}\lambda && (5.28)\\
\nabla_{\Delta x}\mathcal{L}(\Delta\hat{x},\Delta x,\lambda) &= \mu\Delta x-J_{v,\mathcal{A}}^{\top}\lambda && (5.29)\\
\nabla_{\lambda}\mathcal{L}(\Delta\hat{x},\Delta x,\lambda) &= \Delta\hat{x}\,u_{\mathcal{A}}-J_{v,\mathcal{A}}\,\Delta x-v_{\mathcal{A}}\\
&= \Delta\hat{x}\,u_{\mathcal{A}}-J_{v,\mathcal{A}}\,\Delta x-v_{max}\,u_{\mathcal{A}}. && (5.30)
\end{aligned}$$

Assuming that the first order optimality conditions (5.24) to (5.27) for $\mathrm{RRQP}_{\mathcal{A}}(x_k,1)$ are satisfied by the triple ($\Delta\hat{x}^*$, Δx^*, λ^*), that is:

$$\begin{aligned}
1-u_{\mathcal{A}}^{\top}\lambda^* &= 0 && (5.31)\\
\Delta x^*-J_{v,\mathcal{A}}^{\top}\lambda^* &= 0 && (5.32)\\
\Delta\hat{x}^*\,u_{\mathcal{A}}-J_{v,\mathcal{A}}\,\Delta x^*-v_{max}\,u_{\mathcal{A}} &= 0 && (5.33)\\
\lambda^* &\geq 0, && (5.34)
\end{aligned}$$

then the following triple satisfies the first order optimality conditions for $\mathrm{RRQP}_{\mathcal{A}}(x_k,\mu)$:

$$\left(\frac{1}{\mu}\Delta\hat{x}^*+\left(1-\frac{1}{\mu}\right)v_{max},\ \ \frac{1}{\mu}\Delta x^*,\ \ \lambda^*\right). \tag{5.35}$$

This can be verified by inserting the triple (5.35) into the optimality conditions (5.24) to (5.27):

$$\begin{aligned}
\nabla_{\Delta\hat{x}}\mathcal{L} &= 1-u_{\mathcal{A}}^{\top}\lambda^*=0 && (5.36)\\
\nabla_{\Delta x}\mathcal{L} &= \mu\frac{1}{\mu}\Delta x^*-J_{v,\mathcal{A}}^{\top}\lambda^*=\Delta x^*-J_{v,\mathcal{A}}^{\top}\lambda^*=0 && (5.37)\\
\nabla_{\lambda}\mathcal{L} &= \left(\frac{1}{\mu}\Delta\hat{x}^*+\left(1-\frac{1}{\mu}\right)v_{max}\right)\,u_{\mathcal{A}}-J_{v,\mathcal{A}}\,\frac{1}{\mu}\Delta x^*-v_{max}\,u_{\mathcal{A}}\\
&= \frac{1}{\mu}\Delta\hat{x}^*u_{\mathcal{A}}+v_{max}u_{\mathcal{A}}-\frac{1}{\mu}v_{max}u_{\mathcal{A}}-J_{v,\mathcal{A}}\,\frac{1}{\mu}\Delta x^*-v_{max}\,u_{\mathcal{A}}\\
&= \frac{1}{\mu}\left(\Delta\hat{x}^*u_{\mathcal{A}}-J_{v,\mathcal{A}}\,\Delta x^*-v_{max}u_{\mathcal{A}}\right)=0 && (5.38)\\
\lambda^* &\geq 0, && (5.39)
\end{aligned}$$

This means that the solution to $\mathrm{RRQP}_{\mathcal{A}}(x_k,\mu)$ can be expressed in terms of the solution to $\mathrm{RRQP}_{\mathcal{A}}(x_k,1)$, since if Δx^* is a solution $\mathrm{RRQP}_{\mathcal{A}}(x_k,1)$, then $\Delta x^*/\mu$ is a solution to $\mathrm{RRQP}_{\mathcal{A}}(x_k,\mu)$.

The idea is to use this relation to calculate a value for the damping parameter that provides a step length of ρ_k. The length of the step obtained by solving $\mathrm{RRQP}_{\mathcal{A}}(x_k,\mu)$ is thus required to be equal to the trust region radius:

$$\left\|\frac{1}{\mu_k}\Delta x^*\right\|_2=\rho_k, \tag{5.40}$$

which gives:

$$\mu_k = \frac{\|\Delta x^*\|_2}{\rho_k}. \tag{5.41}$$

The damping parameter can thus be calculated using the following two steps:

1. Calculate Δx^* by solving $\mathrm{RRQP}_{\mathcal{A}}(x_k, 1)$
2. Calculate μ_k using (5.41).

5.3.1 An example

The example described in the following concerns an optimization problem with 4 inequality constraints and no equality or domain constraints.

Assume an iterate x_k is found with

$$v_k = \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix} \quad \text{and} \quad J_{v,k} = \begin{bmatrix} -2 & 5 \\ -2 & 0 \\ 2 & 2 \\ 1 & -2 \end{bmatrix}, \tag{5.42}$$

i.e. the first three constraints are violated, and the last constraint is satisfied. The maximum constraint violation is $v_{\max} = 1$, and the set of active constraints is $\mathcal{A} = \{1, 2, 3\}$, which gives:

$$v_{\mathcal{A}} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \text{and} \quad J_{v,\mathcal{A}} = \begin{bmatrix} -2 & 5 \\ -2 & 0 \\ 2 & 2 \end{bmatrix}. \tag{5.43}$$

The solution to $\mathrm{RRQP}_{\mathcal{A}}(x_k, 1)$ is

$$\Delta x^* = \begin{bmatrix} 0.4 \\ -0.8 \end{bmatrix}, \tag{5.44}$$

The damping parameter related to the trust region radius $\rho_k = 1$ is given by

$$\mu_k = \frac{\|\Delta x^*\|_2}{\rho_k} = \sqrt{0.8} \simeq 0.8944 \tag{5.45}$$

In Figure 5.1 are shown the simplified problem $\mathrm{RRQP}_{\mathcal{A}}(x_k, \mu_k)$, where inactive constraints are ignored, and the original problem $\mathrm{RRQP}(x_k, \mu_k)$, where all constraints are included. The trust region is represented by a dashed circle, and the regions where the constraints $\tilde{v}_k(\Delta x) = v_k + J_{v,k}\Delta x$ are active, are also shown.

When applying the damping parameter to $\mathrm{RRQP}_{\mathcal{A}}(x_k, \mu_k)$, it provides a step length of ρ_k, as expected. However, solving $\mathrm{RRQP}(x_k, \mu_k)$ using this damping parameter does not always provide a step length that is equal to ρ_k, since the solution to this problem in some situations is influenced by the constraints that are ignored by $\mathrm{RRQP}_{\mathcal{A}}(x_k, \mu_k)$.

This situation is illustrated in Figure 5.1 (right), where the solution to RRQP(x_k, μ_k) is influenced by $\tilde{v}_4$, which is ignored by RRQP$_{\mathcal{A}}(x_k, \mu_k)$. In this case, the resulting step length is less than ρ_k.

The described approach for calculating the damping parameter therefore seems to provide a step length that is less than or equal to ρ_k, however, a formal proof for this statement is not given. Instead, the step length is compared with ρ_k, and truncated if necessary.

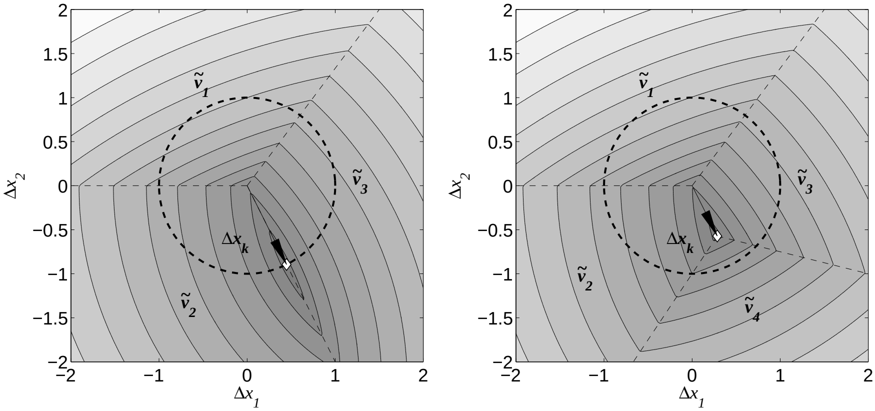


Figure 5.1: Left: The simplified problem RRQP$_{\mathcal{A}}(x_k, \mu_k)$. The dashed circle represents the trust region. Right: The original problem RRQP(x_k, μ_k). The active constraints, and the regions where they are active, are also shown.

5.4 Domain restoration steps

The purpose of the domain restoration steps is to make a step towards the domain when an iterate $x_k \notin \mathcal{D}$ is found. The domain restoration problem is virtually identical to the regular restoration problem (5.9), except that only domain constraints are used for defining the problem:

$$\begin{array}{ll} \text{minimize} & \max\{-d(x)\} \\ \text{with respect to} & x \in \mathbb{R}^n, \end{array} \tag{5.46}$$

The domain restoration subproblems can be derived using the same approach as when deriving the regular restoration problems. This gives the following subproblems:

$$\text{DRLP}(x_k, \rho_k) \left\{ \begin{array}{ll} \text{minimize} & \Delta\hat{x} \\ \text{subject to} & \Delta\hat{x} \geq -d_k - J_{d,k}\,\Delta x \\ & \|\Delta x\|_\infty \leq \rho_k \\ \text{with respect to} & \Delta x \in \mathbb{R}^n \text{ and } \Delta\hat{x} \in \mathbb{R} \end{array} \right. \tag{5.47}$$

and

$$\text{DRQP}(x_k, \mu_k) \begin{cases} \text{minimize} & \Delta\hat{x} + \frac{1}{2}\mu_k \Delta x^\top \Delta x \\ \text{subject to} & \Delta\hat{x} \geq -d_k - J_{d,k}\, \Delta x \\ \text{with respect to} & \Delta x \in \mathbb{R}^n \text{ and } \Delta\hat{x} \in \mathbb{R} \end{cases} \tag{5.48}$$

where $\Delta\hat{x}$ is an auxiliary parameter. The damping parameter μ_k can be calculated in a similar way as with the regular restoration steps. First, the active functions must be determined:

$$\mathcal{A} = \{i \in \mathbb{N} : -d_i(x_k) = \max\{-d(x_k)\}\} \tag{5.49}$$

The approximated QP becomes:

$$\text{DRQP}_{\mathcal{A}}(x_k, \mu) \begin{cases} \text{minimize} & \Delta\hat{x} + \frac{1}{2}\mu\, \Delta x^\top \Delta x \\ \text{subject to} & \Delta\hat{x} \geq -d_{\mathcal{A}} - J_{d,\mathcal{A}}\, \Delta x \\ \text{with respect to} & \Delta x \in \mathbb{R}^n \text{ and } \Delta\hat{x} \in \mathbb{R}, \end{cases} \tag{5.50}$$

where, as before, $d_{\mathcal{A}}$ denotes the subset of d_k only consisting of active domain constraints, and where $J_{d,\mathcal{A}}$ are the corresponding first partial derivatives.

Calculating the damping parameter for the domain restoration subproblem thus consists of the following two steps:

1. Calculate Δx^* by solving $\text{DRQP}_{\mathcal{A}}(x_k, 1)$
2. Calculate μ_k using (5.41).

5.5 The filter concept for nonlinear programming

The filter concept for nonlinear programming is introduced by Fletcher and Leyffer [22], and a slightly different formulation is provided by Fletcher [23].

The aim for all constrained optimization algorithms is to satisfy the following, often conflicting goals:

1. minimize the function value $f(x)$, and
2. minimize the maximum constraint violation $h(x) = \max\{0, \max\{-c_{\mathcal{I}}(x)\}\}$.

The traditional approach is to form a penalty function based on both objectives, and only to accept those iterates that reduce its function value. This approach involves a penalty parameter, which must be updated for all iterations. Details regarding penalty functions for constrained optimization are described by Conn et.al. [14], Chapters 14 and 15.

The filter concept provides a simpler acceptance criteria for the iterates generated by the algorithm, which avoids using penalty functions. Instead, the aforementioned goals are treated as a multi-criteria decision problem based on the pair (h_k, f_k), where $h_k = h(x_k)$ and $f_k = f(x_k)$.

The basic idea is to only accept those iterates x_k that provide a pair (h_k, f_k) that is not dominated, in the Pareto [52] sense of the word, by previous iterates. A pair (h_i, f_i) is said to dominate another pair (h_j, f_j) if both $h_i \leq h_j$ and $f_i \leq f_j$.

The filter is defined as a set of pairs $\{(h_i, f_i)\}$, $i \in \mathcal{F}$, where no pair dominate any other. An iterate x_k is said to be acceptable for inclusion in the filter if its (h_k, f_k) pair is not dominated by any other pair in the filter, i.e. if

$$h_k \leq h_i \ \vee \ f_k \leq f_i \ \text{ for all } \ i \in \mathcal{F} \tag{5.51}$$

In order to ensure convergence for filter algorithms, it is necessary to further increase the requirements (5.51) by introducing a small envelope around the filter, where no iterates are accepted. The criterion for accepting an iterate for inclusion thus become

$$h_k \leq \beta h_i \ \vee \ f_k \leq f_i - \gamma h_k \ \text{ for all } \ i \in \mathcal{F} \tag{5.52}$$

where $1 > \beta > \gamma > 0$, with $\beta \simeq 1$ and $\gamma \simeq 0$ being user-provided parameters. An example of a filter and its envelope is given in Figure 5.2. When an iterate x_k is included in the filter, entries that are dominated by x_k are removed, in order to ensure that no filter entries are dominated by any other.

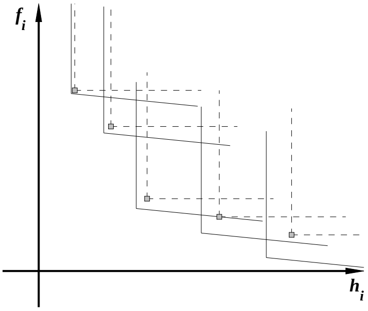


Figure 5.2: A filter and its envelope. The squares represent the filter entries, and the solid lines represent the envelope given by (5.52). The dashed lines represent the criterion (5.51).

The SLP filter algorithm proposed by Fletcher [23] uses a slightly different approach for determining if an iterate is to be included in the filter and used as the next iterate.

The algorithm distinguishes between two types of iterates. A so-called f-type iterate satisfies the following requirements:

$$\Delta f_k \geq \sigma \Delta l_k \ \text{ and } \ \Delta l_k \geq \delta (h_k)^2 \tag{5.53}$$

where

$$\Delta f_k = f_k - f_{k+1} \tag{5.54}$$
$$\Delta l_k = -\nabla f_k^\top \Delta x, \tag{5.55}$$

and where $\sigma \in [\gamma; 1]$ and $\delta > 0$ are used-provided parameters. If the iterate is not an f-type, then it is deemed to be an h-type iterate. In order for an iterate x_k to be included in the filter, it must be acceptable to the filter, and it must be an h-type iterate.

The following measure for the constraint violations is used:

$$h(x) = \max\{0, \max\{v(c)\}\} \tag{5.56}$$

where $v(x)$ is given by (5.10).

5.6 Various details

5.6.1 Updating the trust region radius

Usually ρ_k is updated from one iteration to the next in such a way that if there is a good agreement between the functions that define the optimization problem (2.4), and the linear approximations (5.1) to (5.4), then ρ_k is increased in order to enable the algorithm to take larger steps. If there is a poor agreement, ρ_k is reduced in order to prevent steps that do not improve the iterate x_k.

As a measure for the agreement between the functions and the linear approximations, a prediction factor, or gain factor $r_k \in \mathbb{R}$ is used, which is defined as the ratio between the actual and predicted increment, i.e. $\rho \simeq 1$ if there is a good agreement between the functions and the approximations. The following definition for r_k is used:

$$r_k = \frac{\Delta f_k}{\Delta l_k} \tag{5.57}$$

The definition of Δf and Δl depends on the objective function that the step is based on. When the algorithms are improving the iterates by solving the trust region subproblems, then the definitions (5.54) and (5.55) are used.

When performing regular restoration steps, the following definitions are used:

$$\begin{aligned} \Delta f_k &= h(x_k) - h(x_k + \Delta x_k) && (5.58) \\ \Delta l_k &= h(x_k) - v_{\max}, && (5.59) \end{aligned}$$

where h is given by (5.56), and where $v_{\max}$ is given as part of the solution to (5.15).

When performing domain restoration steps, the following definitions are used:

$$\begin{aligned} \Delta f_k &= \max\{-d(x_k)\} - \max\{-d(x_k + \Delta x_k)\} && (5.60) \\ \Delta l_k &= \max\{-d(x_k)\} - d_{\max}, && (5.61) \end{aligned}$$

where $d_{\max}$ is given as part of the solution to (5.48).

In order to update ρ_k, the following strategy can be used:

$$\rho_{k+1} = \begin{cases} \frac{5}{3}\rho_k & \text{if } r_k > 0.75 \\ \frac{1}{3}\rho_k & \text{if } r_k < 0.25 \\ \rho_k & \text{otherwise} \end{cases} \tag{5.62}$$

which is consistent with the approach suggested by Conn et.al. [14], Algorithm 6.1.1. The practical experience gained by applying the SLPF and SQPF algorithms to the test problems described in appendix C, indicate that the following strategy gives good results:

$$\rho_{k+1} = \rho_k \theta(r_k) \tag{5.63}$$

where

$$\theta(r_k) = \frac{2}{3} \tanh\left(10\left(r_k - \frac{1}{2}\right)\right) + 1 \tag{5.64}$$

The strategies (5.62) and (5.63) are shown in Figure 5.3. The strategy (5.63) for updating ρ_k is similar in concept to the strategy described by Madsen et.al. [41], expression (2.21), except that this expression is used for updating μ_k rather than ρ_k.

A detailed investigation of updating strategies for ρ_k is not conducted, but is a possible topic for further research, since this choice seems to have a significant influence on the performance of the algorithm.

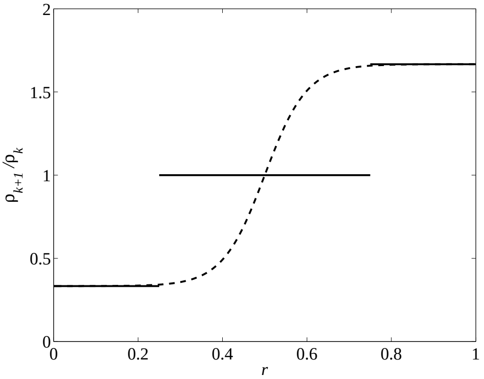


Figure 5.3: The strategies (5.62) (solid lines) and (5.63) (dashed line) for updating ρ_k.

There are other reasons for reducing ρ_k, for instance if a step is made that provides an iterate that is not acceptable to the filter. In these situations, ρ_k is reduced in the following way:

$$\rho_{k+1} = \frac{1}{3}\rho_k, \tag{5.65}$$

otherwise (5.63) is used.

5.6.2 Stopping criteria

The types of criteria considered for deciding when to terminate the algorithms are based on the parameters and the objective function value. The user can specify upper limits on the relative errors for these parameters, so under ideal circumstances, the algorithms should terminate when it suggests an iterate x_{k+1} with

$$\|x_{k+1} - x^*\|_2 < \varepsilon_1 \|x^*\|_2, \tag{5.66}$$

or

$$f_{k+1} - f(x^*) < \varepsilon_2 f(x^*), \tag{5.67}$$

where ε_1 and ε_2 are the tolerance levels provided by the user. Since x^* is obviously not known, it is replaced by the estimated solution x_k. The situations $x^* = 0$ and $f(x^*) = 0$ can be handled by correcting the terms $\|x^*\|_2$ and $f(x^*)$ with a small number, say ε_1 and ε_2, respectively. The stopping criteria thus become:

$$\|\Delta x_k\|_2 < \varepsilon_1(\|x_k\|_2 + \varepsilon_1) \tag{5.68}$$

and

$$f_{k+1} - f(x^*) < \varepsilon_2(f(x^*) + \varepsilon_2), \tag{5.69}$$

where the relation (2.6) is used. Furthermore, as a safeguard feature, it is recommended to specify an upper limit on the number of iterations:

$$k \geq k_{\max} \tag{5.70}$$

5.7 Summary of the gradient-based algorithms

The SLPF and SQPF algorithms require more or less the same steps, except that they use different subproblems for improving the iterate and for performing restoration. Furthermore, the SQPF algorithm requires some additional steps for calculating the damping parameter.

The required steps are summarized in Algorithms 1 and 2, respectively. The algorithms combine the subproblem steps with the restoration steps. Notice that the filter is only used as acceptance criteria when the algorithms are not performing restoration, since the restoration problems are unconstrained problems, where a downhill acceptance criteria is used.

Steps 8 and 11 of Algorithm 1 are used for resetting the trust region radius to its initial value whenever the algorithm enters or leaves the restoration phase.

Evaluating the functions f, $c_{\mathcal{I}}$, $c_{\mathcal{E}}$ and d at x_0 and x_{new} in steps 2, 3, 17 and 19, provides function values as well as the first partial derivatives, which are used for defining the subproblems, and for calculating h using the expression (5.56).

The iterate x_{new} calculated at step 16 is a suggested iterate, which is accepted or rejected as the current iterate, depending on the tests performed in the following steps.

The damping parameter used by Algorithm 2, is calculated in steps 10 and 16. If the step Δx_k calculated by solving a compatible subproblem becomes larger than the trust region radius, then it is truncated in step 18 of Algorithm 2.

Details regarding the initialization and updating of the iteration counter k are omitted for both algorithms.

Algorithm 1 Gradient-based SLP filter algorithm (SLPF)

```
given $x_0$, $\rho_0$, $\beta$, $\gamma$, $\sigma$, $\delta$, $\varepsilon_1$, $\varepsilon_2$ and $k_{\max}$
evaluate $d$ at $x_0$
if $x_0 \in \mathcal{D}$, then evaluate $f$, $c_\mathcal{I}$ and $c_\mathcal{E}$ at $x_0$, end if
while not converged do
    if $x_k \in \mathcal{D}$ then
        try to calculate $\Delta x_k$ by solving LP$(x_k, \rho_k)$
        if subproblem is incompatible then
            if last subproblem was compatible, then $\rho_k \leftarrow \rho_0$, end if
            calculate $\Delta x_k$ by solving RRLP$(x_k, \rho_k)$
        else
            if last iteration was incompatible, then $\rho_k \leftarrow \rho_0$, end if
        end if
    else
        calculate $\Delta x_k$ by solving DRLP$(x_k, \rho_k)$
    end if
    $x_{new} \leftarrow x_k + \Delta x_k$
    evaluate $d$ at $x_{new}$
    if $x_{new} \in \mathcal{D}$ then
        evaluate $f$, $c_\mathcal{I}$ and $c_\mathcal{E}$ at $x_{new}$
        if performing a restoration step then
            calculate $r_k$ using (5.57), (5.58) and (5.59), and update $\rho_k$ using (5.63)
            if $\Delta f_k > 0$, then accept $x_{new}$ as the current iterate, end if
        else if $x_{new}$ is acceptable to the filter then
            calculate $r_k$ using (5.57), (5.54) and (5.55), and update $\rho_k$ using (5.63)
            if $x_{new}$ is h-type iterate then
                add $(h_{new}, f_{new})$ to the filter
                accept $x_{new}$ as the current iterate
            end if
        else
            update $\rho_k$ using (5.65)
        end if
    else
        calculate $r_k$ using (5.57), (5.60) and (5.61), and update $\rho_k$ using (5.63)
        if $\Delta f_k > 0$, then accept $x_{new}$ as the current iterate, end if
    end if
end while
```

Algorithm 2 Gradient-based SQP filter algorithm (SQPF)

```
given $x_0$, $\rho_0$, $\beta$, $\gamma$, $\sigma$, $\delta$, $\varepsilon_1$, $\varepsilon_2$ and $k_{\max}$
evaluate $d$ at $x_0$
if $x_0 \in \mathcal{D}$, then evaluate $f$, $c_{\mathcal{I}}$ and $c_{\mathcal{E}}$ at $x_0$, end if
while not converged do
    if $x_k \in \mathcal{D}$ then
        $\mu_k \leftarrow \|\nabla f_k\|_2/\rho_k$
        try to calculate $\Delta x_k$ by solving QP$(x_k, \mu_k)$
        if subproblem is incompatible then
            if last subproblem was compatible, then $\rho_k \leftarrow \rho_0$, end if
            calculate $\mu_k$ by solving RRQP$_{\mathcal{A}}(x_k, 1)$ and using (5.41)
            calculate $\Delta x_k$ by solving RRQP$(x_k, \mu_k)$
        else
            if last subproblem was incompatible, then $\rho_k \leftarrow \rho_0$, end if
        end if
    else
        calculate $\mu_k$ by solving DRQP$_{\mathcal{A}}(x_k, 1)$ and using (5.41)
        calculate $\Delta x_k$ by solving DRQP$(x_k, \mu_k)$
        if $\|\Delta x_k\|_2 > \rho_k$, then $\Delta x_k \leftarrow \frac{\rho_k}{\|\Delta x_k\|_2}\Delta x_k$, end if
    end if
    $x_{new} \leftarrow x_k + \Delta x_k$
    evaluate $d$ at $x_{new}$
    if $x_{new} \in \mathcal{D}$ then
        evaluate $f$, $c_{\mathcal{I}}$ and $c_{\mathcal{E}}$ at $x_{new}$
        if performing a restoration step then
            calculate $r_k$ using (5.57), (5.58) and (5.59), and update $\rho_k$ using (5.63)
            if $\Delta f_k > 0$, then accept $x_{new}$ as the current iterate, end if
        else if $x_{new}$ is acceptable to the filter then
            calculate $r_k$ using (5.57), (5.54) and (5.55) and update $\rho_k$ using (5.63)
            if $x_{new}$ is h-type iterate then
                add $(h_{new}, f_{new})$ to the filter
                accept $x_{new}$ as the current iterate
            end if
        else
            update $\rho_k$ using (5.65)
        end if
    else
        calculate $r_k$ using (5.57), (5.60) and (5.61), and update $\rho_k$ using (5.63)
        if $\Delta f_k > 0$, then accept $x_{new}$ as the current iterate, end if
    end if
end while
```

5.8 Using approximated gradients

When gradients are not available to the described algorithms, finite difference approximates are used in the first iteration, which are updated in subsequent iterations using the Broyden rank one formula [12].

The accuracy of Broyden approximations deteriorate as iterations progress, which under certain circumstances prevents further progress, even if the current iterate is far from the solution. The accuracy of the Broyden approximations can be restored to using finite difference approximations. However, this approach is not investigated.

Let $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ denote a continuous and smooth vector-valued function. The first order Taylor approximation to g at x_k is:

$$g(x_k + \Delta x_k) \simeq g(x_k) + J_g(x_k)\Delta x_k \tag{5.71}$$

It seems natural to expect that the approximation $B_{k+1} \simeq J_g(x_k + \Delta x_k)$ must satisfy the so-called secant condition:

$$B_{k+1}\Delta x_k = g_{k+1} - g_k, \tag{5.72}$$

where $g_k = g(x_k)$ and $g_{k+1} = g(x_k + \Delta x_k)$. This expression contains m equations and $m\,n$ unknown parameters in B_{k+1}. In order to uniquely determine B_{k+1}, Broyden [12] propose that, in addition to (5.72), to require that the properties of B_{k+1} may not change in directions that are orthogonal to Δx_k, when compared to the approximation $B_k \simeq J_g(x_k)$, which is assumed to be known. These requirements are stated as

$$B_{k+1}\Delta\hat{x} = B_k\Delta\hat{x} \;\text{ for all }\; \Delta\hat{x} \perp \Delta x_k \tag{5.73}$$

Broyden [12] shows that these requirements are satisfied by the following updating formula:

$$B_{k+1} = B_k + \frac{1}{\Delta x_k^\top \Delta x_k}\left(g_{k+1} - g_k - B_k\Delta x_k\right)\Delta x_k^\top \tag{5.74}$$

Dennis and Schnabel [17] show that the formula (5.74) updates B_k from iteration k to $k+1$, in such a way that the change to B_k is minimal in the Frobenius sense, that is:

$$\begin{array}{rlll} B_{k+1} & = & \text{argmin} & \|B - B_k\|_F \\ & & \text{subject to} & B\,\Delta x_k = g_{k+1} - g_k \\ & & \text{with respect to} & B \in \mathbb{R}^{m\times n} \end{array} \tag{5.75}$$

where $\|\cdot\|_F$ is the Frobenius norm:

$$\|B\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} B_{i,j}^2} \tag{5.76}$$

Powell [57] shows that the sequence of Broyden updated approximations B_k, $k = 0, 1, \ldots$ converges to $J_f(x^*)$, independent of the choice of B_0, under the following conditions:

1. The sequence of iterates $x_0, x_1, \dots$ converge to x^*.
2. The last n steps, $\Delta x_{k-n+1}, \Delta x_{k-n+2}, \dots, \Delta x_k$, are linearly independent, for all $k \geq n$, where the steps are given by (2.6).

This result implies that the accuracy of the Broyden updated approximations may deteriorate if the steps calculated by the optimization algorithm are not linearly independent. Methods for enabling the algorithms to calculate steps that are linearly independent are not investigated, but this topic is suggested for further research in the field of building optimization.

5.9 Summary of the gradient-free algorithm

The steps required by the gradient-free SQP filter algorithm are summarized in Algorithm 3. It is similar to Algorithm 2, except that it is extended with steps for initializing and updating the Broyden updated approximations to the first partial derivatives.

The algorithm maintains the following Broyden approximations:

$$B_{f,k} \simeq \nabla f_k \tag{5.77}$$
$$B_{c_{\mathcal{I}},k} \simeq J_{c_{\mathcal{I}},k} \tag{5.78}$$
$$B_{c_{\mathcal{E}},k} \simeq J_{c_{\mathcal{E}},k} \tag{5.79}$$
$$B_{d,k} \simeq J_{d,k} \tag{5.80}$$

These approximations are initialized by calculating forward difference approximations to the first partial derivatives of the functions f, $c_{\mathcal{I}}$, $c_{\mathcal{E}}$ and d. The forward difference approximations for a vector-valued continuous and smooth function $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ calculated at the point x are given by:

$$\frac{\partial g_i}{\partial x_j}(x) \simeq \frac{1}{\alpha}\left(g_i(x + e_j\alpha) - g_i(x)\right) \quad \text{for} \quad i = 1, \dots, m \quad \text{and} \quad j = 1, \dots, n, \tag{5.81}$$

where α is the size of the step length. The following value is used

$$\alpha = \sqrt{\eta}\max\{x_j, 1\}\text{sgn}(x_j), \tag{5.82}$$

where η is a user-provided value, and where sgn is the (slightly modified) signum operator:

$$\text{sgn}(x) = \begin{cases} -1 & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases} \tag{5.83}$$

The expression (5.82) is similar to the value proposed by Dennis and Schnabel [17], Algorithm A5.4.1.

The approximation $B_{d,k}$ is initialized in step 2, and updated in step 22. The approximations $B_{f,k}$, $B_{c_{\mathcal{I}},k}$ and $B_{c_{\mathcal{E}},k}$ are initialized in step 3, if $x_0 \in \mathcal{D}$, otherwise they are initialized in step 25 the first time an iterate $x_k \in \mathcal{D}$ is found, and updated in the same step of the algorithm in subsequent iterations.

Algorithm 3 Gradient-free SQP filter algorithm (GFSQPF)

```
given $x_0$, $\rho_0$, $\beta$, $\gamma$, $\sigma$, $\delta$, $\varepsilon_1$, $\varepsilon_2$, $k_{\max}$ and $\eta$
evaluate $d$ at $x_0$ and initialize $B_{d,k}$
if $x_0 \in \mathcal{D}$, then evaluate $f$, $c_\mathcal{I}$ and $c_\mathcal{E}$ at $x_0$, initialize $B_{f,k}$, $B_{c_\mathcal{I},k}$ and $B_{c_\mathcal{E},k}$, end if
while not converged do
    if $x_k \in \mathcal{D}$ then
        $\mu_k \leftarrow \|\nabla f_k\|_2/\rho_k$
        try to calculate $\Delta x_k$ by solving QP$(x_k, \mu_k)$
        if subproblem is incompatible then
            if last subproblem was compatible, then $\rho_k \leftarrow \rho_0$, end if
            calculate $\mu_k$ by solving RRQP$_\mathcal{A}(x_k, 1)$ and using (5.41)
            calculate $\Delta x_k$ by solving RRQP$(x_k, \mu_k)$
        else
            if last subproblem was incompatible, then $\rho_k \leftarrow \rho_0$, end if
        end if
    else
        calculate $\mu_k$ by solving DRQP$_\mathcal{A}(x_k, 1)$ and using (5.41)
        calculate $\Delta x_k$ by solving DRQP$(x_k, \mu_k)$
        if $\|\Delta x_k\|_2 > \rho_k$, then $\Delta x_k \leftarrow \frac{\rho_k}{\|\Delta x_k\|_2}\Delta x_k$, end if
    end if
    $x_{new} \leftarrow x_k + \Delta x_k$
    evaluate $d$ at $x_{new}$
    update $B_{d,k}$
    if $x_{new} \in \mathcal{D}$ then
        evaluate $f$, $c_\mathcal{I}$ and $c_\mathcal{E}$ at $x_{new}$
        initialize or update $B_{f,k}$, $B_{c_\mathcal{I},k}$ and $B_{c_\mathcal{E},k}$
        if performing a restoration step then
            calculate $r_k$ using (5.57), (5.58) and (5.59), and update $\rho_k$ using (5.63)
            if $\Delta f_k > 0$, then accept $x_{new}$ as the current iterate, end if
        else if $x_{new}$ is acceptable to the filter then
            calculate $r_k$ using (5.57), (5.54) and (5.55), and update $\rho_k$ using (5.63)
            if $x_{new}$ is h-type iterate then
                add $(h_{new}, f_{new})$ to the filter
                accept $x_{new}$ as the current iterate
            end if
        else
            update $\rho_k$ using (5.65)
        end if
    else
        calculate $r_k$ using (5.57), (5.60) and (5.61), and update $\rho_k$ using (5.63)
        if $\Delta f_k > 0$, then accept $x_{new}$ as the current iterate, end if
    end if
end while
```

5.10 Final remarks

The aim of this chapter is to describe an algorithm that is able to solve optimization problems of the form (2.4), without requiring information about the first partial derivatives of the functions that define the problem.

The result of this effort is the GFSQPF algorithm, which is based on the filter SLP algorithm proposed by Fletcher [23]. The GFSQPF algorithm uses a quadratic damping term instead of box constraints, it uses Broyden updated approximations to the first partial derivatives, and it uses domain restoration steps in order to address domain constraints.

In order to investigate how much the performance is influenced by the quadratic damping term, and the Broyden approximations, two additional algorithms are described: The SLPF algorithm, which is a slightly modified version of Fletchers algorithm, and the SQPF algorithm, which uses a quadratic damping term, but requires gradient information.

There are still a number of unresolved issues regarding the algorithms:

1. Intuitively, it seems reasonable that calculating the damping parameter μ_k using the approach described in Section 5.3, provides a step that is at most ρ_k, but a formal proof of this statement is not provided.

2. Convergence theorems for the algorithms are not provided.

3. It is possible that the numerical accuracy of the Broyden updated gradient approximations deteriorate if the steps calculated by the algorithm are linearly dependent. Methods for improving the Broyden approximations are not investigated.

Chapter 6

Evaluating the building optimization method

This chapter provides an evaluation of the building optimization method, conducted through numerical experiments. First, the convergence properties of the gradient-free SQP filter (GFSQPF) algorithm are investigated. Then, the building optimization method, which involves the GFSQPF algorithm, is evaluated through case studies.

The building optimization method and the GFSQPF algorithm are implemented in Matlab [44], and the numerical experiments are conducted on a Windows$^{\text{TM}}$ PC with an AMD Athlon$^{\text{TM}}$ 64 (2 GHz) processor.

6.1 The gradient-free SQP filter algorithm

The following aspects of the GFSQPF are investigated:

1. How much the performance is influenced by using a quadratic damping term instead of box constraints.

2. How much the performance is influenced by using Broyden updated gradient approximations instead of exact gradient information.

3. To determine if the performance of the GFSQPF algorithm is comparable with the gradient-based algorithm SLPF, which uses box constraints.

4. To determine if the domain restoration steps are useful for addressing domain constraints.

The example described in Section 6.1.1 is intended for investigating the performance of the algorithms on a constrained optimization problem without domain constraints, and is also intended for illustrating various aspects of the algorithms, such as the regular restoration subproblems.

The example described in Section 6.1.2 is intended for investigating the performance of the SQPF and GFSQPF algorithms on optimization problems with domain constraints.

Section 6.1.3 provides statistical results obtained by applying the algorithms to the test problems described in Appendix C.

6.1.1 Example 1: A constrained optimization problem

The test problem TP$_2$ described in Appendix C is used for illustrating the difference between the SLPF and SQPF algorithms. The following starting points are considered:

$$\begin{aligned} x_S^{(1)} &= [3.25, -1]^\top \\ x_S^{(2)} &= [1.75, -1]^\top \\ x_S^{(3)} &= [0.25, -1]^\top , \end{aligned} \tag{6.1}$$

which are shown together with TP$_2$ in Figure 6.1.

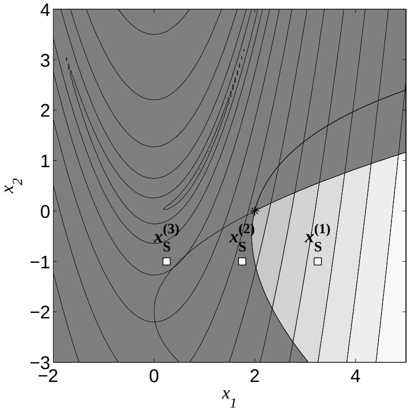


Figure 6.1: The test problem TP$_2$.

The trust region subproblems

In Figure 6.2 are shown the subproblems LP($x_S^{(1)}$, 1) and QP($x_S^{(1)}$, 1). In this case, the steps are solely determined by the trust region radius. The figure indicates that there is a possibility that box constraint restrict the number of possible directions for the solution to (5.5), since it is located in a corner of the trust region. This may cause the algorithm to favor some directions more than others.

Furthermore, small changes in the gradient $\nabla f(x)$ may in some situations cause the solution to (5.5) to change abruptly from one corner to another, which can result in an oscillating behavior.

In Figure 6.3 are shown the subproblems LP($x_S^{(2)}, 1$) and QP($x_S^{(2)}, 1$). In this case, the steps are influenced by the approximations to the inequality constraints.

In Figure 6.4 are shown the subproblems LP($x_S^{(3)}, 1$) and QP($x_S^{(3)}, 1$). In this case, the LP subproblem has no solution due to conflicts between the approximations to the inequality constraints and the box constraints. The problem is therefore referred to as *incompatible*. The purpose of the regular restoration steps described in Section 5.3 is to take a step towards the feasible region in order to avoid compatibility problems.

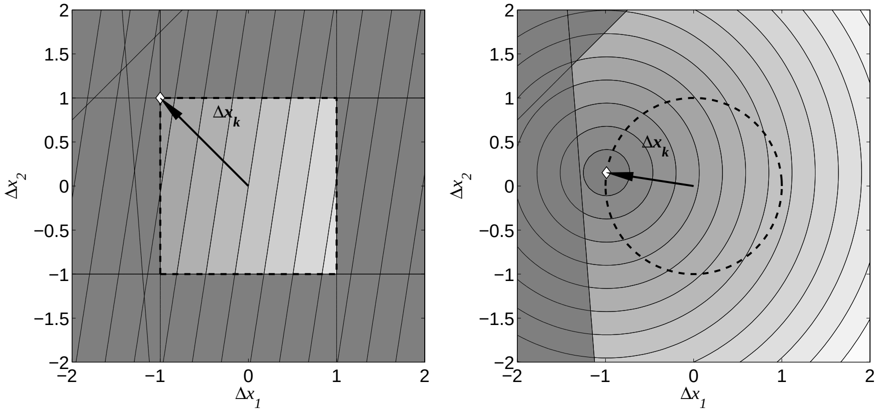


Figure 6.2: Left: The subproblem LP($x_S^{(1)}, 1$). Right: The subproblem QP($x_S^{(1)}, 1$). The dashed lines indicate the trust regions. The "diamonds" indicate the solutions to the subproblems.

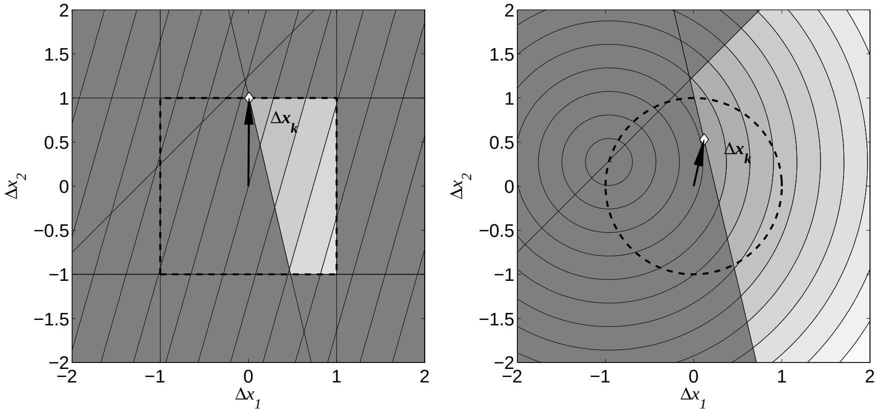


Figure 6.3: Left: The subproblem LP($x_S^{(2)}, 1$). Right: The subproblem QP($x_S^{(2)}, 1$).

The QP problem is still compatible; however, the calculated step is larger than the trust region radius. In this situation, the step is simply truncated to the trust region radius.

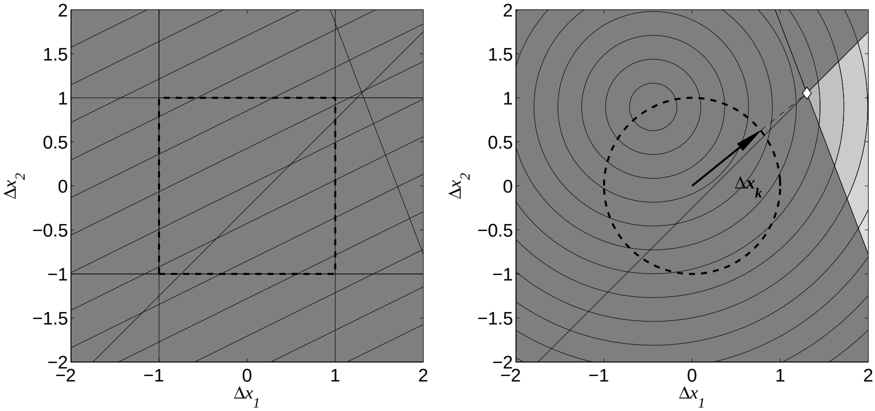


Figure 6.4: Left: The subproblem $LP(x_S^{(3)}, 1)$. Right: The subproblem $QP(x_S^{(3)}, 1)$. The LP problem has no solution, whereas the QP problem has a solution outside the trust region, which is truncated to the boundary of the trust region by the algorithm.

There still exist the possibility that there are conflicts between the approximations to the constraint functions that render the QP subproblem incompatible. It is therefore still necessary to include restoration steps in the algorithm.

Regular restoration subproblems

In Figure 6.5 is shown the regular restoration problem (5.9) for TP$_2$, which is used for estimating a direction towards the feasible region. Estimating one or more steps for this problem forces the iterates towards the feasible region.

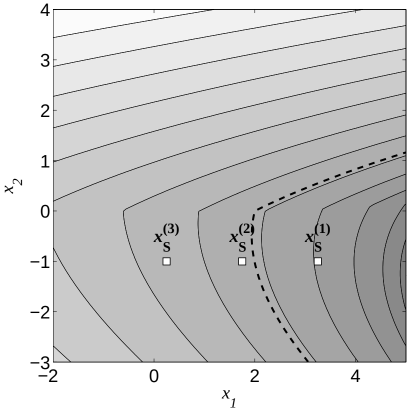


Figure 6.5: The regular restoration problem (5.9) for TP$_2$. The dashed line indicate the boundary of the feasible region.

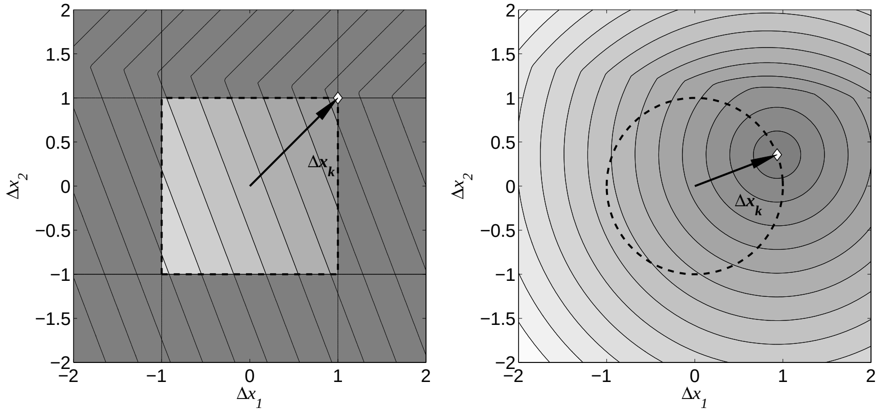


Figure 6.6: Left: The regular restoration subproblem RRLP($x_S^{(3)}$, 1). Right: The regular restoration subproblem RRQP($x_S^{(3)}$, 1).

In Figure 6.6 is shown the two types of regular restoration subproblems for the point $x_S^{(3)}$. The RRQP problem is not necessary in this case, since there are no compatibility problems for the QP subproblem, but is shown for comparison with the RRLP subproblem.

The figure again indicates that the box constraints cause the number of possible directions that the algorithm can suggest to be limited, since the solution again is found in a corner of the trust region.

Convergence properties for the gradient-based algorithms

In order to assess the convergence properties, the algorithms are initialized with the values shown in Table 6.1. The algorithms are allowed to continue until rounding errors are dominating the results. This is achieved by using $\varepsilon_1 = 0$ and $\varepsilon_2 = 0$, which disables the stopping criteria for the relative changes in the iterate and the objective function value.

Parameter	Value
ρ_0	$0.01 \cdot (\|\hat{x}\|_\infty + 1)$
β	0.98
γ	0.02
σ	0.04
δ	0.02
ε_1	0
ε_2	0
k_{max}	1000

Table 6.1: Parameters used for initializing the algorithms.

The parameter $\hat{x}$ is the mid-point of the region of interest, i.e. the mid-point of the topmost plots shown in Figures 6.7, 6.8 and 6.9. The following values are used:

$$\begin{aligned} \hat{x} &= [2.25, -0.50]^\top && \text{for } x_S^{(1)} \\ \hat{x} &= [1.125, -0.125]^\top && \text{for } x_S^{(2)} \text{ and } x_S^{(3)} \end{aligned} \tag{6.2}$$

The Figures 6.7, 6.8 and 6.9 show the iterates generated by the SLPF and SQPF algorithms, and the relative errors $\|e_k\|_2/\|x^*\|_2$ plotted against the iteration counter k.

The figures indicate that the SQPF algorithm has linear convergence. This is further substantiated by the results presented in Table 6.2, where $\|e_{k+1}\|/\|e_k\|$ is shown for the three starting points.

The restricted step directions caused by the box constraints can be seen quite clearly from Figures 6.7 and 6.9. The SLPF algorithm shows an oscillating behavior for the starting points $x_S^{(2)}$ and $x_S^{(3)}$, which is not observed for the SQPF algorithm. Notice that the SQPF algorithm in this case does not require restoration steps.

k	$x_S^{(1)}$	$x_S^{(2)}$	$x_S^{(3)}$
1	9.82454798e-1	9.89920942e-1	9.89564080e-1
2	9.70457221e-1	9.82828819e-1	9.82428323e-1
3	9.49959849e-1	9.69599756e-1	9.70202761e-1
4	9.14612422e-1	9.38387735e-1	9.48841317e-1
5	8.54224034e-1	9.19131069e-1	9.10211445e-1
6	7.69889831e-1	9.26042569e-1	8.35909090e-1
7	7.38133418e-1	9.24799782e-1	6.74971365e-1
8	4.54654175e-1	9.23548561e-1	2.20011405e-1
9	3.80158769e-1	9.22400834e-1	2.38074213e-1
10	3.77198994e-1	9.21357459e-1	3.83284389e-1
11	3.83497447e-1	9.20415837e-1	3.95211447e-1
12	3.85489519e-1	9.19571596e-1	3.95942728e-1
13	3.86099595e-1	9.18819042e-1	3.96162287e-1
14	3.86315788e-1	9.18151593e-1	3.96246128e-1
15	3.86396776e-1	9.17562173e-1	3.96278914e-1
16	3.86427709e-1	9.17043533e-1	3.96291839e-1
17	3.86439609e-1	9.16588516e-1	3.96296950e-1
18	3.86444200e-1	9.16190248e-1	3.96298975e-1
19	3.86445973e-1	9.15842268e-1	3.96299776e-1
20	3.86446658e-1	9.15538611e-1	3.96300094e-1

Table 6.2: The ratio $\|e_{k+1}\|/\|e_k\|$ for $k = 1, \ldots, 20$, for the three starting points, obtained with the SQPF algorithm.

Convergence properties for the gradient-free algorithm

The GFSQPF is initialized using the values shown in Table 6.3, where the values for $\hat{x}$ are given by (6.2). The results obtained with the SQPF algorithm described in Section 6.1.1

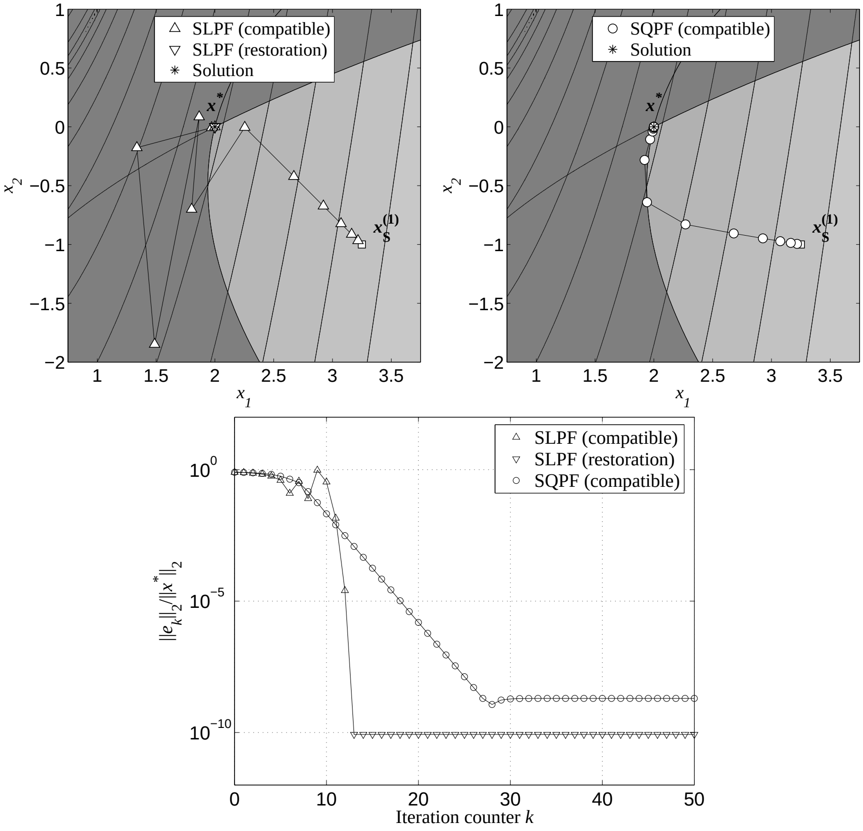


Figure 6.7: Top: The iterates x_k, $k = 1, \dots, 30$ generated by the SLPF algorithm (left) and the SQPF algorithm (right) when started from $x_S^{(1)}$. Bottom: The relative errors $\|e_k\|/\|x^*\|$ for the SLPF and SQPF algorithms.

are used for comparison.

In Figures 6.10, 6.11 and 6.12 are shown the iterates generated by the GFSQPF algorithm for the three starting points, as well as the relative errors for both algorithms.

In Figures 6.10 are the results obtained for the starting point $x^{(1)}$ shown. The GFSQPF algorithm seems to produce some steps of poor quality with regular intervals. The exact reason for this phenomenon is not yet known, but a possible explanation is that the accuracy of the Broyden approximations deteriorates due to linearly dependent steps. This phenomenon is not observed for the SQPF algorithm.

When started from $x_S^{(2)}$, the performance of the GFSQPF algorithm is almost identical to that of the SQPF algorithm, which can be seen from Figure 6.11. When started from $x_S^{(3)}$, the GFSQPF algorithm performs better than the gradient-based algorithm, which can be seen from Figure 6.12. This is, however, uncommon.

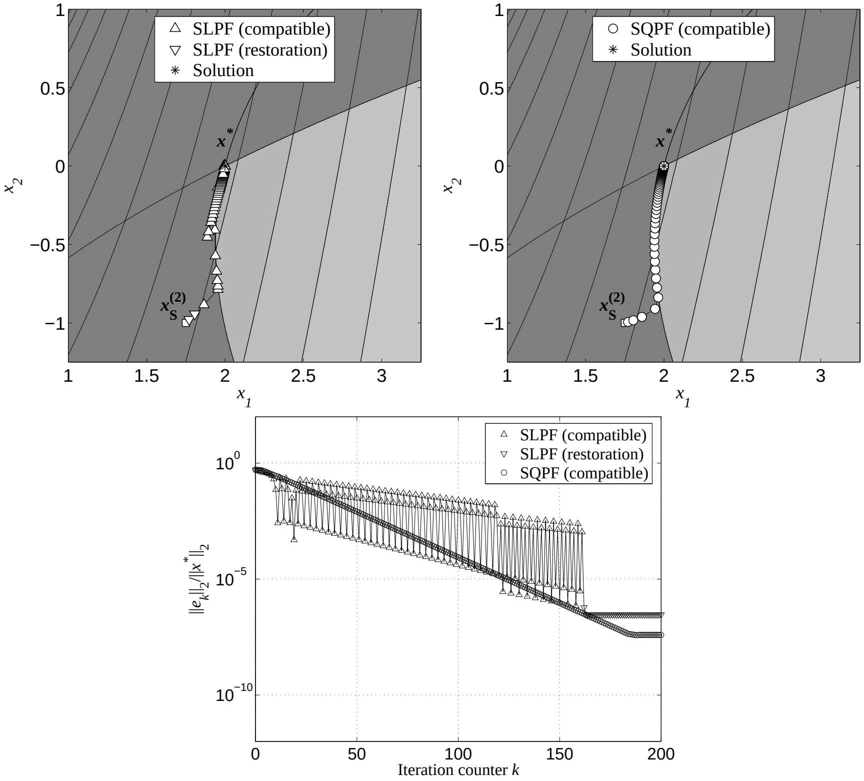


Figure 6.8: Top: The iterates x_k, $k = 1, \ldots, 100$ generated by the SLPF algorithm (left) and the SQPF algorithm (right) when started from $x_S^{(2)}$. Bottom: The relative errors $\|e_k\|/\|x^*\|$ for the SLPF and SQPF algorithms.

Parameter	Value
ρ_0	$0.01{\cdot}(\|\hat{x}\|_\infty + 1)$
β	0.98
γ	0.02
σ	0.04
δ	0.02
ε_1	0
ε_2	0
k_{max}	1000
η	0.02

Table 6.3: Parameters used for initializing the GFSQPF algorithm.

The results shown so far indicate that the algorithms has a tendency to first seek towards

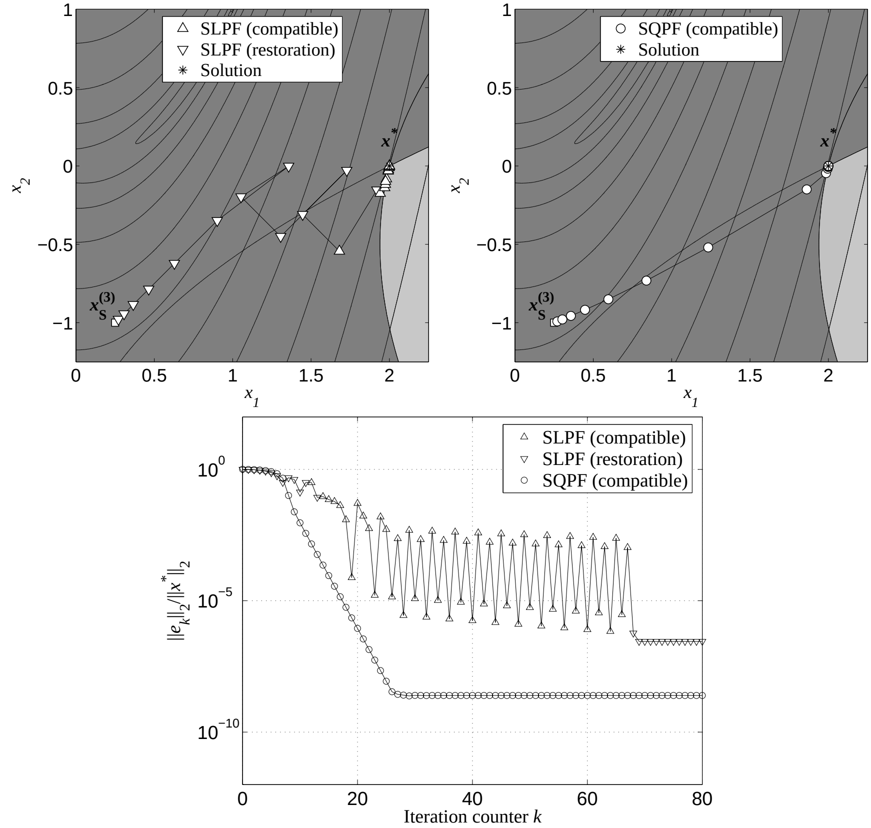


Figure 6.9: Top: The iterates x_k, $k = 1, \ldots, 30$ generated by the SLPF algorithm (left) and the SQPF algorithm (right) when started from $x_S^{(3)}$. Bottom: The relative errors $\|e_k\|/\|x^*\|$ for the SLPF and SQPF algorithms.

the nearest point on the boundary of the feasible region, and then converge to the solution by making steps on, or close to, the boundary of the feasible region. This has implications for the performance of the algorithms when applied to optimization problems with domain constraints, which is investigated in the next section.

6.1.2 Example 2: Optimization problems with domain constraints

The test problems TP_9 and TP_{11} are used for investigating the performance of the SQPF and GFSQPF algorithms when applied to optimization problems with domain constraints. In the following, results from test runs performed with $x_S^{(2)}$ as starting point are described.

In Figure 6.13 is shown the iterates generates by the algorithms when started from $x_S^{(2)}$ for the test problem TP_9. Neither the SQPF nor the GFSQPF algorithm seem to converge

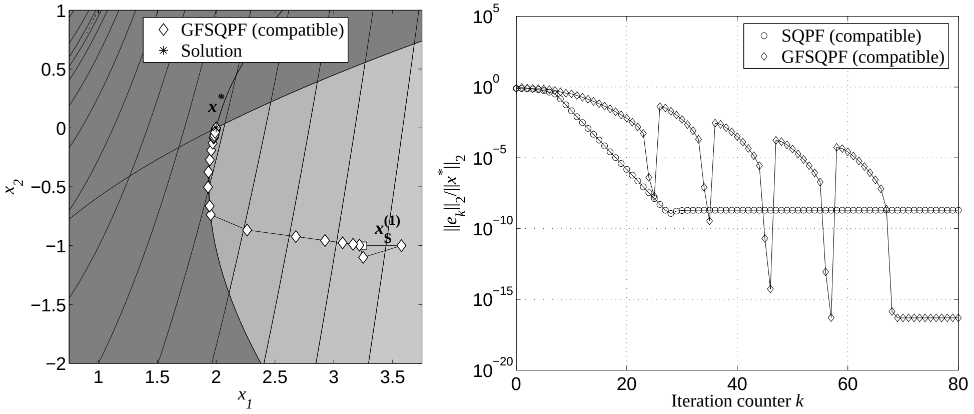


Figure 6.10: Left: The iterates x_k, $k = 1, \ldots, 30$ generated by the GFSQPF algorithm when started from $x_S^{(1)}$. Right: The relative errors $\|e_k\|/\|x^*\|$ for the SQPF and GFSQPF algorithms.

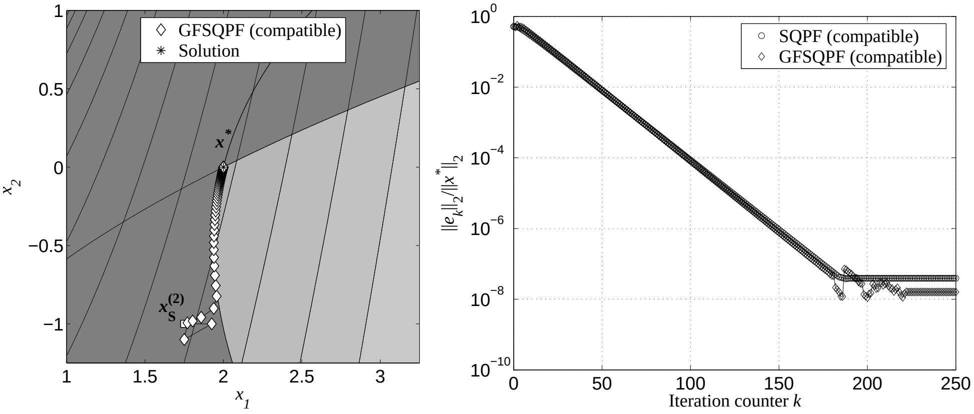


Figure 6.11: Left: The iterates x_k, $k = 1, \ldots, 100$ generated by the GFSQPF algorithm when started from $x_S^{(2)}$. Right: The relative errors $\|e_k\|/\|x^*\|$ for the SQPF and GFSQPF algorithms.

to the solution.

As discussed in the previous section, the algorithm will often converge to the solution by making steps on, or close to the boundary of the feasible region for the problem. In this case, the boundary of the feasible region is in some regions established by a domain constraint function. This means that the algorithm is very likely to end in a situation where it alternates between domain restoration steps calculated by solving the DRQP(x_k, μ_k) problem (5.48), and steps calculated by solving the subproblem QP(x_k, μ_k) (5.7).

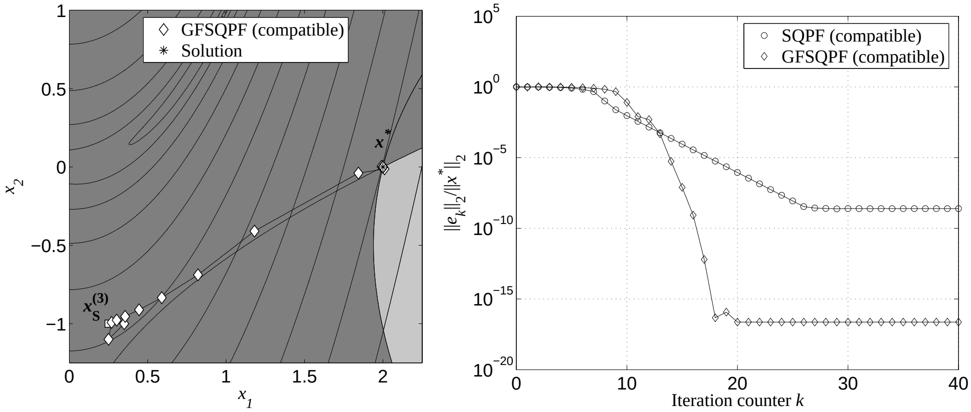


Figure 6.12: Left: The iterates x_k, $k = 1, \ldots, 30$ generated by the GFSQPF algorithm when started from $x_S^{(3)}$. Right: The relative errors $\|e_k\|/\|x^*\|$ for the SQPF and GFSQPF algorithms.

This unfortunate tendency seems to slow down or prevent convergence in situations where the boundary of the feasible region is established by one or more domain constraints. In Figure 6.14 is shown the iterates generates by the algorithms when started from $x_S^{(2)}$ for the test problem TP_{11}. The feasible region for this problem is almost entirely established by the inequality constraints, but the problem has one active domain constraint.

The initial iterates seem to convergence faster, probably because the algorithm in this case does not alternate between the two types of steps. After approximately 20 iterations, the active domain constraint seems to cause the algorithms to alternate between the two types of steps, which has a profound influence on the final convergence rate.

Exactly how to ensure convergence when the considered optimization problem has active domain constraints, or when (parts of) the boundary of the feasible region is established by the domain constraints, is yet unknown.

Developing a gradient-free barrier method for constrained optimization problems is a possible topic for further research. Barrier methods are described by Conn et.al. [14], Chapter 13.

6.1.3 Numerical experiments

This section concerns the numerical experiments performed on the test problems described in Appendix C. The purpose of the experiments is:

1. to determine the benefit, if any, of using the quadratic damping term, and
2. to determine how much the performance of the algorithm is influenced by using Broyden updated gradient approximations instead of exact gradients.

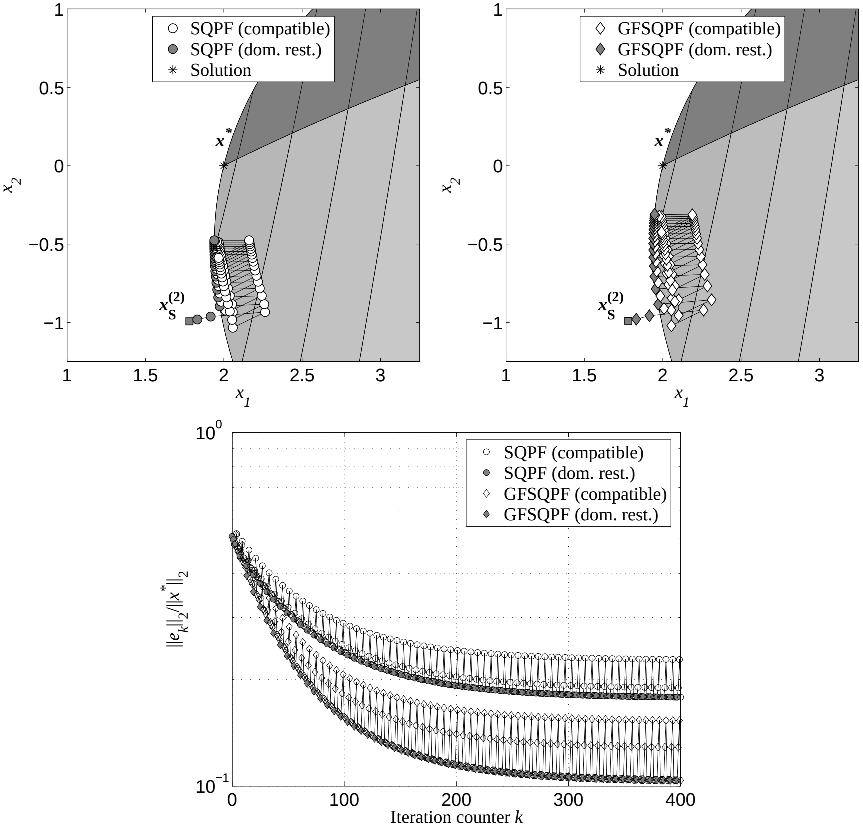


Figure 6.13: Top: The iterates x_k, $k = 1, \ldots, 100$ generated by the SQPF algorithm (left) and the GFSQPF algorithms (right) when started from $x_S^{(2)}$ for the test problem TP_9. Bottom: The relative errors $\|e_k\|/\|x^*\|$ for the SQPF and GFSQPF algorithms.

The benefit of using the quadratic damping term is assessed by comparing the SLPF and SQPF algorithms. The influence of using approximated gradients is assessed by comparing the SQPF and GFSQPF algorithms. Finally, the SLPF and GFSQPF algorithms are compared in order to assess the total influence on the performance caused by introducing quadratic damping terms as well as Broyden updated gradient approximations.

The numerical experiments consist of starting the two algorithms that are compared from 30 different starting points for the 15 test problems, a total of 450 test runs for each pair of algorithms. The algorithms are compared by observing the number of iterations needed to provide a solution estimate with a relative error below a given tolerance level. It is not possible to provide a solution estimate satisfying the tolerance level for all test runs; therefore the number of successful test runs is also observed.

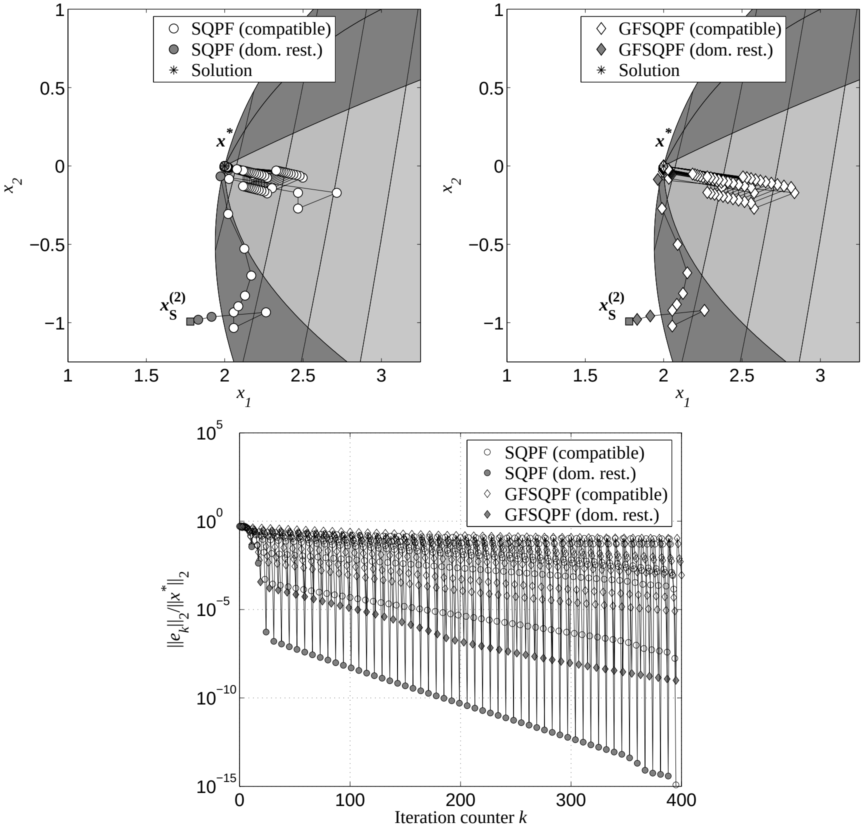


Figure 6.14: Top: The iterates x_k, $k = 1, \ldots, 100$ generated by the SQPF algorithm (left) and the GFSQPF algorithms (right) when started from $x_S^{(2)}$ for the test problem TP$_{11}$. Bottom: The relative errors $\|e_k\|/\|x^*\|$ for the SQPF and GFSQPF algorithms.

The starting points are arranged on three concentric circles with x^* as the centre, and with diameters 1.75, 3.5 and 5.25, respectively. 5, 10 and 15 starting points are arranged on the three circles. In Figure 6.15 is shown the test problem TP$_1$, together with the 30 starting points.

The requirement for a successful test run is that it provides a solution estimate x_k with a relative error

$$\frac{\|e_k\|_2}{\|x^*\|_2} < \varepsilon, \tag{6.3}$$

where

$$\varepsilon = 10^{-6} \quad \text{and} \quad k \leq k_{max}. \tag{6.4}$$

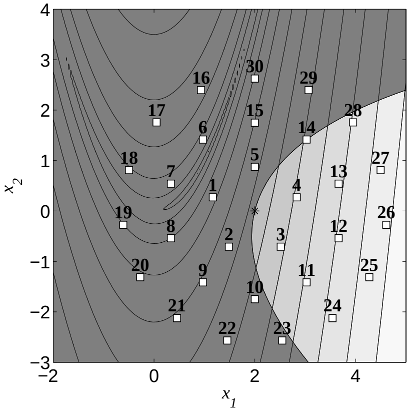


Figure 6.15: Test problem TP_1 and the 30 starting points used for conducting the numerical experiments.

The algorithms are initialized using the values in Table 6.4, where

$$\hat{x} = [1.5, 0.5]^\top . \tag{6.5}$$

Parameter	Value
ρ_0	$0.01 \cdot (\|\hat{x}\|_\infty + 1)$
β	0.98
γ	0.02
σ	0.04
δ	0.02
ε_1	0
ε_2	0
k_{max}	1000
η	0.02

Table 6.4: Parameters used for initializing the three algorithms.

The results of the experiments are provided in Tables 6.5, 6.6 and 6.7. Each row contains results for one test problem. Columns 2 and 3 show the number of successful experiments for the two algorithms that are compared. Column 4 shows the number of starting points where both algorithms succeeded.

Columns 5 and 6 show the average number of iterations needed to provide a solution estimate satisfying the tolerance (6.3). Only the starting points where both algorithms succeeded contribute to the average.

Column 7 shows the improvement, i.e. the ratio between the required number of iterations for the two algorithms.

	No. of successful tests			No. of average iterations		
Problem	SLPF	SQPF	SLPF+ SQPF	SLPF	SQPF	Improve- ment
TP_1	1	30	1	537.00	34.00	15.79
TP_2	28	29	27	46.19	23.56	1.96
TP_3	30	30	30	13.97	14.83	0.94
TP_4	26	30	26	555.88	30.15	18.43
TP_5	28	25	23	29.09	11.52	2.52
TP_6	27	22	20	22.15	11.45	1.93
TP_7	27	21	19	23.21	11.37	2.04
TP_8	0	0	0	-	-	-
TP_9	30	0	0	-	-	-
TP_{10}	0	0	0	-	-	-
TP_{11}	0	30	0	-	-	-
TP_{12}	1	30	1	20.00	238.00	0.08
TP_{13}	1	30	1	20.00	238.00	0.08
TP_{14}	30	30	30	21.53	13.83	1.56
TP_{15}	30	30	30	21.40	13.83	1.55
Total	259	337	208	93.93	18.82	4.99

Table 6.5: Results obtained by comparing the SLPF and SQPF algorithms.

The results presented in Table 6.5 indicate that using a quadratic damping term significantly reduces the number of iterations needed for providing an acceptable solution. In this case, the number of required iterations is almost five times less than when using box constraints. Another advantage is that using a quadratic damping term seems to provide a more stable algorithm, since it succeeded for 337 test runs (74%), whereas using box constraints was successful for 259 test runs (57%).

The results presented in Table 6.6 indicate that when using Broyden updated gradient approximations, the number of required iterations is increased with approximately 30%, on average.

Finally, the results presented in Table 6.7 indicate that the GFSQPF algorithm does not perform worse than the SLPF algorithm. In fact, in this case it requires less than half the number of iterations required by the SLPF algorithm.

The GFSQPF algorithm also seems to be more numerically stable, since it succeeded for 376 test runs (83%), compared to 259 test runs (57%) for the SLPF algorithm.

6.2 Case studies

The performance of the building optimization method is evaluated by applying it to a design decision problem regarding a 3 storey, 2000 m^2 office building, with a main axis

	No. of successful tests			No. of average iterations		
Problem	SQPF	GFSQPF	SQPF+ GFSQPF	SQPF	GFSQPF	Improvement
TP_1	30	28	28	31.75	60.89	0.52
TP_2	29	29	29	22.76	23.55	0.97
TP_3	30	29	29	12.03	16.24	0.74
TP_4	30	28	28	32.36	62.68	0.52
TP_5	25	24	24	11.54	15.71	0.73
TP_6	22	22	22	11.32	15.55	0.73
TP_7	21	20	20	11.20	15.65	0.72
TP_8	0	0	0	-	-	-
TP_9	0	26	0	-	-	-
TP_{10}	0	20	0	-	-	-
TP_{11}	30	30	30	59.37	191.10	0.31
TP_{12}	30	30	30	305.73	404.20	0.76
TP_{13}	30	30	30	305.73	240.97	1.27
TP_{14}	30	30	30	15.07	21.67	0.70
TP_{15}	30	30	30	15.00	21.00	0.71
Total	337	376	330	74.49	97.01	0.77

Table 6.6: Results obtained by comparing the SQPF and GFSQPF algorithms.

oriented in the east-west direction. This means that the building has a north-facing and a south-facing façade. The constant parameters needed for calculating the performance of the building are provided in Appendix D.

Two test runs are described in the following, one run intended for finding a design with minimum construction cost, and one run intended for finding a design with minimum energy consumption.

The problem (3.8) is solved using the GFSQPF algorithm described in Chapter 5. The algorithm is initialized using the parameters in Table 6.8 for all test runs, where x_0 is the initial design decisions.

The algorithm is allowed to make 300 performance calculations. Each performance calculation takes on average 11.345 seconds, which means that the total time consumption is limited to approximately 57 minutes.

6.2.1 Design decisions with minimum construction cost

The first case study concerns finding design decisions that provide the lowest construction cost, but at the same time satisfy the Danish building regulations. This means that the least restrictive energy frame EF_3 must be satisfied ($EF_3 \geq 0$), the heat loss through the building envelope must be below $6\,\mathrm{W/m^2}$ of the façade ($BE \geq 0$), and the requirements

	No. of successful tests			No. of average iterations		
Problem	SLPF	GFSQPF	SLPF+ GFSQPF	SLPF	GFSQPF	Improve-ment
TP_1	1	28	1	537.00	34.00	15.79
TP_2	28	29	27	46.19	23.81	1.94
TP_3	30	29	29	14.03	16.24	0.86
TP_4	26	28	24	572.67	65.08	8.80
TP_5	28	24	22	25.36	15.64	1.62
TP_6	27	22	20	22.15	15.65	1.42
TP_7	27	20	18	23.56	15.78	1.49
TP_8	0	0	0	-	-	-
TP_9	30	26	26	209.23	130.31	1.61
TP_{10}	0	20	0	-	-	-
TP_{11}	0	30	0	-	-	-
TP_{12}	1	30	1	20.00	522.00	0.04
TP_{13}	1	30	1	20.00	22.00	0.91
TP_{14}	30	30	30	21.53	21.67	0.99
TP_{15}	30	30	30	21.40	21.00	1.02
Total	259	376	229	105.36	38.70	2.72

Table 6.7: Results obtained by comparing the SLPF and SQPF algorithms.

Parameter	Value	Description
ρ_0	$0.01 \cdot (\|x_0\|_\infty + 1)$	Initial trust region radius
β	0.98	Parameter used for establishing the filter envelope
γ	0.02	Do.
σ	0.04	Parameter used for distinguishing between f- and h-type iterations
δ	0.02	Do.
ε_1	0	Tolerance level for the decision variables
ε_2	0	Tolerance level for the objective function value
k_{max}	300	Maximum allowed number of function evaluations
η	10^{-6}	Perturbation size used when calculating finite difference approximations

Table 6.8: Parameters used for initializing the GFSQPF algorithm.

to the U-values of the components used in the building envelope must be satisfied. These requirements are described in Section 4.1.5.

Furthermore, in order to ensure a satisfactory indoor thermal environment, the annual number of hours where overheating occurs must be below 100 for both thermal zones ($OH^{(1)} \leq 100$ and $OH^{(2)} \leq 100$). A satisfactory level of natural light is ensured by requiring that the ratio between the depth of the room and the window height is below 4 for both zones ($DH^{(1)} \leq 4$ and $DH^{(2)} \leq 4$). The only requirement to the decision variables is that the number of floors must be 3.

In Table 6.9 is shown the requirements to decision variables and performance measures. The initial values used for initializing the GFSQPF algorithm, and the values returned by the algorithm are also shown.

In order to reduce the cost of constructing the building, the algorithm suggests a more compact design, with a width to length ratio of approximately 0.7. This has a number of consequences. First of all, the area of the building envelope is reduced, which reduces the construction cost. Secondly, it has a negative impact on the room depth to window height ratio. Using a quadratic building is therefore not possible, since this will prevent the requirements to the use of natural light to be fulfilled. This means that there is a limit for how compact the building can be.

The algorithm furthermore suggests increasing the window areas as much as possible. It uses the fact that the windows provided with the window database cost less per m^2 than the external wall construction. The parameters $\sigma^{(1)}$ and $\sigma^{(2)}$ are restricted by the domain constraints. If the domain constraints were less strict, the algorithm may have suggested even larger window areas.

Using the weight factors $\alpha_1^{(i)} = 1$ and $\alpha_2^{(i)} = 0$ means that the first window in the database (the double-glazed window) is selected for façade i, where the weight factors $\alpha_1^{(i)} = 0$ and $\alpha_2^{(i)} = 1$ means that the second window (the triple-glazed window) is selected.

Using the weight factors $\alpha_1^{(i)} = 0.5$ and $\alpha_2^{(i)} = 0.5$ means that average window properties are used as input to the energy performance calculation method.

The weights returned by the optimization means that it suggests using the double-glazed windows, which are the cheapest ones.

Notice that the initial design decisions do not satisfy the requirements, whereas the ones returned by the algorithm do. The design decision found by the algorithm reduces the construction cost of the building with 24%, but increases the energy consumption with 33%. Furthermore, the cost of operating the building is increased with 49%. In general, optimizing one performance measure often has unwanted consequences on other performance measures, which is also the case here.

Decision variable		Requirement	Initial value	Optimum value
ϱ			0.200	0.708
N		$= \quad 3$	3.000	3.000
$\sigma^{(1)}$			0.400	0.900
$\sigma^{(2)}$			0.400	0.900
$d_{g,i}$	(m)		0.200	0.206
$d_{w,i}$	(m)		0.200	0.064
$d_{r,i}$	(m)		0.200	0.201
$\alpha_1^{(1)}$			0.500	1.000
$\alpha_2^{(1)}$			0.500	0.000
$\alpha_1^{(2)}$			0.500	1.000
$\alpha_2^{(2)}$			0.500	0.000
Performance measure		**Requirement**	**Initial value**	**Optimum value**
Q_{tot}	(kWh)		136392.96	181874.94
EF_3	(kWh)	$\geq \quad 0$	74275.48	19513.54
EF_2	(kWh)		-25072.73	-75438.90
EF_1	(kWh)		-58488.80	-107389.71
BE	(W)	$\geq \quad 0$	-51.02	0.00
U_g	(W/m^{2}K)	$\leq \quad 0.30$	0.18	0.18
U_{wall}	(W/m^{2}K)	$\leq \quad 0.40$	0.17	0.40
U_r	(W/m^{2}K)	$\leq \quad 0.25$	0.13	0.13
$U_{win}^{(1)}$	(W/m^{2}K)	$\leq \quad 2.30$	1.59	1.82
$U_{win}^{(2)}$	(W/m^{2}K)	$\leq \quad 2.30$	1.59	1.82
$OH^{(1)}$	(h)	$\leq \quad 100$	40.78	52.47
$OH^{(2)}$	(h)	$\leq \quad 100$	8.82	27.77
$DH^{(1)}$		$\leq \quad 4$	4.66	4.00
$DH^{(2)}$		$\leq \quad 4$	4.66	4.00
C_{con}	(DKR)	minimize	9318393.71	7108482.47
C_{op}	(DKR)		67243.00	100492.73

Table 6.9: The first column shows the decision variables and performance measures, and the second column shows the requirements to these parameters. This particular decision problem involves equality, inequality, as well as optimality requirements. The third column shows the values used for initializing the GFSQPF algorithm, and the final column shows the values returned by the algorithm.

6.2.2 Design decisions with minimum energy consumption

The second case study concerns finding design decisions that provide the lowest amount of energy required annually. The building must satisfy the Danish building regulations, as well as the same requirements regarding the indoor environment as in the first case study.

Decision				Initial	Optimum
variable		Requirement		value	value
ϱ				0.200	0.146
N		$=$	3	3.000	3.000
$\sigma^{(1)}$				0.400	0.396
$\sigma^{(2)}$				0.400	0.396
$d_{g,i}$	(m)	$\leq$	0.5	0.200	0.500
$d_{w,i}$	(m)	$\leq$	0.5	0.200	0.143
$d_{r,i}$	(m)	$\leq$	0.5	0.200	0.500
$\alpha_1^{(1)}$				0.500	0.000
$\alpha_2^{(1)}$				0.500	1.000
$\alpha_1^{(2)}$				0.500	0.000
$\alpha_2^{(2)}$				0.500	1.000
Performance				Initial	Optimum
measure		Requirement		value	value
Q_{tot}	(kWh)	minimize		136392.96	138981.18
EF_3	(kWh)	$\geq$	0	74275.48	71406.90
EF_2	(kWh)			-25072.73	-27808.51
EF_1	(kWh)			-58488.80	-61180.31
BE	(W)	$\geq$	0	-51.02	3105.42
U_g	(W/m^{2}K)	$\leq$	0.30	0.18	0.07
U_{wall}	(W/m^{2}K)	$\leq$	0.40	0.17	0.22
U_r	(W/m^{2}K)	$\leq$	0.25	0.13	0.06
$U_{win}^{(1)}$	(W/m^{2}K)	$\leq$	2.30	1.59	1.36
$U_{win}^{(2)}$	(W/m^{2}K)	$\leq$	2.30	1.59	1.36
$OH^{(1)}$	(h)	$\leq$	100	40.78	44.98
$OH^{(2)}$	(h)	$\leq$	100	8.82	8.84
$DH^{(1)}$		$\leq$	4	4.66	4.00
$DH^{(2)}$		$\leq$	4	4.66	4.00
C_{con}	(DKR)	$\leq$	10000000.00	9318393.71	10000000.00
C_{op}	(DKR)			67243.00	69577.85

Table 6.10: The table shows requirements to design variables and performance measures, related to the problem of finding a design with minimum energy consumption. The third and fourth columns show the initial and optimum values of design variables and performance measures, respectively.

In order to ensure that the design decision problem has a finite solution, an upper limit of 10 million DKR is imposed on the cost of constructing the building. Without such a requirement, the algorithm will increase the amount of insulation indefinitely. The amount of insulation is furthermore subjected to an upper limit of 0.5 m, which reduces the chances that the algorithm suggests unwanted decisions even more.

In Table 6.10 are shown the requirements to decision variables and performance measures related to this decision problem, as well as the initial values and the values returned by the GFSQPF algorithm.

In order to minimize the energy consumption, the algorithm suggests a design that is less compact than the initial value, which may seem counter-intuitive. However, a more compact design requires larger windows in order to satisfy the requirements to the use of natural light, which leads to a higher energy consumption. Therefore, a balance is needed between the compactness of the building and the window areas, and in this case, this balance is provided by a building with a width to length ratio of approximately 0.15, and a window fraction of approximately 0.4.

The algorithm suggests using the maximum allowed amount of insulation in the roof construction and the ground slab, but not in the external walls, which also seems counter-intuitive. This solution probably provides the maximum level of insulation within the given budget.

Not surprisingly, the algorithm suggests using triple-glazed windows on both façades.

The design decision found by the algorithm satisfies all requirements, but the annual amount of energy required by the building is increased with approximately 2%, when compared with the initial design. However, the initial design is not feasible, and the algorithm only searches for the solution within the set of feasible designs.

When compared with the design found in the first case study, the annual energy consumption is reduced with approximately 23%, however, the construction cost is increased with approximately 41%.

6.3 Final remarks

The results regarding the gradient-free SQP algorithm indicate that using a quadratic damping term instead of box constraints improves the performance of the algorithm, when compared with an algorithm using box constraints. The average number of iterations needed for obtaining a solution estimate with a relative error less than 10^{-6} is more than 4 times less when using a damping term, for the considered test problems.

When using Broyden approximations, the number of iterations increases with approximately 30%, which means that the performance of the GFSQPF algorithm seems to be at least as good as the SLPF algorithm. In fact, for the considered test problems, the GFSQPF algorithm needed less than half the number of iterations than the SLPF algorithm, in order to provide a satisfactory solution estimate. This observation is based on the considered test problems, and is not necessarily true in general.

The algorithms do not seem to converge for test problems with active domain constraints, especially if (parts of) the boundary of the feasible region is established by domain constraints. A possible solution is to handle domain constraints by using a barrier methodology.

The algorithms have so far only been tested on 2-dimensional optimization problems. Testing them on more general problems, for instance the CUTE testing environment proposed by Bongartz et.al. [10], is a possible topic for further research.

Two case studies are conducted, one aiming at finding the design with the smallest construction cost, and the other aiming at finding the design with the smallest energy consumption. The case studies illustrate how the requirements to the energy consumption of buildings, described in the Danish building regulations, can be included in the formulation of the building design decision problem.

The building is furthermore subjected to requirements to the indoor environment and the economy. The algorithm is able to find designs that satisfy these requirements within an acceptable time period (at most an hour in both cases).

It is found that a building with low construction costs is not very energy efficient, and vice versa, as one might expect. Multi-criteria optimization methods can be used for conducting a more thorough investigation of the compromise that exists between performance measures.

Chapter 7

Conclusions

The purpose of this study has been to describe a method for optimizing the performance of buildings, and to further improve the understanding of how numerical optimization methods can be used for supporting decision-making, with special focus on design decisions for buildings in the early stages of the design process.

The study is motivated by the fact that it is easier and less costly to change design decisions in the early stages rather than later, and that changes made in early stages have a larger impact on the building performance than changes made later. Furthermore, the parties involved in decision-making for buildings often have different and to some extent conflicting requirements to buildings. It is therefore important to develop methods that focus on design decisions in the early stages, and that are flexible. This study addresses these concerns by combining performance calculation methods for buildings with numerical optimization methods.

Chapter 2 provides a literature survey of optimization-related topics that are relevant for optimizing the performance of buildings, as well as a short survey of methods for calculating the performance of buildings with respect to energy, economy and the indoor environment. Furthermore, a survey of building optimization methods found in the literature is provided. This survey supports the idea that it is advantageous to develop flexible building optimization methods that enable decision makers to optimize any aspect of the building performance.

This issue is addressed in Chapter 3, where an optimization problem is formulated, intended for representing a wide range of design decision problems for buildings. The formulation allows the decision-maker to specify requirements to decision variables and performance measures in a highly flexible way. The decision variables and performance measures can be subjected to equality and inequality requirements, and the performance measures can furthermore be subjected to optimality requirements.

Chapter 4 concerns the details of the proposed building optimization method. The method suggests design decisions by optimizing the performance of a building with a simplified geometry. The method supports design decisions regarding the shape of the building, the window fraction of the façade areas, the window types and the amount of insulation used in the building envelope. The performance calculation methods are described, which involve the energy performance, the economy, and quality of the indoor environment of the building. It is furthermore described how requirements to the energy performance of buildings made by the Danish building regulations can be included in the design decision problem.

Chapter 5 describes a gradient-free SQP filter algorithm (GFSQPF), intended for solving the formulated optimization problem. The algorithm is based on the SLP filter algorithm by Fletcher, but it restricts the step length from one iteration to the next by using a quadratic damping term. Furthermore, the first order partial derivatives of the functions defining the optimization problem are approximated using the Broyden rank one updating formula. The approximations are initialized using finite differences. The algorithm includes so-called domain constraints, which are used for ensuring that the optimization algorithm only calculates the performance measures for design decisions that belong to the domain of the performance measures.

Three algorithms are described, which are used for comparative studies. The first algorithm (SLPF) is a variant of Fletchers algorithm that uses domain constraints, and that updates the trust region radius in the same way as the other algorithms. The step length is restricted by so-called box constraints. The second algorithm (SQPF) uses domain constraints, as well as a quadratic damping term, and requires information regarding the first partial derivatives of the functions that define the optimization problem. The third algorithm is GFSQPF.

The building optimization method is evaluated in Chapter 6. First, numerical experiments are conducted in order to investigate the potential benefits of using a quadratic damping term instead of box constraints, and to investigate the convergence properties of the GFSQPF algorithm. Secondly, the building optimization method, which involves the GFSQPF algorithm, is applied to case studies concerning the design of an office building.

The results for the GFSQPF algorithm can be summarized as follows:

1. Restricting the step length using a quadratic damping term seems to provide faster convergence and a more stable algorithm, when compared to an algorithm using box constraints.

2. Using Broyden updated approximations to the first order partial derivatives seems to provide slightly slower convergence, but more or less the same stability as an algorithm using exact information regarding the partial derivatives.

3. When the optimization problem has active domain constraints, convergence seems to be either deteriorated or prevented. Further research is needed for resolving this issue.

The building optimization method is evaluated by applying it to case studies regarding the design of an office building. The first case study concerns finding design decisions with minimum construction costs. The building is required to satisfy the energy frame EF_3, the requirement regarding the heat loss through the building envelope and the U-value requirements for the components of the building envelope. Furthermore, the building must satisfy requirements to the indoor environment, and to the use of natural light.

The second case study concerns finding design decisions with minimum energy consumption. The building is required to satisfy the same requirements as in the first case study. Furthermore, the cost of constructing the building is subjected to an upper limit of 10 million DKR, in order to ensure that the optimization problem has a finite solution. The amount of insulation used in the building envelope is furthermore subjected to an upper limit of 0.5 m.

Both case studies indicate that the method is able to find design decisions that satisfy all requirements within an hour. The cost of constructing the building is 41% higher for the energy-efficient design found in the second case study, compared with the cost effective design found in the first case study. However, the annual energy consumption is reduced with 23%. Multi-criteria optimization methods can be used for investigating the compromise that exists between performance measures.

Some of the design decisions found by the building optimization method seem to be counter-intuitive. This indicates that optimization in general is a useful approach for finding optimum design decisions for complex systems, such as buildings, where it might be difficult to find such decisions by relying only on engineering intuition.

7.1 Contributions provided by the study

The following contributions have been provided by the present study:

1. A literature survey of optimization-related topics that are relevant for developing building optimization methods.

2. A formulation of an optimization problem that is useful for representing a wide range of design decision problems for buildings.

3. A building optimization method, intended for suggesting design decisions in the early stages of the design process for buildings.

4. A gradient-free SQP filter algorithm intended for solving the formulated optimization problem.

5. An evaluation of the building optimization method through numerical experiments for the filter SQP algorithm, and case studies for the building optimization method.

6. A space mapping interpolating surrogate algorithm, intended for solving optimization problems with time-consuming or costly objective function evaluations.

7. A space mapping modeling technique, intended for improving the accuracy of simplified models of physical systems.

7.2 Unresolved issues and suggestions for further research

The following issues, among others, have not been resolved by the present study, but are suggested as further research in the field of building optimization:

1. Only a single-criteria formulation of the building design decision problem has been provided so far. A multi-criteria formulation will enable the decision-maker to investigate the trade-off between the different performance measures.

2. The present problem formulation does not include reliability analysis, which will enable the building optimization method to include the probability of failure in the problem formulation.

3. The present problem formulation furthermore does not include sensitivity analysis, which is useful for investigating how decisions can be influenced by changes in constant parameters such as prices, physical properties of the building components or climate parameters.

4. The present problem formulation is intended for suggesting optimum design decisions; however, there are other aspects of decision-making that are relevant. For instance, the problem of finding relaxations to a set of requirements, if they render the design decision problem infeasible.

5. The only discrete decision variables included so far concern the windows, however, there are a number of other discrete decisions that are relevant. For instance, selecting active solar shading devices, or selecting combinations of (renewable) energy systems that provide the most desirable performance.

6. Using performance measures such as PMV and PPD, or the environmental impact, has not yet been addressed. Including these performance measures will further increase the usability of the building optimization method.

7. It is relevant to consider calculating the energy performance of buildings using the BE06 software, since it implements the calculation methods required by the Danish building regulations.

8. Further research is needed regarding the convergence properties of the GFSQPF algorithm. Finding efficient updating strategies for the trust region radius ρ_k will improve convergence, and finding methods for ensuring that the steps Δx_k made by the algorithm are linearly independent will make the algorithm more stable. Furthermore, the possibility for developing a gradient-free logarithmic barrier method should be investigated.

9. The optimality conditions for the design decisions found in the case studies were not investigated, which needs to be addressed. This will also be useful for providing a more accurate estimate of the time needed by the building optimization method for finding a satisfactory solution estimate.

10. Programming specific details have not been addressed. For instance, developing a computer-aided design (CAD) modeling environment will provide a simple and easy to use graphical user interface to the building optimization method. A database management system will be useful for managing the large amount of data required for representing the design and performance of buildings.

11. The space mapping interpolating surrogate algorithm has so far only been applied to minimax optimization problems. In order to include it in the building optimization method, it is suggested to develop a space mapping interpolating surrogate method for continuous, constrained optimization problems. This can for instance be accomplished by applying the interpolating surrogate approach to the functions that define continuous, constrained optimization problems, combined with a filter approach as acceptance criteria for the iterates.

References

[1] S. Aggerholm and K. Grau, *Regulation 213 – Energy consumption of buildings (Danish title: Anvisning 213 - Bygningers energibehov)*, Danish Building Research Institute, Dr. Neergaards Vej 15, DK-2970 Hørsholm, Denmark, 2005.

[2] J.W. Bandler, R.M. Biernacki, S.H. Chen, P.A. Grobelny and R.H. Hemmers, *Space mapping technique for electromagnetic optimization*, IEEE Trans. Microwave Theory Tech., vol. 42, no. 12, pp. 2536-2544, 1994.

[3] J.W. Bandler, Q.S. Cheng, D. Gebre-Mariam, K. Madsen, F. Pedersen and J. Søndergaard, *EM-based surrogate modeling and design exploiting implicit, frequency and output space mappings*, in: 2003 IEEE MTT-S International Microwave Symposium Digest, pp. 1003-1006, 2003.

[4] J.W. Bandler, Q. Cheng, S.A. Dakroury, A.S. Mohamed, M.H. Bakr, K. Madsen and J. Søndergaard, *Space mapping: The state of the art*, IEEE Trans. Microwave Theory Tech., vol. 52, no. 1, pp. 337-361, 2004.

[5] J.W. Bandler, Q.S. Cheng, S. Dakroury, D. Hailu, K. Madsen, A.S. Mohamed and F. Pedersen, *Space mapping interpolating surrogates for highly optimized EM-based design of microwave devices*, in: 2004 IEEE MTT-S International Microwave Symposium Digest, pp. 1565-1568, 2004.

[6] J.W. Bandler, Q.S. Cheng, D. Hailu, A.S. Mohamed, M.H. Bakr, K. Madsen and F. Pedersen, *Recent trends in space mapping technology*, in: Proceedings of 2004 Asia-Pacific Microwave Conference, 2004.

[7] J.W. Bandler, D.M. Hailu, K. Madsen and F. Pedersen, *A space mapping interpolating surrogate algorithm for highly optimized EM-based design of microwave devices*, IEEE Trans. Microwave Theory Tech., vol. 52, no. 11, pp. 2593-2600, 2004.

[8] M.H. Bakr, J.W. Bandler, K. Madsen and J. Søndergaard, *Review of the space mapping approach to engineering optimization and modeling*, Optimization and Engineering, vol. 1, no. 3, pp. 241-276, 2000.

[9] M.H. Bakr, J.W. Bandler, K. Madsen and J. Søndergaard, *An introduction to the space mapping technique*, Optimization and Engineering, vol. 2, no. 4, pp. 369-384, 2001.

[10] I. Bongartz, A.R. Conn, N.I.M. Gould and Ph.L. Toint, *CUTE: Constrained and unconstrained testing environment*, ACM Transactions on Mathematical Software, vol. 21, no 1, pp. 123-160, 1995.

[11] N. Bouchlaghem, *Optimising the design of building envelopes for thermal performance*, Automation in Construction, vol. 10, pp. 101-112, 2000.

[12] C.G. Broyden, *A class of methods for solving nonlinear simultaneous equations*, Mathematics of Computation, vol. 19, pp. 577-593, 1965.

[13] *Building regulations 1998*, National Agency for Enterprise and Construction, Copenhagen, Denmark, 1998.

[14] A.R. Conn, N.I.M. Gould and Ph.L. Toint, *Trust-region methods*, Society of Industrial and Applied Mathematics, Philadelphia, USA, 2000.

[15] *Danish electricity price statistics for 2nd quarter 2006 (Danish title: Elprisstatistik 2. kvartal 2006)*, Danish Energy Regulatory Authority, 30 Nyropsgade, DK-1780 Copenhagen V, Denmark (www.energitilsynet.dk).

[16] I. Das and J.E. Dennis, *Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems*, SIAM Journal on Optimization, vol. 8, no. 3, pp. 631-657, 1998.

[17] J.E. Dennis, Jr. and R.B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*, Society of Industrial and Applied Mathematics, Philadelphia, USA, 1996.

[18] Directive 2002/91/EC of the European Parliament and of the Council of 16 December 2002 on the energy performance of buildings.

[19] H. Eschenauer, J. Koski and A. Osyczka (Editor), *Multicriteria Design Optimization: Procedures and Applications*, Springer-Verlag, Berlin, Germany, 1990.

[20] P.O. Fanger, *Thermal Comfort*, Danish Technical Press, Copenhagen, Denmark, 1970.

[21] A.V. Fiacco, *Introduction to sensitivity and stability analysis in nonlinear programming*, Academic Press, New York, USA, 1983.

[22] R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function*, Numerical Analysis Report NA/171, Department of Mathematics, University of Dundee, Scotland, 1997.

[23] R. Fletcher, S. Leyffer and P.L. Toint, *On the global convergence of an SLP-filter algorithm*, Numerical Analysis Report NA/183, Department of Mathematics, University of Dundee, Scotland, 1998.

[24] C.M. Fonseca and P.J. Flemming, *Multiobjective optimization and multiple constraint handling with evolutionary algorithms – part 1: a unified formulation*, IEEE Transactions on Systems, Man and Cybernetics, Part A, vol. 28, no. 1, pp. 26-37, 1998.

[25] L.G. Caldas, L.K. Norford, *A design optimization tool based on a genetic algorithm*, Automation in Construction, vol. 11, pp. 173-184, 2002.

[26] F.W. Gembicki, *Vector optimization for control with performance and parameter sensitivity indices*, Ph.D. Thesis, Case Western Reserve University, Cleveland, Ohio, USA, 1974.

[27] D. Goldberg, *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley Publishing, 1989.

[28] G.H. Golub and C.F. Van Loan, *Matrix Computations* (3rd ed.), The John Hopkins University Press, Baltimore, USA, 1996.

[29] L. Gonzalez-Monroy and A. Cordoba, *Optimization of energy supply systems with simulated annealing: continuous and discrete descriptions*, Physica A, vol. 284, pp. 433-447, 2000.

[30] B.A. Güvenç and L. Güvenç, *Robust repetitive controller design in parameter space*, Journal of Dynamic Systems Measurement and Control – Transactions of the ASME vol. 128, no. 2, pp. 406-413, 2006.

[31] C.E. Hagentoft, *Introduction to building physics*, Studentlitteratur, Box 141, 221 00 Lund, Sweden, 2001.

[32] A. Haldar and S. Mahadevan, *Probability, reliability and statistical methods in engineering design*, John Wiley & Sons, New York, USA, 2000.

[33] F.S. Hillier and G.J. Lieberman, *Introduction to operations research* (7th ed.), McGraw-Hill, New York, USA, 2001.

[34] R. Hooke and T.A. Jeeves, *"Direct search" solution of numerical and statistical problems*, Journal of the ACM, vol. 8, no. 2, pp. 212-229, 1961.

[35] H. Jedrzejuk and W. Marks, *Optimization of shape and functional structure of buildings as well as heat source. Basic theory*, Building and Environment, vol. 37, pp. 1379-1383, 2002.

[36] H. Jedrzejuk and W. Marks, *Optimization of shape and functional structure of buildings as well as heat source utilisation. Partial problems solution*, Building and Environment, vol. 37, pp. 1037-1043, 2002.

[37] H. Jedrzejuk and W. Marks, *Optimization of shape and functional structure of buildings as well as heat source. Utilisation example*, Building and Environment, vol. 37, pp. 1249-1253, 2002.

[38] J. Karlsson and A. Roos, *Modelling the angular behaviour of the total solar energy transmittance of windows*, Solar Energy, vol. 69, no. 4, pp. 321-329, 2000.

[39] J. Kragh, F. Pedersen and S. Svendsen, *Weather test reference year of Greenland*, in: Proceedings of the 7th Symposium on Building Physics in the Nordic Countries, pp. 1147-1154, 2005.

[40] K. Levenberg, *A method for the solution of certain problems in least squares*, Quart. Appl. Math., vol. 2, pp. 164-168, 1944.

[41] K. Madsen, H.B. Nielsen and O. Tingleff, *Methods for non-linear least squares problems (2nd ed.)*, Lecture note, Informatics and Mathematical Modelling, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark, 2004.

[42] K. Madsen and J. Søndergaard, *Convergence of hybrid space mapping algorithms*, Optimization and Engineering, vol. 5, no. 2, pp. 145-156, 2004.

[43] D.W. Marquardt, *An algorithm for least-squares estimation of nonlinear parameters*, Journal of the Society for Industrial and Applied Mathematics, vol. 11, no. 2, pp. 431-441, 1963.

[44] *Matlab^{TM} Version 7.1*, and *Matlab Optimization Toolbox Version 3.0.3*, The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098, USA, 2004.

[45] R.A. Mitchell and J.L. Kaplan, *Non linear constrained optimization by a non-random complex method*, J. Res. Natl. Bur. Stand. C72, 24-258, 1969.

[46] J.A. Nelder and R. Mead, *A simplex method for function minimization*, Computer Journal, vol. 7, pp. 308-313, 1965.

[47] T.R. Nielsen and S. Svendsen, *Life cycle cost optimization of buildings with regard to energy use, thermal indoor environment and daylight*, International Journal of Low Energy and Sustainable Buildings, vol. 2, pp. 1-16, 2002.

[48] T.R. Nielsen, *BuildingCalc – Users Guide*, Lecture note, Department of Civil Engineering, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark, 2003.

[49] T.R. Nielsen, *Optimization of buildings with respect to energy and indoor environment*, Ph.D. Thesis BYG-DTU R-036 (ISBN 87-7877-094-7), Department of Civil Engineering, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark, 2003.

[50] T.R. Nielsen, *Simple tool to evaluate energy demand and indoor environment in the early stages of building design*, Solar Energy, vol. 78, no. 1, pp. 73-83, 2005.

[51] J. Nocedal and S.J. Wright, *Numerical optimization*, Springer, New York, USA, 1999.

[52] V. Pareto, *Manual of political economy*, Augustus M. Kelley Pubs, New York, USA, 1969.

[53] S. V. Patankar, *Numerical heat transfer and fluid flow*, Hemisphere, New York, 1980.

[54] F. Pedersen, P. Weitzmann and S. Svendsen, *Modeling thermally active building components using space mapping*, in: Proceedings of the 7th Symposium on Building Physics in the Nordic Countries, pp. 896-903, 2005.

[55] K. Peippo, P.D. Lund and E. Vartiainen, *Multivariate optimization of design tradeoffs for solar low energy buildings*, Energy and Buildings, vol. 29, pp. 189-205, 1999.

[56] B. Poel, *Integrated design with a focus on energy aspects*, in: Proceedings of the ECEEE Summer Study 2005, pp. 505-512, 2005.

[57] M.J.D. Powell, *A hybrid method for non-linear equations*, in: P. Rabinowitz (ed), Numerical Methods for Non-Linear Algebraic Equations, Gordon and Breach Science, London, pp. 87-144, 1970.

[58] *Rate sheet for the financial year 2006 (Danish title: Takstblad for regnskabsåret 2006)*, Ishøj district heating plant, 34 Industrivangen, DK-2635 Ishøj, Denmark (www.ishoj.dk).

[59] H.H. Rosenbrock, *An automatic method for finding the greatest or least value of a function*, Computer Journal, vol. 3, no. 3, pp. 175-184, 1960.

[60] *The V&S price catalogue 2005, Gross prices for house construction (Danish title: V&S prisbogen 2005, husbygning brutto)*, Byggecentrum, Lautrupvang 1B, DK-2750 Ballerup, Denmark, 2005.

[61] *The V&S price catalogue 2005, Gross prices for renovation and operation (Danish title: V&S prisbogen 2005, renovering & drift brutto)*, Byggecentrum, Lautrupvang 1B, DK-2750 Ballerup, Denmark, 2005.

[62] W. Wang, R. Zmeureanu and H. Rivard, *Applying multi-objective genetic algorithms in green building design optimization*, Building and Environment, vol. 40, pp. 1512-1525, 2005.

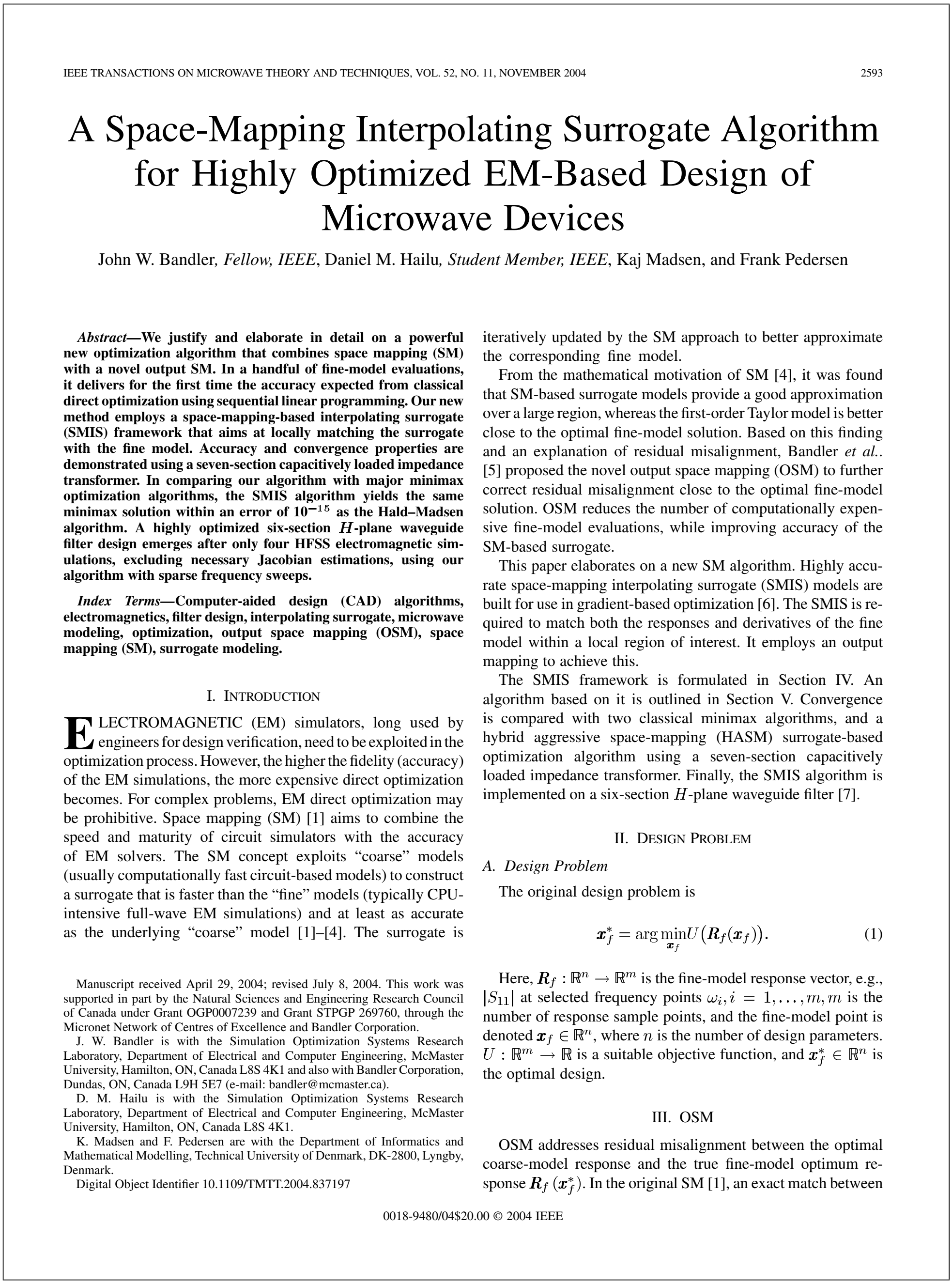
[63] J.A. Wright, H.A. Loosemore and R. Farmani, *Optimization of building thermal design and control by multi-criterion genetic algorithm*, Energy and Buildings, vol. 34, pp. 959-972, 2002.

[64] S. Ye, Q. Sun and E.C. Chang, *Edge directed filter based error concealment for wavelet-based images*, in: Proceedings of the 2004 International Conference on Image Processing, pp. 809-812, 2004.

Appendix A

A space mapping interpolating surrogate algorithm for highly optimized EM-based design of microwave devices

Authors	John W. Bandler, Daniel M. Hailu, Kaj Madsen and Frank Pedersen
Title	A Space Mapping Interpolating Surrogate Algorithm for Highly Optimized EM-Based Design of Microwave Devices
Journal	IEEE Transactions on Microwave Theory and Techniques (ISSN 0018-9480)
Volume	52
Issue	11
Pages	2593-2600
Year	2004



IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES, VOL. 52, NO. 11, NOVEMBER 2004 2593

A Space-Mapping Interpolating Surrogate Algorithm for Highly Optimized EM-Based Design of Microwave Devices

John W. Bandler, *Fellow, IEEE*, Daniel M. Hailu, *Student Member, IEEE*, Kaj Madsen, and Frank Pedersen

***Abstract*—We justify and elaborate in detail on a powerful new optimization algorithm that combines space mapping (SM) with a novel output SM. In a handful of fine-model evaluations, it delivers for the first time the accuracy expected from classical direct optimization using sequential linear programming. Our new method employs a space-mapping-based interpolating surrogate (SMIS) framework that aims at locally matching the surrogate with the fine model. Accuracy and convergence properties are demonstrated using a seven-section capacitively loaded impedance transformer. In comparing our algorithm with major minimax optimization algorithms, the SMIS algorithm yields the same minimax solution within an error of 10^{-15} as the Hald–Madsen algorithm. A highly optimized six-section H-plane waveguide filter design emerges after only four HFSS electromagnetic simulations, excluding necessary Jacobian estimations, using our algorithm with sparse frequency sweeps.**

***Index Terms*—Computer-aided design (CAD) algorithms, electromagnetics, filter design, interpolating surrogate, microwave modeling, optimization, output space mapping (OSM), space mapping (SM), surrogate modeling.**

I. Introduction

ELECTROMAGNETIC (EM) simulators, long used by engineers for design verification, need to be exploited in the optimization process. However, the higher the fidelity (accuracy) of the EM simulations, the more expensive direct optimization becomes. For complex problems, EM direct optimization may be prohibitive. Space mapping (SM) [1] aims to combine the speed and maturity of circuit simulators with the accuracy of EM solvers. The SM concept exploits "coarse" models (usually computationally fast circuit-based models) to construct a surrogate that is faster than the "fine" models (typically CPU-intensive full-wave EM simulations) and at least as accurate as the underlying "coarse" model [1]–[4]. The surrogate is iteratively updated by the SM approach to better approximate the corresponding fine model.

From the mathematical motivation of SM [4], it was found that SM-based surrogate models provide a good approximation over a large region, whereas the first-order Taylor model is better close to the optimal fine-model solution. Based on this finding and an explanation of residual misalignment, Bandler *et al.*. [5] proposed the novel output space mapping (OSM) to further correct residual misalignment close to the optimal fine-model solution. OSM reduces the number of computationally expensive fine-model evaluations, while improving accuracy of the SM-based surrogate.

This paper elaborates on a new SM algorithm. Highly accurate space-mapping interpolating surrogate (SMIS) models are built for use in gradient-based optimization [6]. The SMIS is required to match both the responses and derivatives of the fine model within a local region of interest. It employs an output mapping to achieve this.

The SMIS framework is formulated in Section IV. An algorithm based on it is outlined in Section V. Convergence is compared with two classical minimax algorithms, and a hybrid aggressive space-mapping (HASM) surrogate-based optimization algorithm using a seven-section capacitively loaded impedance transformer. Finally, the SMIS algorithm is implemented on a six-section H-plane waveguide filter [7].

II. Design Problem

A. Design Problem

The original design problem is

$$\boldsymbol{x}_f^* = \arg\min_{\boldsymbol{x}_f} U\big(\boldsymbol{R}_f(\boldsymbol{x}_f)\big). \tag{1}$$

Here, $\boldsymbol{R}_f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is the fine-model response vector, e.g., $|S_{11}|$ at selected frequency points $\omega_i, i = 1, \ldots, m, m$ is the number of response sample points, and the fine-model point is denoted $\boldsymbol{x}_f \in \mathbb{R}^n$, where n is the number of design parameters. $U : \mathbb{R}^m \rightarrow \mathbb{R}$ is a suitable objective function, and $\boldsymbol{x}_f^* \in \mathbb{R}^n$ is the optimal design.

III. OSM

OSM addresses residual misalignment between the optimal coarse-model response and the true fine-model optimum response $\boldsymbol{R}_f\,(\boldsymbol{x}_f^*)$. In the original SM [1], an exact match between

Manuscript received April 29, 2004; revised July 8, 2004. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under Grant OGP0007239 and Grant STPGP 269760, through the Micronet Network of Centres of Excellence and Bandler Corporation.

J. W. Bandler is with the Simulation Optimization Systems Research Laboratory, Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON, Canada L8S 4K1 and also with Bandler Corporation, Dundas, ON, Canada L9H 5E7 (e-mail: bandler@mcmaster.ca).

D. M. Hailu is with the Simulation Optimization Systems Research Laboratory, Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON, Canada L8S 4K1.

K. Madsen and F. Pedersen are with the Department of Informatics and Mathematical Modelling, Technical University of Denmark, DK-2800, Lyngby, Denmark.

Digital Object Identifier 10.1109/TMTT.2004.837197

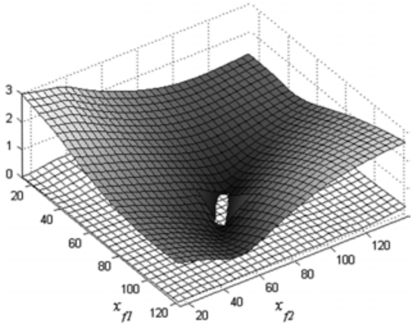


Fig. 1. Error plots for a two-section capacitively loaded impedance transformer [4] exhibiting the quasi-global effectiveness of SM (light grid) versus a classical Taylor approximation (dark grid). See text.

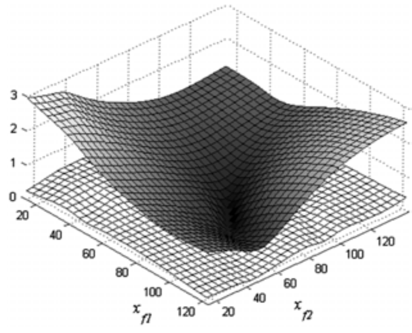


Fig. 2. Error plots for a two-section capacitively loaded impedance transformer [4] exhibiting the quasi-global effectiveness of SM-based interpolating surrogate, which exploits OSM (light grid) versus a classical Taylor approximation (dark grid). See text.

the fine model and mapped coarse model is unlikely. For example, a coarse model such as $R_c = x^2$ will never match the fine model $R_f = x^2 - 2$ around its minimum with any mapping $x_c = P(x_f), x_c, x_f \in \mathbb{R}$. An "output" or response mapping can overcome this deficiency by introducing a transformation of the coarse-model response based on a Taylor approximation [8].

The results of Bakr *et al.* [4] indicate that "input" SM-based surrogates are good approximations to the fine model over a large region, which makes them useful in the early stages of an optimization process. The residual misalignment between the corresponding mapped coarse model(s) and the fine model renders an exact match between them unlikely. Consequently, convergence to $\boldsymbol{x}_f^*$ should not be expected.

Fig. 1 depicts model effectiveness plots [4] for a two-section capacitively loaded impedance transformer at the final iterate $\boldsymbol{x}_f^{(i)}$, approximately $[74.23\ 79.27]^T$. Centered at $\boldsymbol{h} = \boldsymbol{0}$, the light grid shows $\|\boldsymbol{R}_f(\boldsymbol{x}_f^{(i)} + \boldsymbol{h}) - \boldsymbol{R}_c(\boldsymbol{L}_p(\boldsymbol{x}_f^{(i)} + \boldsymbol{h}))\|$. This represents the deviation of the mapped coarse model (using the Taylor approximation $\boldsymbol{L}_p : \mathbb{R}^n \to \mathbb{R}^n$ to the mapping, i.e., a linearized mapping) from the fine model. The dark grid shows $\|\boldsymbol{R}_f(\boldsymbol{x}_f^{(i)} + \boldsymbol{h}) - \boldsymbol{L}_f(\boldsymbol{x}_f^{(i)} + \boldsymbol{h})\|$. This is the deviation of the fine model from its classical Taylor approximation $\boldsymbol{L}_f : \mathbb{R}^n \to \mathbb{R}^m$. The gradient of the two-section capacitively loaded impedance transformer, used in the Taylor approximation, was obtained analytically using the adjoint network method [9]. The light grid surface passing over the dark grid surface near the center of Fig. 1 verifies that the Taylor approximation is most accurate close to $\boldsymbol{x}_f^{(i)}$, whereas the mapped coarse model is best over a larger region. The reason that the Taylor approximation is best in the vicinity of $\boldsymbol{x}_f^{(i)}$ is that the Taylor approximation interpolates at the development point, whereas the mapped coarse model does not.

Based on the above finding, Bakr *et al.* [10] use a surrogate that is a convex combination of a mapped coarse model and a first-order Taylor approximation of the fine model. Madsen and Søndergaard [11] prove convergence of such HASM algorithms.

In this paper, we introduce a novel method to ensure convergence of the SM technique. OSM is incorporated into SMIS to ensure that we obtain the same solution as classical direct gradient-based optimization. Fig. 2 depicts model effectiveness plots for the two-section capacitively loaded impedance transformer corresponding to Fig. 1. Centered at $\boldsymbol{h} = \boldsymbol{0}$, the light grid shows $\|\boldsymbol{R}_f(\boldsymbol{x}_f^{(i)} + \boldsymbol{h}) - \boldsymbol{R}_s(\boldsymbol{x}_f^{(i)} + \boldsymbol{h})\|$. This represents the deviation of the SMIS surrogate from the fine model. The dark grid shows the deviation of the fine model from its classical Taylor approximation as in Fig. 1. Thus, Fig. 2 demonstrates that the SMIS surrogate, because of its interpolating properties, performs better than the first-order Taylor approximation even close to $\boldsymbol{x}_f^{(i)}$.

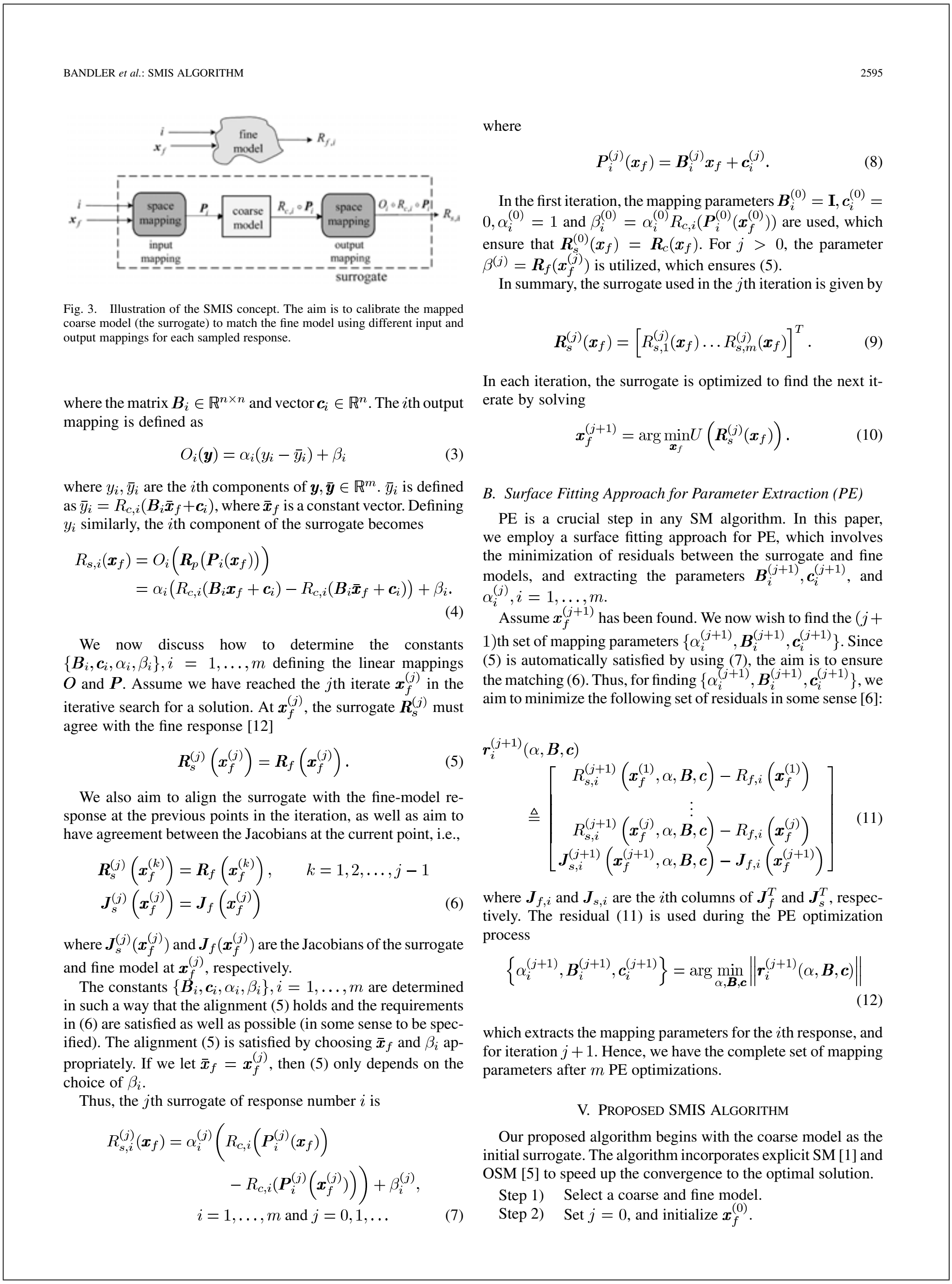
IV. SMIS Framework

A. Surrogate

The SM-based interpolating surrogate $\boldsymbol{R}_s : \mathbb{R}^n \to \mathbb{R}^m$ is defined as a transformation of a coarse model $\boldsymbol{R}_c : \mathbb{R}^n \to \mathbb{R}^m$ through input and output mappings at each sampled response. Fig. 3 illustrates the SMIS framework. Here, $\boldsymbol{P} = [\boldsymbol{P}_1 \ldots \boldsymbol{P}_m]^T$, where $\boldsymbol{P}_i : \mathbb{R}^n \to \mathbb{R}^n, i = 1, \ldots, m$, [1], [2] is an input mapping for the ith coarse response $R_{c,i}$, and $\boldsymbol{O} : \mathbb{R}^m \to \mathbb{R}^m$ [8] is an output mapping applied to the coarse response. Using the function $\boldsymbol{R}_p : \mathbb{R}^{m \cdot n} \to \mathbb{R}^m$ with individually adjusted coarse responses, defined as $\boldsymbol{R}_p(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m) = [R_{c,1}(\boldsymbol{z}_1) \ldots R_{c,m}(\boldsymbol{z}_m)]^T$, where $\boldsymbol{z}_i \in \mathbb{R}^n, i = 1, \ldots, m$, the surrogate can be expressed as a composed mapping $\boldsymbol{R}_s = \boldsymbol{O} \circ \boldsymbol{R}_p \circ \boldsymbol{P}$.

We wish to consider individual mappings of each coarse response $R_{c,i}, i = 1, \ldots, m$. These (nonlinear) mappings will be approximated by a sequence of local linear mappings. The ith linearized input mapping at the jth iteration is assumed to be of the form

$$\boldsymbol{P}_i(\boldsymbol{x}_f) = \boldsymbol{B}_i \boldsymbol{x}_f + \boldsymbol{c}_i \tag{2}$$



i
$\boldsymbol{x}_f$
fine model
$R_{f,i}$
space mapping
input mapping
$\boldsymbol{P}_i$
coarse model
$R_{c,i} \circ \boldsymbol{P}_i$
space mapping
output mapping
$O_i \circ R_{c,i} \circ \boldsymbol{P}_i$
$R_{s,i}$
surrogate

Fig. 3. Illustration of the SMIS concept. The aim is to calibrate the mapped coarse model (the surrogate) to match the fine model using different input and output mappings for each sampled response.

where the matrix $\boldsymbol{B}_i \in \mathbb{R}^{n\times n}$ and vector $\boldsymbol{c}_i \in \mathbb{R}^n$. The ith output mapping is defined as

$$O_i(\boldsymbol{y}) = \alpha_i(y_i - \bar{y}_i) + \beta_i \tag{3}$$

where $y_i, \bar{y}_i$ are the ith components of $\boldsymbol{y}, \bar{\boldsymbol{y}} \in \mathbb{R}^m$. $\bar{y}_i$ is defined as $\bar{y}_i = R_{c,i}(\boldsymbol{B}_i\bar{\boldsymbol{x}}_f + \boldsymbol{c}_i)$, where $\bar{\boldsymbol{x}}_f$ is a constant vector. Defining y_i similarly, the ith component of the surrogate becomes

$$\begin{aligned} R_{s,i}(\boldsymbol{x}_f) &= O_i\Big(\boldsymbol{R}_p\big(\boldsymbol{P}_i(\boldsymbol{x}_f)\big)\Big) \\ &= \alpha_i\big(R_{c,i}(\boldsymbol{B}_i\boldsymbol{x}_f + \boldsymbol{c}_i) - R_{c,i}(\boldsymbol{B}_i\bar{\boldsymbol{x}}_f + \boldsymbol{c}_i)\big) + \beta_i. \end{aligned} \tag{4}$$

We now discuss how to determine the constants $\{\boldsymbol{B}_i, \boldsymbol{c}_i, \alpha_i, \beta_i\}, i = 1, \ldots, m$ defining the linear mappings $\boldsymbol{O}$ and $\boldsymbol{P}$. Assume we have reached the jth iterate $\boldsymbol{x}_f^{(j)}$ in the iterative search for a solution. At $\boldsymbol{x}_f^{(j)}$, the surrogate $\boldsymbol{R}_s^{(j)}$ must agree with the fine response [12]

$$\boldsymbol{R}_s^{(j)}\left(\boldsymbol{x}_f^{(j)}\right) = \boldsymbol{R}_f\left(\boldsymbol{x}_f^{(j)}\right). \tag{5}$$

We also aim to align the surrogate with the fine-model response at the previous points in the iteration, as well as aim to have agreement between the Jacobians at the current point, i.e.,

$$\begin{aligned} \boldsymbol{R}_s^{(j)}\left(\boldsymbol{x}_f^{(k)}\right) &= \boldsymbol{R}_f\left(\boldsymbol{x}_f^{(k)}\right), \qquad k = 1, 2, \ldots, j-1 \\ \boldsymbol{J}_s^{(j)}\left(\boldsymbol{x}_f^{(j)}\right) &= \boldsymbol{J}_f\left(\boldsymbol{x}_f^{(j)}\right) \end{aligned} \tag{6}$$

where $\boldsymbol{J}_s^{(j)}(\boldsymbol{x}_f^{(j)})$ and $\boldsymbol{J}_f(\boldsymbol{x}_f^{(j)})$ are the Jacobians of the surrogate and fine model at $\boldsymbol{x}_f^{(j)}$, respectively.

The constants $\{\boldsymbol{B}_i, \boldsymbol{c}_i, \alpha_i, \beta_i\}, i = 1, \ldots, m$ are determined in such a way that the alignment (5) holds and the requirements in (6) are satisfied as well as possible (in some sense to be specified). The alignment (5) is satisfied by choosing $\bar{\boldsymbol{x}}_f$ and β_i appropriately. If we let $\bar{\boldsymbol{x}}_f = \boldsymbol{x}_f^{(j)}$, then (5) only depends on the choice of β_i.

Thus, the jth surrogate of response number i is

$$\begin{aligned} R_{s,i}^{(j)}(\boldsymbol{x}_f) = \alpha_i^{(j)}\bigg(&R_{c,i}\Big(\boldsymbol{P}_i^{(j)}(\boldsymbol{x}_f)\Big) \\ &- R_{c,i}(\boldsymbol{P}_i^{(j)}\Big(\boldsymbol{x}_f^{(j)})\Big)\bigg) + \beta_i^{(j)}, \\ &i = 1, \ldots, m \text{ and } j = 0, 1, \ldots \end{aligned} \tag{7}$$

where

$$\boldsymbol{P}_i^{(j)}(\boldsymbol{x}_f) = \boldsymbol{B}_i^{(j)}\boldsymbol{x}_f + \boldsymbol{c}_i^{(j)}. \tag{8}$$

In the first iteration, the mapping parameters $\boldsymbol{B}_i^{(0)} = \mathbf{I}, \boldsymbol{c}_i^{(0)} = 0, \alpha_i^{(0)} = 1$ and $\beta_i^{(0)} = \alpha_i^{(0)} R_{c,i}(\boldsymbol{P}_i^{(0)}(\boldsymbol{x}_f^{(0)}))$ are used, which ensure that $\boldsymbol{R}_s^{(0)}(\boldsymbol{x}_f) = \boldsymbol{R}_c(\boldsymbol{x}_f)$. For $j > 0$, the parameter $\beta^{(j)} = \boldsymbol{R}_f(\boldsymbol{x}_f^{(j)})$ is utilized, which ensures (5).

In summary, the surrogate used in the jth iteration is given by

$$\boldsymbol{R}_s^{(j)}(\boldsymbol{x}_f) = \left[R_{s,1}^{(j)}(\boldsymbol{x}_f) \ldots R_{s,m}^{(j)}(\boldsymbol{x}_f)\right]^T. \tag{9}$$

In each iteration, the surrogate is optimized to find the next iterate by solving

$$\boldsymbol{x}_f^{(j+1)} = \arg\min_{\boldsymbol{x}_f} U\left(\boldsymbol{R}_s^{(j)}(\boldsymbol{x}_f)\right). \tag{10}$$

B. Surface Fitting Approach for Parameter Extraction (PE)

PE is a crucial step in any SM algorithm. In this paper, we employ a surface fitting approach for PE, which involves the minimization of residuals between the surrogate and fine models, and extracting the parameters $\boldsymbol{B}_i^{(j+1)}, \boldsymbol{c}_i^{(j+1)}$, and $\alpha_i^{(j)}, i = 1, \ldots, m$.

Assume $\boldsymbol{x}_f^{(j+1)}$ has been found. We now wish to find the $(j+1)$th set of mapping parameters $\{\alpha_i^{(j+1)}, \boldsymbol{B}_i^{(j+1)}, \boldsymbol{c}_i^{(j+1)}\}$. Since (5) is automatically satisfied by using (7), the aim is to ensure the matching (6). Thus, for finding $\{\alpha_i^{(j+1)}, \boldsymbol{B}_i^{(j+1)}, \boldsymbol{c}_i^{(j+1)}\}$, we aim to minimize the following set of residuals in some sense [6]:

$$\begin{aligned} &\boldsymbol{r}_i^{(j+1)}(\alpha, \boldsymbol{B}, \boldsymbol{c}) \\ &\quad\triangleq \begin{bmatrix} R_{s,i}^{(j+1)}\left(\boldsymbol{x}_f^{(1)}, \alpha, \boldsymbol{B}, \boldsymbol{c}\right) - R_{f,i}\left(\boldsymbol{x}_f^{(1)}\right) \\ \vdots \\ R_{s,i}^{(j+1)}\left(\boldsymbol{x}_f^{(j)}, \alpha, \boldsymbol{B}, \boldsymbol{c}\right) - R_{f,i}\left(\boldsymbol{x}_f^{(j)}\right) \\ \boldsymbol{J}_{s,i}^{(j+1)}\left(\boldsymbol{x}_f^{(j+1)}, \alpha, \boldsymbol{B}, \boldsymbol{c}\right) - \boldsymbol{J}_{f,i}\left(\boldsymbol{x}_f^{(j+1)}\right) \end{bmatrix} \end{aligned} \tag{11}$$

where $\boldsymbol{J}_{f,i}$ and $\boldsymbol{J}_{s,i}$ are the ith columns of $\boldsymbol{J}_f^T$ and $\boldsymbol{J}_s^T$, respectively. The residual (11) is used during the PE optimization process

$$\left\{\alpha_i^{(j+1)}, \boldsymbol{B}_i^{(j+1)}, \boldsymbol{c}_i^{(j+1)}\right\} = \arg\min_{\alpha, \boldsymbol{B}, \boldsymbol{c}} \left\|\boldsymbol{r}_i^{(j+1)}(\alpha, \boldsymbol{B}, \boldsymbol{c})\right\| \tag{12}$$

which extracts the mapping parameters for the ith response, and for iteration $j+1$. Hence, we have the complete set of mapping parameters after m PE optimizations.

V. Proposed SMIS Algorithm

Our proposed algorithm begins with the coarse model as the initial surrogate. The algorithm incorporates explicit SM [1] and OSM [5] to speed up the convergence to the optimal solution.

Step 1) Select a coarse and fine model.
Step 2) Set $j = 0$, and initialize $\boldsymbol{x}_f^{(0)}$.

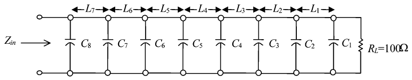


Fig. 4. Seven-section capacitively loaded impedance transformer: "Fine" model.

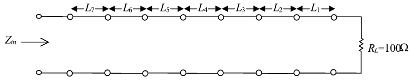


Fig. 5. Seven-section capacitively loaded impedance transformer: "Coarse" model.

TABLE I
Fine Model Capacitances, and the Characteristic Impedances for the Seven-Section Capacitively Loaded Impedance Transformer

Capacitance	Value (pF)	Impedance	Value (Ohm)
C_1	0.025	Z_1	91.9445
C_2	0.025	Z_2	85.5239
C_3	0.025	Z_3	78.1526
C_4	0.025	Z_4	70.7107
C_5	0.025	Z_5	63.9774
C_6	0.025	Z_6	58.4632
C_7	0.025	Z_7	54.3806
C_8	0.025		

Step 3) Optimize the surrogate (9) to find the next iterate $\boldsymbol{x}_f^{(j+1)}$ by solving (10).

Step 4) Evaluate $\boldsymbol{R}_f(\boldsymbol{x}_f^{(j+1)}), \boldsymbol{J}_f(\boldsymbol{x}_f^{(j+1)})$.

Step 5) Terminate if the stopping criteria are satisfied.

Step 6) Update the input and output mapping parameters $\alpha_i^{(j+1)}, \boldsymbol{B}_i^{(j+1)}, \boldsymbol{c}_i^{(j+1)}, i = 1, \ldots, m$ through PE by solving (12).

Step 7) Set $j = j + 1$, and go to Step 3.

As stopping criteria for the algorithm in Step 5, the relative change in the solution vector, or the relative change in the objective function, could be employed.

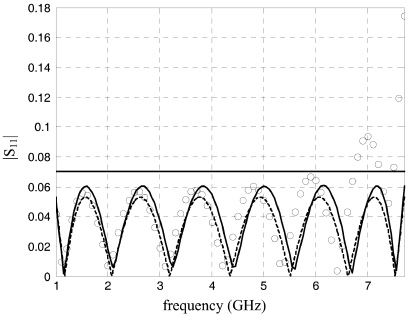


Fig. 6. Seven-section capacitively loaded impedance transformer: optimal coarse-model response (- -), the optimal minimax fine-model response (—), and the fine-model response at the initial solution or at the optimal coarse-model solution (○).

TABLE II
Optimizable Parameter Values of the Seven-Section Capacitively Loaded Impedance Transformer

Parameter	Initial solution (m)	Solution reached by the SMIS algorithm (m)	Solution obtained by direct optimization (m)
L_1	0.01724138	0.01564205	0.01564205
L_2	0.01724138	0.01638347	0.01638347
L_3	0.01724138	0.01677145	0.01677145
L_4	0.01724138	0.01697807	0.01697807
L_5	0.01724138	0.01709879	0.01709879
L_6	0.01724138	0.01723238	0.01723238
L_7	0.01724138	0.01625988	0.01625988

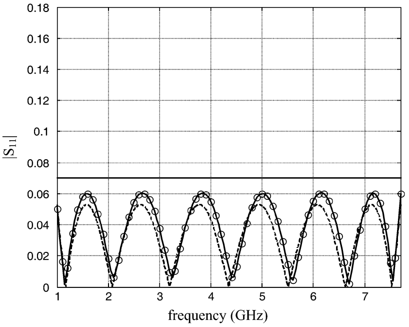


Fig. 7. Seven-section capacitively loaded impedance transformer: optimal coarse-model response (- -), the optimal minimax fine-model response (—), and the fine-model response at the SMIS algorithm solution obtained after five iterations (six fine-model evaluations) (○).

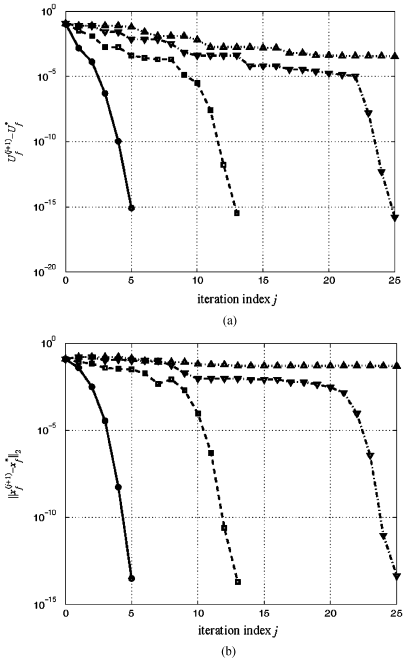
VI. Examples

A. Seven-Section Capacitively Loaded Impedance Transformer

We consider the benchmark synthetic example of a seven-section capacitively loaded impedance transformer [4]. We apply the proposed SMIS algorithm to that example. The objective function is given by $U_f = \max_{1 \le i \le m} |S_{11,i}|$. We consider a "coarse" model as an ideal seven-section transmission line (TL), where the "fine" model is a capacitively loaded TL with capacitors $C_{1,\ldots,8} = 0.025$ pF. The fine and coarse models are shown in Figs. 4 and 5, respectively. Design parameters are normalized lengths $\boldsymbol{x}_f = [L_1\ L_2\ L_3\ L_4\ L_5\ L_6\ L_7]^T$ with respect to the quarter-wave length L_q at the center frequency of 4.35 GHz. Design specifications are

$$|S_{11}| \le 0.07, \qquad \text{for 1 GHz} \le \omega \le 7.7\ \text{GHz} \tag{13}$$

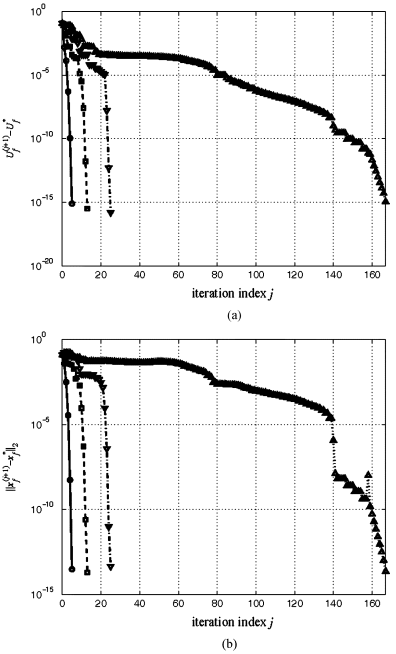
with 68 points per frequency sweep. The characteristic impedances for the transformer are fixed as shown in Table I. The Jacobians of both the coarse and fine models were obtained analytically using the adjoint network method [9]. We solve



(a)

(b)

Fig. 8. (a) First 25 iterations of the difference between the fine-model objective function U_f obtained using the SMIS algorithm (○) and the fine-model objective function at the fine-model minimax solution U_f^* obtained by the Hald–Madsen algorithm (□), the HASM surrogate optimization algorithm using exact gradients (∇), and the HASM surrogate optimization algorithm using the Broyden update (△). (b) The corresponding difference between the designs.



(a)

(b)

Fig. 9. (a) Difference between the fine-model objective function U_f obtained using the SMIS algorithm (○) and the fine-model objective function at the fine-model minimax solution U_f^* obtained by the Hald–Madsen algorithm (□), the HASM surrogate optimization algorithm using exact gradients (∇), and the HASM surrogate optimization algorithm using the Broyden update (△). (b) The corresponding difference between the designs.

the PE problem using the Levenberg–Marquardt algorithm for nonlinear least squares optimization available in the MATLAB Optimization Toolbox.[1]

Optimizing the fine model directly using the gradient-based minimax method of Madsen [13], and Hald and Madsen [14] confirms that the problem has numerous local solutions. Starting from the optimal coarse-model solution (the initial solution for the SMIS method), the Hald–Madsen minimax algorithm needs 13 iterations, or 13 fine-model evaluations, to converge to the fine-model minimax solution. Note that both the direct optimization method of Hald and Madsen and the SMIS approach employ exact gradients.

The fine-model response at the optimal coarse-model solution is shown in Fig. 6. Table II shows the lengths for solutions obtained using the SMIS algorithm and the fine-model direct minimax optimization solution [13], [14]. Our SMIS algorithm took six fine-model evaluations or five iterations to reach the same accurate solution as the Hald–Madsen direct minimax optimization algorithm.

Fig. 7 shows the fine-model response at the SMIS algorithm solution. The difference between the minimax objective function at the optimal minimax fine-model response and the response obtained using the SMIS algorithm is shown in Figs. 8 and 9.

Corresponding results reached by the Hald–Madsen method are shown in Figs. 8 and 9. In these figures, we show the HASM surrogate exploiting exact gradients. The minimax objective function and solution reached by the HASM surrogate optimization approach using the Broyden update [10] are also shown. The four methods converged to the same highly accurate solution.

The optimization methods used for solving (1) and a comparison is shown in Table III. Using the adjoint technique, the SMIS algorithm was able to obtain the same optimum solution as the Hald–Madsen algorithm within an error of 10^{-15} after only five iterations.

[1] MATLAB, ver. 6.1, MathWorks Inc., Natick, MA, 2001.

TABLE III
OPTIMIZATION METHODS USED ON THE SEVEN-SECTION CAPACITIVELY LOADED IMPEDANCE TRANSFORMER

Problem	Method	Number of Iterations	Number of fine model evaluations
(1)	fminimax*	14	153
	HASM	25	26
	Hald-Madsen	13	13
	SMIS	5	6

* The fminimax routine available in the Matlab Optimization Toolbox [13].

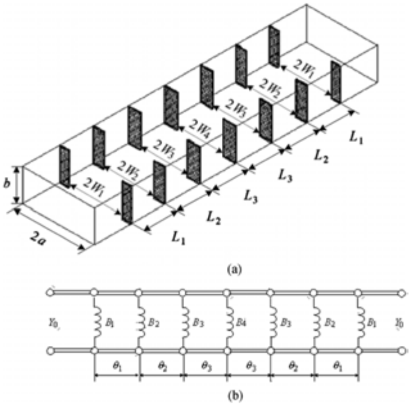


Fig. 10. Six-section H-plane waveguide filter [7]. (a) Physical structure. (b) Coarse model as implemented in MATLAB .

In contrast to SMIS, the standard minimax optimizer available in MATLAB was able to reach the same optimum direct optimization result in 14 iterations (153 fine-model evaluations), while the Hald–Madsen algorithm reached the optimum fine-model solution in 13 iterations (13 fine-model evaluations). The HASM algorithm exploiting exact gradients took 25 iterations (26 fine-model evaluations) to reach the optimum fine-model solution to the same error of 10^{-15}.

The Hald–Madsen algorithm exploits sequential linear programming (SLP) using trust regions, combined with a Newton iteration. The MATLAB minimizer (*fminimax*) exploits a sequential quadratic programming (SQP) method with line searches.

B. Six-Section H-Plane Waveguide Filter

The physical structure of the six-section H-plane waveguide filter is shown in Fig. 10(a) [7]. We simulate the fine model using Agilent High Frequency Structure Simulator (HFSS).[2] The design parameters are the lengths and widths, namely,

$$\boldsymbol{x}_f = \begin{bmatrix} L_1 & L_2 & L_3 & W_1 & W_2 & W_3 & W_4 \end{bmatrix}^T.$$

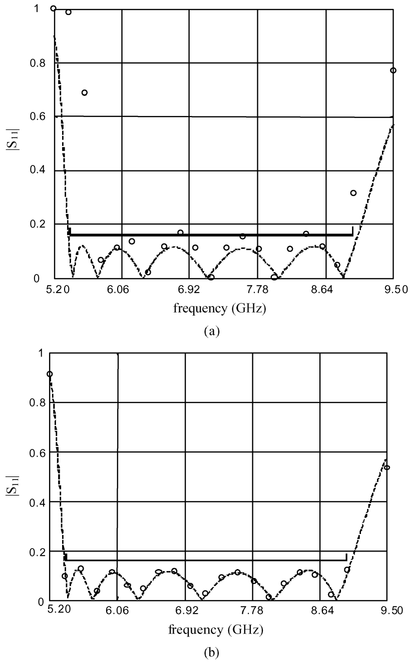


Fig. 11. H-plane filter optimal coarse-model response (—), and the HFSS (fine-model) response: (a) at the initial solution (○) and (b) at the SMIS algorithm solution reached after three iterations (○).

TABLE IV
OPTIMIZABLE PARAMETER VALUES OF THE SIX-SECTION H-PLANE WAVEGUIDE FILTER

Parameter	Initial solution (inches)	Solution reached by the SMIS algorithm (inches)
W_1	0.48583	0.51397
W_2	0.43494	0.47244
W_3	0.40433	0.44501
W_4	0.39796	0.44627
L_1	0.65585	0.63142
L_2	0.65923	0.63922
L_3	0.67666	0.65705

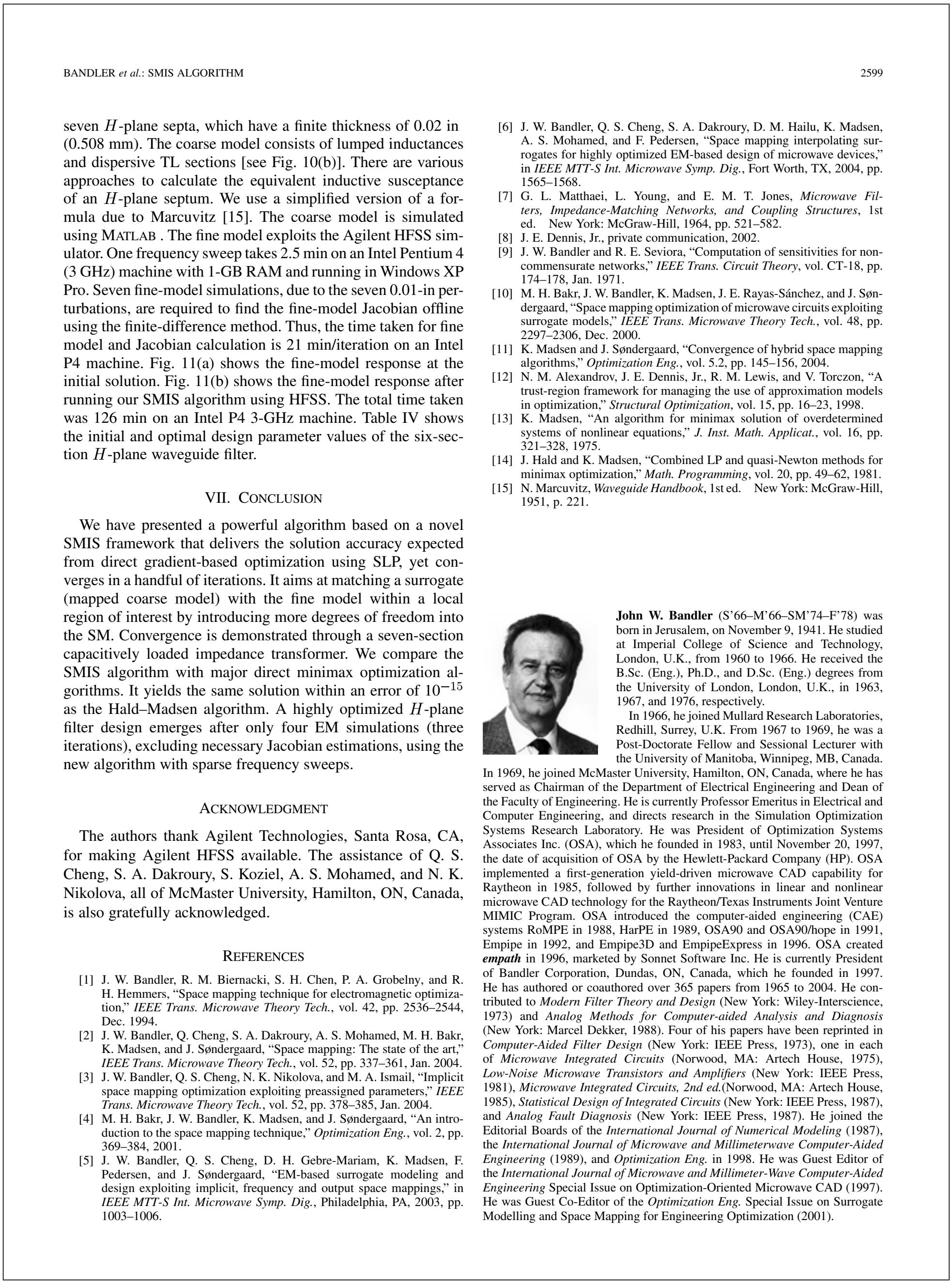
Design specifications are

$$|S_{11}| \le 0.16, \quad \text{for } 5.4 \text{ GHz} \le \omega \le 9.0 \text{ GHz}$$
$$|S_{11}| \ge 0.85, \quad \text{for } \omega \le 5.2 \text{ GHz}$$
$$|S_{11}| \ge 0.5, \quad \text{for } \omega \ge 9.5 \text{ GHz}$$

with 23 points per frequency sweep.

A waveguide with a cross section of 1.372 in × 0.622 in (3.485 cm × 1.58 cm) is used. The six sections are separated by

[2] Agilent HFSS, ver. 5.6, HP EESof, Agilent Technol., Santa Rosa, CA, 2000.



seven H-plane septa, which have a finite thickness of 0.02 in (0.508 mm). The coarse model consists of lumped inductances and dispersive TL sections [see Fig. 10(b)]. There are various approaches to calculate the equivalent inductive susceptance of an H-plane septum. We use a simplified version of a formula due to Marcuvitz [15]. The coarse model is simulated using MATLAB . The fine model exploits the Agilent HFSS simulator. One frequency sweep takes 2.5 min on an Intel Pentium 4 (3 GHz) machine with 1-GB RAM and running in Windows XP Pro. Seven fine-model simulations, due to the seven 0.01-in perturbations, are required to find the fine-model Jacobian offline using the finite-difference method. Thus, the time taken for fine model and Jacobian calculation is 21 min/iteration on an Intel P4 machine. Fig. 11(a) shows the fine-model response at the initial solution. Fig. 11(b) shows the fine-model response after running our SMIS algorithm using HFSS. The total time taken was 126 min on an Intel P4 3-GHz machine. Table IV shows the initial and optimal design parameter values of the six-section H-plane waveguide filter.

VII. Conclusion

We have presented a powerful algorithm based on a novel SMIS framework that delivers the solution accuracy expected from direct gradient-based optimization using SLP, yet converges in a handful of iterations. It aims at matching a surrogate (mapped coarse model) with the fine model within a local region of interest by introducing more degrees of freedom into the SM. Convergence is demonstrated through a seven-section capacitively loaded impedance transformer. We compare the SMIS algorithm with major direct minimax optimization algorithms. It yields the same solution within an error of 10^{-15} as the Hald–Madsen algorithm. A highly optimized H-plane filter design emerges after only four EM simulations (three iterations), excluding necessary Jacobian estimations, using the new algorithm with sparse frequency sweeps.

Acknowledgment

The authors thank Agilent Technologies, Santa Rosa, CA, for making Agilent HFSS available. The assistance of Q. S. Cheng, S. A. Dakroury, S. Koziel, A. S. Mohamed, and N. K. Nikolova, all of McMaster University, Hamilton, ON, Canada, is also gratefully acknowledged.

References

[1] J. W. Bandler, R. M. Biernacki, S. H. Chen, P. A. Grobelny, and R. H. Hemmers, "Space mapping technique for electromagnetic optimization," *IEEE Trans. Microwave Theory Tech.*, vol. 42, pp. 2536–2544, Dec. 1994.

[2] J. W. Bandler, Q. Cheng, S. A. Dakroury, A. S. Mohamed, M. H. Bakr, K. Madsen, and J. Søndergaard, "Space mapping: The state of the art," *IEEE Trans. Microwave Theory Tech.*, vol. 52, pp. 337–361, Jan. 2004.

[3] J. W. Bandler, Q. S. Cheng, N. K. Nikolova, and M. A. Ismail, "Implicit space mapping optimization exploiting preassigned parameters," *IEEE Trans. Microwave Theory Tech.*, vol. 52, pp. 378–385, Jan. 2004.

[4] M. H. Bakr, J. W. Bandler, K. Madsen, and J. Søndergaard, "An introduction to the space mapping technique," *Optimization Eng.*, vol. 2, pp. 369–384, 2001.

[5] J. W. Bandler, Q. S. Cheng, D. H. Gebre-Mariam, K. Madsen, F. Pedersen, and J. Søndergaard, "EM-based surrogate modeling and design exploiting implicit, frequency and output space mappings," in *IEEE MTT-S Int. Microwave Symp. Dig.*, Philadelphia, PA, 2003, pp. 1003–1006.

[6] J. W. Bandler, Q. S. Cheng, S. A. Dakroury, D. M. Hailu, K. Madsen, A. S. Mohamed, and F. Pedersen, "Space mapping interpolating surrogates for highly optimized EM-based design of microwave devices," in *IEEE MTT-S Int. Microwave Symp. Dig.*, Fort Worth, TX, 2004, pp. 1565–1568.

[7] G. L. Matthaei, L. Young, and E. M. T. Jones, *Microwave Filters, Impedance-Matching Networks, and Coupling Structures*, 1st ed. New York: McGraw-Hill, 1964, pp. 521–582.

[8] J. E. Dennis, Jr., private communication, 2002.

[9] J. W. Bandler and R. E. Seviora, "Computation of sensitivities for noncommensurate networks," *IEEE Trans. Circuit Theory*, vol. CT-18, pp. 174–178, Jan. 1971.

[10] M. H. Bakr, J. W. Bandler, K. Madsen, J. E. Rayas-Sánchez, and J. Søndergaard, "Space mapping optimization of microwave circuits exploiting surrogate models," *IEEE Trans. Microwave Theory Tech.*, vol. 48, pp. 2297–2306, Dec. 2000.

[11] K. Madsen and J. Søndergaard, "Convergence of hybrid space mapping algorithms," *Optimization Eng.*, vol. 5.2, pp. 145–156, 2004.

[12] N. M. Alexandrov, J. E. Dennis, Jr., R. M. Lewis, and V. Torczon, "A trust-region framework for managing the use of approximation models in optimization," *Structural Optimization*, vol. 15, pp. 16–23, 1998.

[13] K. Madsen, "An algorithm for minimax solution of overdetermined systems of nonlinear equations," *J. Inst. Math. Applicat.*, vol. 16, pp. 321–328, 1975.

[14] J. Hald and K. Madsen, "Combined LP and quasi-Newton methods for minimax optimization," *Math. Programming*, vol. 20, pp. 49–62, 1981.

[15] N. Marcuvitz, *Waveguide Handbook*, 1st ed. New York: McGraw-Hill, 1951, p. 221.

John W. Bandler (S'66–M'66–SM'74–F'78) was born in Jerusalem, on November 9, 1941. He studied at Imperial College of Science and Technology, London, U.K., from 1960 to 1966. He received the B.Sc. (Eng.), Ph.D., and D.Sc. (Eng.) degrees from the University of London, London, U.K., in 1963, 1967, and 1976, respectively.

In 1966, he joined Mullard Research Laboratories, Redhill, Surrey, U.K. From 1967 to 1969, he was a Post-Doctorate Fellow and Sessional Lecturer with the University of Manitoba, Winnipeg, MB, Canada. In 1969, he joined McMaster University, Hamilton, ON, Canada, where he has served as Chairman of the Department of Electrical Engineering and Dean of the Faculty of Engineering. He is currently Professor Emeritus in Electrical and Computer Engineering, and directs research in the Simulation Optimization Systems Research Laboratory. He was President of Optimization Systems Associates Inc. (OSA), which he founded in 1983, until November 20, 1997, the date of acquisition of OSA by the Hewlett-Packard Company (HP). OSA implemented a first-generation yield-driven microwave CAD capability for Raytheon in 1985, followed by further innovations in linear and nonlinear microwave CAD technology for the Raytheon/Texas Instruments Joint Venture MIMIC Program. OSA introduced the computer-aided engineering (CAE) systems RoMPE in 1988, HarPE in 1989, OSA90 and OSA90/hope in 1991, Empipe in 1992, and Empipe3D and EmpipeExpress in 1996. OSA created ***empath*** in 1996, marketed by Sonnet Software Inc. He is currently President of Bandler Corporation, Dundas, ON, Canada, which he founded in 1997. He has authored or coauthored over 365 papers from 1965 to 2004. He contributed to *Modern Filter Theory and Design* (New York: Wiley-Interscience, 1973) and *Analog Methods for Computer-aided Analysis and Diagnosis* (New York: Marcel Dekker, 1988). Four of his papers have been reprinted in *Computer-Aided Filter Design* (New York: IEEE Press, 1973), one in each of *Microwave Integrated Circuits* (Norwood, MA: Artech House, 1975), *Low-Noise Microwave Transistors and Amplifiers* (New York: IEEE Press, 1981), *Microwave Integrated Circuits, 2nd ed.*(Norwood, MA: Artech House, 1985), *Statistical Design of Integrated Circuits* (New York: IEEE Press, 1987), and *Analog Fault Diagnosis* (New York: IEEE Press, 1987). He joined the Editorial Boards of the *International Journal of Numerical Modeling* (1987), the *International Journal of Microwave and Millimeterwave Computer-Aided Engineering* (1989), and *Optimization Eng.* in 1998. He was Guest Editor of the *International Journal of Microwave and Millimeter-Wave Computer-Aided Engineering* Special Issue on Optimization-Oriented Microwave CAD (1997). He was Guest Co-Editor of the *Optimization Eng.* Special Issue on Surrogate Modelling and Space Mapping for Engineering Optimization (2001).

Dr. Bandler is a Fellow of the Canadian Academy of Engineering, the Royal Society of Canada, the Institution of Electrical Engineers (U.K.), and the Engineering Institute of Canada. He is a member of the Association of Professional Engineers of the Province of Ontario (Canada) and a member of the Massachusetts Institute of Technology (MIT) Electromagnetics Academy. He was an associate editor of the IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES (1969–1974), and has continued serving as a member of the Editorial Board. He was guest editor of the IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES Special Issue on Computer-Oriented Microwave Practices (1974) and guest co-editor of the IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES Special Issue on Process-Oriented Microwave CAD and Modeling (1992). He was guest editor of the IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES Special Issue on Automated Circuit Design Using Electromagnetic Simulators (1997). He is guest co-editor of the IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES Special Issue on Electromagnetics-Based Optimization of Microwave Components and Circuits (2004). He has served as chair of the MTT-1 Technical Committee on Computer-Aided Design. He was the recipient of the 1994 Automatic Radio Frequency Techniques Group (ARFTG) Automated Measurements Career Award and the 2004 Microwave Application Award presented by the IEEE MTT-S.

Daniel M. Hailu (S'99) was born in Winnipeg, MB, Canada, in 1979. He received the B.Eng. degree (with distinction) in computer engineering from McMaster University, Hamilton, ON, Canada, in 2002.

In May 2002, he joined the Simulation Optimization Systems Research Laboratory and the Department of Electrical and Computer Engineering, McMaster University, as a graduate student. His research interests are in CAD and modeling of microwave circuits, EM optimization, SM technology, OSM, device modeling, and CAD methods for antennas.

Mr. Hailu was the recipient of the 2001–2002 Dr. Rudolf de Buda Scholarship for academic achievement and the 2002–2003 Natural Sciences and Engineering Research Council Undergraduate Student Research Award (NSERC USRA).

Kaj Madsen was born in Hjørring, Denmark, in 1943. He received the Cand.Scient. degree in mathematics from the University of Aarhus, Aarhus, Denmark, in 1968, and the Dr.Techn. degree from the Technical University of Denmark (DTU), Lyngby, Denmark, in 1986.

From 1968 to 1988, he was a Lecturer of numerical analysis (aside from the 1973–1974 academic year, when he was with AERE Harwell, Didcot, U.K.). Most of his career has been with the Department for Numerical Analysis, DTU. From 1981 to 1983, he was with the Computer Science Department, Copenhagen University. During the summer of 1978, he visited McMaster University, Hamilton, ON, Canada. In 1988, he became a Full Professor. In 1993, he joined the Department of Mathematical Modelling, DTU, and during 1995–2000, he was Head of that department. In 2000, he took an active part in forming the new Department of Informatics and Mathematical Modelling, DTU, which includes computer science and applied mathematics, and since January 2001, he has headed that department. His primary fields of interest in teaching and research are nonlinear optimization including SM techniques and global optimization, and validated computing using interval analysis.

Dr. Madsen arranged several international workshops on linear programming, parallel algorithms, and surrogate optimization during the 1990s.

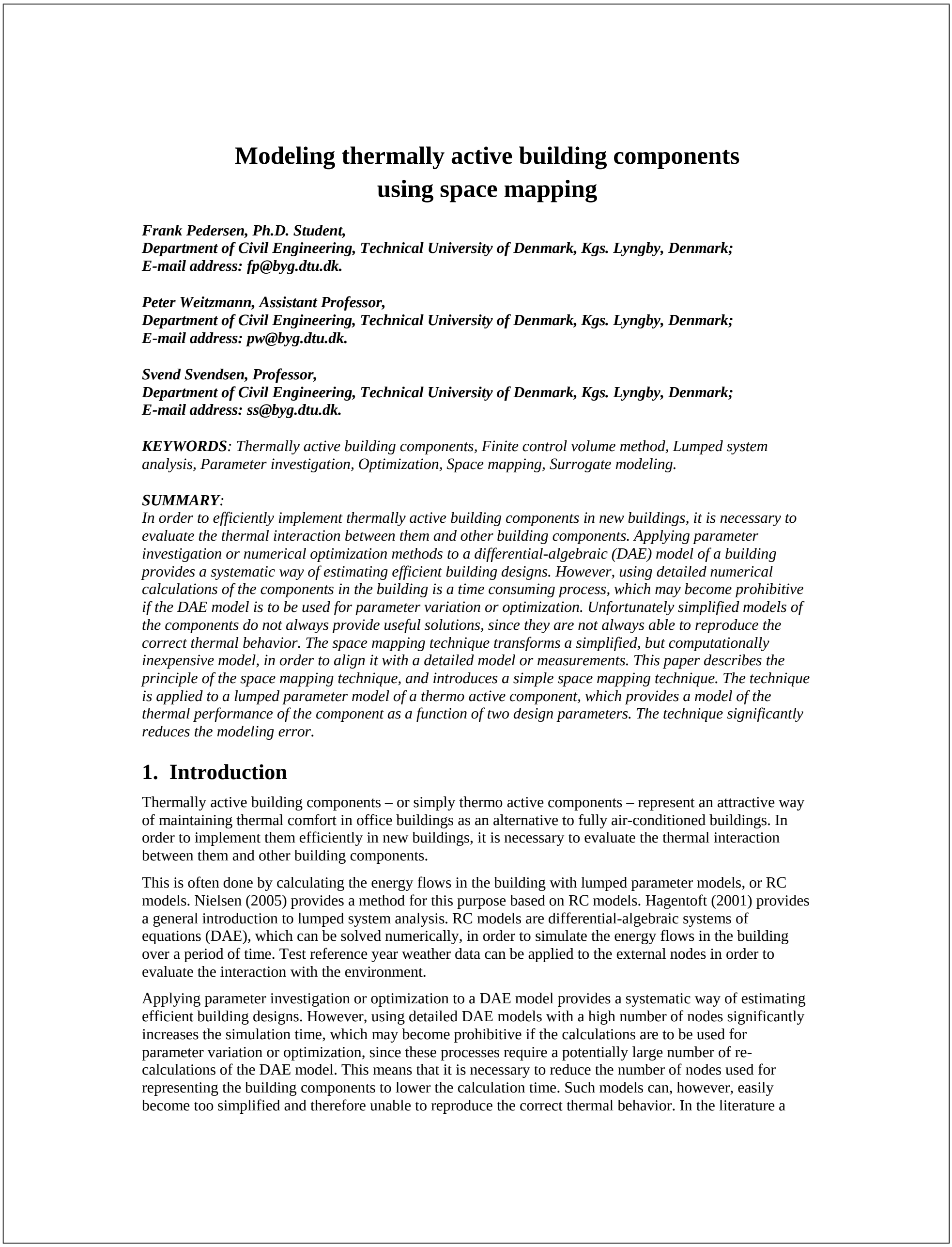
Frank Pedersen was born in Roskilde, Denmark, in 1968. He received the M.Sc. degree in engineering from the Technical University of Denmark, Lyngby, Denmark, in 2001, and is currently working toward the Ph.D. degree in engineering at the Technical University of Denmark.

During 2002, he was a Research Assistant with the Technical University of Denmark. His research concerns the development of nonlinear optimization techniques for design and control problems.

Appendix B

Modeling thermally active building components using space mapping

Authors	Frank Pedersen, Peter Weitzmann and Svend Svendsen
Title	Modeling thermally active building components using space mapping
Part of	Proceedings of the 7th Symposium on Building Physics in the Nordic Countries (ISBN 9979-9174-6-6)
Pages	896-903
Year	2005



Modeling thermally active building components using space mapping

Frank Pedersen, Ph.D. Student,
Department of Civil Engineering, Technical University of Denmark, Kgs. Lyngby, Denmark;
E-mail address: fp@byg.dtu.dk.

Peter Weitzmann, Assistant Professor,
Department of Civil Engineering, Technical University of Denmark, Kgs. Lyngby, Denmark;
E-mail address: pw@byg.dtu.dk.

Svend Svendsen, Professor,
Department of Civil Engineering, Technical University of Denmark, Kgs. Lyngby, Denmark;
E-mail address: ss@byg.dtu.dk.

KEYWORDS*: Thermally active building components, Finite control volume method, Lumped system analysis, Parameter investigation, Optimization, Space mapping, Surrogate modeling.*

SUMMARY*:*
In order to efficiently implement thermally active building components in new buildings, it is necessary to evaluate the thermal interaction between them and other building components. Applying parameter investigation or numerical optimization methods to a differential-algebraic (DAE) model of a building provides a systematic way of estimating efficient building designs. However, using detailed numerical calculations of the components in the building is a time consuming process, which may become prohibitive if the DAE model is to be used for parameter variation or optimization. Unfortunately simplified models of the components do not always provide useful solutions, since they are not always able to reproduce the correct thermal behavior. The space mapping technique transforms a simplified, but computationally inexpensive model, in order to align it with a detailed model or measurements. This paper describes the principle of the space mapping technique, and introduces a simple space mapping technique. The technique is applied to a lumped parameter model of a thermo active component, which provides a model of the thermal performance of the component as a function of two design parameters. The technique significantly reduces the modeling error.

1. Introduction

Thermally active building components – or simply thermo active components – represent an attractive way of maintaining thermal comfort in office buildings as an alternative to fully air-conditioned buildings. In order to implement them efficiently in new buildings, it is necessary to evaluate the thermal interaction between them and other building components.

This is often done by calculating the energy flows in the building with lumped parameter models, or RC models. Nielsen (2005) provides a method for this purpose based on RC models. Hagentoft (2001) provides a general introduction to lumped system analysis. RC models are differential-algebraic systems of equations (DAE), which can be solved numerically, in order to simulate the energy flows in the building over a period of time. Test reference year weather data can be applied to the external nodes in order to evaluate the interaction with the environment.

Applying parameter investigation or optimization to a DAE model provides a systematic way of estimating efficient building designs. However, using detailed DAE models with a high number of nodes significantly increases the simulation time, which may become prohibitive if the calculations are to be used for parameter variation or optimization, since these processes require a potentially large number of re-calculations of the DAE model. This means that it is necessary to reduce the number of nodes used for representing the building components to lower the calculation time. Such models can, however, easily become too simplified and therefore unable to reproduce the correct thermal behavior. In the literature a large number of articles provide comparisons of detailed and simplified models, see for instance Weber and Jóhannesson (2005), Davies et. al. (2001) and Schmidt (2004).

The field of microwave electronics has encountered similar modeling problems. Making reliable calculations of microwave devices require solutions to the governing equations obtained with detailed finite element methods. Parameter investigation or optimization thus becomes a very time consuming task. Using lumped parameter models does not always give useful solutions, since they are not always able to reproduce the correct response of the microwave devices.

These observations motivate the development of the space mapping technique, which aims at transforming a simplified, but computationally inexpensive model, in order to align it with a detailed model or measurements. The purpose is to generate a surrogate model that can be used as replacement for the detailed model. The technique was first described by Bandler et. al. (1994), and since then many variations have been developed, for the purpose of optimization as well as for modeling. An introduction to the technique is given by Bakr et. al. (2001), and the contributions to the development of the technique are reviewed by Bandler et. al. (2004a).

This paper describes the modeling aspects of the space mapping technique. A space mapping modeling technique useful for enhancing the accuracy of simplified models is described. The technique is demonstrated on a thermo active component, where the aim is to model the thermal performance of the component as a function of a set of design parameters.

2. Thermo active building components

In recent years the so-called thermo active components, which are based on embedded pipes in the building structure have been introduced (Meierhans (1993), Meierhans (1996), De Carli and Olesen (2002), Olesen (2000)) as an alternative to mechanical cooling systems in office buildings. In this paper thermo active components are defined as deck floors with embedded pipes in multi-storey buildings. An example of a thermo active deck based on a pre-fabricated hollow deck is shown in Fig. 1.

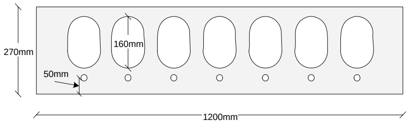


FIG. 1: Example of thermo active deck with integrated pipes and air cavities. The pipes in this figure are placed 50mm above the lower (ceiling) surface of the deck and the deck has a total height of 270mm.

The functionality and operational capabilities of the thermo active components is influenced by a long list of factors including

- Geometry of the component.
- Material properties.
- Environmental parameters (boundary conditions in room and outdoor).
- Operational parameters (control strategy and operation of fluid system).

The performance of the thermo active deck depends on the combination of these parameters.

3. Modeling the performance of thermo-active components

When performing parameter investigation or optimization, it is often convenient to make mathematical models of some quantifiable measure for the performance of the component, which ideally should be as large or small as possible, e.g. the net heat flux or the difference between the desired and actual temperatures. The performance of the component depends on design parameters, as well as control parameters. The set of parameters investigated using parameter variation or optimization is referred to as *model parameters*.

Some space mapping techniques also exploit a second set of parameters, e.g. physical constants or material properties that the designer has little or no control over. These parameters are referred to as the *preassigned parameters*.

The performance of the component also depends on how it is measured. For instance, measuring the transient heat flux through a component gives time-dependent values. A vector-valued performance measure can therefore be formed by sampling the heat flux a finite number of times within a given time interval. The vector-valued performance measure for the component depends on how it is sampled. The parameter that is varied when sampling the performance is referred to as the *sample parameter*, e.g. time, frequency or position.

The vector-valued performance measure can be reduced to a scalar value by applying a suitable merit; e.g. a quadrature rule, which gives an estimate of the total heat flux in a specified time interval, when applied to time sampled values of the heat flux.

The vector-valued performance measure of a component can thus be modeled as a function $f:\mathbb{R}^n\times\mathbb{R}^{\hat{n}}\times\mathbb{R}^m\to\mathbb{R}^m$, with the following arguments:

- Model parameters $x\in\mathbb{R}^n$.
- Preassigned parameters $\hat{x}\in\mathbb{R}^{\hat{n}}$.
- Sample parameters $t\in\mathbb{R}^m$, e.g. points in time, frequency points or positions.

The function value $f(x,\hat{x},t)$ is referred to as the *response vector*.

The merit can be modeled as a function $H:\mathbb{R}^m\to\mathbb{R}$. The scalar performance measure of the component can thus be modeled as a function $F:\mathbb{R}^n\times\mathbb{R}^{\hat{n}}\times\mathbb{R}^m\to\mathbb{R}$, defined as the composed mapping $F=H\circ f$.

The main purpose of f is to model the performance of the component as a function of the model parameters. When using the model for parameter investigation or optimization, it is necessary to calculate them for a sequence $x^{(1)}$, $x^{(2)}$, $\ldots$ of model parameters. These calculations are performed with the same values of $\hat{x}$ and t. Whenever possible, the notation is therefore simplified by omitting the dependency on constant parameters, so e.g. the notation $f(x)$ will be used for $f(x,\hat{x},t)$. These concerns also apply to F.

4. The space mapping technique

Providing reliable but computationally inexpensive mathematical models of a given system is a classical engineering challenge. Simplifying the physics of the system (e.g. disregarding some of the physical properties of the system), or simplifying the numerical methods used for solving the governing equations (e.g. coarsening the computational grid), provides computationally inexpensive models, but often at the expense of the accuracy. However, by transforming such a model, it is often possible to align it, to some degree of accuracy, with a more accurate model through a data fitting process. The space mapping technique provides a systematic way of addressing this issue.

4.1 The principle of space mapping

The space mapping technique requires the following two models of the same system to be supplied by the user:

- A computationally expensive model, referred to as *the fine model*, which accurately models the physical properties of the system. The fine model is represented by the mapping $f:\mathbb{R}^n\to\mathbb{R}^m$ with the argument $x\in\mathbb{R}^n$ (fine model parameters).
- A computationally inexpensive, and less accurate model, referred to as *the coarse model*. The coarse model is represented by the mapping $c:\mathbb{R}^n\times\mathbb{R}^{\hat{n}}\times\mathbb{R}^m\to\mathbb{R}^m$, with the arguments $z\in\mathbb{R}^n$ (coarse model parameters), $\hat{z}\in\mathbb{R}^{\hat{n}}$ (coarse model preassigned parameters) and $q\in\mathbb{R}^m$ (coarse model sample points).

Also a merit $H : \mathbb{R}^m \to \mathbb{R}$ must be specified, which is used for reducing the vector-valued models f and c to scalar values.

The space mapping technique generates a surrogate model $s : \mathbb{R}^n \to \mathbb{R}^m$, by shifting, scaling, and otherwise transforming the coarse model, in order to align it with the fine model. The aim of the space mapping surrogate is to accurately predict the response of the fine model within some region of interest $\Omega \subseteq \mathbb{R}^n$. The surrogate model is generated using one or more of the following approaches:

- Transforming the model parameters.
- Transforming the sample points.
- Transforming the response of the coarse model.
- Modifying a set of coarse model preassigned parameters.

The transformations are accomplished using mappings between the respective parameter spaces, hence the name *space mapping*. The space mapping surrogate is in the literature also referred to as the *mapped coarse model.*

The purpose of Fig. 2 is to illustrate the general principle behind the space mapping techniques developed over the years. All space mapping techniques use one or more of the approaches shown, but usually not all of them.

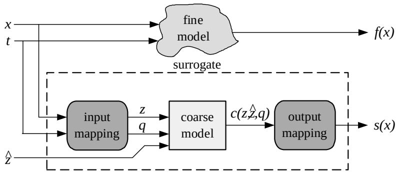


FIG. 2: The approaches used by space mapping techniques to align the coarse model with the fine model.

Fig. 2 does not capture all details of all techniques. Some techniques use a space mapping surrogate only as part of a larger setup. For instance the hybrid space mapping optimization technique, described by Bakr et. al. (2001), Bakr et. al. (2000), and by Madsen and Søndergaard (2004), uses a linear combination of a space mapping surrogate and a linear model of the fine model as the actual surrogate.

Most space mapping methods use the same input and output mappings for all sample points. It is also possible to use different mappings for each sample point, which is utilized by the general space mapping tableau approach by Bandler et. al. (2001), and by the space mapping interpolating surrogate method by Bandler et. al.(2004b).

In the field of microwave electronics, the response vector is usually obtained by sampling the performance of a device for varying frequencies, i.e. frequency is used as sample parameter. For this reason, space mapping techniques that employ mapping of the sample points are referred to as *frequency space mapping techniques.*

4.2 A simple space mapping technique

The proposed space mapping modeling technique uses linear input and output mappings, and only maps model parameters and the response vector. The aim is to align the surrogate model with a set of user provided fine model calculations $D = \{(x^{(1)}, f^{(1)}), \ldots, (x^{(n_d)}, f^{(n_d)})\}$, with $f^{(i)} = f(x^{(i)})$. The dataset consists of $M = n_d \cdot m$ performance calculations of the fine model. The surrogate model used by the technique is the composed mapping $s = o \circ c \circ p$, where the input mapping $p : \mathbb{R}^n \to \mathbb{R}^n$ is defined as

$$p(x) = B \cdot x + d \ , \qquad (1)$$

with $B \in \mathbb{R}^{n\times n}$ and $d \in \mathbb{R}^{n}$. The output mapping $o:\mathbb{R}^{m} \to \mathbb{R}^{m}$ is defined as

$$o(y) = \operatorname{diag}\{a\}\cdot y + b\,, \tag{2}$$

where $\operatorname{diag}\{a\} \in \mathbb{R}^{m\times m}$ is a diagonal matrix with the vector $a \in \mathbb{R}^{m}$ on the diagonal, and where $b \in \mathbb{R}^{m}$. Substituting the definition of the surrogate with (1) and (2) gives

$$s(x) = \operatorname{diag}\{a\}\cdot c\left(B\cdot x + d\right) + b\,. \tag{3}$$

The mapping parameters that needs to be estimated in order to align the surrogate model with the fine model calculations are B, d, a and b, a total of $N = n^2 + n + 2m$ unknown parameters. These parameters are estimated using the function $r:\mathbb{R}^{N} \to \mathbb{R}^{M}$, which is defined as the residuals between the surrogate model and the fine model calculations in the set D, as a function of the mapping parameters:

$$r\left(B,d,a,b\right) = \begin{bmatrix} s(x^{(1)}) - f^{(1)} \\ \vdots \\ s(x^{(n_d)}) - f^{(n_d)} \end{bmatrix} \tag{4}$$

The residual vector (4) can be reduced to a scalar value using one of the vector norms $\|y\|_1 = \sum_{i=1}^{m}|y_i|$ or $\|y\|_2 = \sum_{i=1}^{m} y_i^2$, which gives the scalar measure $R:\mathbb{R}^{N} \to \mathbb{R}$ for the size of the residual vector:

$$R\left(B,d,a,b\right) = \left\| r\left(B,d,a,b\right)\right\|. \tag{5}$$

The mapping parameters that give the smallest residual can thereby be estimated by solving the following data fitting problem:

$$\begin{aligned}&\text{minimize } R\left(B,d,a,b\right)\\ &\text{with respect to } B,\ d,\ a \text{ and } b.\end{aligned} \tag{6}$$

The data fitting problem (6) can be solved using numerical optimization methods, such as the function “lsqnonlin” that comes with Matlab® Optimization Toolbox (2004). The mapping parameters that solve (6) are denoted B^*, d^*, a^* and b^*, respectively.

5. Numerical results

The proposed space mapping method is applied to a thermo-active component as shown in Fig. 1. The purpose is to generate a space mapping surrogate that can be used for modeling the transient heat flux through the component as a function of the height of the pipe above the ceiling surface, referred to as x_1, and the resistance of the floor covering, referred to as x_2.

5.1 The fine and coarse models

The fine model uses the Finite Control Volume (FCV) method, which estimates a numerical solution to a PDE. The domain is a two-dimensional model of the deck shown in Fig. 1, with a numerical mesh of 1200 nodal points. The cavities of the hollow deck are modeled as air with an equivalent thermal resistance. The sides of the deck are adiabatic, and the pipes are included as hydronic pipes. The simulation is based on typical physical material properties of concrete used in thermo active components. A detailed description of the model can be found in Weitzmann (2004), based on e.g. Patankar (1980).

The coarse model is a lumped thermal network shown in Fig. 3. Two nodal points are used for the internal conditions in the floor construction and one for each of the upper and lower surfaces. The internal nodal points have a thermal capacity while it is set to zero for the surface nodal points. This means that effectively there are only two nodal points, as the surface points are just algebraic relations, which can be calculated by the known surrounding temperatures. Therefore only two ODEs need to be solved.

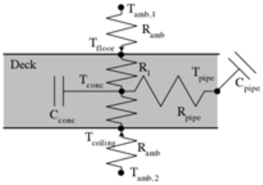


FIG. 3: Lumped thermal network used as the coarse model.

Both models use the following boundary conditions:

- The fluid in the pipe is given as a sinusoidal temperature variation with 24 hour period. The average is 15°C and an amplitude of 3K. No temperature drop between supply and return is included.
- A constant ambient temperature of 20°C is applied to the floor and ceiling surface of the thermo active component along with a combined constant radiant and convective boundary condition.

Both models sample the transient heat flux through the floor above the component and the ceiling below the component. The heat flux is sampled a total of 48 times in the time interval from $t = 49\text{h}$ to $t = 96\text{h}$, and the sampled values are organized in a vector with 96 elements.

5.2 The data fitting problem

In order to generate the space mapping surrogate, a set of fine model calculations is required. The fine model is calculated in the 25 design points obtained by combining the following parameter values:

$$\begin{aligned} &x_1 \in \{0.05,\quad 0.075,\quad 0.1,\quad 0.125,\quad 0.15\}\left[\text{m}\right] \\ &x_2 \in \{0,\quad 0.1,\quad 0.2,\quad 0.3,\quad 0.4\}\left[\text{m}^2\text{K/W}\right] \end{aligned} \tag{7}$$

For all these designs, the heat flux through the floor above the thermo-active component and through the ceiling below is sampled in the specified time interval.

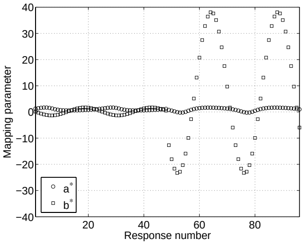


FIG. 4: The output mapping parameters.

Since $n = 2$, $m = 96$ and $n_d = 25$, the data fitting problem (6) consists of $M = 2400$ equations and $N = 198$ unknown mapping parameters, i.e. (6) is overdetermined, and overfitting of the surrogate does not seem likely to occur. Solving (6) using the vector norm $\|y\|_2 = \sum_{i=1}^{m} y_i^2$ gives the following input mapping parameters (rounded to 6 decimals):

$$\begin{aligned} B^* &= \begin{bmatrix} 1.253477\cdot 10^{0} & -8.155003\cdot 10^{-4} \\ -4.474478\cdot 10^{-1} & 8.027254\cdot 10^{-1} \end{bmatrix} \\ d^* &= \begin{bmatrix} -3.647651\cdot 10^{-1} \\ -1.949180\cdot 10^{-1} \end{bmatrix} \end{aligned} \qquad (8)$$

The output mapping parameters found by solving (6) are shown in Fig. 4.

5.3 Evaluation of the space mapping surrogate

In Fig. 5 (left) are shown calculations of the heat flux performed with the fine, coarse and surrogate models for design A, given by $x = [0.1\text{m} \quad 0.2\,\text{m}^2\text{K}/\text{W}]$. The coarse model is significantly misaligned compared to the fine model, whereas the space mapping surrogate quite accurately predicts the fine model response. The modeling error $e(x) = f(x) - s(x)$ for the surrogate model, calculated for all 25 designs is shown in Fig. 5 (right). The absolute value of all errors is below 1 W/m^2 for all 25 designs.

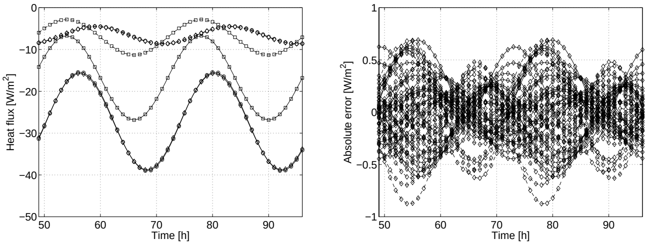


FIG. 5: Solid lines: calculations for the ceiling below the thermo-active component. Dashed lines: calculations for the floor above the thermo-active component. ○: calculations with the fine model. □: calculations with the coarse model. ◇: calculations with the space mapping surrogate model. Left: Heat flux calculated for design A. Right: The error when calculating the heat flux with the surrogate model.

6. Conclusion

The basic thermal and physical properties of thermo active components are described, and it is described how the performance of thermo active components can be expressed as a function of model parameters, preassigned parameters, and sample parameters.

The principle of the space mapping modeling technique is described, and a simple space mapping technique using linear input and output mappings is described. The technique is applied to a lumped parameter model of a thermo active component, where the aim is to model the thermal performance of the component as a function of the height of the pipe above the ceiling surface and the resistance of the floor covering. The technique provides a space mapping surrogate model with a modeling error less than 1 W/m^2 for all designs used in the data fitting problem.

7. Acknowledgments

The following persons have provided many useful comments and suggestions, that have improved the quality of the paper: J.W. Bandler, Q.S. Cheng, A.S. Mohamed and D.M. Hailu, all of McMaster University, Hamilton, ON, Canada, and Kaj Madsen of Technical University of Denmark, Kgs. Lyngby, Denmark. Their contributions are gratefully acknowledged.

8. References

M.H. Bakr, J.W. Bandler, K. Madsen, J.E. Rayas-Sánchez and J. Søndergaard (2000), Space mapping optimization of microwave circuits exploiting surrogate models, IEEE Trans. Microwave Theory Tech., vol. 48, no. 12, pp. 2297-2306.

M.H. Bakr, J.W. Bandler, K. Madsen and J. Søndergaard (2001), An introduction to the space mapping technique, Optimization and Engineering, vol. 2, no. 4, pp. 369-384.

J.W. Bandler, R.M. Biernacki, S.H. Chen, P.A. Grobelny and R.H. Hemmers (1994), Space mapping technique for electromagnetic optimization, IEEE Trans. Microwave Theory Tech., vol. 42, no. 12, pp. 2536-2544.

J.W. Bandler, N. Georgieva, M.A. Ismail, J.E. Rayas-Sánchez and Q.J. Zhang (2001), A generalized space mapping tableau approach to device modeling, IEEE Trans. Microwave Theory Tech., vol. 49, no. 1, pp. 67-79.

J.W. Bandler, Q. Cheng, S.A. Dakroury, A.S. Mohamed, M.H. Bakr, K. Madsen and J. Søndergaard (2004a), Space mapping: the state of the art, IEEE Trans. Microwave Theory Tech., vol. 52, no. 1, pp. 337-361.

J.W. Bandler, D.M. Hailu, K. Madsen and F. Pedersen (2004b), A space mapping interpolating surrogate algorithm for highly optimized EM-based design of microwave devices, IEEE Trans. Microwave Theory Tech., vol. 52, no. 11, pp. 2593-2600.

M. De Carli and B.W. Olesen (2002), Field measurements of operative temperatures in buildings heated or cooled by embedded water-based radiant systems, ASHRAE Transactions, vol. 108, no. 2, pp. 714-725.

M. Davies, S. Zoras and M.H. Adjali (2001), Improving the efficiency of the numerical modelling of built environment earth-contact heat transfers, Applied Energy, vol. 68, pp. 1-42.

C.E. Hagentoft (2001), Introduction to building physics, Studentlitteratur, Box 141, 221 00 Lund, Sweden.

K. Madsen and J. Søndergaard (2004), Convergence of a hybrid space mapping algorithm, Optimization and Engineering, vol. 5, no. 2, pp. 145-156.

Matlab® 7.0 and Matlab® Optimization Toolbox 3.0 (2004), The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098, USA.

R.A. Meierhans (1993), Slab cooling and earth coupling, ASHRAE Transactions, vol. 99, pp. 511-518.

R.A. Meierhans (1996), Room air conditioning by means of overnight cooling of the concrete ceiling, ASHRAE Transactions, vol. 102, no. 1, pp. 693-697.

T.R. Nielsen (2005), Simple tool to evaluate energy demand and indoor environment in the early stages of building design, Solar Energy, vol. 78, pp. 73-83.

B.W. Olesen (2000), Cooling and heating of buildings by activating the thermal mass with embedded hydronic pipe systems, in: Proceedings of CIBSE/ASHRAE joint conference “20/20 Vision”, Dublin, September 2000.

S. V. Patankar (1980), Numerical heat transfer and fluid flow, Hemisphere, New York.

D. Schmidt (2004), Methodology for the modelling of thermally activated building components in low exergy design, Doctoral Thesis, KTH Civil and Architectural Engineering, Stockholm, Sweden.

T. Weber and G. Jóhannesson (2005), An optimized RC-network for thermally activated building components, Build. Environ., vol. 40, pp. 1-14.

P. Weitzmann (2004), Modelling building integrated heating and cooling systems, Ph.D. Thesis, Department of Civil Engineering, Technical University of Denmark.

Appendix C

Test problems

This appendix concerns 15 problems used for testing the algorithms described in Chapter 5. The aim is to evaluate the performance of the algorithms on problems with a varying number of active inequality, equality and domain constraints. The number of parameters is $n = 2$ for all test problems.

All problems are defined such that they have the following solution:

$$x^* = [2, 0]^\top \tag{C.1}$$

The Rosenbrock function [59] is used as objective function for all test problems:

$$f_1(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \tag{C.2}$$

The following functions are used as inequality, equality or domain constraint functions:

$$\begin{array}{rcl} c_1(x) & = & -\frac{1}{2}x_2^2 + x_1 - 2x_2 - 2 \\ c_2(x) & = & -\frac{1}{2}x_2^2 + x_1 - 2 \\ c_3(x) & = & -x_1^2 - x_2^2 + 10x_1 + 8x_2 - 16 \\ c_4(x) & = & -x_1x_2 + 2x_1 + x_2 - 4 \\ c_5(x) & = & \frac{1}{\alpha_3}\big(-\frac{1}{4}\alpha_1^2x_1^2 - \frac{1}{4}\alpha_2^2x_2^2 - \frac{1}{2}\alpha_1\alpha_2x_1x_2 \ldots \\ & & \quad + \left(\alpha_1^2 + \sqrt{\alpha_3}\alpha_2\right) x_1 + \alpha_1 \left(\alpha_2 - \sqrt{\alpha_3}\right) x_2 - \alpha_1^2 - 2\sqrt{\alpha_3}\alpha_2 \big) \end{array} \tag{C.3}$$

The parameters used for defining c_5 are

$$\begin{array}{rcl} \alpha_1 & = & 800 \\ \alpha_2 & = & 3202 \\ \alpha_3 & = & \alpha_1^2 + \alpha_2^2 \end{array} \tag{C.4}$$

The function c_5 is defined in such a way that $\nabla c_5(x^*) = \nabla f_1(x^*)$, which ensures that the first order optimality conditions for the test problems TP_1, TP_4 and TP_8 are satisfied by x^*.

The test problems, denoted $\mathrm{TP}_1 \ldots \mathrm{TP}_{15}$, are defined in Table C.1, and are illustrated in Figures C.1 to C.8.

Problem	Domain constraints	Equality constraints	Inequality constraints
TP$_1$	-	-	c_5
TP$_2$	-	-	c_1, c_5
TP$_3$	-	-	c_1, c_2, c_5
TP$_4$	-	c_5	-
TP$_5$	-	c_3	c_4
TP$_6$	-	c_3	c_1, c_4
TP$_7$	-	c_3	c_1, c_2, c_4
TP$_8$	c_5	-	-
TP$_9$	c_5	-	c_1
TP$_{10}$	c_5	-	c_1, c_4
TP$_{11}$	c_5	-	c_1, c_2, c_4
TP$_{12}$	c_5	c_3	-
TP$_{13}$	c_5	c_3	c_1
TP$_{14}$	c_5	c_3	c_1, c_4
TP$_{15}$	c_5	c_3	c_1, c_2, c_4

Table C.1: The 15 test problems.

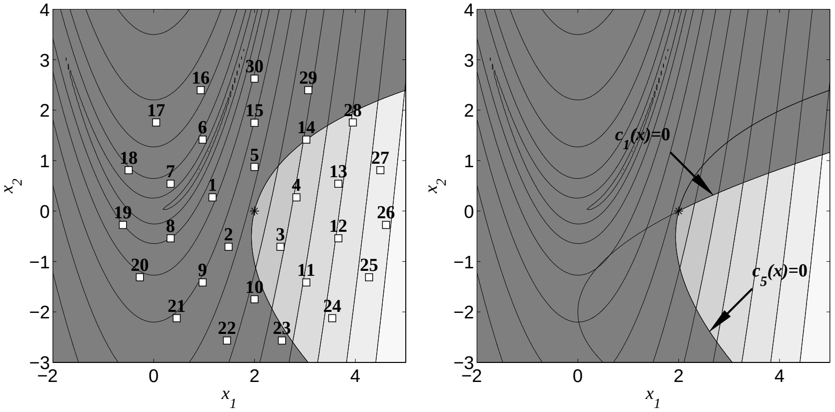


Figure C.1: Left: TP$_1$. Right: TP$_2$.

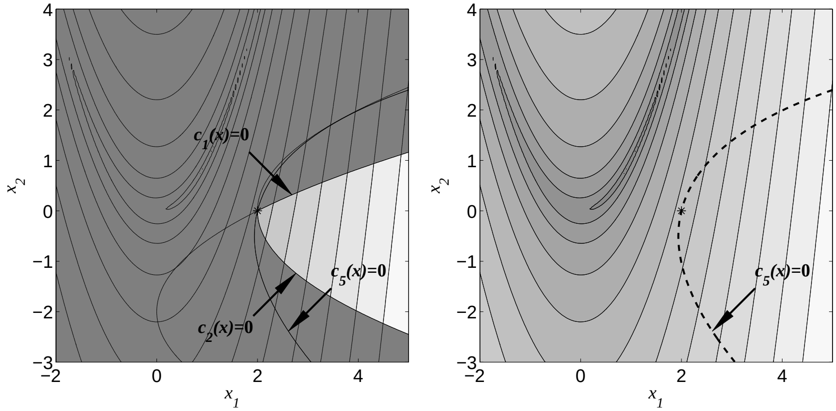


Figure C.2: Left: TP$_3$. Right: TP$_4$.

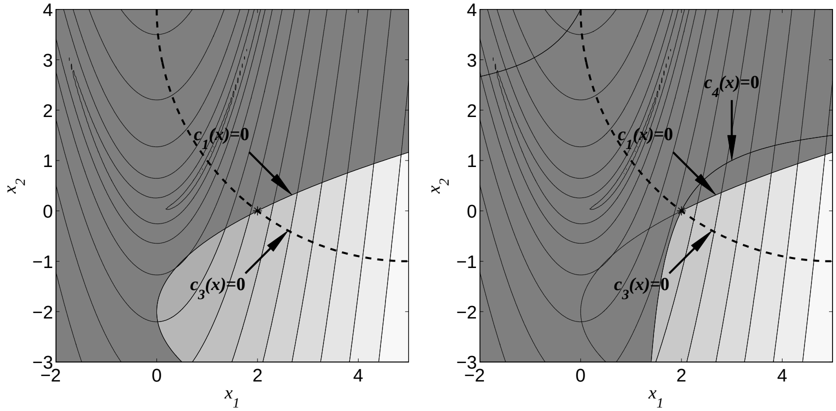


Figure C.3: Left: TP$_5$. Right: TP$_6$.

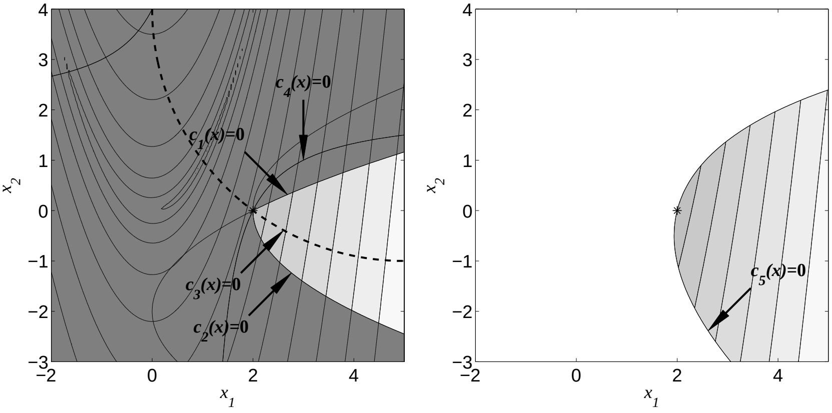


Figure C.4: Left: TP_7. Right: TP_8.

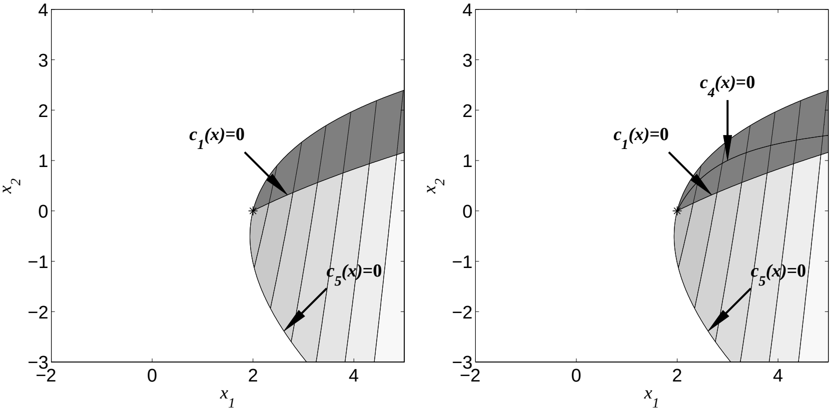


Figure C.5: Left: TP_9. Right: TP_{10}.

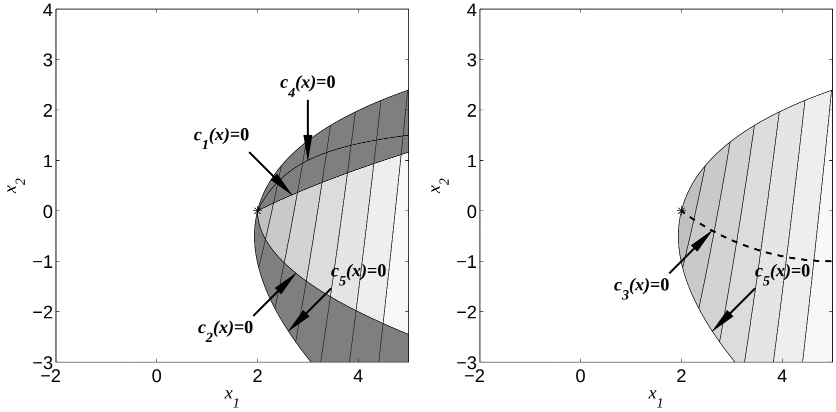


Figure C.6: Left: TP_{11}. Right: TP_{12}.

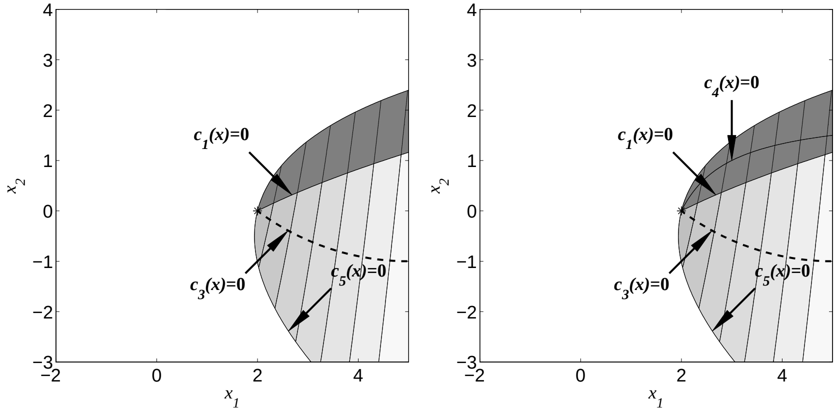


Figure C.7: Left: TP_{13}. Right: TP_{14}.

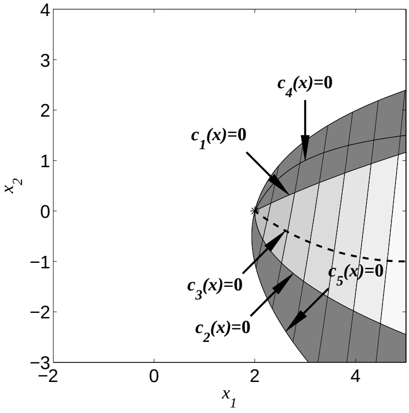


Figure C.8: TP_{15}.

Appendix D

Constant parameters

Parameter	Value	Description
A_i	$2000\,\mathrm{m}^2$	Internal floor area of the building
d_{int}	$0.10\,\mathrm{m}$	Thickness of internal wall
h_{ext}	$3.00\,\mathrm{m}$	External height of floors
C_{tot}	$4{\cdot}10^7$ J/K	Total heat capacity of building contents[1]
Ψ_{ww}	$0.06\,\mathrm{W/mK}$	Linear thermal transmittance for the thermal interaction between the windows and the external walls
Ψ_{fw}	$0.40\,\mathrm{W/mK}$	Linear thermal transmittance for the thermal interaction between the foundation and the external walls
ϱ_r	$7870\,\mathrm{kg/m}^3$	Density of reinforcement rods
c_w	$4.32 \cdot 10^5\,\mathrm{J/m^2K}$	Specific effective heat capacity of constructions
ΔT_{hl}	32°C	Design temperature difference used for calculating the heat loss through the building envelope

Table D.1: Constant parameters representing general properties of the simplified building.

[1] Corresponds to 20 kg of furniture per m^2 of internal floor area, with a specific heat capacity of 1 kJ/kgK.

Constant parameters

Parameter	Value	Description
Latitude	55.4°N	Corresponds to Copenhagen, Denmark
Longitude	12.19°E	
Time meridian	15°	Longitude for the local time zone
Albedo	0.2	Percentage of the solar energy that is reflected from the surroundings
Weather data		Danish design reference year[2]
Orientation	90°	Orientation of the main axis of the building relative to due south

Table D.2: Constant parameters regarding the position, time zone and the surroundings of the building.

Parameter	Value	Description
h_f	0.90 m	Height of the foundation
d_f	0.60 m	Width of the foundation
$n_{f,m}$	2	Number of reinforcement meshes
$d_{f,m}$	0.25 m	Distance between rods in reinforcement mesh
$d_{f,r}$	0.025 m	Diameter of reinforcement rods
τ_f	30 MPa	Compressive strength of concrete

Table D.3: Constant parameters for the annular foundation.

Parameter	Value	Description
h_{roof}	0.50 m	Height of structural layer
$d_{r,u}$	0.50 m	Thickness of the uninsulated layer
$\lambda_{r,u}$	0.200 W/mK	Thermal conductivity of the uninsulated layer
$\lambda_{r,i}$	0.039 W/mK	Thermal conductivity of the insulated layer
$n_{roof,m}$	2	Number of reinforcement meshes
$d_{roof,m}$	0.25 m	Distance between rods in reinforcement mesh
$d_{roof,r}$	0.025 m	Diameter of reinforcement rods
τ_{roof}	30 MPa	Compressive strength of concrete

Table D.4: Constant parameters for the roof construction.

[2] Provided with the BuildingCalc program described by Nielsen [48].

Parameter	Value	Description
$d_{w,u}$	0.21 m	Thickness of the uninsulated layer
$\lambda_{w,u}$	0.310 W/mK	Thermal conductivity of the uninsulated layer
$\lambda_{w,i}$	0.039 W/mK	Thermal conductivity of the insulated layer
$n_{wall,m}$	2	Number of reinforcement meshes
$d_{wall,m}$	0.25 m	Distance between rods in reinforcement mesh
$d_{wall,r}$	0.025 m	Diameter of reinforcement rods
τ_{wall}	30 MPa	Compressive strength of concrete

Table D.5: Constant parameters for the external walls.

Parameter	Value	Description
$d_{g,u}$	0.189 m	Thickness of the uninsulated layer
$\lambda_{g,u}$	0.454 W/mK	Thermal conductivity of the uninsulated layer
$\lambda_{g,i}$	0.039 W/mK	Thermal conductivity of the insulated layer
d_{cb}	0.30 m	Thickness of the capillary-breaking layer
d_{ws}	0.05 m	Thickness of the wearing surface
τ_{gs}	30 MPa	Compressive strength of concrete used for ground slab
τ_{ws}	30 MPa	Compressive strength of concrete used for wearing surface

Table D.6: Constant parameters for the ground slab.

Parameter	Value	Description
h_{deck}	0.50 m	Height of concrete decks
$n_{deck,m}$	2	Number of reinforcement meshes
$d_{deck,m}$	0.25 m	Distance between rods in reinforcement mesh
$d_{deck,r}$	0.025 m	Diameter of reinforcement rods
τ_{deck}	30 MPa	Compressive strength of concrete

Table D.7: Constant parameters for the concrete decks.

Parameter	Value	Description
η_c	2.50	COP-value for the cooling system
ε_v	1000 J/m^3	Specific fan power for the ventilation system
ΔT_w	55°C	Temperature difference required for heating the domestic hot water
DF_{avg}	1%	Average daylight factor
φ_1	917 h	Annual number of hours[3] where $100\,\text{lux} \leq I_{avg} < 500\,\text{lux}$
φ_2	1143 h	Annual number of hours[3] where $I_{avg} < 100\,\text{lux}$

Table D.8: Constant parameters used for calculating the energy related performance measures.

Parameter	Value
Set point for the heating system	20°C
Set point for the air conditioning system	26°C
Minimum shading factor	0.2
Minimum amount of mechanical ventilation	$0.5\,\mathrm{h}^{-1}$
Maximum amount of mechanical ventilation	$0.5\,\mathrm{h}^{-1}$
Maximum amount of natural ventilation	$0\,\mathrm{h}^{-1}$
Heat exchanger efficiency	90%
Check for bypass	Yes
Internal load[4]	5 kW
Variable insulation	Not used
Infiltration	$0.1\,\mathrm{h}^{-1}$
Period where the settings are used	Every weekday of the year from 8am to 6pm

Table D.9: Settings for the HVAC systems when the building is occupied.

Parameter	Value
Set point for the heating system	12°C
Set point for the air conditioning system	Not used
Minimum shading factor	0.2
Minimum amount of mechanical ventilation	$0\,\mathrm{h}^{-1}$
Maximum amount of mechanical ventilation	$0\,\mathrm{h}^{-1}$
Maximum amount of natural ventilation	$0\,\mathrm{h}^{-1}$
Heat exchanger efficiency	90%
Check for bypass	Yes
Internal load	0 W
Variable insulation	Not used
Infiltration	$0.1\,\mathrm{h}^{-1}$
Period where the settings are used	When the settings in Table D.9 are not used

Table D.10: Settings for the HVAC systems when the building is empty.

[3] Only includes the hours where the building is used.

[4] Corresponds to 50 people, each generating 100 W.

Parameter	Window 1	Window 2
No. of panes	2	3
Glazing category	1.803861	1.473416
Total U-value (W/m^{2}K)	1.82	1.36
Total g-value	0.653	0.592
β_1	2.830662e+3	5.009999e+3
β_2	−3.389326e+0	−3.351915e+0
β_3	2.514105e+2	4.217602e+2
β_4	−6.173207e−2	−5.957476e−2
β_5	9.670630e+2	1.780656e+3

Table D.11: Window database. The thermal transmittance, or U-value, includes the interaction between the frame and the glazing unit. The solar transmittance, or g-value, includes the effect of the window frame. The last five rows consist of price model parameters for the construction jobs related to the windows. The construction jobs are given in Table D.12. The parameters apply to the model (4.27).

Window	Description	Job no.
Window 1	Double-glazed window of type 4-15-4, with air-filled gap	04.35.11,01-06
Window 2	Triple-glazed window of type 4-12-4-12-4, with gas-filled gaps	04.35.13

Table D.12: Description of the windows in the database.

Parameter	Value	Description
R_{int}	0.13 W/mK	Internal surface resistance
R_{ext}	0.04 W/mK	External surface resistance
ϱ_{air}	1.205 kg/m^3	Density of air at 20°C
c_{air}	1005 J/kgK	Specific heat capacity of air at 20°C
ϱ_w	980.7 kg/m^3	Density of water at 65°C
c_w	4183.28 J/kgK	Specific heat capacity of water at 65°C

Table D.13: Physical constants.

Parameter	Value	Description
R_{eq}	0.1 m^{2}K/W	Equivalent thermal resistance
w_a	20%	Fraction of solar energy absorbed by the air
w_w	80%	Fraction of solar energy absorbed by the internal surfaces

Table D.14: Various constants used when calculating the energy performance of the building, using the thermal network shown in Figure 4.6.

Constant parameters

Parameter	Value	Description
T_{max}	26°C	Maximum allowed internal air temperature
ε	10^{-3}	Tolerance level used when calculating the number of hours with overheating
p_{el}	1.92 DKR/kWh	Energy price for electricity[5]
p_{dh}	0.57 DKR/kWh	Energy price for district heating[6]

Table D.15: Various constants.

[5] Danish electricity price (including VAT) for the 2nd quarter 2006, provided by the Danish Energy Regulatory Authority [15].

[6] The calculation of this price is based on the district heating energy price for 2006, for the Ishøj district heating plant [58]. The energy price includes constant and variable prices, as well as VAT.

Appendix E

Optimization related nomenclature

Symbol	Description
f	Objective function
c	Constraint function
m	Number of constraints
x	Decision variables
n	Number of decision variables
$\mathcal{D}$	Domain of f and c
x^*	Solution to an optimization problem
i, j	Indices
$\mathcal{I}$	Index set referring to inequality constraints
$c_\mathcal{I}$	Inequality constraints
$n_\mathcal{I}$	Number of inequality constraints
$\mathcal{E}$	Index set referring to equality constraints
$c_\mathcal{E}$	Equality constraints
$n_\mathcal{E}$	Number of equality constraints
$\mathcal{S}$	Index set
$c_\mathcal{S}$	Constraint functions referred to by $\mathcal{S}$
$n_\mathcal{S}$	Number of indices in $\mathcal{S}$
$P_\mathcal{S}$	Matrix used for calculating $c_\mathcal{S}$
d	Domain constraint functions
$n_\mathcal{D}$	Number of domain constraint functions
$\mathcal{F}$	Feasible region
$\emptyset$	The empty set
k	Iteration counter
x_k	Solution estimate (iterate) for the kth iteration
Δx_k	Increment to x_k, also referred to as a step
x_0	First iterate, or starting point
$\lVert \cdot \rVert$	Unspecified vector norm
$\mathcal{R}_k$	Trust region for the kth iteration
ρ_k	Trust region radius for the kth iteration

Symbol	Description
e_k	Error in x_k
q	Performance measures
n_q	Number of performance measures
$A_{\hat{\mathcal{I}}}$, $b_{\hat{\mathcal{I}}}$	Matrix and vector used for specifying inequality requirements to decision parameters
$n_{\hat{\mathcal{I}}}$	Number of inequality requirements to decision parameters
$A_{\hat{\mathcal{E}}}$, $b_{\hat{\mathcal{E}}}$	Matrix and vector used for specifying equality requirements to decision parameters
$n_{\hat{\mathcal{E}}}$	Number of equality requirements to decision parameters
$a_{\mathcal{O}}$	Vector used for specifying optimality requirements to performance measures
$A_{\mathcal{I}}$, $b_{\mathcal{I}}$	Matrix and vector used for specifying inequality requirements to performance measures
$n_{\mathcal{I}}$	Number of inequality requirements to performance measures
$A_{\mathcal{E}}$, $b_{\mathcal{E}}$	Matrix and vector used for specifying equality requirements to performance measures
$n_{\mathcal{E}}$	Number of equality requirements to performance measures
∇f	Gradient of f
$J_{c_{\mathcal{I}}}$	Jacobian matrix for the function $c_{\mathcal{I}}$
$J_{c_{\mathcal{E}}}$	Jacobian matrix for the function $c_{\mathcal{E}}$
J_d	Jacobian matrix for the function d
μ_k	Damping term for the kth iteration
v	Vector used for minimizing the largest constraint violation
J_v	Jacobian matrix for v
$\Delta\hat{x}$	Auxiliary parameter
$\mathcal{A}$	Index set referring to the active constraints
$v_{\mathcal{A}}$	Vector containing the active subset of v
$\mathcal{L}$	Lagrange function
λ	Lagrange multipliers
h	Measure for constraint violation
β, γ, σ, δ	Constants
Δf_k	Decrease in objective function value in kth iteration
Δl_k	Decrease in linear model of objective function value in kth iteration
r_k	Gain factor for kth iteration
$theta$	Function used for updating trust region radius
ε_1, ε_2	Tolerance levels used as stopping criteria
k_{max}	Maximum allowed number of objective function evaluations
x_{new}	Suggested iterate for next iteration
B_k	Approximation to Jacobian matrix
$x_S^{(1)}$, $x_S^{(2)}$, $x_S^{(3)}$	Starting points
$\hat{x}$	Mid-point of the region of interest
ε	Tolerance level

Appendix F

Building related nomenclature

Symbol	Description
ϱ	Width to length ratio of building
w_{ext}	External width of building (m)
w_{int}	Internal width of building (m)
l_{ext}	External length of building (m)
l_{int}	Internal length of building (m)
N	Number of floors
$\sigma^{(1)}$, $\sigma^{(2)}$	Window fraction of the two façades
$A_{win}^{(1)}$, $A_{win}^{(2)}$	Window areas for the two façades (m^2)
n_{win}	Number of windows in window database
$\alpha^{(1)}$, $\alpha^{(2)}$	Weight factors for the windows of the two façades
i, j, k	Indices
$d_{g,i}$	Thickness of insulated layer of ground slab (m)
$d_{w,i}$	Thickness of insulated layer of external walls (m)
$d_{r,i}$	Thickness of insulated layer of roof construction (m)
n_d	Number of decision variables
Q_{tot}	Total amount of energy required by the building (kWh)
EF_3	Energy frame calculation required by the Danish building regulations (kWh)
EF_2	Energy frame calculation for acquiring the low energy class 2 label (kWh)
EF_1	Energy frame calculation for acquiring the low energy class 1 label (kWh)
BE	Heat loss through building envelope, excluding windows and doors (W)
U_g	Thermal transmittance for the ground slab ($\mathrm{W/m^2K}$)
U_{wall}	Thermal transmittance for the external walls ($\mathrm{W/m^2K}$)
U_r	Thermal transmittance for the roof construction ($\mathrm{W/m^2K}$)
$U_{win}^{(1)}$, $U_{win}^{(2)}$	Thermal transmittance for the windows of the two façades ($\mathrm{W/m^2K}$)
$OH^{(1)}$, $OH^{(2)}$	Annual number of hours with overheating for the two thermal zones (h)

Symbol	Description
$DH^{(1)}$, $DH^{(2)}$	Ratio between the depth of the room and the window height for the two thermal zones
C_{con}	Cost of construction the building (DKR)
C_{op}	Annual cost of operating the building (DKR)
A_{tot}	Total heated floor area (m^2)
A_e	Area of the building envelope, excluding windows and doors (m^2)
Q'_e	Heat loss through building envelope, excluding windows and doors (W)
T_{ext}	External air temperature (°C)
T_a	Internal air temperature (°C)
T_s	Internal surface temperature (°C)
T_w	Temperature of the thermal mass (°C)
K_w	Conductance between the thermal mass and the surface (W/K)
K_i	Conductance between the surface and internal air (W/K)
UA	Conductance between the internal and external environment through façade (W/K)
K_r	Conductance between the internal and external environment through roof construction (W/K)
K_g	Conductance between the internal and external environment through ground slab (W/K)
C_w	Effective heat capacity of thermal mass (J/kg K)
C_i	Heat capacity of internal air and property contents (J/kg K)
Q'_s	Energy absorbed by internal surfaces (W)
Q'_{sun}	Transmitted solar energy (W)
Q'_l	Internal loads (W)
Q'_h	Energy provided by heating system (W)
Q'_c	Energy removed by cooling system (W)
Q'_a	Energy delivered to internal air (W)
w_s	Fraction of transmitted solar energy absorbed by internal surfaces
w_a	Fraction of transmitted solar energy absorbed by internal air
S	Shading factor
$\widehat{UA}$	Conductance between the internal and external environments (W/K)
b_g	Temperature factor
V'	Mechanical ventilation rate (m^3/s)
p	Unit price for construction job (DKR)
β	Price model parameters
u	Number of purchased units
s	Secondary parameter
n_β	Number of price model parameters
$\hat{p}_{ji}$	Unit price at jth row and ith column for a construction job (DKR)
n_u	Number of columns used for organizing unit prices in price catalogue
n_s	Number of rows used for organizing unit prices in price catalogue

Symbol	Description
Δp_{kl}	Difference between price model and catalogue price (DKR)
P	Vector with unit prices for all required construction jobs (DKR)
n_{jobs}	Number of construction jobs
d_{wall}	Thickness of external walls (m)
d_{int}	Thickness of internal wall (m)
$d_{w,i}$	Thickness of insulated layer of external walls (m)
$d_{w,u}$	Thickness of uninsulated layer of external walls (m)
A_{int}	Internal heated floor area (m^2)
B, C, D	Auxiliary parameters
h_{ext}	External floor height (m)
$h_{win}^{(1)}$, $h_{win}^{(2)}$	Window heights for the two façades (m)
$O_{win}^{(1)}$, $O_{win}^{(2)}$	Circumference of windows for the two façades (m)
$A_{wall}^{(1)}$, $A_{wall}^{(2)}$	Areas of external walls for the two thermal zones (m^2)
A_{ext}	External area of a single floor (m^2)
A_{tot}	Total heated floor area (m^2)
$A_s^{(1)}$, $A_s^{(2)}$	Internal surface area for the two thermal zones (m^2)
h_{int}	Internal floor height (m)
h_{deck}	Height of concrete decks (m)
V_{int}	Internal air volume (m^3)
O_{ext}	External circumference of building (m)
$w^{(1)}$, $w^{(2)}$	Vectors with window properties for the two thermal zones
Ψ_{ww}	Linear thermal transmittance for thermal interaction between external wall and windows (W/m K)
Ψ_{fw}	Linear thermal transmittance for thermal interaction between external wall and foundation (W/m K)
R_{int}	Internal surface resistance (m^2 K/W)
R_{ext}	External surface resistance (m^2 K/W)
$\lambda_{w,u}$	Thermal conductivity of uninsulated layer of external walls (W/m K)
$\lambda_{w,i}$	Thermal conductivity of insulated layer of external walls (W/m K)
$U_{win}^{(1)}$, $U_{win}^{(2)}$	Thermal transmittance of windows for the two thermal zones (W/m^2 K)
$d_{r,i}$	Thickness of insulated layer of roof construction (m)
$d_{r,u}$	Thickness of uninsulated layer of roof construction (m)
$\lambda_{r,u}$	Thermal conductivity of uninsulated layer of roof construction (W/m K)
$\lambda_{r,i}$	Thermal conductivity of insulated layer of roof construction (W/m K)
$d_{g,i}$	Thickness of insulated layer of ground slab (m)
$d_{g,u}$	Thickness of uninsulated layer of ground slab (m)
$\lambda_{g,u}$	Thermal conductivity of uninsulated layer of ground slab (W/m K)
$\lambda_{g,i}$	Thermal conductivity of insulated layer of ground slab (W/m K)
$C_{i,tot}$	Total thermal capacity of property contents (J/K)
ϱ_{air}	Density of air (kg/m^3)
c_{air}	Specific heat capacity of air (J/kg K)

Symbol	Description
C_i	Thermal capacity of property contents per floor (J/K)
d_f	Width of foundation (m)
$m_{f,r}$	Weight of reinforcement rods used in foundation (kg)
$n_{f,m}$	Number of reinforcement meshes used in foundation
A_f	External area of foundation (m^2)
h_f	Height (m)
$d_{f,r}$	Diameter of reinforcement rods used in foundation (m)
$d_{f,m}$	Separation of reinforcement rods in mesh used in foundation (m)
$V_{r,tot}$	Volume of reinforcement rods used in foundation (m^3)
V_f	Volume of foundation (m^3)
d_{cb}	Thickness of capillary-breaking layer (m)
τ_{gs}	Compression strength of concrete used in ground slab (MPa)
V_{ws}	Volume of wearing surface (m^3)
τ_{ws}	Compression strength of concrete used in wearing surface (MPa)
d_{ws}	Thickness of wearing surface (m)
$A_{wall,tot}$	Total area of external walls (m^2)
$m_{wall,r}$	Weight of reinforcement rods used in external walls (kg)
$d_{wall,m}$	Separation of reinforcement rods in mesh used in external walls (m)
$n_{wall,m}$	Number of reinforcement meshes used in external walls
$d_{wall,r}$	Diameter of reinforcement rods used in external walls (m)
τ_{wall}	Compression strength of concrete used in external walls (MPa)
V_{wall}	Volume of concrete wall (m^3)
A_{deck}	Total area of concrete decks (m^2)
$m_{deck,r}$	Weight of reinforcement rods used in concrete decks (kg)
$d_{deck,m}$	Separation of reinforcement rods in mesh used in concrete decks (m)
$n_{deck,m}$	Number of reinforcement meshes used in concrete decks
$d_{deck,r}$	Diameter of reinforcement rods used in concrete decks (m)
τ_{deck}	Compression strength of concrete used in decks (MPa)
V_{deck}	Volume of concrete decks (m^3)
$m_{roof,r}$	Weight of reinforcement rods used in roof construction (kg)
$d_{roof,m}$	Separation of reinforcement rods in mesh used in roof construction (m)
$n_{roof,m}$	Number of reinforcement meshes used in roof construction
$d_{roof,r}$	Diameter of reinforcement rods used in roof construction (m)
τ_{roof}	Compression strength of concrete used in roof construction (MPa)
V_{roof}	Volume of roof construction (m^3)
h_{roof}	Height of concrete deck used in roof construction (m)
Q_{dh}	Annual amount of energy delivered from district heating system (kWh)
Q_{el}	Annual amount of electric energy delivered to the building (kWh)
Q_h	Annual amount of energy for heating the building (kWh)
Q_w	Annual amount of energy for producing domestic hot water (kWh)
Q_c	Annual amount of energy removed by the cooling system (kWh)

Symbol	Description
$Q_{c,el}$	Annual amount of electric energy for cooling the building (kWh)
Q_v	Annual amount of energy for ventilating the building (kWh)
Q_l	Annual amount of energy for artificial lighting (kWh)
Δt	Sample interval (h)
η_c	Coefficient of performance for the cooling system
V_w	Volume of domestic hot water required annually (m^3)
m_w	Mass of domestic hot water required annually (m^3)
m_w	Mass of domestic hot water required annually (m^3)
ϱ_w	Density of water ($\mathrm{kg/m}^3$)
c_w	Specific heat capacity of water (J/kg K)
ε_v	Specific fan power for the ventilation fan ($\mathrm{J/m}^3$)
Q'_v	Power needed for ventilating the building (W)
I_{avg}	Average internal illuminance (lux)
I_h	Global illuminance (lux)
DF_{avg}	Average daylight factor
φ_1	Annual number of hours where $100\,\mathrm{lux} \leq I_{avg} < 500\,\mathrm{lux}$
φ_2	Annual number of hours where $I_{avg} < 100\,\mathrm{lux}$
Q'_e	Heat loss through the building envelope, excluding windows and doors (W)
A_e	Area of the building envelope, excluding windows and doors (m^2)
ΔT_{hl}	Design temperature difference for building envelope (K)
T_{max}	Maximum allowed internal air temperature (°C)
ε	Tolerance level (K)
Δt_i	Contribution from time step i to the annual number of hours with overheating (h)
$\hat{T}_a$	Interpolated internal air temperature ($^\circ C$)
Δt^*	Time where interpolated temperature is equal to T_{max} (h)
p_{el}	Unit price for electric energy (DKR/kWh)
p_{dh}	Unit price for energy supplied by the district heating system (DKR/kWh)

Report no R-148
ISSN 1601-2917
ISBN 87-7877-220-6