Multiobjective Optimization in Engineering Design

Applications to Fluid Power Systems

Linköping Studies in Science and Technology. Dissertations.
No. 675

Multiobjective Optimization in Engineering Design

Applications to Fluid Power Systems

Johan Andersson

INSTITUTE OF TECHNOLOGY
LINKÖPINGS UNIVERSITET

Division of Fluid and Mechanical Engineering Systems
Department of Mechanical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

Linköping 2001

On the cover a set of chromosomes is depicted. Chromosomes are essential for evolution as well as for genetic algorithms, which is an optimization algorithm employed within this thesis.

ISBN 91-7219-943-1
ISSN 0345-7524

Department of Mechanical Engineering
Linköpings Universitet
SE-581 83 Linköping, Sweden

Printed in Sweden by UniTryck Linköping, 2001.

Abstract

THIS THESIS FOCUSES on how to improve design and development of complex engineering systems by employing simulation and optimization techniques. Within the thesis, methods are developed and applied to systems that combine mechanical, hydraulical and electrical subsystems, so-called multi-domain systems. Studied systems include a landing gear system for a civil aircraft, electro-hydrostatic actuation systems for aircraft applications as well as hydraulic actuation systems.

The usage of simulation and optimization in engineering design is gaining wider acceptance in all fields of industry as the computational capabilities of computers increase. Therefore, the applications for numerical optimization have increased dramatically. A great part of the design process is and will always be intuitive. Analytical techniques as well as numerical optimization could however be of great value and can permit vast improvements in design.

Within the thesis, a framework is presented in which modeling and simulation are employed to predict the performance of a design. Additionally, non-gradient optimization techniques are coupled to the simulation models to automate the search for the best design.

Engineering design problems often consist of several conflicting objectives. In many cases, the multiple objectives are aggregated into one single objective function. Optimization is then conducted with one optimal design as the result. The result is then strongly dependent on how the objectives are aggregated. Here a method is presented in which the Design Structure Matrix and the relationship matrix from the House of Quality method are applied to support the formulation of the objective function.

Another approach to tackle multiobjective design problems is to employ the concept of Pareto optimality. Within this thesis a new multiobjective genetic algorithm is proposed and applied to support the design of a hydraulic actuation system. The outcome from such a multiobjective optimization is a set of Pareto optimal solutions that visualize the trade-off between the competing objectives. The proposed method is capable of handling a mix of continuous design variables and discrete selections of individual components from catalogs or databases.

In real-world situations, system parameters will always include variations to some extent, and this fact is likely to influence the performance of the system. Therefore we need to answer not only the question “What is best?”, but also “What is sufficiently robust?” Within this thesis, several approaches to handle these two different questions are presented.

Acknowledgments

Nog finns det mål och mening med vår färd -
med det är resan som är mödan värd.
Karin Boye, 'I rörelse'.

THERE ARE SEVERAL people who have in one way or another made this thesis possible, and to whom I whish to express my gratitude. First, I would like to thank associate Professor Petter Krus for his great supervision, support and encouragement during this work. I would also like to thank Professor Jan-Ove Palmberg, head of division, for his support and for fruitful discussions during my work to complete this thesis.

My gratitude is extended to all members of the division of Fluid and Mechanical Engineering Systems for their help and support. The innumerable discussions, regarding not only research but everything under the sun have created a stimulating and enjoyable atmosphere to work in. I would like to express my special gratitude to Jochen Pohl, co-author of a number of papers in this thesis. Thanks for the many valuable discussions during the completion of this thesis and for the companionship in traveling the bumpy road towards the Ph.D. degree.

I would also like to thank my other co-authors: Dr. Katarina Nilsson at CNS Systems AB (formerly at Saab AB), for her help and support and for being an extraordinary industrial advisor. Professor Steven Eppinger at Massachusetts Institute of Technology (MIT), Boston, for introducing me to a new discipline, and for his kind supervision.

During this work, I have had the pleasure to co-operate with Professor David Wallace at the MIT CADlab, for which I am very grateful. The co-operation has been very valuable and given birth to many new ideas. I would like to extend the gratitude to all members of the CADlab and to Dr. Matthew Wall at Oculus Technologies Corporation for his support while using his GAlib genetic algorithm package.

Founding for this work was provided by the Swedish Foundation for Strategic Research through the ENDREA research program and is gratefully acknowledged.

Last but not least I would like to express the warmest thank you to my family Eva, Rolf and Anna and to my girlfriend, Kia, for always backing me up and encouraging me to struggle on.

Linköping in March 2001

Johan Andersson

Bryt upp, bryt upp! Den nya dagen gryr.
Oändligt är vårt stora äventyr.
Karin Boye, 'I rörelse'.

Papers

THE FOLLOWING SEVEN papers are appended and will be referred to by their Roman numerals. The papers are printed in their originally published state except for changes in format and minor errata.

[I] ANDERSSON J., POHL J. AND EPPINGER S. D., "A Design Process Modeling Approach Incorporating non-Linear Elements", in *Proceedings of the ASME Design Theory and Methodology Conference*, Atlanta, USA, September 13-16, 1998.

[II] ANDERSSON J., POHL J. AND KRUS P., "Design of Objective Functions for Optimization of Multidomain Systems", in *Proceedings of the ASME Annual Winter meeting*, Fluid Power System and Technology, Anaheim, USA, November 15-20, 1998.

[III] NILSSON K., ANDERSSON J. AND KRUS P., "Method for Integrated Systems Design – A Study of EHA Systems", in *Proceedings of Recent Advances in Aerospace Hydraulics*, Toulouse, France, November 24-25, 1998.

[IV] ANDERSSON J. AND WALLACE D., "Pareto Optimization Using the Struggle Genetic Crowding Algorithm", submitted for international publication, 2000.

[V] ANDERSSON J., KRUS P. AND WALLACE D., "Multiobjective Optimization of Hydraulic Actuation Systems, in *Proceedings of ASME Design Automation Conference*, Baltimore, USA, September 11-13, 2000.

[VI] ANDERSSON J. AND KRUS P., "Multiobjective Optimization of Mixed Variable Design Problems", in *Proceedings of 1st International Conference on Evolutionary Multi Criteria Optimization*, Zurich, Switzerland, March 7-9, 2001.

[VII] ANDERSSON J. AND KRUS P., "Metamodel Representations for Robustness Assessment in Multiobjective Optimization", accepted publication in *Proceedings of the 13th International Conference on Engineering Design, ICED 01*, Glasgow, UK August 21-23, 2001.

The following papers are not included in the thesis but constitute an important part of the background.

[VIII] ANDERSSON J., POHL J. AND KRUS P., "Design of Multi-Domain Systems Using Optimization", in *Proceedings of NordDesign 98*, Stockholm, Sweden, August 26-28, 1998.

[IX] ANDERSSON J., KRUS P. AND NILSSON K., "Optimization as a Support for Selection and Design of Aircraft Actuation Systems", in *Proceedings of the Seventh AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, St. Louis, USA, September 2-4, 1998.

[X] SVENSON H., ANDERSSON J. AND RYDBERG K.-E., "Modelling of Losses and Temperature Calculations in Fluid Power Systems", in P*roceedings of the Sixth Scandinavian International Conference on Fluid Power*, Tampere, Finland, May 26-28, 1999.

[XI] ANDERSSON J., KRUS P., NILSSON K. AND STORCK, K., "Modelling and Simulation of Heat Generation in Electro-Hydrostatic Actuation Systems", in *Proceedings of the 4:th JHPS International Symposium on Fluid Power*, Tokyo, Japan, November 15-17, 1999.

[XII] PERSSON P., KAMMERLIND P., BERGMAN B. AND ANDERSSON J., "A Methodology for Multi-Characteristic System Improvement with Active Expert Involvement", *Quality and Reliability Engineering International*, vol. 16, 2000, pp. 405-416.

[XIII] ANDERSSON J., "A Survey of Multiobjective Optimization in Engineering Design", Technical report LiTH-IKP-R-1097, Department of Mechanical Engineering, Linköping University, Linköping, Sweden, 2000.

Contents

Appended papers

1 Introduction

THERE IS A clear trend in industry towards more complex products spanning over several engineering domains. Simultaneously, there is a pressure on developing products faster, at competitive prices, and to a high quality standard. In order to meet these demands, manufacturing companies have been forced to focus their efforts on the development process. In that respect, one issue has been to ensure the efficiency of the development process, which has resulted in methods to analyze and manage the design process. Another issue has been to develop tools and techniques that support the design of complex products, which has produced a wealth of computerized engineering tools. As the computational capabilities of the computers increase, the scope for simulation and numerical optimization is enlarged. A great part of the design process will always be intuitive. However, analytical techniques, simulation models and numerical optimization could be of great value and can permit vast improvements in design.

The first issue is to ensure an efficient design process. Within this thesis, a design process modeling approach is presented where simulation is employed in order to predict the performance of the design process in terms of lead-time and cost. Design process modeling gives enhanced understanding of the properties of the process, which is important as a thorough understanding of the design process forms the basis for further process improvements. With the help of design process models, different competing design processes can be compared and evaluated based on process lead-time and costs. The design process modeling method presented in this thesis is described in chapter 3.3 and in Paper [I].

The second issue focuses on how to improve the design of complex systems by employing simulation and optimization techniques. As widely recognized, engineering design is an iterative process where new design proposals are generated and evaluated. According to Roosenburg and Eekels [70], the iterative part of the design process consists of *synthesis, simulation, evaluation* and *decision*. For each provisional design, the expected properties are predicted using simulation models, which are then compared to the requirements on the system. If the design does not meet the requirements it is modified and evaluated again in the search for the best possible design. Based on this description, it could be seen that design is essentially an optimization process, as stated already in 1967 by Simon [76]. In order to raise the level of automation, and thereby speed up parts of the process, the optimization could be formalized and an optimization algorithm introduced.

This thesis focuses on a set of problems concerning the reformulation of the design problem as an optimization problem. First, a framework is presented where simulation and optimization are introduced to support the design process. Then optimization methods suitable for these types of applications are discussed. One component of employing simulation and optimization techniques is to gain increased insight into the properties of the system. The other component is to gain a better understanding of ourselves, meaning our priorities among the objectives and our expectations of the system. Perhaps what we wish for is unrealistic or ill conceived. Conversely, our wishes might not be imaginative enough. When employing simulation we want it to answer some of our questions, however a simulation model raises new questions as well. The balance between questions asked and questions raised is critical to our success, Schrage [73].

The presence of several conflicting objectives is typical for engineering design problems. In many cases where optimization techniques are utilized, the multiple objectives are aggregated into one single objective function. Optimization is then conducted with one optimal design as the result. The result is then strongly dependent on how the objectives are aggregated. Here a method is presented in which the Design Structure Matrix, presented by Steward [84], and the relationship matrix from the House of Quality method, see Hauser and Clausing [34] and Sullivan [85], are applied to support the formulation of the objective function. The method is applied to the design of a landing gear system, which constitutes a mechanical structure and a hydraulic actuation system.

Another approach to handle multiobjective design problems is to employ the concept of Pareto optimality. Pareto optimality was introduced in the late eighteen hundreds by the economist Vilfredo Pareto, and defined as follows: A solution is said to be Pareto optimal if there exists no other solution that is better in all attributes. This implies that in order to achieve a better value in one objective at least one of the other objectives is going to deteriorate if the solution is Pareto optimal. Thus, the outcome of a Pareto optimization is not one optimal point, but a set of Pareto optimal solutions that visualize the trade-off between the objectives.

Within this thesis, a multiobjective genetic algorithm is proposed and applied to support the design of a hydraulic actuation system. The outcome from this optimization is a set of Pareto optimal solutions that visualizes the trade-off between system performance and cost. The proposed method is extended to be able to handle a mix of continuous design variables and discrete selections of individual components from catalogs or databases.

In real-world situations, system parameters will always include variations to some extent, and this fact is likely to influence the performance of the system. However, we want the system to be robust and perform well under a wide range of operational conditions. Therefore we need to answer not only the question “What is best?”, but also “What is sufficiently robust?” Within this thesis, several approaches to handle the issue of robustness in design optimization are presented, see chapter 7.

The seven appended papers constitute the bulk of this thesis. In this introduction the methods developed in the papers are discussed briefly and presented as parts of a greater whole. For a more thorough reading of the methods developed, the reader is referred to the appended papers.

2
Aims

THE PRINCIPAL AIM of this thesis is to support the employment of simulation and optimization techniques in engineering design. The first aim is to present a framework where optimization is employed in order to speed up and improve the design of complex systems based on simulations. The focus is on real applications where computer simulations are employed to predict the properties of a system.

The second aim is to support the formulation of the optimization problem, partly by supporting the selection of optimization parameters, but also by supporting the formulation of the objective function. The design problem is often multiobjective in nature, it is therefore natural to formulate the problem as a multiobjective optimization problem. Consequently, another aim is to develop a reliable multiobjective optimization algorithm.

Another important issue is that the system should perform well under a wide range of working conditions. An additional aim is therefore to present methods that take both system optimality and system robustness into account.

No matter how good the tools we employed in order to support the design process, the design process itself has to be managed as well. Thus, a further aim is to provide means to increase the insight into the properties of the design process. Within this thesis, a design process modeling approach is presented where simulation is employed in order to predict the performance of the design process. Simulation formerly supported the designer in improving the design. Here, design process simulation supports the management in managing the design process.

3
The engineering design process

ENGINEERING DESIGN IS a special form of problem solving where a set of frequently unclear objectives has to be balanced without violating any given constraints. Therefore, it seems natural to look upon a design problem as an optimization problem. By employing modern modeling, simulation and optimization techniques, vast improvements could be achieved in design. However, there will always be parts of the design process that require human or inquantifiable judgment that is not suited for automation with any optimization strategy. Within this chapter, different design theories will be discussed and it is striking how similar they are to a general optimization procedure. On this basis, a system design model is presented, which introduces simulation and optimization as tools that support the design process.

However good tools we employ to support the design of a system, the design process itself has to be managed. The last part of this chapter is assigned to describing a method that could be employed to model the design process itself. Once such a model is created, simulation could give increased insights to the properties of the design process.

3.1 Literature

This thesis focuses on the design of engineering systems based on numerical simulation models. In the literature, there are several definitions of what a system is, see for instance Blanchard and Fabrycky [5], Bruns [9], Hubka and Eder [38], Ljung and Glad [52] and Pahl and Beitz [62]. A common meaning of the word system is a set of interrelated components intended to achieve a common objective. The system is also characterized by a boundary, which cuts across the links to the environment, thus creating inputs and outputs to the system. The properties and behaviors of each component contribute to the system behavior as a whole.

In a system design perspective we are not focusing on detailed design of individual components, but on the interrelation of components, which each could be described by a limited set of key parameters. Thus, the focus of this thesis is between the conceptual design and early detail design of the phase model described below.

There are many models of the engineering design process. A common model is the phase type model, see for instance Hubka and Eder [38], Pahl and Beitz [62] and Pugh [66]. The phase model is a top-down iterative process, here represented in the nomenclature of Pahl and Beitz [62]:

- Clarification of the task
- Conceptual design
- Embodiment design
- Detailed design

Within each phase a set of activities is performed. Naturally, one whishes to perform each phase only once, thus ending up with the final design without any iterations. However, the design process is very iterative, as stated by several authors, see Cross [15], Hubka [38], Roozenburg and Eekels [70] and Smith and Eppinger [78]. Therefore, many iterations are often required before the final design is achieved. An iterative model of a basic design process is presented by for example Roozenburg and Eekels [70] as depicted in Figure 1. This iterative design process could be found within each of the three later phases of the phase type model.

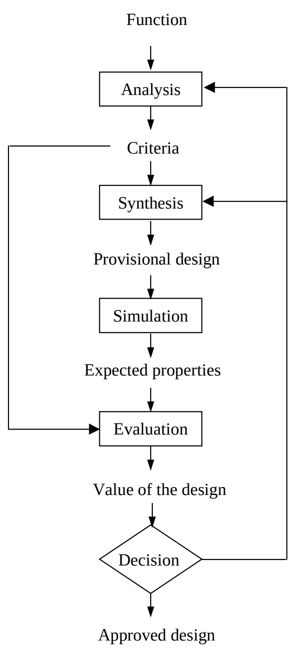


Figure 1. The basic design cycle according to Roozenburg and Eekels [70].

The model in Figure 1 states that design is an iterative process where new design proposals are generated and evaluated. According to Roosenburg and Eekels, the iterative part consists of *analysis, synthesis, simulation, evaluation* and *decision*. For each provisional design the expected properties are compared to the criteria. If the design does not meet the criteria it is modified and evaluated again in the search for the best possible design. From this model, it could be seen that design is essentially an optimization process, as stated already in 1967 by Simon [76]. In order to raise the level of automation, and thereby speed up parts of the process, the optimization could be formalized and an optimization algorithm introduced, see the model in Figure 2.

3.2 System design

Figure 2 below depicts a system design process where modeling, simulation and optimization are introduced to support and speed up the design process. The focus of the process is on the two middle phases of the phase type model, namely conceptual and embodiment design. In the proposed system design process, the iterative part of the general design process in Figure 1 is formalized and automated with the help of an optimization algorithm.

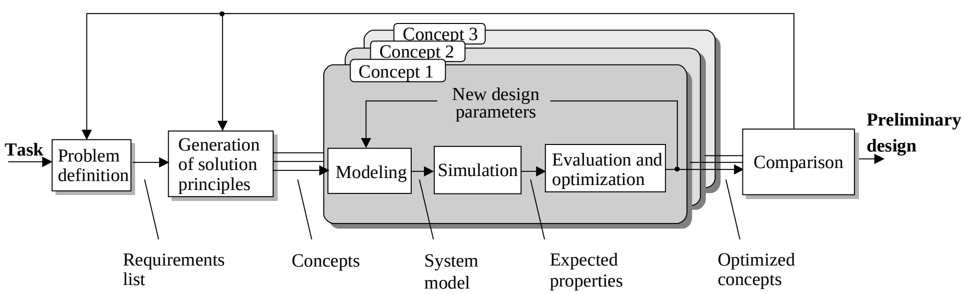


Figure 2. The system design process.

The 'problem definition' in Figure 2 results in a requirements list which is used in order to generate different solution principles/concepts. Although the first two tasks are both tedious and important, they are not the focus of this work. Once the concepts have reached a sufficient degree of refinement, modeling and simulation are employed in order to predict the properties of particular system solutions. Each solution is evaluated with the help of an objective function, which acts as a figure of merit. Optimization is then employed in order to automate the evaluation of system solutions and to generate new system proposals. The process continues until the optimization is converged and an optimal system is found.

Often the first optimization run does not result in the final design. One essential aspect of using modeling and simulation is to understand the system we are designing. The other aspect is to understand our expectations from the system, and our priorities among the objectives. Both aspects are equally important. It is essential to engineering design to manage the dialog between specification and prototype, as stated by Schrage [73]. Often simulations confirm that what we wish for is unrealistic or ill conceived. Conversely, they can also reveal that our whishes are not imaginative enough. If the optimization does not converge to a desired system, the concept has to be modified or the problem reformulated, which results in a new objective function. In Figure 2 this is visualized by the two outer loops back to 'generation of solution principles' and 'problem definition' respectively.

Naturally the activity 'generation of solution principles' produces a number of conceivable concepts, which each one is optimized. Thus each concept is brought to maximum performance; optimization thereby provides a solid basis for concept selection, as will be discussed further in chapter 6.2. The parts of the evaluation that include human and inquantifiable judgment are performed outside the optimization loop in the activity named 'comparison'. If the concepts do not fulfill these requirements, they have to be modified as indicated with the outer loop. The modification can include both changes to the actual concepts but also to the objective function formulation, e.g. by introducing new constraints.

3.2.1 Modeling and simulation

Roozenburg and Eekels [70] understand the term simulation as *"forming an image of the behavior and properties of a designed product by reasoning and/or testing models"*. This is an excellent definition of simulation, although somewhat broader than otherwise used in this thesis. Here simulation always refers to the execution of a model in order to predict the properties of a design proposal.

There is an abundance of definitions for the word 'model'. A common definition, here expressed in the words of Neelamkavil [58], is very illustrative. *"A model is a simplified representation of a system intended to enhance our ability to understand, predict and possibly control the behavior of the system."* A model can be of a mental, verbal, physical or mathematical nature. This thesis focuses on mathematical models, typically those implemented in a computer environment.

Systems today usually combine different engineering disciplines, and each discipline uses their own tools to create models of their parts of the system. Furthermore, the engineers developing the different models may also be situated in geographically disperse locations. In order to predict the properties of the system as a whole, the different models have to be interconnected. Therefore, in order to manage cross-functional teams we have to managing multi-domain simulation. Thus, engineers working with different software packages have to be able to communicate with each other. A framework that allows this is the DOME framework presented by Wallace et al. [94]. Other techniques are presented in works by Larsson [50] and Papalambros et al. [63].

This thesis does not focus on how to model complex systems, or on how to facilitate engineering software to communicate. However, the optimization techniques that are being developed should be capable of optimizing systems based on such system descriptions. The way the design is represented determines which optimization techniques could be used.

The simulation models employed in the systems design process are all deterministic, i.e. they always give the same result at repeated calculations when a steady-state condition could be obtained. However, in reality the system parameters would always include variations to some extent. These variations in system parameters would most likely influence the performance of the system. In chapter 7 the issue about robustness versus optimality is discussed, and new approaches to handle this problem are presented.

3.2.2 Optimization

Optimization as it is employed here is based on simulation results, possibly from a large number of different simulation environments. The problem is also characterized by the presence of both continuous parameters and discrete selections of individual components from catalogs or databases. Thus, the problem is non-linear and there are now derivatives of the objective functions available in a straightforward manner. This is one reason for applying non-gradient methods such as the Complex method or genetic algorithms. Another major reason is that these methods are more robust in locating the global optima in multi-modal search spaces. These methods could be applied to a wide range of problems without any modifications to the algorithms.

One part of the optimization is the evaluation of design proposals. The second part is the generation of new and hopefully better designs. Thus, optimization consists of both analysis (evaluation) and synthesis (generation of new solutions).

The evaluation is usually done by means of an objective function which consists of a figure of merit describing how good a design proposal is. As in any optimization problem, the formulation of the objective function is very crucial to the outcome of the optimization. In engineering design, neither the objectives nor the constraints are clearly defined. Within this thesis, different methods for creating objective functions and handling constraints are presented, see chapter 4 and 6. Optimization could however be conducted without an explicit objective function, for example by an iterative genetic algorithm as in Dawkins [18] and Smyth [80] or by active expert involvement as in Paper [XII]. Within this thesis however, the focus is on optimization with explicitly stated objective functions.

The generation of new solutions depends on the optimization strategy. Within this thesis two different optimization methods are used, namely the Complex method and genetic algorithms. Both methods use a set of design proposals, which evolves as the optimization progresses. At initial inspection, one can argue that there is no synthesis and nothing creative involved in the solutions generated by the optimization strategy. Technically speaking we are just finding solutions that are already out there waiting to be found. Surely, there is nothing creative about searching a small space, but the solution space could be huge, and as it gets larger, increasingly sophisticated search methods are needed in order to find the best or even a good solution.

In *The Blind Watchmaker*, Dawkins [18] describes evolution as a creative process where ‘predefined designs’ are found. In Kroo et al. [45] a genetic algorithm ‘invents’ the C-shaped wing. Schrage has also discovered the powers of evolution. In his book *Serious Play* [73], he states that: “*In many respects, evolution is the ultimate prototyping and simulation methodology. Evolution’s power and versatility are inarguable; its ability to innovate and surprise is overwhelming.*” Therefore, under certain premises, optimization could be seen as a technique for innovation. Naturally, this is depending on how the optimization problem is formulated. During this work optimization has shown these creative properties on several occasions.

Although better methods and tools are constantly developed to support the design process, the design process itself has to be managed in order to be effective. In the following chapter, a method is presented that facilitates analysis of the performance of a design process as well as it proposes appropriate actions to improve it.

3.3 Designing the design process

Industry today faces the challenge of having to develop high quality products faster than ever before. This situation has put a stronger focus on the development process, see for instance Eppinger et al. [20], Wheelwright and Clark [95] and Whitney [96]. A well-managed product development process is an important factor in order to stay competitive. When the repetitive nature of the development processes is recognized, the process can be modeled and simulation can be employed in order to predict the performance of the process. Design process modeling can be one course of action to discover key activities that have great impact on process lead-time and cost.

Iterations are fundamental to the engineering design process, but common management tools such as PERT charts do not handle iterations in a satisfactory manner. There are different approaches for handling these design iterations in the literature, see Adler et al. [1], Bell et al. [3], Christian and Seering [10] and Smith and Eppinger [78] and [79].

The method presented in this thesis follows most closely the signal flow graph approach presented by Eppinger et al.[21], and applied to evaluate different design processes in Isaksson et al. [41]. This method extends the signal flow graph approach by introducing non-linear elements, which reduces the modeling effort significantly and also increases the clarity of the process model.

3.3.1 Modeling approach

Design process modeling as implemented here is based on the observation that a design process comprises a number of smaller design activities. The process can be modeled by tracking design information that is exchanged between different design tasks. In this type of model, both parallel and sequential work flows are often observed. This modeling approach includes two types of elements, namely design tasks and design reviews.

With every design task, characteristics of execution time and task cost per time unit are associated, as described in Figure 3. To create a flexible and accurate model, the task characteristics are allowed to vary with the number of iterations done. Consider for example a design process with a large amount of CAD modeling. In the first design iteration, the CAD models have to be created, but in the second iteration they only need to be modified, which is less time-consuming. This would correspond to a step reduction in task time as shown in Figure 3. A task in which the execution time decreases with every design iteration can be modeled as a “learning by doing” task with an associated learning curve function.

The design review model element, see Figure 3, models the probability of proceeding forward to the next design task, otherwise the process flows back to an earlier task. The design review is evaluated with the help of a random function. The characteristics of the design review can also be a function of the number of iterations done.

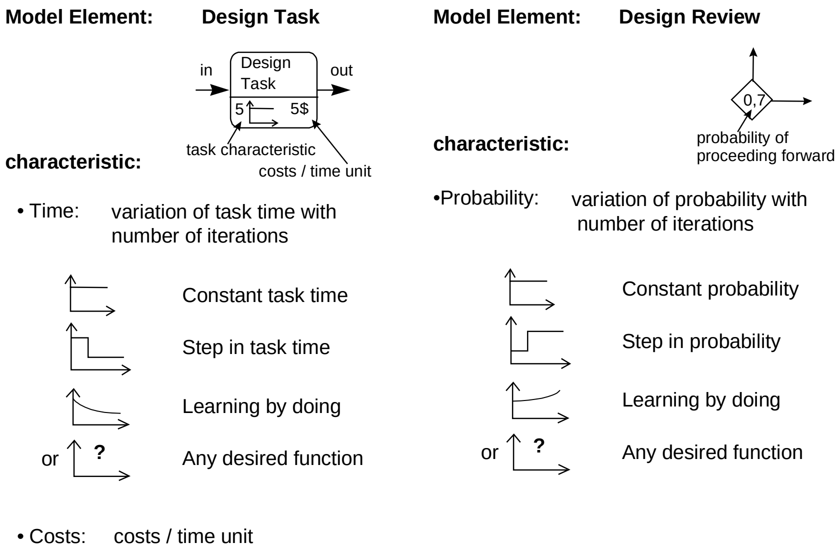


Figure 3. The modeling elements design task and design review. The figure also shows the characteristics of the elements and how they may vary with the number of iterations.

By combining these basic modeling elements, an arbitrary design process can be modeled. In Figure 4 below, a sample process model for the development process of a hydraulic pump is depicted in order to explain the modeling approach.

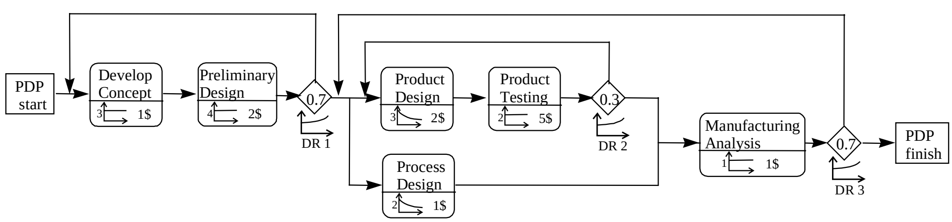


Figure 4. Hydraulic pump development process, starting with conceptual and preliminary design. Followed by product design and testing in parallel with process design, and finally manufacturing analysis. Please note the dynamic change in task and design review characteristics.

The lead-time and cost distributions of the process are calculated numerically by employing different types of modified Monte Carlo simulation, as explained in Paper [I]. The lead-time distribution for the pump development process in Figure 4 is shown in Figure 5 below.

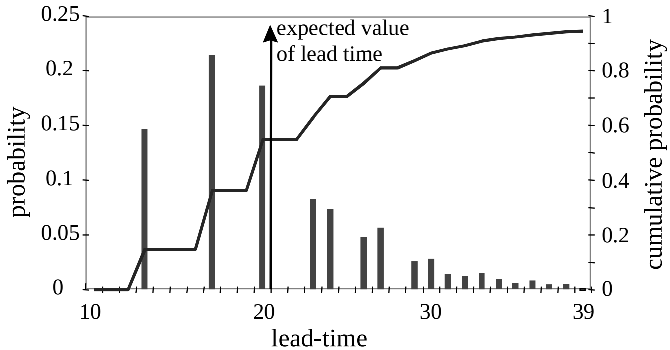


Figure 5. Lead-time distribution of the hydraulic pump development process, together with the expected value of lead-time and cumulative probability. All paths with a lead-time shorter than 40 are shown.

From Figure 5 it could be seen that the shortest lead-time possible is 13 time units and the expected value is 20.4 time units. With the help of the cumulative probability, it is possible to estimate the probability of the process finishing within a certain time. These measures could be used to estimate the risk associated with the project. By modeling different conceivable processes, they could be compared based on both lead-time and cost.

A sensitivity analysis of the design process could be performed in order to provide insights as to how each task and design review influences overall lead-time and cost. The sensitivity can be calculated as the relative change in expected value due to a small change in a parameter, e.g. a task characteristic. If L represents the lead-time and k a parameter of interest, the sensitivity of L to changes in k is given by equation (1).

$$S_k^L = \frac{\Delta E[L]/E[L]}{\Delta k/k} \tag{1}$$

A sensitivity analysis of the sample process is performed and the result is shown in Figure 6.

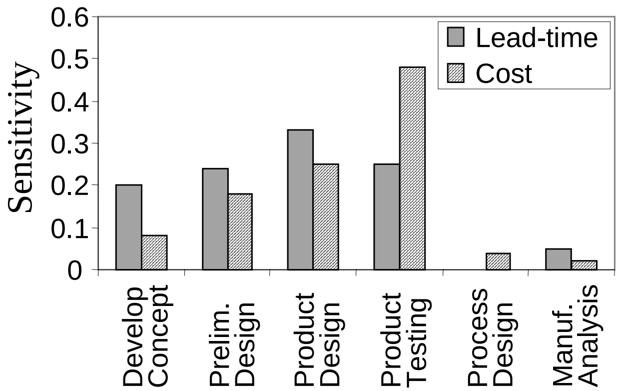


Figure 6. Lead-time and cost sensitivities due to changes in task duration.

The sensitivity analysis confirms a general insight that tasks performed frequently are more sensitive to changes in task parameters. The positive sensitivity values indicate to what extent lead-time and cost increase for positive variations of the task times. The highest cost sensitivity value is the sensitivity to the lead-time of product testing, which is the most expensive task. The highest time sensitivity value is not for the longest-duration task. The highest time sensitivity instead is to changes in the duration of product design, which is embedded within the most frequently performed iteration loop. Another insight is that the process design task has no influence on overall lead time because it is carried out in parallel with product design and testing, which together have a longer lead time. However, process design still affects the total cost.

Studying the lead time probability distribution and the sensitivity analysis yields a profound understanding of the process. The tasks with the greatest influence on lead-time and costs can be identified and thereby focused upon when improving the process.

4 Optimization in engineering design

MANY ASPECTS HAVE to be taken into account when reformulating the design problem as an optimization problem. First, we have to consider what properties the system should have, i.e. what are the values that we want to create, and how shall we be able to measure them? Secondly, which are the design variables, or system parameters that we could manipulate in order to achieve the best possible design? Finally, how do we articulate what is actually the best possible design? These types of questions always confront the designer. However, when formulating the design problem as an optimization problem the answers have to be mathematically formalized.

This chapter starts with a discussion of the concept of value, giving an idea of the difficulties in formulating the optimization problem. Thereafter the design variables, the means by which we modify the design, are investigated. Here parts of the framework presented by Sidall [75] are adopted. Then the optimization problem is formulated. Finally, the focus is on how to express what is desired of the system, i.e. the creation of the objective function. Different approaches are discussed and their pros and cons are emphasized.

4.1 The concept of value

The concept of value is central to decision theory — the measure about what is good or desirable about a design. At a first glance, one would say that it is no problem. If two designs are comparable, simply chose the cheapest one. However, consider the designing of a car; which must not only be cheap but safe, have a long life, be both quiet and fast. How shall we then choose? Which characteristics contribute the most to the overall value of the design? This is very crucial to decision-making, and in general also to design.

For any given design, the designer has to give the different characteristics such as low initial cost, long life and good performance a weighting value. This is usually not done explicitly, but intuitively the designer does just that, however unaware he might be of it. During the design process, the designer must tradeoff characteristics against each other. How much is longer life worth in terms of higher manufacturing costs. One purpose of conducting multiobjective optimization is to make these trade-offs visible. It would indeed be an interesting task to estimate what different ratings gave the final design.

Value is an inherent property of the design, which could be defined as that which satisfies desire. It remains however to be determined whose desires we should try to satisfy, and how we can articulate them to the designer. It might be hard to value a design even when one has the physical artifact; it is even harder to do it in the earlier phases of the design process. However, in order to employ optimization to support the designer this is exactly what we have to do. Usually the designer employs a set of modeling and simulation tools in order to predict the properties of a design.

Often when we say value of a design we refer to the utility value which relates to the function or usefulness of the design. There are however many other values that the designer must take into account. Here, we are just focusing on the function of a design. This is without saying that the others are not important, they are however left out of the optimization and have to be considered once the function is approved.

4.2 The design variables

Design variables are parameters that the designer might "adjust" in order to modify the system he is designing. There are many types of design variables.

Independent design variables are the actual quantities the designer deals with directly, such as geometry, material properties, production volume, surface finish, configuration of components, lubrication properties and many more. Independent design variables are usually called just design variables or design parameters. Here the term system parameters will be used also.

Dependent variables are variables the designer can not directly assign values to but he works with them through the design parameters. The dependent variables are usually named *characteristics* or *attributes* of the design. Examples of system characteristics are energy consumption, control error and cost. The value of a design is largely a function of the characteristics of the design. In optimization, the objective function value corresponds to the value of a particular characteristic. An objective function is thus the relation between the design parameters and the value of a particular characteristic. For a general design problem, it might be very difficult or even impossible to represent all such relations analytically. For once, the characteristic might be the outcome of a complex simulation, or they might include inquantifiable human judgment.

State variables are an intermediate type of design variables between dependent and independent design variables, such as the pressure in a hydraulic cylinder or the current to an electric motor. State variables cannot directly be assigned values, and they do not directly contribute to the value of the design, as the characteristics do.

Operating variables are variables that can be changed by the operator after the design has been actually built. The *environmental variables* or the external variables are the environmental factors that affect the design when in use, e.g. changing loads, extreme temperature and wear. The designer has to determine the working conditions of the design in order to assess both the environmental and the operational variables.

The design problem could be formulated as to assign values to the system parameters to ensure that the state variables and the characteristics are as good as possible during a wide range of operating and environmental variables. This is indeed an intricate multiobjective optimization problem.

4.3 The multiobjective optimization problem

A general multiobjective design problem is expressed by equations (2) and (3)

$$\min \mathbf{F}(\mathbf{x}) = \left[f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x}) \right] \qquad s.t.\ \mathbf{x} \in S \tag{2}$$

$$\mathbf{x} = (x_1, x_2, \ldots, x_n)^T \tag{3}$$

where $f_1(x), f_2(x), \ldots, f_k(x)$ are the k objective functions, $(x_1, x_2, \ldots, x_n)$ are the n optimization parameters, and $S \in R^n$ is the solution or parameter space. Obtainable objective vectors, $\{\mathbf{F}(\mathbf{x}) | x \in S\}$, are denoted by *Y*, so *S* is mapped by **F** onto *Y*. $Y \in R^k$ is usually referred to as the attribute or criteria space, where ∂Y is the boundary of *Y*. For a general design problem, **F** is non-linear and multi-modal, and *S* might be defined by non-linear constraints and may contain both continuous and discrete member variables.

$f_1^*, f_2^*, \ldots, f_k^*$ will be used to denote the individual minima of each objective function respectively. The utopian solution is defined as $\mathbf{F}^* = \left[f_1^*, f_2^*, \ldots, f_k^* \right]$. As $\mathbf{F}^*$ minimizes all objectives simultaneously, it is an ideal solution, however it is rarely feasible.

In this formulation, minimize **F**(**x**), lacks clear meaning as the set {**F**(**x**)} for all feasible **x** lacks a natural ordering, whenever **F**(**x**) is vector-valued. In order to determine whether $\mathbf{F}(\mathbf{x_1})$ is better then $\mathbf{F}(\mathbf{x_2})$, and thereby order the set {**F**(**x**)}, the subjective judgment from a decision-maker is needed.

One property commonly considered as necessary for any candidate solution to the multiobjective problem is that the solution is not dominated. Considering a minimization problem and two solution vectors **x**, $\mathbf{y} \in S$. **x** is said to dominate **y**, denoted $\mathbf{x} \succ \mathbf{y}$, if:

$$\forall i \in \{1, 2, \ldots, k\} : f_i(\mathbf{x}) \le f_i(\mathbf{y}) \quad and \quad \exists j \in \{1, 2, \ldots, k\} : f_j(\mathbf{x}) < f_j(\mathbf{y}) \tag{4}$$

The Pareto subset of ∂Y contains all non-dominated solutions. The space in R^k formed by the objective vectors of Pareto optimal solutions is known as the Pareto optimal front, $\mathcal{P}$.

If the final solution is selected from the set of Pareto optimal solutions, there would not exist any solutions that are better in all attributes. It is clear that any final design solution should preferably be a member of the Pareto optimal set. If the solution is not in the Pareto optimal set, it could be improved without degeneration in any of the objectives, and thus it is not a rational choice. This is true as long as the selection is done based on the objectives only. Pareto optimal solutions are also known as non-dominated or efficient solutions. Figure 7 provides a visualization of the presented nomenclature.

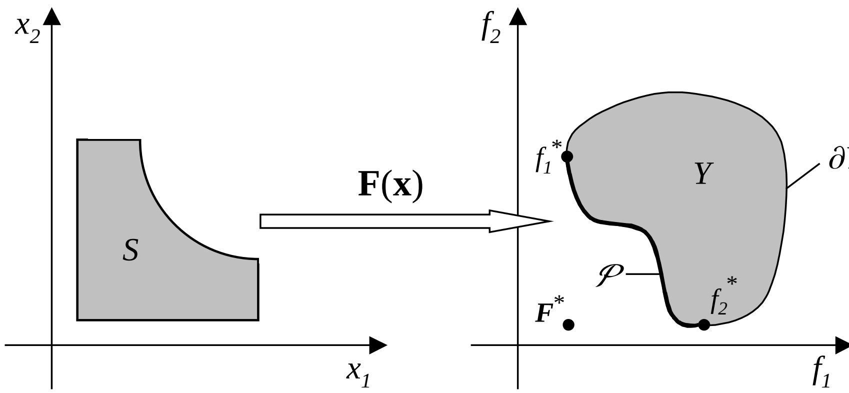


Figure 7. Solution and attribute space nomenclature for a problem with two design variables (x_1 and x_2) and two objectives (f_1 and f_2), which should both be minimized.

The attribute space, Y, looks the same regardless of how the objectives are aggregated to an overall objective function. Depending on how the overall objective function is formulated, the optimization will result in different points on the Pareto front. The remains of this chapter are designated to methods that support the formulation of the overall objective function, thus ordering $\mathbf{F}(\mathbf{x})$ so that the multiobjective problem could be solved.

4.4 Formulating the objective

As most optimization problems are multiobjective in nature, there are many methods available to tackle this kind of problems. References to multiobjective optimization could be found in Hwang et al. [40], Ringuest [69] and Steuer [82] and with applications to engineering design in Eschenauer et al. [22] and Osyczka [61]. Generally, the multiobjective optimization problem (MOOP) can be handled in four different ways depending on when the decision-maker articulates his preference concerning the different objectives: never, before, during or after the actual optimization procedure.

In the first two approaches, the different objectives are aggregated to one overall objective function. Optimization is then conducted with one optimal design as the result. The result is then strongly dependent on how the objectives were aggregated. In the literature, different methods have been developed to support the decision-maker in aggregating the objectives.

The third approach is an iterative process where the decision-maker progressively articulates his preferences on the different objectives. The underlying assumption is that once the search for an optimal solution has started and the decision-maker has been presented with some alternatives, he will be better equipped to value the objectives.

In the fourth and final approach, optimization is conducted without the decision-maker articulating any preferences among the objectives. The outcome of this optimization is a set of Pareto optimal solutions which elucidate the trade-off between the objectives. The decision-maker then has to trade the objectives against each other in order to select the final design. Thus, optimization is conducted before the decision-maker articulates his preferences.

The methods developed in this thesis belong to the second and the fourth approach. The four different approaches, exemplified with suitable methods are shown in Figure 8.

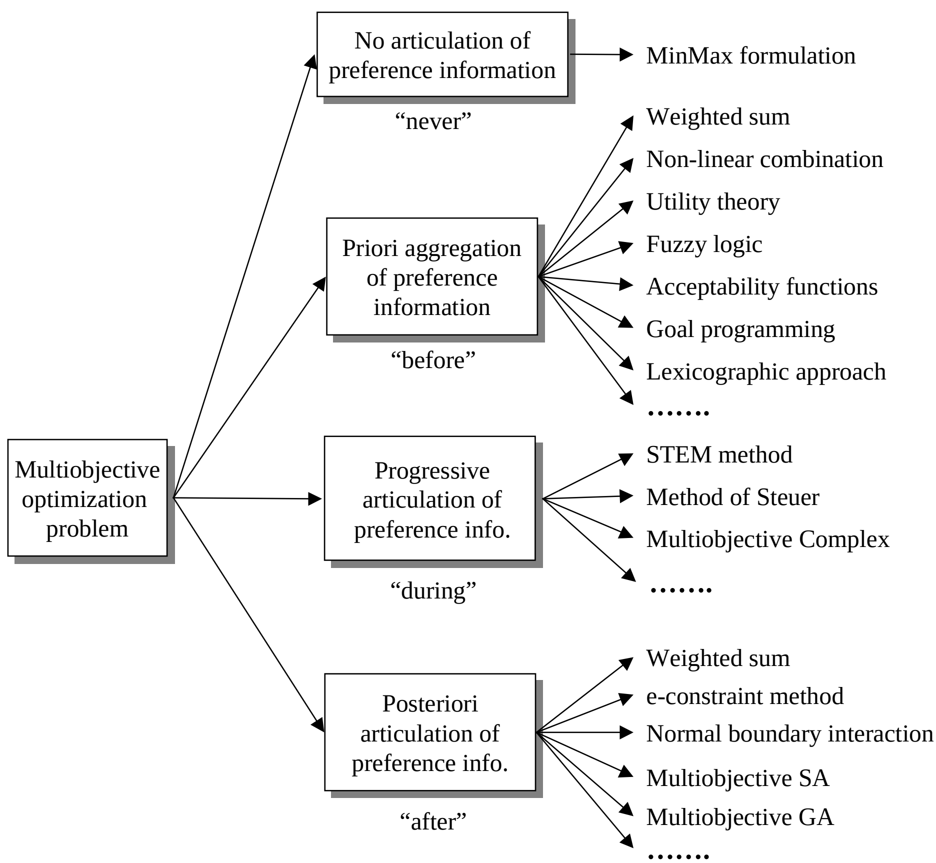


Figure 8. A classification of some methods to conduct multiobjective optimization, after Hwang et al. [40].

Although the classification gives a far from complete description of all available techniques to conduct multiobjective optimization, it constitutes a good framework for a discussion of the most common methods suitable for engineering optimization. An interesting observation that one can make is that the later the decision-maker articulates his preference among the objectives the more the problems moves from being specification driven to prototype driven. A more thorough discussion of these methods is presented in Paper [XIII].

4.4.1 No preference articulation

There are methods that do not use any preference information, e.g. the Min-Max formulation and global criterion method, see Hwang et al. [40], Osyczka [61] and Steuer [82].

The Min-Max formulation is based on minimization of the relative distance from a candidate solution to the utopian solution $\mathbf{F}^*$, see Figure 7. The distance between a solution vector and the utopian vector is typically expressed as a $\mathcal{L}_p$-norm.

4.4.2 Priori articulation of preference information

The most common way of conducting multiobjective optimization is by priori articulation of the decision-maker's preferences. This means that before the actual optimization is conducted the different objectives have to be aggregated to one single objective function. This can be done in many ways; some of which are shown in Figure 8.

Weighted-sum approaches

The easiest and perhaps most widely used method is the weighted-sum approach. The objective function is formulated as a weighted $\mathcal{L}_1$-metric, see equation (5).

$$\min \sum_{j=1}^{k} \lambda_j f_j(\mathbf{x}) \tag{5}$$

$$\text{s.t.} \quad \mathbf{x} \in S$$

$$\lambda \in \mathrm{R}^k \,\middle|\, \lambda_i > 0, \sum \lambda_i = 1$$

By choosing different weightings, λ_i, for the different objectives, the preference of the decision-maker is taken into account. As the objective functions are generally of different magnitudes, they might have to be normalized first. Although the formulation is simple, the method is somewhat ad-hoc, as there is no clear relation between the weightings and the obtained solution. How to determine the weightings from the decision-maker's preferences is also an ad-hoc procedure. Another drawback of this method is that it is not possible to locate solutions at non-convex parts of the Pareto-front. These drawbacks are discussed in Das and Denis [17], and Steuer [82].

Non-linear approaches

Although many methods might be referred to as non-linear, e. g. all the ones mentioned hereafter, it here refers to higher order of $\mathcal{L}$p-metrics formulations and their equals. Equation (6) below represents one such approach presented by Krus et al. [48].

$$\min \sum_{j=1}^{k} \left(\frac{f_j(\mathbf{x})}{f_{j0}} \right)^p \tag{6}$$

$$s.t. \quad \mathbf{x} \in S$$

In this formulation each objective is normalized by f_{j0}, which represents the value of *j:th* objective for an initial solution. For consistency the whole expression could be raised to the power of *1/p*. The exponent *p* expresses how much an improvement in f_i is worth and how much a poorer value penalizes the overall objective function. The graph in Figure 9 below depicts $(f_i / f_{i0})^p$ as a function of f_i.

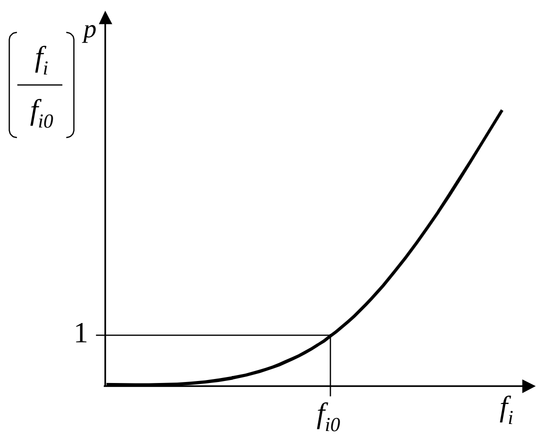


Figure 9. The form $(f_i / f_{i0})^p$ as a function of f_i. *for* p=3.

Advantages of this method are that the objectives do not need to be calculated in absolute terms, since the problem is formulated as to achieve improvement from an initial state. It is an attractive method from an engineering perspective, as it expresses what is wanted from the system with a minimum of parameters, compared to the more rigorous methods discussed below. This method has been applied in Paper [II], [III], [VIII] and [IX], and will be discussed further in chapter 6.

Utility theory and related methods

Utility theory forms the basis of decision-making and dates back to von Neumann and Morgenstern in 1944 [92], although the basic reference can be found in Keeney and Raiffa [43]. In utility theory, a utility function is established which expresses the value of the design as a function of each attribute. An extensive interaction with the decision-maker is needed in order to determine the utility function. A formal method that allows this is presented by Thurston [90]. However, in order achieve an overall utility function, a set of assumptions has to be made. For instance, it is usually assumed that the different utilities are mutually independent, and either additive or multiplicative. Although utility theory is very mathematically rigorous, it is not widely used in engineering optimization, because of the great effort needed to establish the utility functions.

Many methods have been developed which address the shortcomings of utility theory, e.g. fuzzy logic approaches [98] and the method of using acceptability functions, see Wallace et al. [93].

The method of acceptability functions is a goal-oriented design evaluation model that employs the same goals and targets that are commonly used in engineering design to evaluate the performance of each solution. The acceptability function represents the subjective probability that a designer will accept a design based upon each objective. This is explained in Figure 10, where the acceptability function for a fluid power system is expressed as function of the losses in the system.

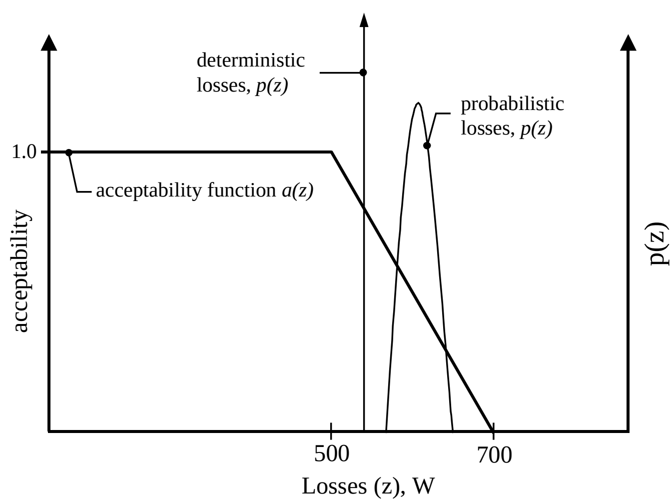


Figure 10. Acceptability function as well as probability density functions as a function of the losses in a fluid power system.

a(z) is the acceptability function which defines the probability that different levels of the performance characteristic *z* will be acceptable. A system with losses below 500 W has a probability of 1.0 of being accepted, whereas a system with losses of 700 W is surely rejected. The function *p(z)* is a probability density function of unit area which quantifies the design's performance for the characteristic *z*. This formulation allows the designer to quantify a design's performance either deterministically or probabilistically. In the deterministic case, the probability density function is an infinite spike, with the area unity. In the framework presented in this thesis, the performance of a design is deterministic, determined by the outcome of the simulation.

The probability, P_i, of accepting the design based upon the *i:th* characteristic is expressed in equation (7), and the overall probability of accepting the design based on all objectives is calculated by multiplying the individual probabilities, see equation (8). The optimization problem is then formulated so as to maximize the probability of acceptance, see equation (9).

$$P_i = \int_x a(z)\, p(z)\, dz \tag{7}$$

$$P_{acc} = \prod_{i=1}^{k} P_i \tag{8}$$

$$\begin{aligned} &\max \ P_{acc}(\mathbf{x}) \\ &s.t. \ \ \mathbf{x} \in S \end{aligned} \tag{9}$$

4.4.3 Progressive articulation of preference information

Methods that rely on progressive articulation of preference information are generally referred to as interactive methods. These methods work according to the hypothesis that the decision-maker is unable to indicate preferences information 'a priori' due to the complexity of the problem. However, as the search moves on and the decision-maker learns more about the problem, he/she is capable of giving directions in which to look for improvements. Advantages of these types of methods are:

- there is no need for 'a priori' preference information,
- only local preference information is needed,
- it is a learning process where the decision-maker gets a better understanding of the problem,
- as the decision-maker takes an active part in the search it is more likely that he accepts the final solution.

The disadvantages are:

- The solutions are depending on how well the decision-maker can articulate his preferences
- A high effort is required from the decision-maker during the whole search process.
- The solution is depending on the preferences of one decision-maker. If the decision-maker changes his preferences or if there is a new decision-maker, the process has to be restarted.

- The required computational effort is higher than in the previous methods.

These methods usually progress by either changing the weights in a weighted-sum approach, e.g. the method by Steuer and Choo [83], or by progressively reducing the search space. In the STEM method [4], an initial Pareto optimal solution is found by a weighted-sum approach. The decision-maker then has to determine a relaxation to some objectives in order to achieve improvements in others. The relaxed objectives are moved from the objective function and added as constraints to limit the solutions space.

These types of methods are not commonly used in engineering optimization, at least not in connection with non-gradient methods. Interactive methods that use gradient information are more widely used, see for example the method presented by Tappeta et al. [88].

There are also other types of iterative methods, such as interactive genetic algorithms. Genetic algorithms (GAs) evolve a design by mimicking natural selection, see chapter 5.2. In interactive methods, the designer/decision-maker selects the best design in each evolution step to form the basis for further breeding. These methods are well suited for problems where the objectives are very hard to express mathematically. The most famous example is of course Richard Dawkin's *The Blind Watchmaker* [18], whereas a similar method has also been developed by Smith [77]. A more recent example is Smyth and Wallace [80] were a GA is employed to evolve aesthetic product forms.

A multiobjective interactive complex method has been developed as well, see Ringuest [69]. In contrast to interactive GAs, the decision-maker is asked to point out the worst solution, which is then reflected through the centeroid of the complex according to the normal procedure of the method (see chapter 5.1).

4.4.4 Posteriori articulation of preference information

There are a number of techniques that allow to first search the solution space for a set of Pareto optimal solutions and then present them to the decision-maker. The big advantages with these types of methods are that the solutions are independent from the decision-maker's preferences. The analysis has only to be performed once, as the Pareto set would not change as long as the problem description remains unchanged. However, some of these methods suffer from a large computational burden. Another disadvantage might be that the decision-maker has too many solutions to choose from. There are however methods that support screening of the Pareto set in order to cluster optimal solutions that have similar properties, see Morse [57] and Rosenmann and Gero [71].

By sampling a set of discrete points on the Pareto front the decision-maker could get a feeling for the form of the front and thereby the possible trade-off between the objectives. The simplest way of doing this is to repeatedly change the weightings in a weighted sum. There are however some disadvantages associated with this approach. Linear combinations of the objectives cannot produce solutions on non-convex parts of the Pareto front. Furthermore, there is no guideline as how to choose the weightings to ensure an even spread on the Pareto front, see Das and Dennis [17].

An approach that overcomes these drawbacks is normal boundary interaction presented by Das and Dennis [16]. Another approach is the e-constraint method where one objective is selected for optimization and the others are reformulated as constraints. By progressively changing the constraint values, different points on the Pareto front could be sampled. These methodsare generally computational expensive as each point requires a completely new optimization run.

An appealing thought is to be able to conduct just one optimization and still sample a set of discrete points on the Pareto front. As genetic algorithms manipulates a population as they evolve they can be modified to accomplish this. These modified genetic algorithms are known as multiobjective genetic algorithms and will be discussed in chapter 5.3. Within this thesis, a multiobjective genetic algorithm is developed, see Paper [IV], and later applied to the design of fluid power systems in Paper [V] and [VI].

In the literature as well as in this thesis, the focus is on problems with only two objectives, because they are so much easier to visualize. 3-Dimensional problems could be visualized, but for problems of higher dimensions, other techniques must be applied. One way is to aggregate some objectives e.g., energy consumption and cost could be aggregated to an overall cost objective. Another possibility is to cluster solutions in regions with different properties and present them to the decision-maker, who has to point out the most interesting regions where we should concentrate our efforts. Methods that visualize the properties of problems with many objectives are discussed briefly in chapter 8.

5 Optimization methods

OPTIMIZATION METHODS COULD be divided into derivative and non-derivative methods. This thesis focuses on non-derivative methods, as they are better suited for general design problems. Non-gradient methods are more robust in locating the global optima and are applicable in a broader set of problem areas, see Goldberg [28]. Another advantage of non-derivative methods is that they do not require any derivatives of the objective function in order to find the optimum. Hence, they are also known as black-box methods. Here the objectives are results of complex computer simulations, thus the derivatives of the objective function are not explicitly known. The disadvantages are however that we cannot prove that we have found the actual optima. This is partly true for gradient methods also as they might get caught in local optima. By conducting several optimizations with different initial conditions, it could be made probable that the global optimum is truly found. Another disadvantage with non-gradient methods is that they usually require more function calls than gradient methods, and are thus more computational expensive. However, as the capacities of the computers are increasing this disadvantage is diminishing. Furthermore, most non-gradient methods are well suited for implementation on parallel processors.

There is a large number of non-derivative methods. For example, the Complex method developed by Box [8] in the 60’s, genetic algorithms [35] or the similar evolutionary algorithms [68], both developed in the early 70’s by Holland and Rechenberg respectively. Simulated annealing was then developed by Kirkpatrick [44] in the early 80’s. Methods that are more recent include Tabu search, developed by Glover [27] in 1989, which have been applied to the design of fluid power circuits by Connor and Tilley [14]. Apart from these methods, there are also other promising techniques to conduct engineering optimization, for instance response surface approximations [55], as well as Taguchi methods [13].

In most comparison studies different methods come out on top depending on the problem and how well the different methods were tuned to fit that particular problem. Comparative studies of different types of non-derivative methods could be found in for instance Borup [6], Hajela [31], Jansson [42] and Mongeau [56]. In this thesis, genetic algorithms and the Complex method are applied. Genetic algorithms because they are known to be most robust in finding the global optimum and they have the broadest field of applications. The Complex method is employed because it is fast, but also because it is easy to implement and parameterize, see Borup [6] and Jansson [42].

In order to bring out what is best in each method, there is a vast amount of hybrids, both between gradient and non-gradient methods but also between different non-gradient methods, see Yen [97].

5.1 The Complex method

The Complex method was first presented by Box [8], and later improved by Guin [30]. The method is a constraint simplex method developed from the Simplex method by Spendley et al [81] and Nelder Mead [59]. Similar related methods are named Nelder-Mead Simplex and flexible polyhedron search. These methods also have similar properties, although the Complex performs slightly better.

In the Complex method, a complex consisting of several possible problem solutions (sets of design parameters) is manipulated. Each set of parameters represents one single point in the solution space. The number of points in the Complex, *m* has to be greater than the number of optimization parameters, i.e. $m \geq n+1$. Typically, the complex consists of twice as many points as the number of optimization parameters. The starting points are randomly generated, and it is checked that both explicit and implicit constraints are not violated. Let the points x^h and x^l represent the points with maximal and minimal function values. The centroid, $\bar{x}$ is calculated according to equation (10).

$$\bar{x} = \frac{1}{n-1}\sum_{i=1}^{n} x^i \ , \quad x^i \neq x^l \tag{10}$$

The main idea of this algorithm is to replace the worst point by a new and better point. The new point is calculated as the reflection of the worst point through the centroid of the remaining points in the complex, see equation (11).

$$x^r = \bar{x} + \alpha\left(\bar{x} - x^l\right) \tag{11}$$

The reflection coefficient, $\alpha = 1.3$ according to Box. The point x^r is examined with regard to explicit and implicit constraints, and if it is feasible, it replaces x^l unless it repeats being the worst. In that case, it is moved halfway towards the centroid of the reaming points. This is repeated until it stops repeating the worst point. However, as pointed out by Guin this can not handle the situation where there is a local minimum at the centeriod. Here the method has been modified so that the point is gradually moved towards the maximum value if it continues to be the lowest value, see Krus et al. [46]. This might however lead to the two points coming very close to each other, with a risk of collapsing the complex. Therefore, a random value is also added to the new point. In this way the algorithm will take some extra effort in searching for a better point in the neighborhood of the maximum value. The modified algorithm could be described according to equation (12).

$$x^{r(new)} = \left[x^{r(old)} + \varepsilon \bar{x} + (1-\varepsilon) x^h \right] / 2 + \left(\bar{x} - x^h \right)(1-\varepsilon)(2R-1)$$

$$\text{where } \varepsilon = \left(\frac{n_r}{n_r + k_r - 1} \right)^{\frac{n_r + k_r - 1}{n_r}} \qquad (12)$$

k_r is the number of times the point has repeated itself as lowest value, and n_r is a constant. Here $n_r = 4$ has been used. R is a random number in the interval [0,1]. If the point violates an implicit constraint, the reflection follows a similar scheme. An example of the complex method is shown in Figure 11 below for a two dimensional parameter space. The circles in the graph indicate the objective function value for different solutions, with the best value in the middle.

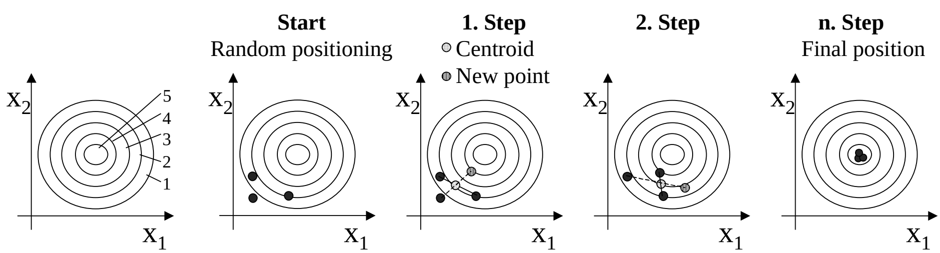


Figure 11. The progress of the Complex method for a two dimensional example, with the optimum located in the middle of the circles.

The Complex method has been applied to a wide range of problem areas such as physics [72], structural engineering [26] and [32], fluid power system design [42], aerospace engineering [49], and others [53] and [54]. The Complex method was originally developed for problems with continuous variables but Haque [32] has shown that the Complex method could also be applied to mixed continuous and discrete variable problems.

5.2 Genetic algorithms

Genetic algorithms (GAs) and the closely related evolutionary algorithms are a class of non-gradient methods which has grown in popularity ever since Rechenberg [68] and Holland [35] first published their work on the subject in the early 70's. For a more comprehensive study of genetic algorithms, see Goldberg's [28] splendid book on the subject.

Genetic algorithms are modeled after mechanisms of natural selection. Each optimization parameter (x_n) is encoded by a gene using an appropriate representation, such as a real number or a string of bits. The corresponding genes for all parameters $x_1,..x_n$ form a chromosome capable of describing an individual design solution. A set of chromosomes representing several individual design solutions comprises a population where the fittest are selected to reproduce. Mating is performed using crossover to combine genes from different parents to produce children. The children are inserted into the population and the procedure starts over again, thus creating an artificial Darwinian environment as depicted in Figure 12.

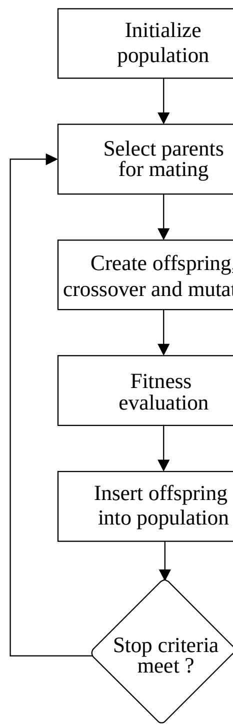


Figure 12. An example of a simple genetic algorithm. More sophisticated algorithms include other techniques in order to enhance the performance of the algorithm.

5.3 Multiobjective genetic algorithms

When the population of an ordinary genetic algorithm is evolving, it usually converges to one optimal point. It is however tempting to adjust the algorithm so that it spreads the population over the entire Pareto optimal front instead. As this idea is quite natural, there are many different types of multiobjective genetic algorithms. For a review of genetic algorithms applied to multiobjective optimization, readers are referred to work by Fonseca and Fleming [23]. Literature surveys and comparative studies on multiobjective genetic algorithms are also provided by several other authors, see Coello [11], Horn [37], Tamaki et al. [87] and Zitzler and Thiele [99]. In Paper [IV] a discussion of some of the most common algorithms is presented. Here just the multiobjective GA (MOGA) is described, since it is one of the cornerstones of the new multiobjective genetic algorithm being proposed.

In the MOGA presented by Foseca and Fleming [24] and [25], each individual is ranked according to their degree of dominance. The more population members that dominate an individual, the higher the ranking for the individual. An individual's ranking equals the number of individuals that it is dominated by plus one (see Figure 13). Individuals on the Pareto front have a ranking of one, as they are non-dominated. The rankings are then scaled to score individuals in the population. In MOGA both sharing and mating restrictions are employed in order to maintain population diversity. Fonseca and Fleming also include preference information and goal levels to reduce the Pareto set to those that simultaneously meet certain attribute values.

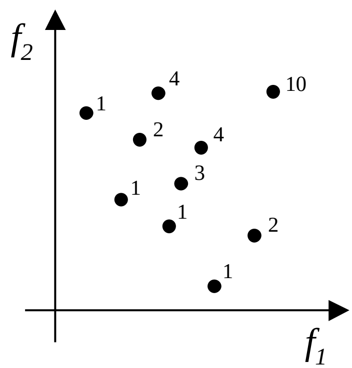


Figure 13. Population ranking according to Fonseca and Fleming.

Although there is a substantial body of research on Pareto multiobjective genetic algorithms, there are still important issues that current methods address with only partial success. The methods typically require extensive genetic algorithm parameter tuning on a problem-by-problem basis in order for the algorithm to perform well. However, in a real-world problem there is little knowledge about the shape of attribute space, which makes it difficult to assess problem-specific parameters. Additionally, existing methods do not handle the location of multiple Pareto frontiers in multi-modal problem spaces consistently. This thesis attempts to develop a reliable algorithm that distributes solutions evenly across Pareto frontiers in a variety of multi-modal problems without problem-specific tuning.

5.4 A new multiobjective genetic algorithm

The multiobjective struggle genetic algorithm (MOSGA) combines the struggle crowding genetic algorithm presented by Grüninger and Wallace in [29] with Pareto-based ranking as devised by Fonseca and Fleming in [24].

In the struggle algorithm, a variation of restricted tournament selection, Harik [33], two parents are chosen randomly from the population, and crossover/mutation is performed to create a child. The child then has to compete with the most similar individual in the entire population, and replaces it if the child has a better fitness. This replacement strategy counteracts genetic drift that spoils population diversity. To assure diversity in population is necessary in order to avoid inbreeding, and to spread the population evenly on the Pareto front.

There is no single objective function to determine the fitness of the different individuals in a Pareto optimization. Therefore, the ranking scheme presented by Fonseca and Fleming [24] is employed. Each individual is given a rank based on the number of individuals in the population that are preferred to it, i.e. for each individual the algorithm loops through the population counting the number of preferred individuals. "Preferred to" could be implemented in a strict Pareto optimal sense or extended to include goal levels on the objectives in order to limit the frontier.

The principle of the MOSGA algorithm is outlined below.

Step 1: Initialize the population.

Step 2: Select parents randomly from the population.

Step 3: Perform crossover and mutation to create a child.

Step 4: Calculate the rank of the child, and a new ranking of the population that considers the presence of the child.

Step 5: Find the most similar individual, and replace it with the new child if the child's ranking is better.

Step 6: Update the ranking of the population if the child has been inserted.

Step 7: Perform steps 2-6 until the mating pool is filled.

Step 8: If the stop criterion is not met go to step 2 and start a new generation.

In order to assess the performance of the algorithm a set of test problems from Deb [19] was explored.

5.4.1 Test Function

Deb developed a set of problems to highlight difficulties that multiobjective genetic algorithms may encounter. For visualization reasons, the focus is on two-dimensional problems defined generally by equations (13) and (14).

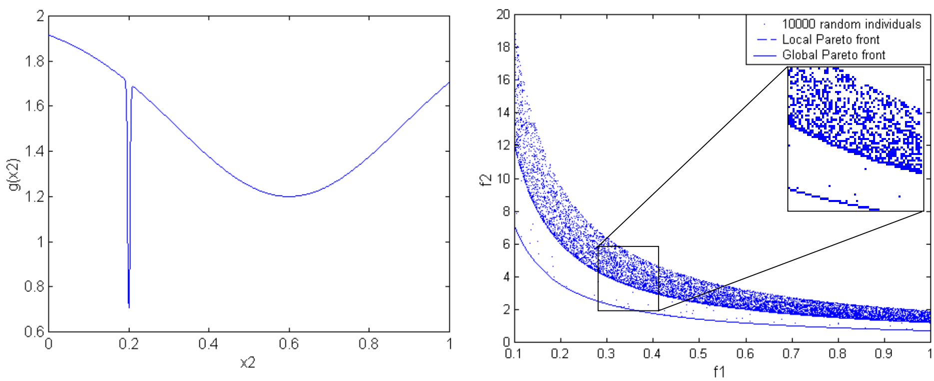
$$f_1(x_1, x_2) = x_1 \tag{13}$$

$$f_2(x_1, x_2) = \frac{g(x_2)}{x_1}, \quad g(x_2) > 0, \quad x_1 > 0 \tag{14}$$

If the function g is multi-modal, the corresponding multiobjective problem will have global and local Pareto-optimal frontiers. A multi-modal g function is defined in equation (15).

$$g(x_2) = 2 - \exp\left\{-\left(\frac{x_2 - 0.2}{0.004}\right)^2\right\} - 0.8\exp\left\{-\left(\frac{x_2 - 0.6}{0.4}\right)^2\right\} \tag{15}$$

Figure 14(a) shows the *g* function for $0 \leq x_2 \leq 1$ with the global optimum located at $x_2=0.2$ and a local optimum at $x_2=0.6$. Figure 14(b) shows a plot of f_1 and f_2 in the attribute space with the global and local Pareto optimal solutions. 10000 randomly chosen solutions are generated and plotted in Figure 14(b) to illustrate that the problem is biased—the solution density is higher towards the local Pareto-optimal front.



(a) (b)

Figure 14. (a) shows the multi-modal function $g(x_2)$, where the global optimum is situated at $x_2=0.2$ and the local optimum at $x_2=0.6$. For the multiobjective problem, a f_1-f_2 plot for 10000 random solutions is shown in (b). Note the low solution density at the global Pareto optimal front.

The optimization was conducted with a population size of 60 individuals and ran for 200 generations. Deb reported that the NSGA was trapped in the local Pareto front in 59 out of 100 runs. The original MOSGA algorithm used an attribute based distance function resulting in the algorithm converging to the local Pareto front in only 7% of 100 optimizations. The algorithm found the preferred global Pareto optimal front in 86% of the optimizations, as shown in Figure 15(a) and (b). In 7% of the optimizations, it converged to both frontiers. Thus, the MOSGA seems more robust in locating the global Pareto optimal front.

However, the algorithm should ideally be capable of identifying both fronts in every optimization run. By changing to a parameter based distance function this can be achieved. However, the parameter based distance function was slower and less exact in its convergence to the frontier.

In the MOSGA, the new child has to compete with the individual most similar to itself. When the comparison is done in parameter space, a portion of the population will find and maintain local optima, where solutions close in the parameter space are all dominated. When using an attribute based distance function, solutions at local optima might have to compete with solutions at the global optima, as they might be close in attribute space. Therefore, local optima would not be maintained.

By combining equally weighted attribute-based and a parameter-based distance function to form a mixed distance measure, the advantages of fast convergence and the ability of finding multiple fronts were realized. Figure 15 shows how the algorithm spreads the population evenly on both fronts when using the mixed distance function. To summarize, the attribute distance function performs well on problems with one Pareto front. For problems with multiple frontiers, a mixed distance function is preferred. A more detailed discussion about the properties of the algorithm is given in Paper [IV].

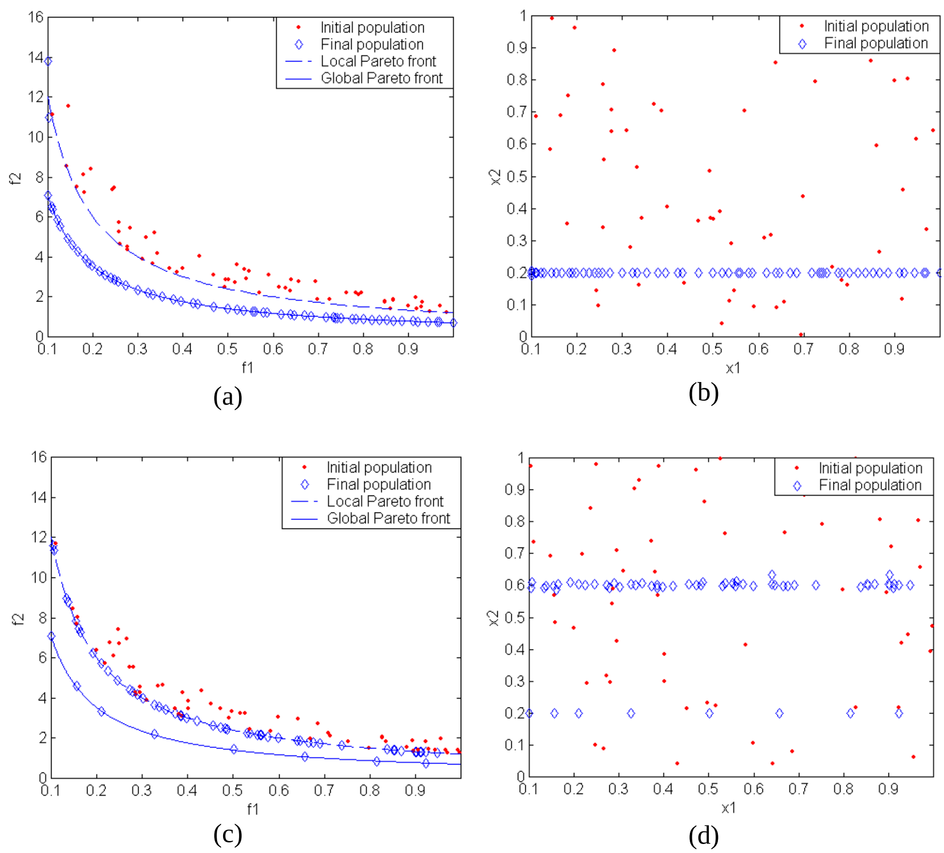


Figure 15. Optimization results using different distance functions. In (a) and (b) an attribute-based distance function is used and the population has converged to the global Pareto front. In (c) and (d) the mixed distance function is used and the population converges to both the global and the local front. (a) and (c) show the result in attribute space, whereas (b) and (d) show the result in parameter space.

The method is capable of reliably identifying multiple Pareto fronts in a single optimization run, thus outperforming other techniques. For an engineering problem, the optimization formulation is often a simplification of the real world problem, which in part requires human or inquantifiable judgement. When deciding upon the final design there are usually more criteria to consider than just the optimization objectives, e.g. the robustness of the system. Therefore, knowledge of the existence of local Pareto optimal solutions is very valuable.

Another advantage of the proposed method is that it does not require problem specific parameter settings. The only GA parameters that have to be determined are population size and number of generations. This is an important strength, cause in real word problems there are little or no knowledge about the properties of the attribute space. Therefore, determination of problem specific parameters might be a tedious task. The method has thereby fulfilled the ambitions as it performs well, is robust and is to easy use on a wide range of problems. This will be exemplified in the upcoming application chapter.

6 Applications

THIS CHAPTER DESCRIBES in principle two different applications of the described optimization techniques to engineering design problems, mainly in the field of fluid power. As these real design problems are discussed, new techniques will be introduced that support the employment of optimization to engineering design problems. In the first application, the multiple objectives are aggregated to one figure of merit before the optimization is conducted, whereas the second application utilizes the proposed multiobjective genetic algorithm. For each problem, a simulation model is established in order to predict the properties of different design proposals. Therefore, the optimization strategy has to be connected to the simulation environment.

The first application is the design of a landing gear system gathered from Paper [II]. Here a method is presented that structures the optimization problem with the help of the so-called Design Structure Matrix (DSM). In addition, the relationship matrix from the House of Quality is introduced in order to support the formulation of the overall objective function.

The second application constitutes two concepts of hydraulic actuation systems, gathered from Paper [V] and [VI]. The different concepts are studied with the help of Pareto optimization, and it will be shown how optimization could be utilized to facilitate concept selection. Real design problems usually show a mixture of determining continuous parameters as well as selecting existing components from catalogs or databases. Therefore, the multiobjective genetic algorithm has been extended to handle a mixture of continuous variables as well as discrete catalog selections.

6.1 Landing gear system

The landing gear concept chosen for this study is shown in Figure 16. The landing gear system consists of the actual landing gear, the hydraulic actuator that creates the retardation movement and the hydraulic supply system. This study focuses on a concept with a local hydraulic system that supports the landing gear system only. This is not common in today's aircraft. However, the trend in modern aircraft design is towards more decentralized hydraulic systems, with locally powered actuation systems situated directly at the various control surfaces, as discussed in Paper [III] and [IX]. The landing gear system would therefore need its own hydraulic power supply.

In order to analyze the behavior of the landing gear system, a simulation model was established. The most natural choice for the hydraulic part was to make the model in the HOPSAN simulation package. For the mechanical structure on the other hand, there is a choice of modeling it in HOPSAN or some other modeling environment. Since HOPSAN's standard library only contains some basic mechanical components, it would require quite a modeling effort to model the mechanical structure. Therefore, the geometric model of the landing gear was implemented in the simulation package Pro Mechanica Motion, which is a Multi-Body-Simulation (MBS) environment. This model includes all mechanical parts and external forces such as landing gear drag.

The different simulation models and their interconnection are shown in Figure 16. A more detailed picture is given in Figure 1 in Paper [II]. Figure 16 also shows how the optimization strategy is connected to the simulation environment. The optimization method, in this case the Complex method, generates a set of system parameters, which are fed to the simulation model. The system is then evaluated and the resulting system characteristics are sent to the optimization strategy, which calculates a new objective function value. Then the procedure starts over again and the complex method generates a new set of system parameters, resulting in a system with better characteristics.

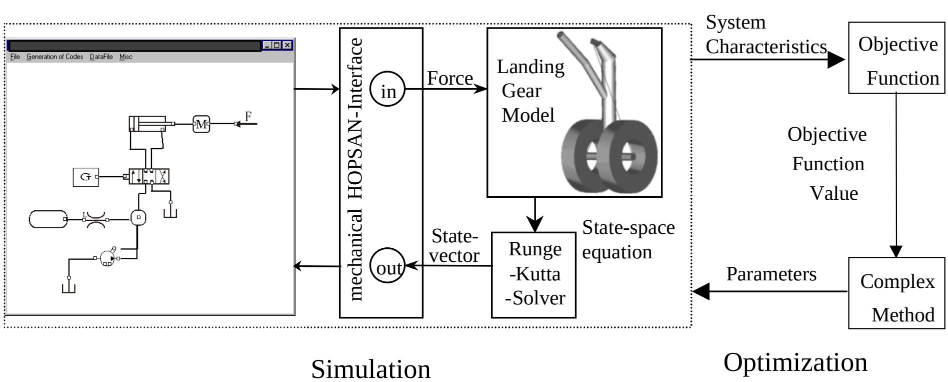


Figure 16. The coupling between HOPSAN, the landing gear subroutine and the optimization strategy.

6.1.1 Methods that support objective function formulation

This chapter deals with reformulating the design problem as an optimization problem introducing support tools such as the Design Structure Matrix (DSM) and the relationship matrix from the House of Quality method. These methods were applied in Paper [II], [III], [VIII] and [IX] in order to support the formulation of the objective function. Here the DSM is employed to choose relevant optimization parameters. Then, the relationship matrix supports the formulation of the objective function depending on the parameters chosen. Furthermore, the method is self-documenting.

Design structure matrix

The design structure matrix was originally developed by Stewart [84], but has been further developed by for instance Eppinger et al. [20]. Eppinger et al. extended the original binary DSM with measures of the degree of dependency between different tasks, and included the task duration to the matrix. Here however, the matrix is employed in its binary form.

The DSM is often used to visualize information requirements between various teams involved in the development process. It can also be used to describe couplings between different system parameters, and thereby support the selection of optimization parameters.

The strength of the design structure matrix is that it visualizes the couplings between different design parameters. The crosses in each row identify which parameters that have to be determined before that particular parameter could be decided upon. The dots in the matrix represent the decision on each particular parameter and are only included in order to separate the upper and lower triangles of the matrix, see Figure 17.

	Landing gear leg					Movement control								
						Actuation				Supply				
	A	B	C	D	E	F	G	H	I	J	K	L	M	
A	●	X												Wheel diameter
B	X	●												Number of wheels
C	X		●	X										Leg length
D			X	●										Leg diameter
E			X	X	●									Supporting beam
F	X	X	X	X		●	X	X		X		X		Length of lever
G	X	X	X	X		X	●	X		X		X		Angle of lever
H	X	X	X	X		X	X	●		X				Piston Area
I	X		X			X	X		●					Piston stroke
J								X		●				System pressure
K								X	X	X	●			Valve size
L								X	X			●	X	Pump size
M								X	X	X		X	●	Accumulator size

Figure 17. Design structure matrix for the landing gear system.

In an ideal situation all the crosses would be situated underneath the diagonal, which indicates that the system can be designed in a top-down fashion. This is often not the case. On the contrary, engineering system design is often very iterative which results in numerous crosses above the diagonal, see Figure 17.

An example from the DSM in Figure 17 is that the piston area (H) cannot be designed without knowing the diameter (A) and the number (B) of the wheels, the length (C) and diameter (D) of the leg, as well as the length (F) and the angel (G) of the lever. On the other hand, the designer needs to know the piston area and the lever angle in order to decide on the lever length. Thus, the parameters F, G and H are tightly coupled with the parameters A, B, C, D as input parameters.

As can be seen from Figure 17, the landing gear development process is block-lower- triangular. The blocks indicate design tasks that are executed in sequence. The marks above the diagonal in each block indicate feedback that leads to a strong coupling between the parameters within that particular block. The two larger blocks are added in order to visualize the major design activities. Three main activities can be derived from the DSM, namely the landing gear leg design, the actuation design and the design of the hydraulic supply system.

As can be seen from the DSM, the parameters for the system that creates the movement are totally dependent on the landing gear parameters but not vice versa. This indicates that these two design processes can be executed in sequence. Optimization of the movement control parameters can therefore be conducted with constant values of the landing gear leg parameters without running the risk of sub-optimization. This is no limitation or simplification of the actual design problem since the design processes are done purely sequentially.

Thus, the DSM supports choosing optimization parameters and thereby demarcates the optimization problem. At first sight, one could think that every system parameter should be included as optimization parameter in order to avoid sub-optimization. This is not necessarily the case and the DSM could support the designer in selecting the appropriate optimization parameters, see also Pohl et al. [65].

Relationship matrix

The House of Quality method has proven to be a useful tool in providing means for the translation of customer requirements into critical product control characteristics, as stated by for instance Cohen [12], Hauser and Clausing [34] and Sullivan [85]. Here the relationship matrix from the House of Quality method is employed in order relate system requirements, or system characteristics, to system parameters. Figure 18 shows how this can be done for the landing gear system.

			System Parameters													
			Landing gear						Movement control							
				Wheel			Landing gear leg				Actuator		Hyd. Supply			
		Wheighting	Supporting beam	Wheel dimensions	Number of wheels	Brakes	Leg length	Leg diameter	lever length	Lever angle	Piston area	Piston stroke	Valve size	Pump size	Accumulator size	Pressure level
System Characteristics	Descending velocity	D					5	5								
	Retraction time	D							5	5	5	5				
	Weight	3		5	5	5	5	5			1	1	1	3	3	
	Energy consumption	3		5	5	5	5	5	1	1	3	3	3	3	3	3
	Price	1	3	3	3	5	5	5			1	1	1	3	1	
	Ground carrying capacity	D		5	5											
	Locking landing gear	D	5													
	Brake distance	D				5										

Figure 18. Relationship matrix for the landing gear system.

First the requirements or characteristics of the system are established and listed on the vertical axis of the relationship matrix. The characteristics are then given a weighting, indicating their relative importance. The higher the figure the more important the characteristic. The letter "D" indicates that this particular characteristic is a demand that just has to be fulfilled. The demands are usually connected to a constraint in the objective function formulation.

Then the system parameters are listed on the horizontal axis. The relationship matrix evolves in a team activity where engineers from different disciplines relate the system parameters to the system characteristics. The strength of the relationship matrix lies in the visualization of the relation between system characteristics and system parameters. The figures in the matrix express how strong the relation between one particular system characteristic and a system parameter is. A high figure indicates a strong relationship. The boxes that lack a figure are as important as the other ones, since they indicate a missing relationship. A further advantage of the relationship matrix lies in the fact that it fosters team activities, which is crucial when weighting system characteristics against each other and expressing the strength of the different relationships.

An example from Figure 18 shows that with the movement control parameters as optimization parameters, the retraction time, weight, cost and energy consumption are the system characteristics that we can influence. Therefore, these characteristics should be reflected in the objective function. For a more detailed discussion, see Paper [II].

6.1.2 Objective function formulation

The design problem can be described as a multi-variable constrained optimization problem. The problem is to minimize the function

$$\begin{gathered} F\left(x_1, x_2, ..x_n\right) \\ \mathbf{x} \in S \end{gathered} \tag{16}$$

The objective function has to reflect all relevant system characteristics found with the help of the relationship matrix. Each system characteristic, f_i, is expressed as a function of the optimization parameters, i.e.

$$f_i = f_i\left(x_1, x_2, ..x_n\right), \tag{17}$$

either explicitly or implicitly through the simulation, see equation (1)-(3) in Paper [II]. When the objective is to minimize the system characteristics $f_1, f_2, .., f_i$, the objective function F can be represented as

$$F = \left(\frac{f_1}{f_{10}}\right)^{\gamma_1} + \left(\frac{f_2}{f_{20}}\right)^{\gamma_2} + .. + \left(\frac{f_i}{f_{i0}}\right)^{\gamma_i} \tag{18}$$

where $f_{10}, .., f_{i0}$ are the function values from an evaluation of one initial acceptable system, i.e.

$$f_{i0} = f_i\left(x_{10}, x_{20}, ..x_{n0}\right) \tag{19}$$

where $x_{10}\ldots,x_{n0}$ are the parameters of the initial solution. $\gamma_1,..,\gamma_i$ represent the relative importance of the different objective functions and can be expressed as functions of the weighting factors from the relationship matrix in Figure 18, one example is shown in equation (20).

$$\gamma_i = 1 + \frac{w_i + 1}{2} \tag{20}$$

The functions $f_{10},..,f_{i0}$ normalize the different characteristics and reduce the problem to finding one acceptable solution that the optimization strategy tries to improve. Another advantage with this formulation is that each characteristic could be expressed in relative terms.

In order to handle design constraints a constraint polynomial is multiplied with the original objective function. Thus the objectives and the constraints are separated in the objective function, see equation (21).

$$F = \underbrace{\left[\left(\frac{f_1}{f_{10}}\right)^{\gamma_1} + \left(\frac{f_2}{f_{20}}\right)^{\gamma_2} + .. + \left(\frac{f_i}{f_{i0}}\right)^{\gamma_i}\right]}_{\text{Objectives}} \cdot \underbrace{\left[(1+c_1)^{\alpha_1} + (1+c_2)^{\alpha_2} + .. + (1+c_j)^{\alpha_j}\right]}_{\text{Constraints}} \tag{21}$$

In equation (21), c_j is a function that equals zero if the *j:th* constraint is not violated and significantly greater than unity otherwise. The exponent α_j indicates the strength of the *j:th* constraint.

The landing gear system was optimized using this objective function formulation and the result is shown in Figure 19, please not that the problem was reformulated from min(F) to the equivalent max(-F). For a more thorough discussion of the optimization result, see Paper [II].

This objective function formulation has successfully been employed in Paper [II] and [VIII] in order to support the design of the landing gear system. In Paper [III], [IX] and [XI] this formulation is employed in order to optimize two different concepts of electro-hydrostatic actuation systems for aircraft applications. These systems constitute a good example of multi-domain systems as they combine electrical, hydraulical and mechanical sub systems. The different systems have been studied during authentic duty-cycles and optimization has been introduced in order to form the basis for concept selection.

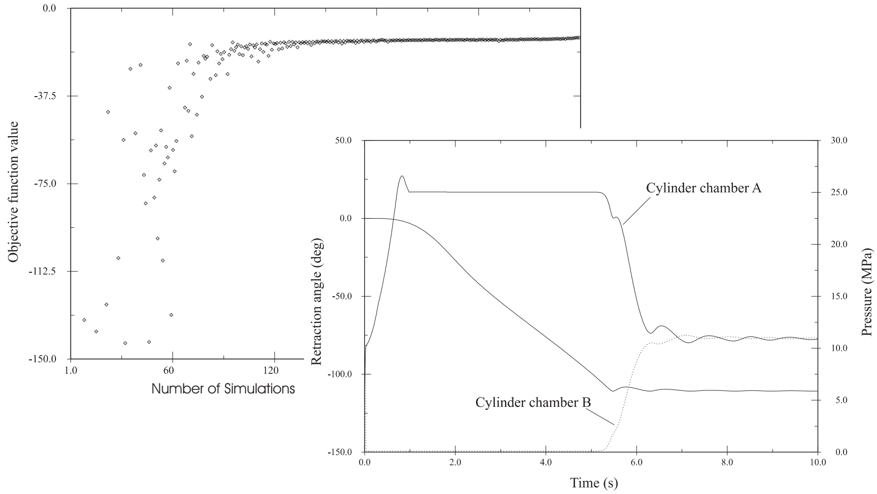


Figure 19. The optimization result for the landing gear system. The graphs show the progression of the objective function as well as the performance of the optimized system.

6.2 Multiobjective optimization

The objects of study for this application are two different concepts of hydraulic actuation systems. Both systems consist of a hydraulic cylinder that is connected to a mass of 1000 kilograms. The objective is to follow a pulse in the position command with a small control error and simultaneously obtain low energy consumption. Naturally, these two objectives are in conflict with each other. The problem is thus to minimize both the control error and the energy consumption from a Pareto optimal perspective.

Two different ways of controlling the cylinder are studied. In the first more conventional system, the cylinder is controlled by a directional valve, which is powered from a constant pressure system. In the second concept, the cylinder is controlled by a servo pump. Thus, the systems have different properties. The valve concept has all that is required for a low control error, as the valve has a very high bandwidth. On the other hand, the valve system is associated with higher losses, as the valve constantly throttles fluid to the tank. The different concepts have been modeled in the simulation package HOPSAN [36]. The system models are depicted in Figure 20 and Figure 21 respectively.

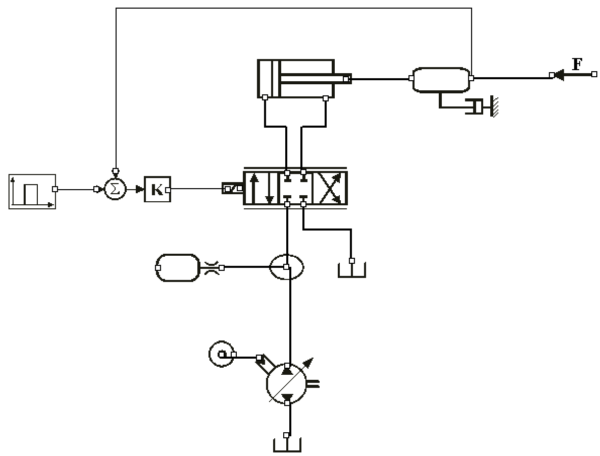


Figure 20. The valve concept for hydraulic actuation.

The valve system consists of the mass and the hydraulic cylinder, the directional valve and a p-controller to control the motion. The directional valve is powered by a constant pressure pump and an accumulator, which keeps the system pressure at a constant level. The optimization parameters are the sizes of the cylinder, valve and the pump, the pressure level, the feedback gain. Furthermore, a leakage parameter is added to both systems in order to guarantee sufficient damping. Thus, this problem consists of six optimization parameters and two objectives.

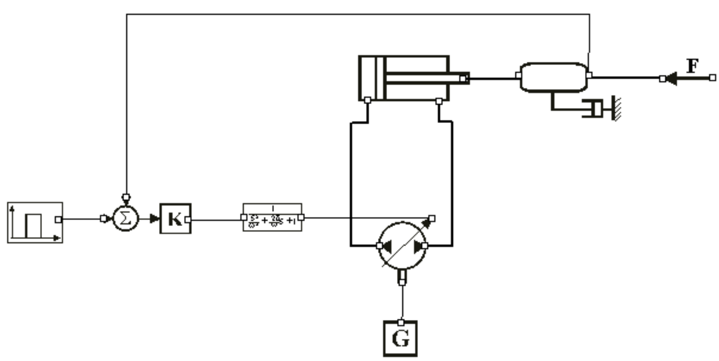


Figure 21. The pump concept of hydraulic actuation.

The pump concept contains fewer components: the cylinder and the mass, the controller and the pump. A second order low-pass filter is added in order to model the dynamics of the pump. The pump system consists of only four optimization parameters. The performance of a relatively fast pump system is depicted in Figure 22.

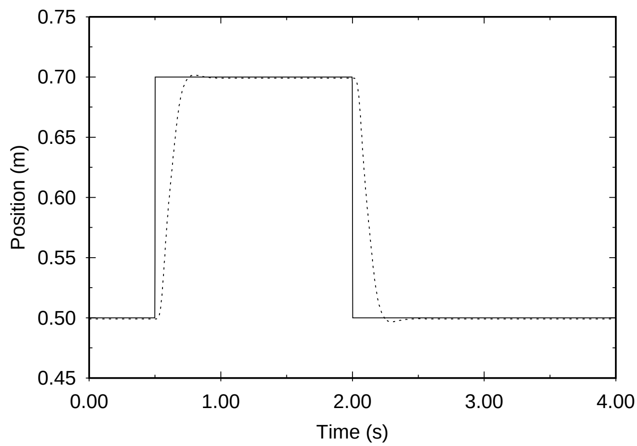


Figure 22. The pulse response for a relatively fast pump system, i.e. the control error = 0.05sm.

6.2.1 Optimization results

Both systems where optimized in order to simultaneously minimize the control error f_1 and the energy consumption f_2. The optimization is conducted with the multiobjective genetic algorithm with a population size of 30 individuals over 200 generations.

As a Pareto optimization searches for all non-dominated individuals, the final population will contain individuals with a very high control error, as they have low energy consumption. It is possible to obtain an energy consumption of close to zero, if the cylinder does not move at all. However, these solutions are of no interest, as we want the system to follow the pulse. Therefore, a goal level on the control error is introduced. The optimization strategy is modified so that solutions below the goal level are always preferred to solutions above it regardless of their energy consumption. In this manner, the population is focused on the relevant part of the Pareto front.

The obtained Pareto optimal fronts for both systems are depicted in Figure 23. In order to achieve fast systems, and thereby low control errors, large pumps and valves are chosen by the optimization strategy. A large pump delivers more fluid, which enables higher speed of the cylinder. However, bigger components consume more energy, which explains the shape of the Pareto fronts.

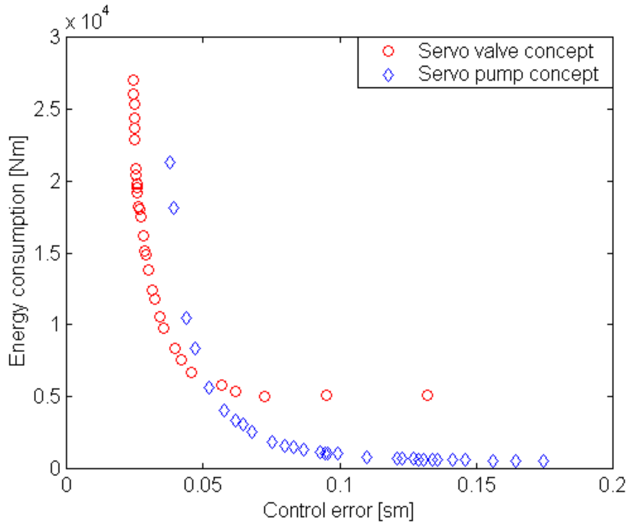


Figure 23. The Pareto optimal fronts for the servo valve and the servo pump concepts respectively.

It is evident that the final design should preferably be on the overall Pareto front, which elucidates when to change between concepts. The servo pump system consumes less energy and is preferred if a control error larger than 0.05sm is acceptable. The servo valve system is fast but consumes more energy. If a lower control error than 0.05sm is desired, the final design should preferably be a servo valve system. In order to choose the final design, the decision-maker has to select a concept and then study the trade-off between the control error and the energy consumption and select a solution point on the Pareto front.

This application shows how Pareto optimization can be employed to support concept selection, by visualizing the pros and cons of each concept. However, optimization could also support concept selection when the objectives are aggregated into one figure of merit. Paper [IX] constitutes a good example of such an approach where two different concepts of electro-hydrostatic actuation systems were optimized with the same objective function formulation.

6.2.2 Mixed variable design problems

Real design problems usually show a mixture of determining continuous parameters as well as selecting existing components from catalogs or databases, see Senin et al. [74]. Therefore, the multiobjective genetic algorithm has been extended to handle a mixture of continuous variables as well as discrete catalog selections. The extensions made to the algorithm are presented in Paper [VI]. The objectof study for this example is the valve actuation system depicted in Figure 20.

The objective is again to design a system with good controllability, but this time at low cost. To achieve good controllability we can choose a fast servo valve, which is more expensive than a slower proportional valve. Therefor, there is a trade-off between cost and controllability. The cost for a particular design is composed of the cost for the individual components as well as the cost induced by the energy consumption.

When designing the system, cylinders and valves are selected from catalogs of existing components. Other parameters such as the control parameter, a leakage coefficient and the maximum flow of the supply system have to be determined as well. Thus the problem is multiobjective with two objectives and five optimization variables of which two are discrete catalog selections and three are continuous variables.

Component catalogs

For the catalog selections, catalogs of valves and cylinders have been included in the simulation program. For the directional valve, the choice is between a slow but inexpensive proportional valve or an expensive and fast servo valve. Valves from different suppliers have been arranged in two ordered sub-catalogs as depicted in Figure 24. The same structure applies to the cylinders as they are divided into sub-catalogs based on their maximum pressure level. The pressure in the system has to be controlled so that the maximum pressure for the cylinder is not exceeded. A low-pressure system is cheaper but has inferior performance compared to a high-pressure system. Each catalog element contains a complete description of that particular component, i.e. the parameters that describe the dynamics of the component, which is needed by the simulation model as well as information on cost and weight etc.

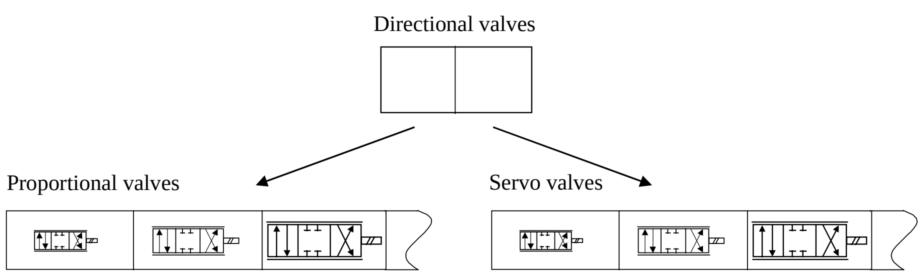


Figure 24. The catalog of directional valves is divided into proportional valves and servo valves. Each sub-catalog is ordered based on the valve size. For each component, a set of parameters describing the component is stored together with information on cost and weight.

Optimization results

The system has been optimized using a population of 40 individuals and 400 generations. In order to limit the Pareto frontier a goal level on the control error was introduced. The result could be divided into four distinct regions depending on valve type and pressure level, see Figure 25.

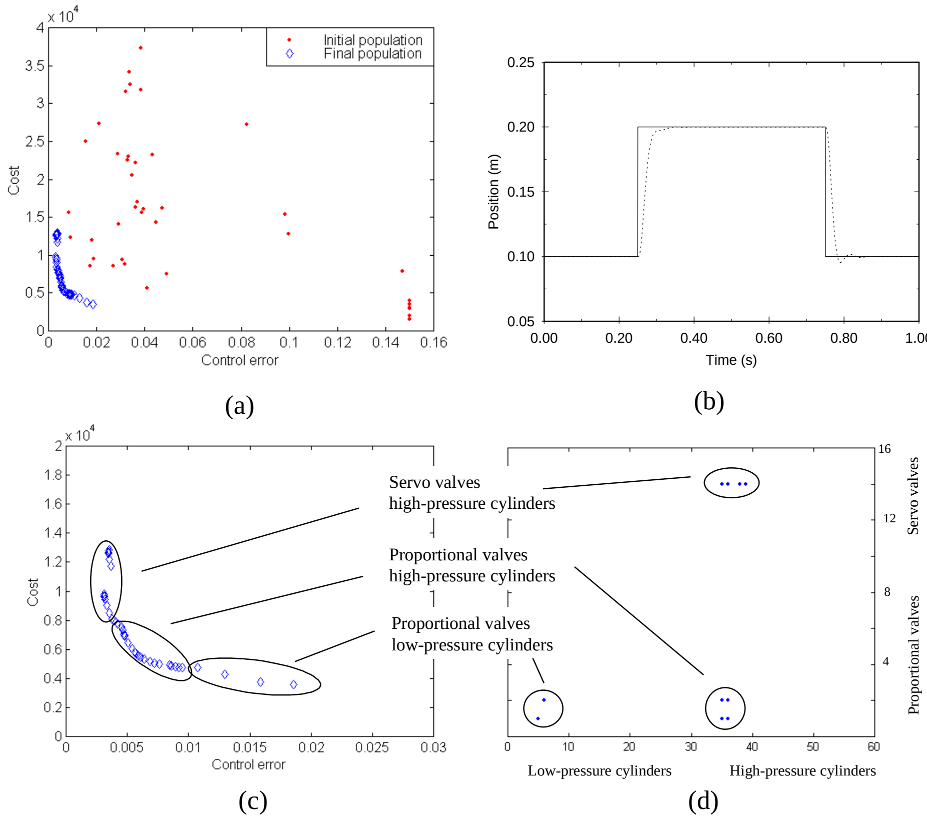


Figure 25. Optimization results. In (a) the initial and final population of the optimization are shown. In (b) the simulated pulse response for a reasonably fast solution is depicted. Figure (c) shows an enlargement of the Pareto front where different regions have been identified based on valve and cylinder selections, as shown in (d).

As can be seen from Figure 25, there is a trade-off between system performance (control error) and system cost. By accepting a higher cost, better performance could be achieved. The cheapest designs consist of small proportional valves and low-pressure cylinders. By choosing larger proportional valves and high-pressure cylinders, the performance could be increased at the expense of higher cost. If a still better performance is desired, a servo valve has to be chosen, which is more expensive but has better dynamics.

The continuous parameters, such as the control parameter, tend to smoothen out the Pareto front. For a given valve and cylinder, different settings on the continuous parameters affect the pulse response. A faster response results in a lower control error, but also a higher energy consumption and thereby higher cost. Therefore, there is a local trade-off between cost and performance for each catalog selection.

7
Robustness versus optimality

IN REALITY, SYSTEM parameters will always include variation to some extent and this fact is likely to influence the result. The system we are designing should have an optimal performance and still be robust. These two aspects are often in conflict with each other, and therefore we need to answer not only the question “What is best?”, but also “What is sufficiently robust?”

The concept of robustness is rather intuitive and is here illustrated in Figure 26, where the sensitivity to changes in a design parameter is shown for two hypothetical designs. Design 1 shows a better optimal performance but is more sensitive to disturbances. In robust design, the designer tries to determine the design parameters in order to obtain desirable values on the objectives, while at the same time minimize their variance. The most well known method that achieves this was presented by Taguchi [86].

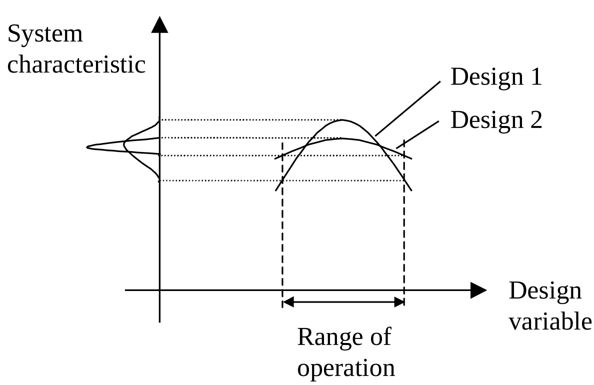


Figure 26. Sensitivity to changes in a design variable for two hypothetical designs.

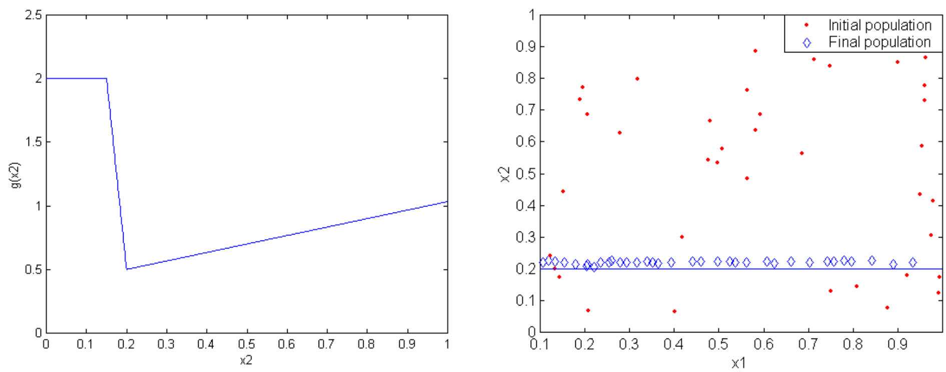
In this chapter, three approaches to handle robustness in design optimization are presented. In the first approach, the anticipated variations of the design parameters are introduced before each solution is evaluated. With this method, designs like Design 1 are avoided. The second approach is a design by experience method, were the robustness is analyzed around the achieved optimum. Here, the optimal design (Design 1) is first identified. Then disturbances are introduced and the experimental designs guide the search towards more robust designs (Design 2). The third approach involves a recursive least square method that identifies the sensitivities, and thereby the robustness, as the optimization evolves. Here both Designs 1 and 2 are identified and their sensitivities are estimated by two response surfaces.

7.1 Disturbing the design variables

The first approach is well suited for optimization methods where a population is used and where individual solutions have to repeatedly prove fit, such as in genetic algorithms. We then assume that we know what parameters are subjected to disturbances and how big these disturbances are. The optimization process gives normative values for each design parameter, but before each solution is evaluated, the disturbances are added to the design parameters. Thus, the anticipated variations are considered during the optimization process. As the optimization progresses, one lucky shot at the narrow optima is not enough to guarantee survival. Only solutions that repeatedly show good performance under the influence of disturbances will survive. Therefore, solutions on narrow optima or close to sharp edges will not survive. This method has been applied to a single objective problem as presented by Tsutsui and Ghosh in [91]. In order to clarify this approach, an example will follow.

Consider the multiobjective problem of equations (15)-(16), with the *g*-function depicted in Figure 14. Now assume that the parameter x_2 is subjected to noise of the form $x_2' = x_2 \pm 0.02$. This could correspond to the tolerance of a manufacturing process, where the disturbance could have any distribution. The disturbance is added to x_2 before each solution is evaluated. Solutions at the global narrow peek, $x_2 = 0.2$, will have a small disturbance before the objective function is calculated. Therefore, solutions that should have been on the narrow peek will fall off due to the disturbance. This optimization problem has been solved 100 times with the MOSGA, and every time the population converged to the local more robust optimum.

Another advantage of the method of adding noise to the design variables is to avoid sharp edges. Consider the multiobjective problem again, but now with a *g*-function according to Figure 27(a). At the sharp edge, solutions are very sensitive to disturbances. Therefore, solutions a bit to the right of the optimum are more robust and have almost the same function value. This problem is solved with the same disturbance on x_2. In Figure 27(b) it can be seen how the population avoids the sharp edge and converges to a more robust region.



(a) (b)

Figure 27. (a) shows a *g*-function with a sharp edge. In (b) the optimization result is shown when a disturbance is added to x_2. The population avoids the sharp edge and converges to a more robust area.

A disadvantage of using the method of adding noise to the design variables is that each time an objective function value is needed it has to be recalculated, as it is probabilistic and not deterministic. Depending on the method used, this might be more or less computationally expensive. In the multiobjective genetic algorithm presented in this thesis, many function calls are needed. Naturally, the new child has to be evaluated, but then it is compared to the others in the population, and thus they have to be re-evaluated as well. In an ordinary genetic algorithm, the objective function values are only used once when the child is initiated. Therefore, this method might be more suited together with other genetic algorithms such as MOGA [24] or SPEA [99]. However, the performance of these methods goes down when the ranking is no longer deterministic. Recent work by Huges [39] and Teich [89] present methods for dealing with Pareto optimization under uncertainty for MOGA and SPEA respectively.

7.2 The design by experiments approach

Among system parameters, it is possible to distinguish between controllable and non-controllable factors. The controllable factors can be given a value around which variation may occur, i.e. these are the typical design parameters such as the size of a hydraulic pump. On the other hand, the non-controllable factors and their variation cannot be directly affected, such as changes in the environment variables or increased leakage in the pump. These factors are also known as disturbing factors.

In the proposed method, optimization and simulation are combined with design of experiments, as indicated in Figure 28. The method originates from Nilsson et al. [60]. The foundations for design of experiments can be found in for instance Box et al. [7] and Phadke [64].

The basic idea is to evaluate the behavior of the response variable around the optimum with a polynomial including controllable and non-controllable factors. The response variable could be any system characteristic of interest, such as the objective function. The coefficients of the polynomial are estimated using regression analysis with the method of least squares. By examining the coefficient values, the influence of the controlling and the disturbing factors could be estimated.

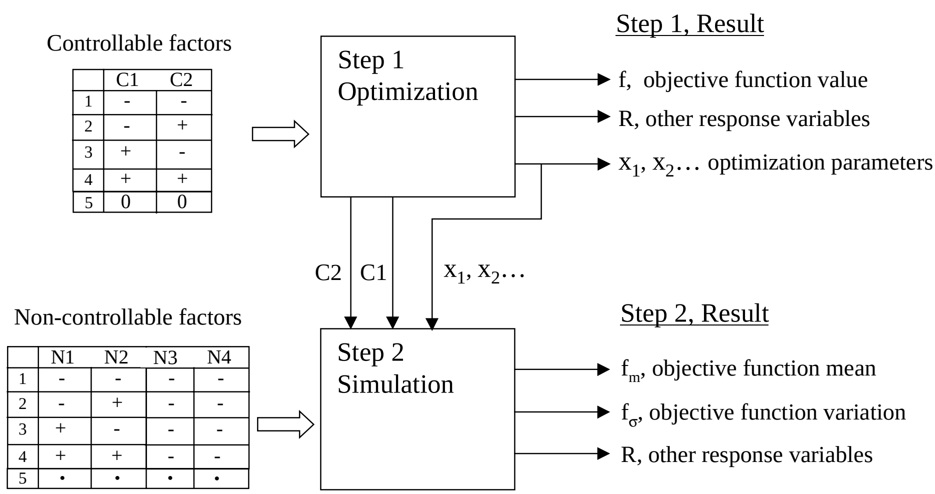


Figure 28. Calculation procedure to evaluate system robustness, considering the effect of non-controllable factors.

Before applying the procedure in Figure 28 the system is optimized without considering any variations. Then an orthogonal array is used in step 1 in order to vary the controllable factors around the values achieved for the optimal solution. For each set of parameters of the orthogonal array, optimization of the remaining system parameters is performed. This first calculation step gives a measure of how narrow the optimum is considering changes in the controllable factors. By studying the coefficients of the resulting polynomial, it could be seen which are the main contributing factors to the value of the response variable. However, no effect of the disturbing factors is considered.

The outcome of the optimization in Step 1 acts as an input to the simulations in Step 2. In Step 2 an orthogonal array of the non-controllable factors is evaluated for each optimized system solution, i.e. for each row in the first orthogonal array. The levels for the non-controllable factors can be determined as plus/minus the standard deviation, corresponding to an assumed real-life distribution around a mean value. The outcome of Step 2 is the mean and the standard deviation of the response variable considering the effects of the disturbing factors. It could then be seen which disturbing factors have the greatest influence on the response variable. By comparing the mean and the standard deviation for the different designs from Step 1, greater insight into the properties of the system could be gained. Some of the settings of the controllable factors may result in designs with poorer value for the response variable but with a smaller variation. Thus, system performance could be traded for robustness.

Another approach to analyze the result is to cross the two arrays, thus giving one large experimental design. By crossing the arrays, interaction effects between the controllable and the non-controllable factors could be found. A result from this analysis is how the controllable factors should be chosen in order to minimize the influence of the non-controllable factors. Thus, by assigning appropriate values to the controllable factors, the system robustness is increased.

The design of experiments approach is discussed further in Paper [III] where it was applied to the design of two different electro-hydrostatic actuation systems. From this study, it could be seen how a larger cylinder area could reduce the influence of the disturbing factors. It was also stated that the load sequence and the pump leakage were the disturbing factors that had the greatest influence on both the value and the deviation of the objective function. Analyses of different electro-hydrostatic actuation systems were also performed in Paper [IX], where optimization is introduced in order to support concept selection, and in Paper [XI], where the thermodynamic properties of the two concepts have been studied.

7.3 Metamodel representations

In the third approach, metamodels are introduced in order to assess the properties of the optimal designs. The metamodel is typically a second order polynomial describing a system characteristic, such as the objective functions, in terms of the design parameters, see equation (22). In order to keep the number of parameters in the model low; the cross-product terms are omitted. Although this is a simplification, the robustness could still be assessed. If the cross-product terms were not omitted, the required number of calculations for the estimation would be larger than the number of calculations needed for the actual optimization.

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_1^{\,2} + \theta_3 x_2 + \theta_4 x_2^{\,2} + ... \tag{22}$$

In order to estimate the parameters, θ_i, a Recursive Least Square (RLS) scheme is employed, see for example Ljung [51]. As the GA evolves a population of individuals, there is a large number of evaluations that could be utilized to estimate the model parameters. The RLS method continuously estimates the parameters of the metamodel and is ready to present the estimate when the optimization has converged. The RLS method could be described as:

$$\begin{array}{ccccccccc} \hat{\theta}(t) & = & \hat{\theta}(t-1) & + & L(t) & \Big[& y(t) & - & \varphi^T(t)\hat{\theta}(t-1) \Big] \\ \text{new estimate} & = & \text{old estimate} & + & \text{correcting vector} & \Big[& \text{new y-value} & - & \text{Estimated y using old } \hat{\theta} \Big] \end{array} \tag{23}$$

$$\text{where } L(t)=\frac{P(t-1)\varphi(t)}{\lambda(t)+\varphi^T(t)P(t-1)\varphi(t)} \tag{24}$$

$$\text{and } P(t)=\frac{1}{\lambda(t)}\left[P(t-1)-L(t)\varphi^T(t)P(t-1)\right] \tag{25}$$

λ is the forgetting factor and $\varphi^T=\begin{bmatrix}1 & x_1 & x_1^2 & x_2 & x_2^2 & ....\end{bmatrix}$ is the data vector. To use the recursive algorithm, initial values for the start-up are required. Here we use $\hat{\theta}(0)=0$ and $P(0)=\alpha\,\mathrm{I}$, where α is a large number. If the design parameters are of different magnitude, the value of α has to be scaled, so that $P(0)=diag(\alpha_1, \ldots, \alpha_{2n+1})$

The MOSGA was modified to incorporate the RLS estimation. For each individual, the current estimate, $\hat{\theta}(t)$, and the covariance matrix $P(t)$ are stored. The RLS estimation is then added to Step 5 of the algorithm, so that the child is used to update the estimate of the most similar individual. If the child is better then the old individual, the child replaces it. $\hat{\theta}(t)$ and $P(t)$ are then transferred to the child.

In Paper [VII], this method is applied to a mathematical test problem similar to the problem defined by equations (13) - (15). However, the original *g*-function has been modified so that the global spike is not as narrow. The metamodels have been applied to estimate a polynomial for the *g*-function at the identified Pareto optimal points, see Figure 29. In Paper [VII], metamodels for the objectives f_1, *and* f_2 are estimated as well.

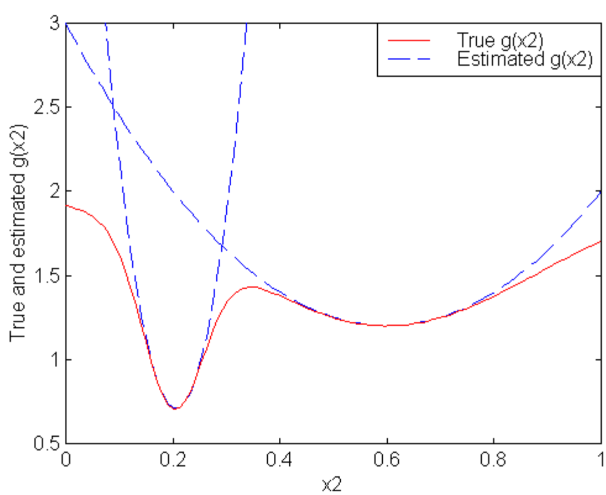


Figure 29. The modified multimodal g-function (solid line) and the model estimation at the global and local optima (dashed lines).

The metamodel could be seen as a response surface, which visualizes the shape of the Pareto optimal front. By studying the coefficients of the metamodel, the robustness of different optima could be estimated. This approach constitutes a good example of how information could be extracted from the evolving population of the genetic algorithm. This information would otherwise have been neglected. The method is computationally inexpensive, as no extra function evaluations are needed. A similar approach for the single objective problem has been presented by Krus [47].

7.4 Summary

It is obvious that we want the system we are designing to have the best possible performance. However, we want the system to be robust as well. In this chapter, three different approaches to handle robustness in design optimization were presented. The different approaches have different properties and are therefore suited for different kinds of applications. The first approach is best suited for genetic algorithm optimization, whereas the second method, and to some extent the third, works with other optimization techniques as well.

The first method is easy to implement and performs well if the disturbances of the design parameters are known. The result of the optimization would be insensitive to known variation in the design variables. However, no information about the shape of optimum is given.

The second approach presents an estimate of the shape of the optimum, both due to changes in the optimization parameters as well as due to disturbing factors. Thereby the system could be designed so that it is less sensitive to disturbances. A disadvantage of this method is that it requires a set of extra calculations to be performed after the actual optimization. This method could however be applied to any optimization technique.

The metamodel method is slightly more complex to implement, as the RLS method has to be included in the optimization strategy. However, when this is done, useful information could be extracted from the evolving population. The method only involves a minor amount of extra calculations, and no extra function evaluations. A disadvantage of this method is that it only considers the optimization parameters and no other disturbing factors. The advantage of this method is that it can be used to estimate the properties of different characteristics of the system and not just the objective function. The resulting response surface can be employed to support decision-making in multiobjective optimization, so that the final solution should be not only Pareto optimal but also robust.

8 Discussion and conclusions

WITHIN THIS THESIS, methods have been presented that support the employment of simulation and optimization techniques in engineering design. Applications were mainly gathered from the field of fluid power system design. Studied systems include a landing gear system for a civil aircraft, electro-hydrostatic actuation systems for aircraft applications as well as hydraulic actuation systems. The focus has been twofold. The first issue was to employ simulation in order to enhance our understanding of the design process, and the second to develop optimization techniques that support the design of complex systems based on simulations.

In the first issue, the main goal is to ensure an efficient design process. However, in order to achieve that a thorough understanding of the process is needed. Here a design process modeling approach is presented where simulation is employed in order to predict the performance of the design process in terms of lead-time and process cost. By studying the outcome of such simulations, increased insight into the properties of the design process could be gained. Furthermore, a sensitivity analysis could be performed in order to identify the tasks which have the largest impact on the performance of the process. Thus, the outcome from such models could be used to estimate the risk of the project. Furthermore, by modeling different competing design processes they can be compared and evaluated based on process lead-time and cost. The design process modeling method presented in this thesis is described in detail in Paper [I].

The second issue, which is the major part of the thesis, has been to facilitate the employment of optimization techniques to engineering design problems. The focus was on the design of complex systems where a set of simulation tools is employed in order to predict the properties of the system. Optimization is then conducted based on the outcome of these simulations. This particular environment puts specific demands on the optimization algorithms that are going to be applied.

In this thesis, it has been concluded that non-gradient optimization methods are best suited for these types of applications, as they do not need derivatives of the objective functions. Furthermore, they are more robust in finding the global optimum, and they can be applied to a wide range of problems without alterations to the algorithm. Another advantage is that they can handle a mix of continuous and discrete parameters in a straightforward manner. This is important, as engineering design typically constitutes a mix of determining continuous parameters as well as selecting individual components from catalogs or databases. Therefore, by choosing a non-gradient optimization method, a wide range of problems might be solved, without having to tailor the method to the problem. The disadvantage of using non-gradient methods is however that they are computationally expensive, as they require many function evaluations. Here however the advantages are considered to outweigh the disadvantages.

The optimization methods applied within this thesis are the Complex method and genetic algorithms. Both methods are characterized by a population that evolves as the optimization progresses. When the design problem is reformulated as an optimization problem, it is moved from assigning values on individual design parameters to formulating the objectives, i.e. towards selecting evolutionary pressure.

Design problems are often characterized by the presence of several conflicting objectives. When optimization is employed in order to support engineering design, these objectives are usually aggregated to one overall objective function. The outcome of the optimization is then strongly dependent on how the objectives are aggregated. Within this thesis, different ways of aggregating the objectives are discussed. Furthermore, a method is presented that uses the Design Structure Matrix and the relationship matrix from the House of Quality method. The method formalizes the formulation of the objective function, it is self-documenting and it guarantees traceability. The proposed method was applied to support the design of a landing gear system, which combines a mechanical structure with a hydraulic actuation system. The landing gear system was successfully optimized with the help of the Complex method.

Another way to handle the multiple objectives of a design problem is to introduce the concept of Pareto optimality. The search is then not for one optimal solution but for a set of solutions that are optimal in a broader sense, i.e. they are Pareto optimal. An advantage of conducting Pareto optimization is that the arbitrariness of the decision-maker is left out of the optimization. The search is for the Pareto set, which includes all rational choices, among which the decision-maker has to select the final solution by trading the objectives against each other. In the two-objective case, the Pareto front is a curve that clearly visualizes the trade off between the competing objectives. As the dimension of the problem increases, this visualization becomes harder as the Pareto front now is a surface. This demonstrates one weakness of Pareto optimization, namely how to visualize the trade-off as the dimension of the problem increases.

One way of addressing this problem is to aggregate some objectives, e.g. cost and energy consumption could be aggregated to one overall cost objective. Another possibility is to filter and cluster the solutions so that the set of Pareto optimal solutions is reduced to a set of clusters, see Morse [57] and Rosenmann and Gero [71]. Each cluster consists of solutions with similar properties, and therefore the decision-maker only has to investigate one solution per cluster. Once the most interesting cluster has been identified, this is where to focus further analysis. Ways to visualize the trade-off between more than to objectives have been presented by several authors see for example Tappeta et al. [88] and Ball et al. [2]. Figure 30 shows an example with four objectives and five distinct regions on the Pareto front.

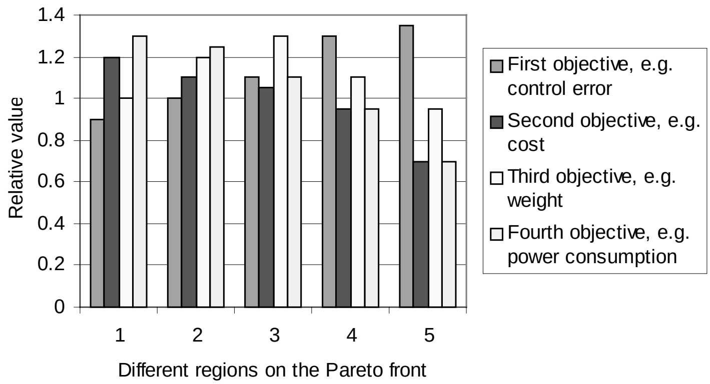


Figure 30. A sample problem with four different objectives visualized for five distinct regions on the Pareto front.

Another possibility is to visualize the different Pareto optimal regions within the framework of the House of Quality method, see Figure 31. Usually the right side of the House of Quality is used for customer evaluation of competitive products, see Hauser and Clausing [34] and Sullivan [85]. Here however it is used to visualize how the different Pareto optimal solutions perform on the systems characteristics, i.e. the objectives. By comparing the 'Pareto traces' with the relative importance of the system characteristics it could be determined which solution is the most suitable.

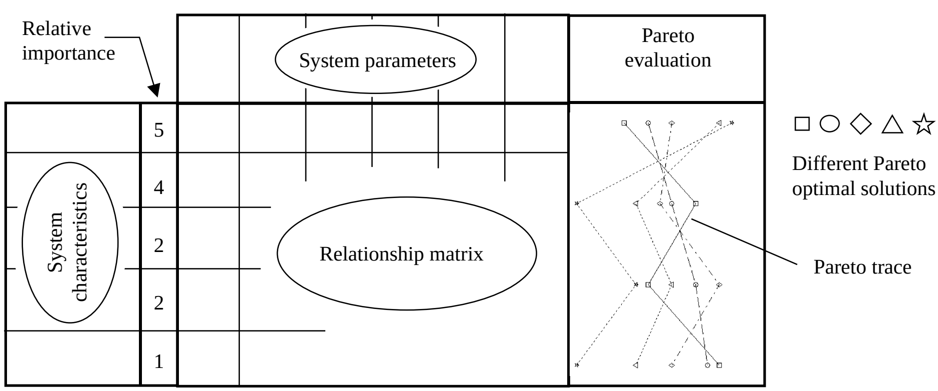


Figure 31. The same sample problem here visualized with the help of the House of Quality.

Within the thesis different methods to identify Pareto optimal solutions are discussed. It is concluded that the most appealing methods are within the group of multiobjective genetic algorithms. These methods are well suited for application in engineering design, and they are capable of identifying the Pareto front in one single optimization run.

Within the thesis, a new multiobjective genetic algorithm is proposed. The method was tested on a set of benchmark problems gathered from the literature, and it has proven to perform well. It has also been applied to the design of a hydraulic actuation system. Here two different concepts were studied and it was shown how studying the resulting Pareto fronts for the competing concepts constitute a valuable support for concept selection. Furthermore, the method was extended to handle problems with a mix of continuous and discrete parameters. The enhanced algorithm was tested on the design of the same actuation system, this time with catalogs of directional valves and hydraulic cylinders. The catalogs were established by collecting data from suppliers of fluid power components. The optimization resulted in a discrete Pareto front that visualizes the trade-off between the performance and the cost of the system.

An important issue in engineering design is to ensure the function of the system for a wide range of operational conditions. The system should preferably perform well even if there are variations in the system parameters, e.g. due to imperfections in the manufacturing process, as well as under the influence of disturbances, e.g. changing environmental conditions. Thus, the system should have good (optimal) performance and simultaneously be robust. Within this thesis, three methods were presented that address the issue of robustness versus optimality. In the first method, variation in system parameters is considered during the optimization process. Thereby the optimization avoids narrow optima where variations have great influence on the performance. This method has proven to perform well together with genetic algorithms. However, it does not provide any information on the shape of the objective function and it does not consider the effect of disturbing factors. It also remains to be tested on a real design problem.

The second approach combines design of experiments with simulation and optimization. In this method, post-optimal analysis is performed with the help of designed experiments and statistical methods. Based on the analysis it could be seen which disturbing factors have the greatest influence on the design, and how the design variables could be selected in order to minimize the influence of the disturbing factors.

In the third approach the use of metamodels is introduced in order to develop a response surface that visualizes the shape of the Pareto front. As the population of the multiobjective genetic algorithm evolves, there are a large number of solutions that could be used to estimate the shape of the Pareto front. Here the recursive least square method is applied to estimate the coefficients of the metamodel. By studying these coefficients it could be seen which parameters have the greatest influence on the objective functions on different regions on the Pareto front. This can be a useful support in selecting the final design. For once, the most robust solutions could be identified, together with the parameters that contribute the most to the design. At one part of the Pareto front, one component may be critical, but at other parts, another component might have the greatest influence, and our effort should thus be focused accordingly. This method was successfully applied to a mathematical test problem. As the metamodel is only a function of the optimization parameters, the influences of other disturbing factors are not taken into account.

To conclude, the thesis presents methods that address many of the important aspects of design optimization. Furthermore, they are presented together in a framework, which elucidates how simulation and optimization techniques could be introduced in the design process in a rewarding manner. Thus, the thesis constitutes a step towards an overall framework for employing optimization in engineering design.

9 Outlook

A NATURAL AREA for future work is to estimate what impact the presented optimization techniques could have on an actual development process by employing the presented design process modeling technique. This could help to obtain a measure of the value provided by simulation and optimization techniques in engineering design.

Another interesting area is to develop an integrated design environment where systems could be optimized and their robustness evaluated in an easy manner. Within this framework, the simulation model and the formulation of the optimization problem would evolve simultaneously as one unit. As the simulation model evolves, the optimization parameters are determined, their anticipated variation is estimated and the disturbing factors identified. Furthermore, the formulation of the optimization problem would be an integral part of the simulation model.

Within this framework, there is a need to develop techniques that visualize the robustness of design proposals, and the influence of different design parameters and disturbing factors on the objectives. There is also a need to incorporate methods that support multiobjective optimization of large problems and problems with many objectives. This would include techniques to visualize the trade-off between more than two objectives, and methods to filter and cluster Pareto optimal solutions. A natural desire is then to study authentic problems gathered from the industry, which would include both more design parameters as well as more objectives than otherwise studied within this thesis.

A natural extension for design problems with discrete choices is a tighter connection to database technology. Here component catalogs were created in the simulation program, where the elements of the catalogs contain the parameters needed by the simulation program. These catalogs could be interchanged for databases, where each elements of the database could be extended to contain the entire simulation model for a particular component. These models could either be made by the designer, or be provided by the supplier in such a form that proprietary information is not jeopardized. In this way, the supplier does not only supply a component but also the simulation model describing the component. Thus, the knowledge of the supplier is incorporated in the development process.

Within this thesis a method was presented in which the relationship matrix from the House of Quality method is used to determine which characteristics should be incorporated in the objective function formulation. The figures in the matrix are determined in a team activity where engineers consider what impact different design parameters may have on different system characteristics. However, the presented metamodel technique identifies a mathematical relation between system parameters and system characteristics. Hence, the initial figure in the matrix could be compared to the once calculated during the optimization. Thus, the relationship matrix could also be employed in order to visualize the computed relation between system parameters and system characteristics.

10
Review of papers

Paper I

A Design Process Modeling Approach Incorporating Nonlinear Elements

The subject of the first paper is to analyze the performance of a design process with the help of simulation based on design process models. A method is presented where the design process is decomposed to a set of interconnected design activities with design information flowing between them. Among the design activities, design tasks and design reviews could be distinguished. Design work is performed within design tasks, which are characterized by completion time and cost. Design reviews control the information flow of the process, and they are characterized by the probability to proceed forward to the next design task. The task characteristics can change with the advance of the design process, so that the second time a task is performed it is completed faster.

The model provides information on the expected value and the probability distribution of process lead-time and costs. Furthermore, by conducting sensitivity analysis, the design tasks with the greatest impact on the overall performance of the process could be identified. Summing up, design process modeling provides greater insight to the properties of the design process.

Paper II

Design of Objective Functions for Optimization of Multidomain Systems

The second paper addresses a multiobjective optimization problem where the different objectives are aggregated to one overall objective function. The paper presents a method to support objective function formulation through the incorporation of tools such as the Design Structure Matrix (DSM) and the House of Quality method. The DSM structures the design problem and supports the choosing optimization parameters. After that, the relationship matrix from the House of Quality method is employed in order to relate system parameters to system characteristics, thus supporting the formulation of the objective function. The method presented is applied to a landing gear system, where simulation and optimization are employed in order to design an actuation system, which retracts a given landing gear. The hydraulic actuation system is modeled in the HOPSAN simulation package, whereas the mechanical structure is modeled in Pro-Mechanica Motion.

Paper III

METHOD FOR INTEGRATED SYSTEMS DESIGN - A STUDY OF EHA SYSTEMS

The third paper presents a method for robust design, which combines design by experiments with simulation based optimization. By employing two orthogonal arrays, variations in both controllable and disturbing factors are considered. The aim was to find system solutions, which have a good performance (close to the optimum) but are insensitive to disturbances.

Within the paper, two concepts of electro-hydrostatic actuation systems for aircraft applications were studied with the optimization strategy outlined in Paper [II]. By applying the method, it is shown that system performance can be traded for increased system robustness. The method also gives valuable insights into which disturbing factors have the greatest influence on the system performance. It also suggests how the controllable factors should be tuned in order to minimize the impact of the disturbing factors.

Paper IV

PARETO OPTIMIZATION USING THE STRUGGLE GENETIC CROWDING ALGORITHM

The fourth paper presents a new multiobjective genetic algorithm based on the struggle crowding algorithm and Pareto based ranking. The method is tested on a set of benchmark problems gathered from the literature and it has proven to perform well. The properties of the method are analyzed and enhancements are proposed.

The method is capable of identifying multiple Pareto fronts in multi modal search spaces, something that no other methods are capable of. For an engineering problem, the knowledge of the existence of global as well as local Pareto optimal fronts is very valuable. An another advantage of the method is that it requires a minimum of parameters to be set for each problem, which makes it robust and easy to use.

Paper V

MULTIOBJECTIVE OPTIMIZATION OF HYDRAULIC ACTUATION SYSTEMS

In the fifth paper the multiobjective genetic algorithm presented in Paper [IV] is applied to the design of a hydraulic actuation system. Two different concepts of hydraulic actuation systems are studied with the help of simulation models. Therefore, the optimization strategy is coupled to the simulation program. The outcome of the optimization is a Pareto front elucidating the trade-off between system performance and energy consumption. There is a trade-off between performance and energy consumption because a system with good performance consumes more energy. By comparing the resulting Pareto fronts for the two concepts, optimization is introduced to support concept selection. Thereby, it becomes evident under which preferences one concept is to be preferred to another.

Paper VI

MULTIOBJECTIVE OPTIMIZATION OF MIXED VARIABLE DESIGN PROBLEMS

In the sixth paper, the multiobjective genetic algorithm is extended to handle a mixture of discrete and continuous parameters. Real-world engineering design problems usually consist of a mixture of determining continuous parameters as well as selecting individual components from catalogs or databases. The extended method was again applied to the design of a hydraulic actuation system, with the intention of simultaneously minimizing the control error and the cost. For this application, catalogs of valves and cylinders were created by collecting data from suppliers of hydraulic components. The system was optimized resulting in a discrete Pareto front elucidating the trade-off between performance and cost.

Paper VII

METAMODEL REPRESENTATIONS FOR ROBUSTNESS ASSESSMENT IN MULTIOBJECTIVE OPTIMIZATION

In the last paper, a method to assess the properties of Pareto optimal solutions is presented. The method utilizes information from the evolving population of the genetic algorithm by means of a recursive least square method. Thereby metamodels are created that represent a response surface of the Pareto front. By studying the coefficients of these metamodels, the robustness of individual solutions could be assessed. It could also be seen which parameters contributed the most to the system performance at different locations of the Pareto front. Thus, metamodels constitute an extra support in choosing the final design. One part of selecting the final design is to trade the objectives against each other by analyzing the Pareto front. The second part is to assure a robust design, which is facilitated by the metamodels. Thus, metamodels support the employment of multiobjective optimization in engineering design.

References

[1] ADLER P.S., MANDELBAUM A.., NGUYEN V., AND SCHWERER E., "From Project to Process Management: An Empirically-based Framework for Analyzing Product Development Time", *Management Science*, Vol. 41, No. 3, 1995.

[2] BALL M., FLEISCHER M., CHURCH D., "A Product Design System Employing Optimization-based Tradeoff analysis", in *Proceedings of ASME DETC Design Theory and Methodology Conference*, Baltimore, USA, September 10-13, 2000.

[3] BELL D.G., KANNAPAN S.M., AND TAYLOR D.L., "Product Development Process Dynamics", in *Proceedings of ASME Design Theory and Methodology Conference*, DE-Vol. 2, pp. 257-266, 1992.

[4] BENAYOUN R., DE MONTGOLFIER J., TERGNY J., AND LARITCHEV O., "Linear programming with multiple objective functions: Step Method (STEM)," *Mathematical Programming*, vol. 1, pp. 366-375, 1971.

[5] BLANCHARD B. AND FABRYCKY W., *Systems Engineering and Analysis*, Prentice Hall, New Jersey, 1990.

[6] BORUP L. AND PARKINSON A., "Comparison of four non-derivative optimization methods on two problems containing heuristic and analytic knowledge," in *Proceedings of at ASME Advances in Design Automation*, Scottsdale, Arizona, 1992.

[7] BOX G., HUNTER W. AND HUNTER S., *Statistics for Experiments*, John Wiley & Sons Inc, 1978.

[8] BOX M. J., "A new method of constraint optimization and a comparison with other methods", *Computer Journal*, 8:42-52, 1965.

[9] BRUNS M., *Systemtechnik Methoden zur interdisziplinären Systementwicklung*, Springer-Verlag, Berlin, 1991.

[10] CHRISTIAN A. D. AND SEERING W. P., "A Model of Information Exchange in the Design Process", *in Proceedings of ASME Design Engineering Technical Conferences*, DE-Vol. 83, pp. 323-328, 1995.

[11] COELLO C., *An empirical study of evolutionary techniques for multiobjective optimization in engineering design*, Dissertation, Department of Computer Science, Tulane University, 1996.

[12] COHEN L., *Quality function deployment: how to make QFD work for you*, Addison-Wesley, Reading, 1995.

[13] CONNOR A. M., "Parameter Sizing for Fluid Power Circuits Using Taguchi Methods", *Journal of Engineering Design*, Vol. 10, No. 4, 1999.

[14] CONNOR A. M. AND TILLEY D. G., "A tabu search method for the optimization of fluid power circuits," *Journal of Systems & control engineering*, vol. 212, pp. 373-381, 1998.

[15] CROSS N., *Engineering Design Methods*, John Wiley & Sons Inc, 1994.

[16] DAS I. AND DENNIS J., "Normal-boundary interaction: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM Journal of Optimization*, vol. 8, pp. 631-657, 1998.

[17] DAS I. AND DENNIS J. E., "Closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems," *Structural Optimization*, vol. 14, pp. 63-69, 1997.

[18] DAWKINS R., *The Blind Watchmaker*, Harlow, Longman, 1986.

[19] DEB K., 1999, "Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems," *Evolutionary Computation*, vol. 7, pp. 205-230.

[20] EPPINGER S. D., WHITNEY D. E., SMITH R. P., AND GEBALA D. A., "A Model-Based Method for Organizing Tasks in Product Development", *Research in Engineering Design*, vol. 6, no. 1, pp. 1-13, 1994.

[21] EPPINGER S. D., NUKALA M. V., AND WHITNEY D. E., "Generalized Models of Design Iteration Using Signal Flow Graphs", *Research in Engineering Design*, Vol. 9, no. 2, pp. 112-123, 1997.

[22] ESCHENAUER H., KOSKI J., AND OSYCZKA A., *Multicriteria Design Optimization*, Berlin: Springer-Verlag, 1990.

[23] FONSECA C. AND FLEMING P., "An overview of evolutionary algorithms in multiobjective optimization", *Evolutionary Computation*, vol. 3, pp. 1-18, 1995.

[24] FONSECA C. M. AND FLEMING P. J., "Multiobjective optimization and multiple constraint handling with evolutionary algorithms - Part I: a unified formulation," *IEEE Transactions on Systems, Man, & Cybernetics Part A: Systems & Humans*, vol. 28, pp. 26-37, 1998.

[25] FONSECA C. M. AND FLEMING P. J., "Multiobjective genetic algorithms made easy: Selection, sharing and mating restriction," in *Proceedings of 1st IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems*, Sheffield, England, 1995.

[26] FU K.-C. AND LEVEY G. E., "Discrete frame optimization by Complex-Simplex procedure," *Computers & Structures*, vol. 9, pp. 207-217, 1978.

[27] GLOVER F., "Tabu Search - Part I," *ORSA Journal on Computing*, vol. 1, pp. 190-206, 1989.

[28] GOLDBERG D. E., *Genetic Algorithms in Search and Machine Learning*, Addison Wesley, Reading, 1989.

[29] GRUENINGER T. AND WALLACE D., "Multi-modal optimization using genetic algorithms," Technical Report 96.02, CADlab, Massachusetts Institute of Technology, Cambridge, 1996.

[30] GUIN J. A., "Modification of the Complex method of constraint optimization," *Computer Journal*, vol. 10, pp. 416-417, 1968.

[31] HAJELA P., "Nongradient methods in multidisciplinary design optimization - status and potential," *Journal of Aircraft*, vol. 36, pp. 255-265, 1999.

[32] HAQUE M. I., "Optimal frame design with discrete members using the complex method," *Computers & Structures*, vol. 59, pp. 847-858, 1996.

[33] HARIK G., "Finding multimodal solutions using restricted tournament selection," in *Proceedings of Sixth International Conference on Genetic Algorithms*, 1995.

[34] HAUSER J., R. AND CLAUSING D., "The House of Quality", *Harvard Business Review*, May-June, 1988.

[35] HOLLAND H. J., *Adaptation in Natural and Artificial Systems, an introductory analysis with application to biology, control and artificial intelligence,* The university of Michigan Press, Ann Arbor, USA, 1975.

[36] HOPSAN, *Hopsan, A Simulation Package, User's Guide,* Technical Report LITH-IKP-R–704, Div. of Fluid and Mech. Eng. Systems, Dept. of Mech. Eng., Linköping University, Sweden, 1991.

[37] HORN J., "Multicriterion decision making," in *Handbook of evolutionary computation*, T. Bäck, D. Fogel, and Z. Michalewicz, Eds., IOP Publishing Ltd. and Oxford University Press, 1997.

[38] HUBKA W. AND EDER E., *Design Science,* Springer-Verlag, London, 1996.

[39] HUGES E., "Evolutionary Multi-objective Ranking with Uncertainty and Noise", in *Proceeding of the 1st International Conference on Multi Criteria Evolutionary Optimization*, Zurich, Switzerland, March 7-9, E. Zitzler et al. (eds.), Lecture Notes in Computer Science No. 1993, Springer Verlag, Berlin, 2001.

[40] HWANG C., PAIDY S., AND YOON K., "Mathematical programming with multiple objectives: a tutorial," *Computers & Operations Research*, vol. 7, pp. 5-31, 1980.

[41] ISAKSSON O., KESKI-SEPPÄLÄ S., EPPINGER S., "Evaluation of design process alternatives using signal flow graphs", *Journal of Engineering Design*, September 2000, Vol. 11 Issue 3, pp. 211-224.

[42] JANSSON A., *Fluid Power System Design - A Simulation Approach*, Dissertation, Linköping University, Linköping, Sweden, 1994.

[43] KEENEY R. AND RAIFFA H., *Decisions with multiple objectives - preferences and value tradeoffs*, John Wiley & Sons, New York, USA, 1976.

[44] KIRKPATRICK S., GELATT C. D., AND VECCHI M. P., "Optimization by simulated annealing," *Science*, vol. 220, pp. 671-680, 1983.

[45] KROO I., MCMASTERS J. AND SMITH S., Highly Nonplanar Lifting Systems, in *Proceedings of Transportation Beyond 2000: Technologies Needed for Engineering Design*, NASA Langley Research Center, September 26-28, 1995.

[46] KRUS P., JANSON A., PALMBERG J.-O., "Optimization Based on Simulation for Design of Fluid Power Systems, in *Proceedings of ASME Winter Annual Meeting*, Anaheim, USA, 1992.

[47] KRUS P., “Post optimal system analysis using aggregated design impact matrix”, in *Proceeding of ASME Design Automation Conference*, Baltimore, September 11-13, 2000.

[48] KRUS P., PALMBERG J.-O., LÖHR F. AND BACKLUND G., “The Impact of Computational Performance on Optimization in Aircraft System Design”, *in Proceedings of IMechE ‘Aerotech 95’*, Birmingham, UK, 1995.

[49] KRUS P., JANSSON A., BERRY P., HANSSON E., AND OVREBO K., “A Generic Model Concept for Optimisation in Preliminary Design,” in *Proceeding of World Aviation Congress and exposition*, Los Angeles, California, USA, 1996.

[50] LARSSON J., *Concepts for Multi-Domain Simulation with Applications to Construction Machinery*, Licentiate thesis, Linköping Studies in Science and Technology, Thesis no. 870, Linköping University, Linköping, Sweden, 2001.

[51] LJUNG L., *System Identification: Theory for the user,* 2nd ed., Prentice Hall, 1999.

[52] LJUNG L. AND GLAD T., *Modellbygge och simulering*, Studentlitteratur, Lund, Sweden, 1991.

[53] MANETSCH T. J., “Toward efficient global optimization in large dynamic systems - The adaptive complex method,” *IEEE Transactions on Systems, Man & Cybernetics*, vol. 20, pp. 257-261, 1990.

[54] MANETSCH T. J. AND CABRERA A. D., “Use of concurrent processing with the adaptive complex method for global optimization of large dynamic systems,” *IEEE Transactions on Systems, Man & Cybernetics*, vol. 21, pp. 442-445, 1991.

[55] MARVIS D., QIU S., “An Improved Process for the Generation of Drag Polars for use in Conceptual/Preliminary Design”, in *Proceedings of SAE World Aviation Conference*, San Francisco, USA, October 19-21,1999.

[56] MONGEAU M., KARSENTY H., ROUZÉ V., AND HIRIART-URRUTY J.-B., “Comparison of public-domain software for black-box global optimization,” Technical report LAO 98-01, Laboratoire Approximation et Optimisation, Université Paul Sabatier Toulouse, Toulouse, France, 1998.

[57] MORSE J. N., “Reducing the size of the nondominated set: pruning by clustering,” *Computers & Operations Research*, vol. 7, pp. 55-66, 1980.

[58] NEELAMKAVIL F., *Computer Simulation and Modeling*, John Wiley & Sons Inc, 1987.

[59] NELDER J. A. AND MEAD R., “A simplex method for function minimization,” *Computer Journal*, vol. 7, pp. 308-313, 1965.

[60] NILSSON K., ÅKERLUND O. AND HYNÉN A., “Using Optimisation and Statistical Methods to Analyse the Robustness of Integrated Hardware/Software Systems”, in *Proceedings of the fifth Scandinavian International Conference on Fluid Power SICFP’97*, Linköping, Sweden, 1997.

[61] OSYCZKA A., *Multicriterion Optimization in Engineering - with FORTRAN Programs,* Ellis Horwood, Chichester, 1984.

[62] PAHL G. AND BEITZ W., *Engineering Design – A Systematic Approach*, Springer-Verlag, London, 1996.

[63] PAPALAMBROS P., MICHELENA N., KIKUCHI N., "Distributed Cooperative Systems Design", in *Proceedings of the International Conference on Engineering Design*, Tampere, Finland, August 19-21,1997.

[64] PHADKE M. S., *Quality Engineering Using Robust Design*, Prentice Hall, 1989.

[65] POHL J., KRUS P., PALMBERG J.-O., "A Model Based Approach for the Conceptual Design of a Hydraulic Valve Train System", accepted for publication in *Proceedings of Bath Workshop on Power Transmission & Motion Control*, Bath, UK, September 12-14, 2001.

[66] PUGH S., *Total Design*, Addison-Wesley Publishers Ltd., 1991.

[67] RAO S. S., *Engineering Optimization*, John Wiley & Sons Inc, 1996.

[68] RECHENBERG I., *Evolutionsstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, Friedrich Frommann Verlag, 1973.

[69] RINGUEST J., *Multiobjective Optimization: Behavioral and Computational Considerations*, Kluwer Academic Publishers, Boston, 1992

[70] ROOZENBURG N. AND EEKELS J., *Product Design: Fundamentals and Methods*, John Wiley & Sons Inc, 1995.

[71] ROSENMAN M. A. AND GERO J. S., "Reducing the Pareto opimal set in multicriteria optimization (With application to Pareto optimal dynamic programming)," *Engineering Optimization*, vol. 8, pp. 189-206, 1985.

[72] SAHOO N. K. AND APPARAO K. V. S. R., "Modified complex method for constrained design and optimization of optical multilayer thin-film devices," *Applied Physics A-Solids & Surfaces*, vol. 59, pp. 317-326, 1994.

[73] SCHRAGE M., *Serious Play: How the world's best companies simulate to innovate*, Harvard Business School Press, 2000.

[74] SENIN N., WALLACE D. R., AND BORLAND N., "Mixed continuous and discrete catalog-based design modeling and optimization," in *Proceeding of teh 1999 CIRP International Design Seminar*, University of Twente, Enschede, The Netherlands, 1999.

[75] SIDDALL J., *Analytical decision-making in engineering design*, Prentice-Hall, Englewood Cliffs, New Jersey, 1972.

[76] SIMON H., *The Sciences of the Artificial*, MIT Press, 1969.

[77] SMITH J., R., "Designing Biomorphs with an interactive genetic algorithm", in *Proceedings of the 4th International Conference on Genetic Algorithms*, July 13-16, Publ. by Morgan-Kaufmann Publ., Palo Alto, USA, 1991.

[78] SMITH R. P. AND EPPINGER S. D., "Identifying Controlling Features of Engineering Design Iteration", *Management Science*, Vol. 43, no. 3, pp. 276-293, 1997.

[79] SMITH R. P. AND EPPINGER S. D., "A Predictive Model of Sequential Iteration in Engineering Design", *Management Science*, Vol. 43, no. 8, pp. 1104-1120, 1997.

[80] SMYTH S., WALLACE D., "Towards the Synthesis of Aesthetic Product Form", in *Proceedings of ASME Design Theory and Methodology Conference*, Baltimore September 10-13. 2000.

[81] SPENDLEY W., HEXT G. R., AND HIMSWORTH F. R., "Sequential application of Simplex designs in optimisation and evolutionary operation," *Technometrics*, vol. 4, pp. 441-462, 1962.

[82] STEUER R., *Multiple criteria optimization: theory, computation and application*, John Wiley & Sons, Inc, New York, 1986.

[83] STEUER R. E. AND CHOO R.-U., "An interactive weighted Tchebycheff procedure for multiple objective programming," *Mathematical Programming*, vol. 26, pp. 326-344, 1983.

[84] STEWARD D. V., The design Structure System: A Method for Managing the Design of Complex Systems, *IEEE Transaction on Engineering Management,* vol. EM-28, no. 3 pp. 71-74, August, 1981.

[85] SULLIVAN L. P., "Quality Function Deployment", *Quality Progress*, June, 1986.

[86] TAGUCHI G., WU Y., *Introduction to Off-Line Quality Control*, Central Japan Quality Control Association, Nagoya, Japan, 1980.

[87] TAMAKI H., KITA H., AND KOBAYASHI S., "Multi-objective optimization by genetic algorithms: a review," in *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, ICEC'96, Nagoya, Japan, 1996.

[88] TAPPETA R., RENAUD J., MESSAC A., SUNDARARAJ G., "Interactive Physical Programming: Tradeoff Analysis and Decision Making in Multicriteria Optimization", *AIAA Journal*, vol. 38, no. 5, pp. 917-926, May 2000.

[89] TEICH J., "Pareto-Front exploration with Uncertain Objectives", in *Proceeding of the 1st International Conference on Multi Criteria Evolutionary Optimization*, Zurich, Switzerland, March 7-9, E. Zitzler et al. (eds.), Lecture Notes in Computer Science No. 1993, Springer Verlag, Berlin, 2001.

[90] THURSTON D., "A formal method for subjective design evaluation with multiple attributes," *Research in engineering design*, vol. 3, pp. 105-122, 1991.

[91] TSUTSUI S., GHOSH A., "Genetic Algorithms with a Robust Solution Searching Scheme," *IEEE Trans. on Evolutionary Computation*, vol. 1 no. 3, September 1997.

[92] VON NEUMANN J., MORGENSTERN O, *Theory of games and economic behavior*, Princeton, Princeton Univ. Press, 1944.

[93] WALLACE D. R., JAKIELA M. J., AND FLOWERS W. C., "Multiple criteria optimization under probabilistic design specifications using genetic algorithms," *Computer-aided Design*, vol. 28, pp. 405-421, 1996.

[94] WALLACE D., ABRAHAMSON S., SENIN N., SFERRO P., "Integrated Design in a Service Marketplace," *Computer-aided Design*, volume 32, no 2, pp. 97-107, 2000.

[95] WHEELWRIGHT S. AND CLARK K., *Revolutionizing Product Development – Quantum Leaps in Speed, Efficiency, and Quality*, The Free Press, 1992.

[96] WHITNEY D., "Designing the Design process", *Research in Engineering Design*, 2:3-13, 1990.

[97] YEN J., LIAO J., LEE B., AND RANDOLPH D., "A hybrid approach to modeling metabolic systems using a genetic algorithm and simplex method," *IEEE Transaction on systems, man and cybernetics*, vol. 28, pp. 173-191, 1998.

[98] ZIMMERMANN H.-J. AND H.-J. S., "Intelligent system design support by fuzzy-multi-criteria decision making and/or evolutionary algorithms," in *Proceedings of IEEE International Conference on Fuzzy Systems*, Yokohama, Japan, 1995.

[99] ZITZLER E. AND THIELE L., "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," *IEEE Transaction on evolutionary computation*, vol. 3, pp. 257-271, 1999.

Paper I

A Design Process Modeling Approach Incorporating non-Linear Elements

Johan Andersson, Jochen Pohl and Steven Eppinger

This paper has been published as:
ANDERSSON J., POHL J. AND EPPINGER S. D., "A Design Process Modeling Approach Incorporating non-Linear Elements", in *Proceedings of ASME Design Theory and Methodology Conference,* Atlanta, USA, September 13-16, 1998.

A Design Process Modeling Approach Incorporating non-Linear Elements

Johan Andersson[*], Jochen Pohl[*] and Steven Eppinger[**]

[*]Department of Mechanical Engineering
Linköping University
SE-581 83 Linköping, Sweden

[**]Sloan School of Management
Massachusetts Institute of Technology
Cambridge, MA 02142, USA

Abstract

This paper extends the literature of engineering design process modeling. We focus on the modeling of design iterations using a task-based description of a development project. We present a method to compute process performance and to relate this outcome to critical activities within the process.

Design tasks are modeled as discrete-event activities with design information flowing between them. With every design task, we associate process characteristics such as the completion time and cost per time unit for the task. These characteristics can change with the advance of the design process. The method is especially suited for comparison of different design processes on the basis of overall process costs and lead time.

In order to illustrate the method a simple design process was modeled as an example. Based on this model, the process lead-time distribution and the process costs were simulated.

1 Introduction

The costs of a product development project are roughly proportional to the number of people involved and the duration of the project [10]. For today's manufacturing firms a well managed product development process is therefore an important factor to stay competitive. Process modeling can be one course of action to discover key activities that influence process lead time and process costs.

In order to be able to compare different design processes, it is important to estimate their expected lead times. It is also helpful to understand why the lead-time varies. Sensitivity studies can be a complement to design process modeling in order to gain further insight into the iteration process. Key activities which strongly influence lead time and process costs can be identified through sensitivity analysis.

When engineering costs are also incorporated in the model, the costs of the development process can be calculated. This makes the method well suited for comparison of different processes. The fastest process is not necessarily the cheapest one. For example, a process that involves several parallel activities may be more expensive than one where the work is done sequentially.

2 Design Process Modeling

Iteration is fundamental to the design process, as is stated by several authors [5] - [9]. An increased understanding of design iteration will enlarge our understanding of the design process. Iterations result from a coupled design task structure, one in which the (coupled) tasks require information from each other [5].

Generally there are two different ways to execute coupled design tasks: sequential iteration or parallel iteration. Prior models, Eppinger et al. [6], and Smith and Eppinger [8], describe methods to depict sequential iterations where coupled tasks are performed in sequence and rework is considered by a probabilistic chance of feedback to earlier tasks. Both the task time and the rework probability are constant with time. A parallel iteration model is presented by Smith and Eppinger in [7], where the iterations are carried out in parallel and the amount of rework decreases in a linear manner. Adler et al. [1], model a scenario with concurrent projects and resource constraints. Engineering resources are modeled as workstations and projects as jobs. A job in their model is either receiving service from a workstation or queuing. Iterations in their modeling approach are purely sequential with fixed characteristics.

Related work has also been done by Austin et al. [2], where the construction design process is modeled as a discrete event system. Bell et al. [3] modeled a product development process using the dynamic systems metaphor. The design process is modeled as purely sequential or parallel. They focus on computational design, considering process lead time, costs, and design quality expressed in terms of objective functions. In parallel-task scenarios, iteration is required to resolve conflicting goals. Christian and Seering [4], model design process dynamics based on a detailed representation of activities taking place among several designers in a team. This approach allows both sequential and parallel iterations as well as overlapping.

The approach presented in this paper follows most closely from Eppinger et al. [6]. We combine both sequential and parallel execution of design tasks in a more general process model. This is possible due to the way that process lead times are computed. Some non-linear properties can be modeled, i.e. the task time and the probability of rework can vary with the number of iterations completed. The introduction of a cost factor adds another novel dimension to the process model.

3 Modeling approach

Design process modeling, as implemented here, is based on the observation that a design process is comprised of a number of smaller design activities. The process can be modeled by tracking design information that is exchanged between different design tasks. Work is executed as information flows to the design tasks. In such models, both parallel and sequential flows can often be observed.

3.1 Model elements

The model elements include two types: design tasks and design reviews. We connect these by the design information/work flows.

Design Tasks- With every design task, we associate characteristics of execution time and task cost per time unit, as described in Figure 1.

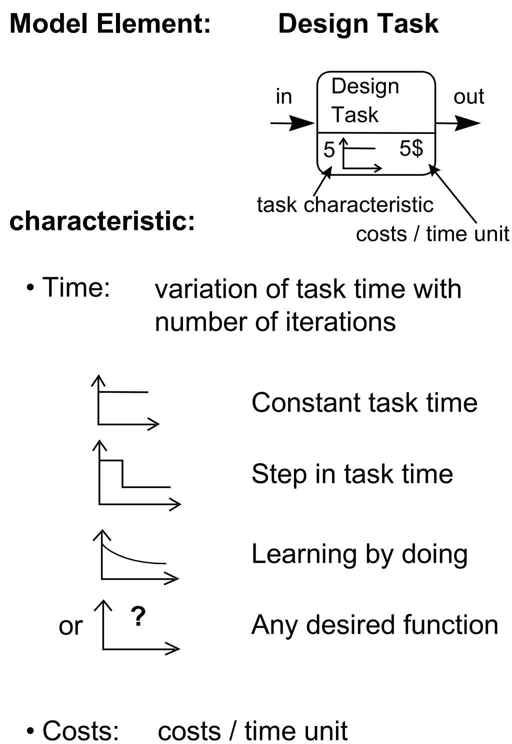


Figure 1. Characteristics of the model element **design task.**

To create a flexible and accurate model, we allow the task characteristics to vary with the number of iterations done. Consider for example a design process with a large amount of CAD modeling. In the first design iteration, the CAD models have to be created, but in the second iteration they need only to be modified, which is less time intensive. This would correspond to a step reduction in task time as shown in Figure 1. A task in which the execution time decreases with every design iteration can be modeled as a "learning-by-doing" task with an associated learning curve function.

Since the analysis method is based on a non-linear approach, even an arbitrary functional or random relationship between an individual task duration and the number of design iterations is conceivable. However, the simplest case is also possible— a constant task time.

Design Reviews- The design review model element (shown in Figure 2) depicts the probability of proceeding forward to the next design task, otherwise the process flows back to an earlier task. The design review is evaluated with the help of a random function. The characteristics of the design review can also be a function of the number of iterations. Again, the relationship between number of iterations and design review probability can be a step in probability, a "learning-by-doing" function, or simply constant.

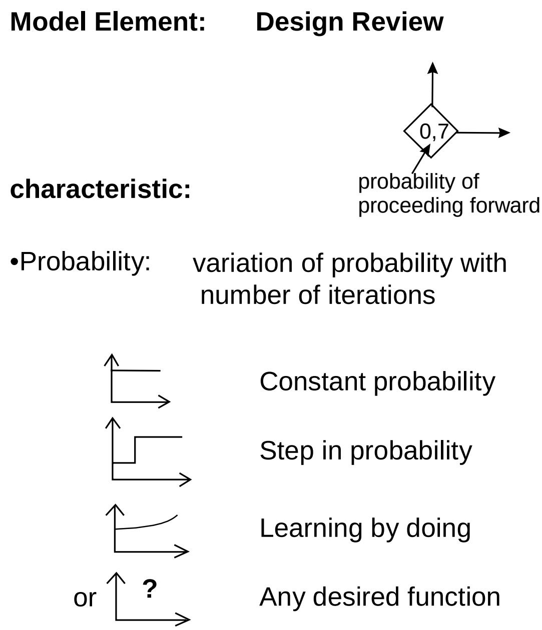


Figure 2. Characteristics of the model element **design review.**

3.2 Computation of lead time and process costs

The nonlinear, probabilistic design process model is analyzed using one of two numerical methods: modified Monte-Carlo simulation or depth-first search, depending upon the functional form of the model elements.

In the modified Monte-Carlo method, the input signal to the first model element is an impulse. As the impulse passes through the model network, the appropriate task execution times and costs are accumulated. Each time the impulse passes a design review, a probabilistic choice is made to determine the direction to proceed. The likelihood of the path is calculated by multiplying the probabilities at each design review passed by the impulse. When the final task is reached, the path is stored together with its lead-time, cost, and probability, unless this specific path has already been found earlier. By calculating the exact probability of each path found, we do not rely on a large number of simulation runs to determine the likelihood of the paths found. Different paths through the process model can result in different probabilities with the same lead-time. In this case, the path probabilities are added up to one single probability for that specific lead-time. The probabilities of all paths found are then summed up to an accumulated probability which is used as a measure to stop simulation when close to 100% is reached (say 99%). Since the number of paths is infinite, it is impossible to find them all. The paths that have not been found have a very low probability of occurring.

In order to speed up the simulation, one could introduce an additional probability in the design review elements. This can be used to steer the impulse propagating through the model in a more efficient manner. This figure can be changed as a function of the accumulated probability, so that with increasing probability the impulse discovers the less likely paths. However, for the calculation of process probability the original (model) probability value is used. After a few simulations, the most common paths are found. With this method, the likelihood of finding the paths with low probabilities is increasing with the number of simulations done and the time to reach a certain accuracy decreases significantly.

In the depth-first search method, the network is fully explored by enumerating (almost) all possible paths. The analysis begins by tracing the impulse through the process, accumulating time, cost, and probability, until a design review is reached. One path is chosen at this point, but the alternate path is noted (on a stack) for future exploration. A path is followed until either the final task is reached or a very low probability is reached (say 0.1%). All paths of interest are thus identified in a rather efficient manner. Note that this depth-first method is only appropriate where the design task execution times and design review probabilities are explicit functions of the state (number of iterations and/or accumulated duration), not random functions.

The outcome of these analyses is the list of all paths found, their lead times, costs and probabilities. From these data, the expected lead-time and costs can be computed. Not all possible paths in the model can be found, therefore the expected lead time can only be calculated approximately. The paths that not have been found are likely to have low probability values and long lead times, which leads to a slight underestimation of the expected lead-time.

The complete lead-time distribution can also be plotted, as shown in figure 4 for the example in the next section. It also shows how the lead-time varies. If the accumulated probability is plotted on the same graph, as shown in figure 4, one can say what the likelihood is that the design process will be finished within a certain time. Combining this with the expected cost of each lead-time, as shown in figure 4, one can understand the expected cost of the development process. The results shown are computed with the help of the modified Monte-Carlo method.

The model analysis handles design tasks executed in parallel. The beginning of a parallel activity flow is called a fork and the finish is a joint. These paths are depicted as arrows in the information flow model diagram. When the impulse passes a fork it splits up into as many impulses as there are arrows. The impulses propagate through the model until a joint is reached. At the joint, the incoming impulses of each parallel path are delayed until all impulses have arrived, and one is passed through. After the joint, only one impulse is used for evaluation.

3.3 Sensitivity analysis

A sensitivity analysis of the design process provides insight as to how each task and design review influences overall lead time and cost, allowing us to focus improvement efforts accordingly. The expected value and distribution of lead-time and cost are dependent on the task characteristics and the probabilities in the design reviews. The sensitivity of the expected value of the lead time or cost can be calculated as the relative change in expected value due to a small change in a parameter, e.g. a task characteristic or a design review probability. If for instance L represents the lead-time and k a parameter of interest, the sensitivity of L to changes in k is given by:

$$S_k^L = \frac{\Delta E[L]/E[L]}{\Delta k/k}$$, according to Eppinger et al (1997)

3.4 Test case

Here an industrial development process is taken as a test case to illustrate the modeling approach and the analysis methods. The input data to this basic model are obtained by interviewing the engineers involved.

The development process studied is that of hydraulic pump design at a manufacturer of heavy mobile equipment. The model is depicted in Figure 3 below.

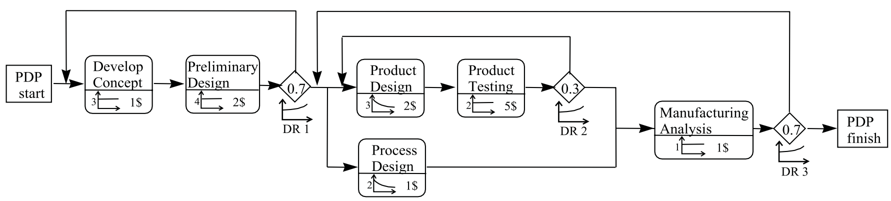


Figure 3. Hydraulic pump development process

Inputs to the process are constraints such as the fluid to be used, working conditions, rotational speed, pressure and so forth. In the concept design and preliminary design tasks, parameters such as pump type, the material to use, lubrication issues, bearings, and physical layout are established. Both of these tasks have constant lead-time and are relatively inexpensive. The probability of rework is low (30% to start) and decreasing with the number of iterations executed.

The next phase is where the detailed design takes place. The product design task is not very time consuming because most parameters are already set. Product testing includes prototyping and is the most expensive task due to the large amount of hardware and engineering time involved. Because of the uncertainties in the analysis methods used in detailed design the likelihood of having to repeat the product design and testing phase is high.

In parallel with product design and testing, the manufacturing process design is performed. When doing these tasks in parallel the lead-time is only dependent upon the most time consuming path (product design and testing). On the other hand, both paths contribute to the total process cost. The process continues with the manufacturing analysis and eventually a final design review before completion.

3.5 Results

The lead time distribution of the development process is shown in Figure 4, together with the cumulative probability and the expected value of lead time. With the help of such a graph it is possible to get a sense of the performance variation within a development process.

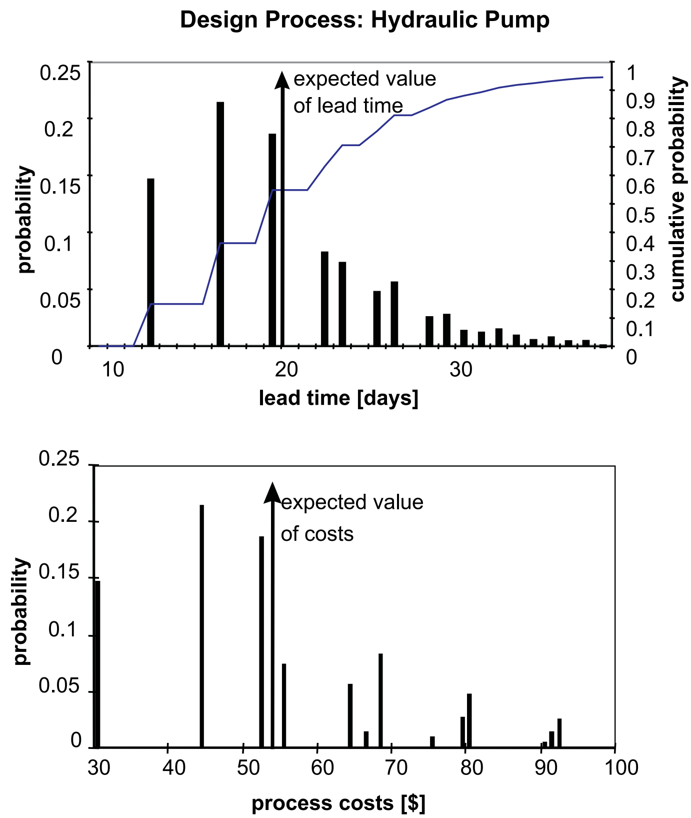


Figure 4. Lead time and cost distribution of the pump development process.

The graph shows the lead times of all paths shorter than 40 time units. The shortest lead-time possible is 13 time units and the expected value of lead-time is 20.4 time units. The likelihood of finishing within a certain lead time can also be read from this graph, e.g. the likelihood of completing the development process within 25 time units is approximately 70%. This measure helps to understand the variation of the process lead-time and the schedule risk of the development process.

The associated cost distribution and the expected cost of the development process are graphed in Figure 4. The cost and the lead time distribution are similar, because the cost is implemented as proportional to the task lead-time. When analyzing just one development process this might be superfluous, but when more development processes are compared it adds a useful dimension to the comparison.

The results of a sensitivity analysis explain the relative importance of the parameter values in the model. The sensitivity of overall lead-time and cost are calculated for changes in task lead time and design review probabilities, as shown in Table 1 and 2.

Table 1. Lead-time and cost sensitivity due to changes in task lead-time

Design Task s	Develop Concept	Prelim. Design	Product Design	Product Testing	Process Design	Manuf. Analysis
Lead Time	0.20	0.24	0.33	0.25	0	0.05
Cost	0.08	0.18	0.25	0.48	0.04	0.02

Table 2. Lead-time and cost sensitivity due to changes in design review probability.

Design Review	DR1	DR2	DR3
Lead Time	-0.35	-0.15	-0.25
Cost	-0.22	-0.24	-0.38

The sensitivity analysis confirms a general insight that tasks performed frequently are more sensitive to changes in task parameters. The positive sensitivity values indicate to what extent lead times and costs increase for positive variations of the task times. The negative sensitivity values identify that increasing forward probabilities in the design reviews shorten the process lead time and cost.

The highest cost sensitivity value is the sensitivity to the lead-time of product testing, which is the most expensive task. The highest time sensitivity value is not for the longest-duration task. The highest time sensitivity instead is to changes in the duration of product design, which is embedded within the most frequently performed iteration loop. Another insight is that the process design task has no influence on overall lead time because it is carried out in parallel with product design and testing, which together have a longer lead time. However, process design still affects the total cost.

The sensitivity analysis on design reviews shows that changes in the success rate of the first design review (labeled DR1) has the strongest impact on lead-time. This is because an iteration in the DR1 loop takes longest time. The rate of the third design review (DR3) has the greatest impact on the process cost, because one iteration of this loop is more expensive than a repetition through the other loops in the model.

4 Discussion

In this section, we discuss the assumptions and limitations of the modeling approach and the insights that can be gained by using this method to model design processes.

4.1 Assumptions and limitations

The modeling approach is based on the assumption that the workflow of a design process can be described by a probabilistic rule governing the likelihood that tasks have to be executed or repeated during the design process. We assume that there are no time delays due to lack of information. (Such delays can be included in the task lead-time.)

The model presented here does not consider any queuing effects. As observed by others, Adler et al. [1] and Eppinger et al. [6], queuing effects can be significant. In some cases, the delays due to queuing can be longer then the actual task lead-time. This is likely to happen when engineers are involved in several parallel development projects or many process steps. An approach to handle the effects of queuing using the Monte-Carlo analysis method is to model several development processes performed simultaneously. Queuing can then be taken into consideration by tracking tasks that share resources and assuring that when one task is executed the others have to queue up. The queue may then be treated using a first-in-first-out rule or any other job prioritization rule.

Computation of the expected value and the variation is done numerically and thereby always with a certain amount of uncertainty. By using the modified Monte-Carlo method, and by calculating the accumulated probability we keep track of the uncertainty.

4.2 Insights gained by process modeling

This modeling approach provides engineering teams insight into their development processes through computation of lead-time probability distributions and cost variations and by sensitivity analyses. It is a powerful aid to compare and evaluate different development processes. Some of the insights and positive effects are suggested below:

- *Understanding of the process:* Studying the lead-time probability distribution and the sensitivity analysis yields a deep understanding of the process. The tasks which have the greatest influence on lead-time and costs can be identified and thereby focused upon when improving the process. The sensitivity analysis also identifies which design reviews launch iterations with the largest impact.

- *Evaluation of risk:* The variation of the lead-time and the cost helps in estimating the budget and schedule risk of the project.

- *Comparing alternative design processes:* Our approach makes it possible to compare different design processes in terms of lead time and development cost distributions. For example, performing more tasks in parallel may reduce lead time but may raise development costs.

5 Conclusion

The modeling approach presented here provides a powerful and flexible method for modeling and analysis of development processes. The modeling method is non-linear and incorporates dynamic changes of design conditions in a straightforward manner without expanding the model. Both sequential and parallel workflow can be modeled in a natural way. By incorporating both process lead time and cost, this method is well suited for comparison of different development processes. The model provides information of the expected value and the probability distribution of lead-time and costs. By conducting sensitivity analyses on the lead-time and cost due to changes in model parameters, a deeper insight into the iterative development process is gained.

6 References

[1] Adler P.S., Mandelbaum A., Nguyen V., and Schwerer E., 1995, "From Project to Process Management: An Empirically-based Framework for Analyzing Product Development Time", *Management Science*, Vol. 41, No. 3.

[2] Austin S.A., Baldwin A.N., Thorpe A., and Hassan T.M., 1995, "Simulating the Construction Design Process by Discrete Event Simulation", International Conference on Engineering Design, Praha, Czech Republic.

[3] Bell D.G., Kannapan S.M., and Taylor D.L., 1992, "Product Development Process Dynamics", ASME Design Theory and Methodology Conference, DE-Vol. 2, pp. 257-266.

[4] Christian A.D. and Seering W.P., 1995, "A Model of Information Exchange in the Design Process", ASME Design Engineering Technical Conferences, DE-Vol. 83, pp. 323-328.

[5] Eppinger S.D., Whitney D. E., Smith R.P., and Gebala D.A., 1994, "A Model-Based Method for Organizing Tasks in Product Development", *Research in Engineering Design*, Vol. 6, no. 1, pp. 1-13.

[6] Eppinger S.D., Nukala M.V., and Whitney D.E., 1997, "Generalized Models of Design Iteration Using Signal Flow Graphs", *Research in Engineering Design*, Vol. 9, no. 2, pp. 112-123.

[7] Smith R.P. and Eppinger S.D., 1997a, "Identifying Controlling Features of Engineering Design Iteration", *Management Science*, Vol. 43, no. 3, pp. 276-293.

[8] Smith R.P. and Eppinger S.D., 1997b, "A Predictive Model of Sequential Iteration in Engineering Design", *Management Science*, Vol. 43, no. 8, pp. 1104-1120.

[9] Steward D.V., 1981, "The Design Structure System: A Method for Managing the Design of Complex Systems", *IEEE Transactions on Engineering Management*, Vol. EM-28, no 3, pp. 71-74.

[10] Ulrich K.T. and Eppinger S.D., 1995, "Product Design and Development", McGraw-Hill, New York.

Paper II

Design of Objective Functions for Optimization of Multi-domain Systems

Johan Andersson, Jochen Pohl and Petter Krus

This paper has been published as:
ANDERSSON J., POHL J. AND KRUS P., "Design of Objective Functions for Optimization of Multidomain Systems", in *Proceedings of ASME Annual Winter Meeting*, Fluid Power Systems and Technology, Anaheim, USA, November 15-20, 1998.

Design of Objective Functions for Optimization of Multi-domain Systems

Johan Andersson, Jochen Pohl and Petter Krus

Department of Mechanical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Abstract

Engineering problems are often characterized by many conflicting objectives which span over several engineering domains. In this paper, an approach to design of objective functions for engineering problems is presented.

Multi-domain systems can be characterized as complex systems which combine different fields of engineering. Here an aircraft landing gear is studied which is a true example of a multi-domain simulation problem due to the presence of mechanical, electrical and hydraulic sub systems. On this simulation model a non-gradient optimization strategy is applied where the formulation of the objective function is supported by the House of Quality method. The design of an objective function for optimization is in many cases a complex procedure involving a great amount of expertise. The House of Quality method is used here both to facilitate team activities and to elucidate the relation between system characteristics and system parameters.

1 Introduction and background

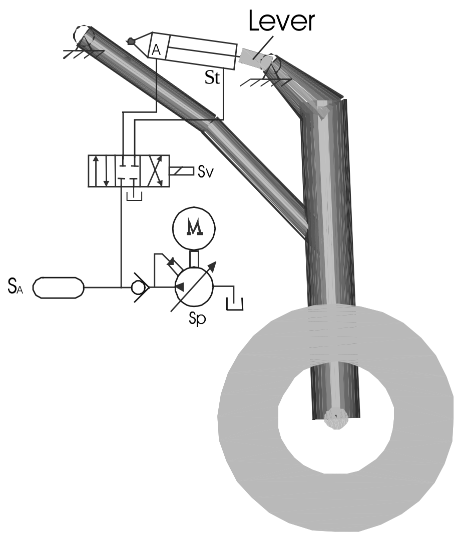
The engineering design process may be understood as an interactive feedback process incorporating the phases clarification of the task, conceptual design, embodiment design and detailed design, see for instance Pahl and Beitz [10]. The performance of the design is compared with the performance specification and changes are made according to the outcome of the comparison. Usually, this is a manual process in which the designer makes a prototype system that is tested and modified until satisfactory. With the help of simulation, the prototyping can be reduced significantly. If the desired behavior can be expressed as a figure of merit, i.e. an objective function, it is possible to introduce optimization as a tool to reach an optimal solution. In this context, optimization can be seen as a semi automation of the design process.

Optimization of multi domain systems based on simulation is characterized by a nonlinear system representation. Gradient optimization methods are unsuitable for these kinds of systems since the objective function is only given by simulation results, i.e. it can not be derived analytically. Optimization methods that can handle these types of problems are non-gradient methods such as Genetic Algorithms (GA's) as presented by for instance Goldberg [4] or the Complex method presented by Box [1]. These methods have been used in simulation optimization problems by for instance Jansson [7], Krus et al. [8] and Pohl et al. [11].

Quality function deployment (QFD) has proven to be a useful tool in providing means for translating customer requirements into critical product control characteristics. This is stated by Sulivan [13], Hauser et al. [5] and Cohen [2]. In this paper, the House of Quality method has been employed in order to support objective function design. The advantages of the House of Quality method are that it facilitates both team activity and an understanding of customer requirements.

2 Landing gear design process

The landing gear concept chosen for this study is shown in Figure 1. This is a very common landing gear of today's regular aircraft. The landing gear system consists of the actual landing gear, the hydraulic actuator that creates the retardation movement and the hydraulic supply system.



Pump parameters
speed: n_{pump}= 1500rpm
displacement: Dp = 10 cm^3/rev
pressure: p=250 bar

Piston parameters
diameter: D_1 = 19mm
area ratio: α = 0.5

Valve parameters
diameter: sd= 7mm
max. opening: xv_{max}=1.8mm

Figure 1. The landing gear system and data of the optimized system.

This study focuses on a concept with a local hydraulic system that supports only the landing gear system. This is not common in today's aircraft. However, the trend in modern aircraft design is going towards more decentralized hydraulic systems.

In order to understand the landing gear design process a parameter based Design Structure Matrix (DSM) was created. The Design Structure Matrix was originally developed by Stewart [14] but has been further developed by for instance Smith et al. [12] and Eppinger et al. [3]. These authors extended the originally binary matrix with measures for the degree of dependencies and task duration. However, here the DSM is used in its binary form where the crosses indicate task dependencies and the dots symbolize the actual design task. The dots are included to distinguish the upper and lower triangles of the matrix from each other.

The strength of the DSM method is the visualization of the coupling between different design tasks. It thereby makes it possible to reorganize the design process in order to make it perform more successfully. Here the matrix is employed to graphically display the information flow of the landing gear design process.

The marked elements in each row identify which parameters have to be determined before that particular parameter could be determined. When analyzing a DSM, the first step is to try to make the matrix lower triangular, which decouples design tasks from each other. In Figure 2 a dummy DSM is depicted in order to explain different sequences of a design process.

	A	B	C	D	E	F	G	H	
A	●	X		X					Coupled
B	X	●	X						
C		X	●						
D	X		X	●					
E		X	X	X	●				Series
F				X	X	●			
G		X			X	X	●		Parallel
H			X	X		X		●	

Figure 2. Binary design structure matrix.

In the coupled blocks, there are marks above the diagonal which indicate feedbacks which lead to iterations. Iterations are very significant for engineering design problems. An example from the DSM in Figure 3 is that the piston area can not be designed without knowing the lever angle and length. On the other hand does the designer have to know the piston area and the lever angle in order to decide on the lever length.

	Landing gear leg					Movement control								
						Actuation				Supply				
	A	B	C	D	E	F	G	H	I	J	K	L	M	
A	●	X												Wheel diameter
B	X	●												Number of wheels
C	X		●	X										Leg length
D			X	●										Leg diameter
E			X	X	●									Supporting beam
F	X	X	X	X		●	X	X	X	X		X		Length of lever
G	X	X	X	X		X	●	X	X	X		X		Angle of lever
H	X	X	X	X		X	X	●		X				Piston Area
I	X		X			X	X		●					Piston stroke
J								X		●				System pressure
K								X	X	X	●			Valve size
L								X	X			●	X	Pump size
M								X	X	X		X	●	Accumulator size

Figure 3. Design Structure Matrix for the landing gear design process.

As seen from Figure 3, the landing gear development process is block- lower- triangular. The blocks indicate design tasks that are executed in sequence. In each block, the marks above the diagonal indicate feedback that leads to a strong coupling between these parameters within that block. The two larger blocks are added in order to visualize the major design activities. From the DSM three main activities can be derived, namely the landing gear leg design, the actuation design and the design of the hydraulic supply system.

As can be seen from the DSM the parameters for the system that creates the movement are totally dependent on the landing gear parameters but not the other way around. This indicates that these two design processes can be done in sequence. Optimization of the movement control parameters could therefore be conducted with constant values of the landing gear leg parameters. This is no limitation or simplification of the actual design problem since the design processes are done purely sequentially.

3 The employment of QFD

In this paper the ideas of quality function deployment (QFD) were employed in order to elucidate the requirements on a landing gear system. The strength of QFD is that it provides means for relating system characteristics to appropriate system parameters.

			System Parameters													
			Landing gear						Movement control							
				Wheel			Landing gear leg				Actuator		Hyd. Supply			
		Wheighting	Supporting beam	Wheel dimensions	Number of wheels	Brakes	Leg length	Leg diameter	lever length	Lever angle	Piston area	Piston stroke	Valve size	Pump size	Accumulator size	Pressure level
System Characteristics	Descending velocity	D					5	5								
	Retraction time	D							5	5	5	5				
	Weight	3		5	5	5	5	5			1	1	1	3	3	
	Energy consumption	3		5	5	5	5	5	1	1	3	3	3	3	3	3
	Price	1	3	3	3	5	5	5			1	1	1	3	1	
	Ground carrying capacity	D		5	5											
	Locking landing gear	D	5													
	Brake distance	D				5										

Figure 4. Relationship matrix for the landing gear system.

This study constitutes a technical application of the House of Quality method on a system level. First, the characteristics of the landing gear system were established. These characteristics have been listed on the vertical axis of the relationship matrix in Figure 4. Some characteristics come from aircraft regulations (FAR) that have to be fulfilled, i.e. these are demands that the system has to meet, for instance minimum descending velocity or maximum brake distance. Other characteristics express the customer requirements.

Thereafter the landing gear system parameters were listed on the horizontal axis, given the specific landing gear concept from Figure 1. By relating these parameters to the system characteristics the relationship matrix evolves. The strength of the relationship matrix is the visualization of the relation between the system characteristics and the system parameters. The figures in the matrix express how strong the relation between a particular requirement and a product characteristic is, a high figure indicates a strong relationship. The boxes that lack a figure are as important as the other ones, since they indicate a missing relationship between the specific requirement and characteristic. The weighting factors express the relative importance of different system characteristics and the letter “D” indicates that this particular characteristic is a demand, which has to be fulfilled.

The descending velocity determines the forces that act on the landing gear during landing. The required descending velocity is therefore dimensioning for the landing gear structure, i.e. required length and diameter of the leg in order to manage the ground contact. Retardation time is determined by both drag and weight of the landing gear as well as the actuation system. Energy consumption is depending on factors such as drag and weight of the landing gear and type of actuation system. The braking distance is only influenced by the size of the brakes. The forces on the landing gear due to braking are of secondary importance in this study.

The value for a particular relation is of minor importance, the fact that it is nonzero indicates that a parameter does influence a particular system characteristic. If such a parameter is changed during the design process, it might be considered to include the corresponding system characteristic in the objective function.

Weak dependencies indicate that a requirement does not necessarily have to be included in the objective function. Thereby the relationship matrix supports the engineer when deciding which requirements should be taken into account in the objective function.

In this study, the movement control parameters (see Figure 3) are the free design parameters that can be manipulated in order to achieve an “optimal” design. This implies that the system characteristics that can be affected are weight, energy consumption and system costs, see Figure 4. The demand on retardation time acts as a constraint on the design.

When using optimization as a tool in the design process it is obvious that the “optimal” design solution has to minimize weight, costs and energy consumption without violating the retardation time demand.

4 The design problem

The design problem is to design an actuation system that retards a given landing gear within a certain time and simultaneously minimizes weight, cost and energy consumption.

The system weight can be expressed as a function of the size of each component. Since the optimization is based on component size selection rather than component design, it is assumed that each component is in some sense already optimized. A consequence of this assumption is that most parameters of a component or sub system can be described as a simple function of a very limited set of performance parameter, i.e. the size.

The weight function of the actuation system can be expressed as:

$$f_{weight} = K_1 \cdot s_{Pump} + K_2 \cdot s_{Acumulator} + K_3 \cdot s_{valve} + K_4 \cdot s_{Piston} \tag{1}$$

where $K_1,..,K_4$ are constants that relate component size to component weight, and s_i is the size of a particular component. The weight of the landing gear lever is negligible in this context. The remaining landing gear system parameters, that we can not influence, do not have to be included in the objective function.

Assuming that the component cost is proportional to the component size the cost function could be expressed as

$$f_{\cos t} = C_1 \cdot s_{Pump} + C_2 \cdot s_{Acumulator} + C_3 \cdot s_{valve} + C_4 \cdot s_{Piston} \tag{2}$$

where $C_1,..,C_4$ relates component size to cost.

The systems energy consumption can be expressed in the following manner.

$$f_{Energy} = \int_0^{t_{ret}} p_s \cdot q_s \cdot dt \tag{3}$$

were: p_s = the system pressure, $q_s =$ system flow and $t_{ret} =$ the retardation time.

The design problem to minimize f_{Emergy}, f_{Emergy} and f_{Cost} subjected to the retardation constraint can be expressed as a minimization of

$$F\left(X_1, X_2,..X_N\right) \tag{4}$$

subjected to the constraints

$$G_i \leq X_i \leq H_i \tag{5}$$

where i=1,2,..M. The implicit variables $X_{N+1},.., X_M$ are dependent functions of $X_1,.., X_N$. The constraints G_i and H_i are either constants or functions of $X_1,.., X_N$. In our design study, the retardation time is an example of an implicit constraint given by the system parameters $X_1,.., X_N$ through the simulation. For the component size parameters, $X_1,.., X_N$, the explicit constraints G_i and H_i are all constants.

When the objective is to minimize the functions $f_1, f_2,.., f_i$ the objective function F can be represented as

$$F = \left(\frac{f_1}{f_{10}}\right)^{\gamma_1} + \left(\frac{f_2}{f_{20}}\right)^{\gamma_2} + .. + \left(\frac{f_i}{f_{i0}}\right)^{\gamma_i} \tag{6}$$

Here $f_{10},.., f_{i0}$ are the function values from an evaluation of one initial acceptable system and $\gamma_1,.., \gamma_i$ represent the relative importance of the different objective functions. $\gamma_1,.., \gamma_i$ can be expressed as functions of the weighting factors form the relationship matrix in Figure 4. According to Krus et al. [9], γ can typically be chosen in the interval 2 to 5. Using

$$\gamma_i = 1 + \frac{w_i + 1}{2} \tag{7}$$

where w_i is the weighting of the i:th system characteristic. This definition for γ has proven to work well.

This formulation shows that the design can be divided into two parts. The first objective is to find an acceptable system solution and thereafter the optimization itself can be performed. A strength with this approach is that each objective function could be expressed in relative terms and thereby simplifies the composition of each objective function. This implies that the cost and weight factors $K_1,..,K_4$ and $C_1,..,C_4$ just have to be related to each other.

In order to handle the design constraints a constraint polynomial is multiplied with the original objective function, see equation (8), so that the objective function constitutes of both objectives and constraints. In this manner the constraints are separated from the original objectives.

$$F = \underbrace{\left[\left(\frac{f_1}{f_{10}}\right)^{\gamma_1} + \left(\frac{f_2}{f_{20}}\right)^{\gamma_2} + .. + \left(\frac{f_i}{f_{i0}}\right)^{\gamma_i}\right]}_{\text{objectives}} \cdot \underbrace{\left[\left(1 + \left|\frac{\Delta c_1}{c_{10}}\right|\right)^{\alpha_1} + \left(1 + \left|\frac{\Delta c_2}{c_{20}}\right|\right)^{\alpha_2} + \left(1 + \left|\frac{\Delta c_j}{c_{j0}}\right|\right)^{\alpha_j}\right]}_{\text{constraints}} \tag{8}$$

In equation (8), $\Delta c_1,..,\Delta c_j$ indicate a deviation from the desired value. $c_{10},..,c_{j0}$ is a tolerance value for that particular constraint, i.e. the energy difference of the accumulator before and after a load cycle should lie within a certain tolerance. The exponent α_i indicates the strength of the i:th constraint.

In contradiction to the objectives, which can directly be derived from the relationship matrix, not all relevant constraints on the system may be obvious right from the start. Naturally, the need for a constraint arises during the progress of the optimization procedure. Only in few cases, the first objective function will lead to a design satisfying all demands.

5 Modeling and optimization

In order to analyze the retardation movement of the system, a geometric model of the landing gear was implemented in the simulation package Pro Mechanica Motion, which is a Multi-Body-Simulation (MBS) environment. This model includes all mechanical parts and even landing gear drag is taken into account. The drag changes naturally both in size and point of attack during the retardation movement. In order to analyze the entire landing gear system, the hydraulic system was modeled as well using the HOPSAN simulation package. With the help of the Multi-Body-Simulation software the equations of motion of the landing gear were generated and then linked to the HOPSAN package. Both the mechanical and the hydraulic sub systems were simulated simultaneously.

Optimization of multi domain systems is in most cases based on a nonlinear system representation. Since the objective function value is given by simulation, derivatives of the objective function are not available. This makes non-gradient methods, such as the Complex method [1], or genetic algorithms as presented in for instance Goldberg, [4], well suited for this kind of problem. The optimization presented in this paper was carried out with help of the complex method.

In the complex method a complex consisting of several possible problem solutions (sets of system parameters) is manipulated. When all points in the complex have converged the optimal solution is found. Each set of parameters represents one point in the solution space. The starting points are randomly generated and it is checked that both implicit and explicit constrains are fulfilled. The main idea of this algorithm is to replace the worst point by a new and better point. The new point is calculated as the reflection of the worst point trough the centeroid of all the points in the complex.

6 Results of the optimization

The optimization process can be studied in figure 5 to 9, where the variation of some of the optimization parameters is shown.

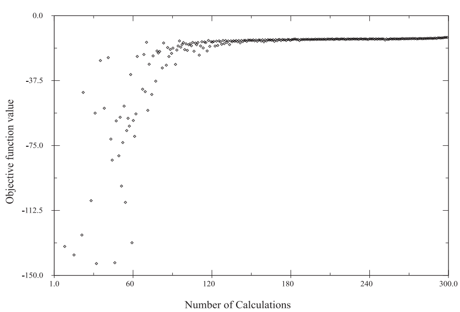


Figure 5. Objective function value.

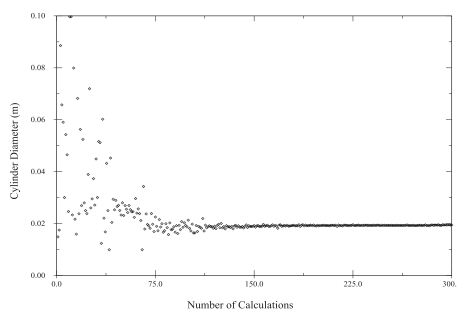


Figure 6. Cylinder diameter.

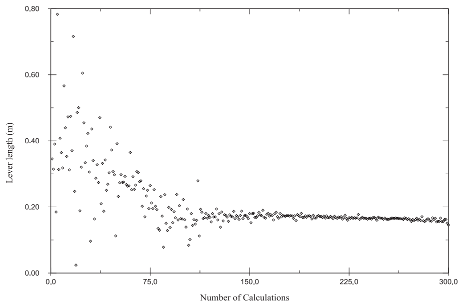


Figure 7. Lever length.

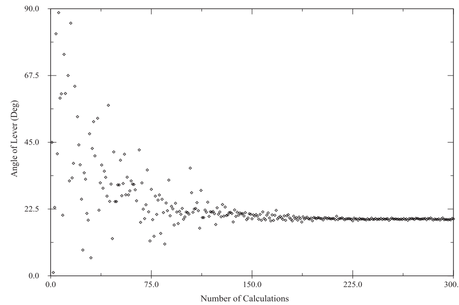


Figure 8. Lever angle.

300 simulations were needed in order to reach the optimal solution. However, a system with acceptable performance is reached after only 200 simulations, which takes about 15 minutes on a Pentium II 300Mhz computer.

The behavior of the system with optimal parameters is shown in figure 9 below.

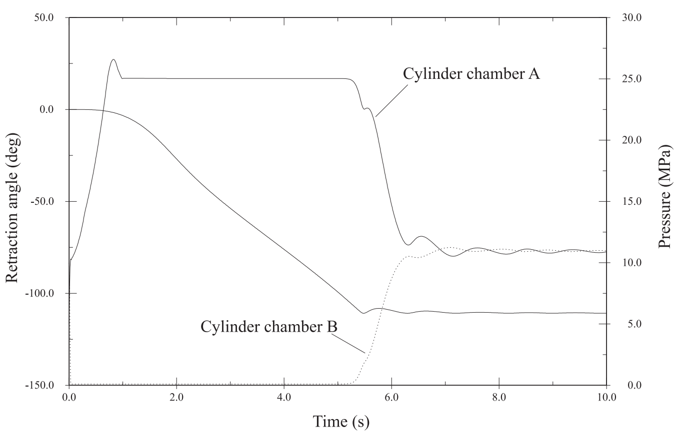


Figure 9. Landing gear retraction and pressure levels.

As can be seen from the figure, the landing gear retracts smoothly and it is fully retracted within five seconds.

7 Discussion and Conclusion

A structured way of looking at a design problem incorporating the Design Structure Matrix (DSM) and the House of Quality as supporting tools of the optimization problem has been presented in this paper. The approach presented here fosters good documentation during the entire design process and traceabillity is guaranteed. Furthermore, the objective function evolves from the DSM and the House of Quality matrices.

The value of an objective function can be regarded as the fingerprint of a design. Thereby the outcome of the optimization process, the final design, is via the House of Quality method directly related to the customer requirements.

The optimization in this paper was carried out on a landing gear of today's aircraft, where both mechanical structure and hydraulic actuation system have been modeled. The mechanical part has been modeled using Multi-Body-Simulation software and was then linked to the HOPSAN simulation package, where the hydraulic part was modeled. In this manner, both the mechanical structure and the hydraulic system have been modeled in each modeling environment rather than in one common simulation package. This can be seen as a step towards integration of modeling and simulation tools.

8 References

[1] Box M. J. (1965), A new method of constraint optimization and a comparison with other methods, Computer Journal 8:42-52.

[2] Cohen L. (1995), Quality function deployment: how to make QFD work for you, Addison-Wesley, Reading.

[3] Eppinger S. D., Whitney D. E., Smith R. P., and Gebala D. A. (1994), "A Model-Based Method for Organizing Tasks in Product Development", Research in Engineering Design, vol. 6, no. 1, pp. 1-13.

[4] Goldberg D. E. (1989), Genetic Algorithms in Search and Machine Learning, Addison Wesley, Reading.

[5] Hauser J., R., Clausing D. (1988), The House of Quality, Harvard Business Review, May-June.

[6] Hopsan (1991), Hopsan, a simulation package, user's guide, Technical Report LITH-IKP-R-704, Div. of Fluid and Mech. Eng. Systems, Dept. of Mech. Eng., Linköping University, Sweden.

[7] Jansson A. (1994), Fluid Power System Design A Simulation Approach, Dissertation, Linköping University, Linköping

[8] Krus P., Jansson A., Palmberg J.-O. (1993), Optimization based on Simulation for Design of Fluid Power Systems, Proceedings of IMECH International Conference on Aircraft Hydraulics and Systems, London, UK.

[9] Krus P., Palmberg J.-O., Löhr F., Backlund G. (1995), The Impact of Computational Performance on Optimization in Aircraft System Design, Birmingham

[10] Pahl G., Beitz W. (1996), Engineering Design – A Systematic Approach, Springer, London.

[11] Pohl J., Jansson A., Palmberg J.-O. (1997), Design of a boom suspension on a Wheel loader – A Simulation and Optimization Approach, Land Transport Hydraulics, ImechE, Bath 15 September 1997.

[12] Smith R. P., Eppinger S. D. (1997), Identifying Controlling Features of Engineering Design Iteration, Management Science, Vol. 43, No. 3, March 1997, pp. 276-293

[13] Sullivan L. P. (1986), Quality Function Deployment, Quality Progress, June.

[14] Steward D. V. (1981), The design Structure System: A Method for Managing the Design of Complex Systems, IEEE Transaction on Engineering Management, vol. EM-28, no.3 August 1981, pp. 71-74.

Paper III

Method for Integrated Systems Design
A Study of EHA Systems

Katarina Nilsson, Johan Andersson and Petter Krus

This paper has been published as:
NILSSON K., ANDERSSON J. AND KRUS P., ”Method for Integrated Systems Design – A Study of EHA Systems”, in *Proceedings of Recent Advances in Aerospace Hydraulics*, Toulouse, France, November 24-25,1998.

Method for Integrated Systems Design
A Study of EHA Systems

Katarina Nilsson[*], Johan Andersson[**] and Petter Krus[**]

[*]General Systems Research & Development
Saab AB
S-581 88 Linköping, Sweden

[**]Department of Mechanical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Abstract

In the development and design of general aircraft systems, the focus of today is to reach more efficient, integrated and robust system solutions in a shorter time and at a lower cost than has been in demand before. Since requirements are tougher, there is a need for powerful tools and structured methods. This is especially true when traditional system solutions are replaced with new ideas that involve integration of several engineering domains and a robust system performance is required. A structured method for systems development and design is described, that includes simulation and optimization as well as statistical methods for analysis of robustness with reference to system variation. This method has been used in a case study of different concepts of electrically powered actuation systems.

1 Introduction

A major task in the industrial design and development work of today is the demand for lower costs and tighter time schedules. There is also a demand for designs that are more robust and able to meet life cycle performance requirements. At the same time, requirements are pushed further than before and the integration of multiple domain aspects are becoming increasingly important. In this complex environment, the need for powerful tools and methods in the design effort is evident.

The future development of general aircraft systems is directed towards highly integrated and power efficient systems. The More Electric Aircraft (MEA) technology is one of the promising techniques for the future. One part of the MEA technology is replacement of the traditional hydraulic system with local electrically powered actuation systems for primary flight control. In order to take full advantage of the MEA technology, a change of design rules and standards is necessary, as is stated by Clough [4]. It is also important to accurately define the operational and environmental requirements and take into account the integration of systems in the total aircraft design.

In this paper, an integrated design method for multi-domain engineering systems is described. A platform for systems design is provided by a combination of development tools, where system solutions can be modeled and compared in detail and evaluated by simulation and optimization. As is described by Andersson et al. [1], a systematic approach for solving complex engineering design issues can be introduced by formulating them as optimization problems. By use of a structured way of setting up the problem, it is possible to find the coupling between different design parameters and the corresponding engineering domains. It is also possible to relate system parameters to system characteristics, which will facilitate formulation of the objective function used in the optimization. When optimization is used as a design tool, it is always essential to evaluate the quality of the result. With statistical methods, the influence of variation among system parameters can be introduced in the design procedure. When system variation is considered in the design process, the resulting system performance will become more robust. As is shown by Nilsson et al. [10], the result will be a trade-off between system performance and system robustness, that aims at an optimized system with a stable performance for the conditions that are concerned.

2 Actuator systems

In the traditional hydraulic system for primary flight control, power is generated by the engine and transmitted to the hydraulic system that serves the actuators. The actuators are controlled by electrical commands in a Fly-By-Wire (FBW) system. The foreseen future development in this area is to replace the traditional hydraulic power distribution with electrical wires. In such a Power-By-Wire (PBW) system, both the distribution of power and the actuator control is performed electrically. A step further in the future development of these systems is the combination of PBW for the distribution of power and Fly-By-Light (FBL) technology for the flight control system, where electrical wires are replaced by optical fiber data interfaces. An actuation system of this kind is described by Roach [11]. There are several anticipated advantages with a PBW system, like weight reduction, reduced fuel consumption and reduced life cycle cost, see for example Bildstein [2]. It will also contribute to improved reliability and supportability of the aircraft. Since the whole actuator system is assembled in one package, it can easily be removed and replaced. In the design of PBW systems, some areas that not cause problems in the traditional hydraulic system will have influence. As is described by Clough [4], some examples of such areas are electromagnetic interference, local heat rejection, requirements of large hydraulic flow capacities, residual actuator stiffness, tailoring of load-rate curves and high-peak horsepower demands.

The method described in this paper has been applied in the study of two basic configurations of PBW actuator systems, which are shown in Figure 1. One of these concepts is an EHA-FP (Electro-Hydrostatic Actuator-Fixed displacement Pump) system, where the actuator movement is controlled by a reversible variable speed DC motor and a pump with constant displacement. The control signal is in this case a high power voltage controlling the motor. The other concept is an EHA-VP (Electro-Hydrostatic Actuator-Variable displacement Pump) system, using a constant speed AC induction motor and a variable displacement pump with a separate motor-driven displacement control. The control signal is in this case the displacement of the pump, which is a low power signal. The concepts are described in more detail in [1].

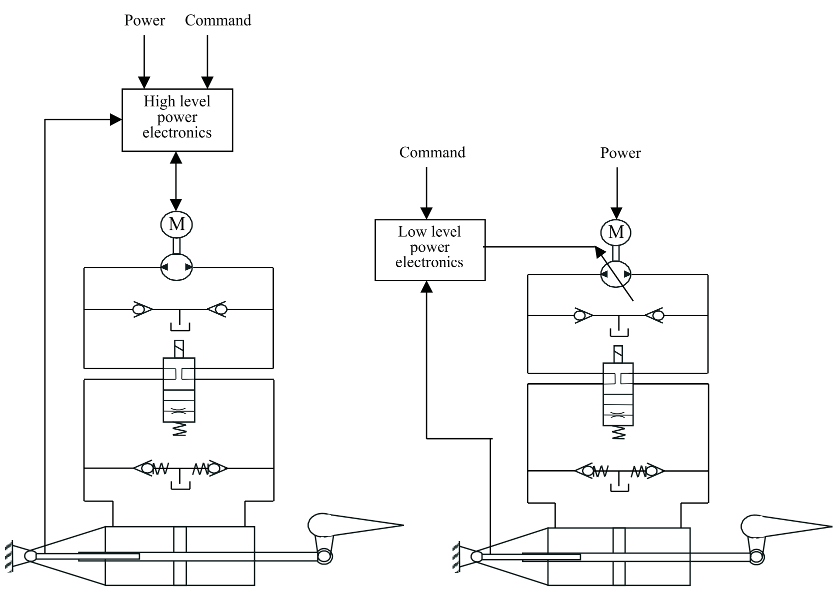


Figure 1. The studied concepts of electrical actuation.

3 Simulation and optimization

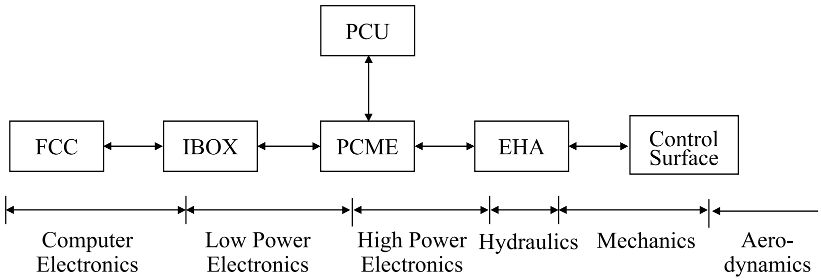
The two actuator systems described in Figure 1 have been modeled in the HOPSAN simulation package [6], which is a simulation software developed for fast simulation of hydraulic circuits. The simulation is carried out using distributed modeling, described in [8], and it is based on the transmission line modeling method presented by Johns et al. [7]. Transmission line modeling implies that each component can be regarded as a separate unit numerically isolated from the rest of the system.

Optimization of hydraulic systems will require a non-linear system representation and this can be accomplished in an iterative calculation procedure, where the system is modeled in a simulation tool that is coupled to an optimization facility. Optimization requires the system being simulated each time the objective function is evaluated. Since the value of the objective function is given only by simulation, derivatives of the objective function are not available without a large number of additional function evaluations. This implies that ordinary gradient optimization methods are not suited for this kind of problem. Non-gradient optimization methods, like the Complex method, [3] or genetic algorithms [5], are well suited to handle the problem at hand. These methods have a large probability of finding the global optimum by scanning large solution spaces.

The optimization results presented in this paper were calculated with the Complex method. The optimization search is then started from an initial population of points randomly spread in the solution space. This population forms a complex, where each point represents a whole set of solution parameters and a specific objective function value. The optimization search procedure involves moving the points with poor objective function value to an assumed better position until convergence is obtained.

4 The design problem

It is clear that the development of a PBW system is a complex engineering design problem, among other things because it will involve issues from several engineering disciplines. In order to reach an understanding of the actuation design problem, the architecture of a typical PBW system and its related engineering disciplines has been studied. The PBW architecture originates from the NASA System Research Aircraft, which is described by Navarro [9], see Figure 2. Since the EHA system and its surroundings are of special interest here, relations to areas such as mechanics and high power electronics can be anticipated, as is indicated in Figure 2.



FCC=Flight control computer
IBOX=Interface box
PCME=Power control and monitoring unit
PCU=Power conversion unit

Figure 2. Typical architecture of a PBW system with the corresponding engineering domains.

In setting up the optimization problem, several choices can be made in the problem formulation. These choices are essential for the quality of the result. A structured way of dealing with this will facilitate problem formulation and lead to increased quality, see also Andersson et al. [1].

In order to attain a systematic approach in the formulation of the optimization problem, a Design Structure Matrix (DSM) is used. The DSM, shown in Table 1, makes it possible to graphically display the information flow in the actuation design process, thus providing a tool to find the coupling between different design parameters and thereby also the different engineering domains.

Table 1. Design Structure Matrix for design of an actuation system.

		A	B	C	D	E	F	G	H	I	J	K	L	M
Airframe	A	●			X			X	X					X
Control surface	B	X	●											X
Hinge lever	C		X	●										
Piston area	D		X	X	●		X	X	X					
Piston stroke	E		X	X		●		X	X					
System pressure	F		X	X	X		●							
Pump size	G		X	X	X	X		●	X					
Motor size	H				X		X	X	●		X			
Feed back gain	I				X	X	X	X	X	●				
Power Control and Monitoring	J								X	X	●		X	X
Power Converter Unit	K								X		X	●	X	X
Interface Box	L												●	X
Flight Control Computer	M	X	X	X	X	X	X	X	X		X	X	X	●

Each row in the matrix represents one parameter to be considered in the optimization. The inserted crosses on each row indicate a strong influence of parameters in the corresponding columns. The dots just separate the upper and the lower triangles of the matrix. The crosses above the diagonal indicate feed-backs, which contribute to the iterative nature of the engineering design. The DSM is used to determine which parameters to take into consideration in the optimization procedure.

The actuation part of the matrix is described in more detail, since it is of main focus in this study. The matrix shows, for example, that the airframe has a strong influence on the control surface. The control surface will influence the piston area and the stroke and so forth. But there is also a feed-back to the airframe from the piston area, pump and motor size, i.e. the actual size of the actuator package is likely to influence the airframe. The squares, shown in the matrix, marks the coupled blocks, where the big square in the middle is the actuator where mechanical, hydraulic and electrical engineering is combined.

The method of Quality Function Deployment (QFD) can be used to clarify system requirements and their relation to system parameters. The relationship matrix from the first House of Quality has proven to be helpful in relating system parameters to system characteristics, as is shown in Table 2.

Table 2. Relationship Matrix for an EHA system.

			System parameters											
		Weighting	Control surface area	Hinge lever	Piston Area	Piston stroke	System pressure	Pump size	Motor size	Feed back gain	Power Control and Monitoring	Power Converter Unit	Interface Box	Flight Control Computer
System Characteristics	Weight	5		1	5	5	5	5	5		3	3	1	1
	Energy consumption	3	3		3	3	1	5	5		5	5		
	Cost	1			3	3	3	5	5		5	5	1	1
	Envelope dimensions	3		1	5	5	3	5	5		5	5	1	1
	Controlability	5	5	3	5	3	5	1	3	5	1	3		5
	Maintainability	1			1		1	5	5			3	3	3
	Reliability	1				1		3	3				5	5
	Temperature	D												

The matrix is established by listing requirements on the vertical axis and system parameters on the horizontal axis. The system parameters are then related to the system characteristics by a number in the matrix. A higher number indicates a stronger relation. The relative importance of the system characteristics is given by the value of a weight factor or simply by a 'D', a demand that the system has to fulfil but is difficult to express in terms of system parameters.

The value of the number given in the matrix is of minor importance, it merely indicates that a parameter does influence a system characteristic. If one of the parameters that is related to a system characteristic is subject to changes during the design process, it might be considered to be included in the objective function.

In a traditional (central) hydraulic system, the cost characteristic would refer to component cost during the life cycle period of the aircraft and include purchase as well as maintenance cost. Maintainability could be added either as a separate characteristic or be included in the cost. Since the local actuation system in this case can be regarded as a LRU (Line Replaceable Unit), maintainability is depending on where the system is situated on the aircraft but also on accessibility to the individual components within the system.

The optimization problem was formulated as a study of the system characteristics weight, energy consumption, cost and controllability. The envelope dimension was considered as being indirectly given by the weight of the system. The formulation of the objective function is described in detail in [1].

5 Method for robust design

When a model of a system is used in the design process, the natural variation of the system variables can not be directly included. A simulation will give the same result at repeated calculations when steady-state conditions can be obtained and similar input is given. In reality, system parameters will always include variation to some extent and this fact is likely to influence the result. There are several ways to investigate the influence of system variation, leading to results of different value. In the design process, it is essential to include the actual grade of variation that is anticipated for each variable in order to evaluate the way it will affect system performance. It is then important not only to locate the influencing variables and their interaction, but also to evaluate the result of their specific variation and be able to include this information in the design process. The method described here includes these aspects and has proven to contribute to increased knowledge about the studied system, thus forming a tool for decisions in the design process, see Nilsson et al. [10].

Su and Renaud [12], have shown that variation can be considered in system design by including sensitivity derivatives in the optimization procedure. This approach would require the use of a gradient based (derivative) optimization method or require a considerable effort in additional function evaluation. An alternative is to study the influence of variation as a post-optimization analysis. The ideas of Taguchi and Wu [13], show that direct experimentation techniques can be used to improve the quality of a product by a parameter design that reduces the system sensitivity to variation. The strategy to use experimental design in the planning of simulation analyses is also described by Wild and Pignatiello [14], where the approach of crossing the design arrays is used (see the single response approach described in the following text).

The method for robust design that is described here includes optimization and simulation performed in two steps, where parameter variation is included in each step by use of full-scale or reduced factorial experiments, see Figure 3. Among system parameters, it is possible to distinguish between controllable and non-controllable factors. The controllable factors can be given a value around which variation may occur, while the non-controllable factors and their variation can not be directly affected. Experimental design is introduced here as two orthogonal arrays, describing variation among controllable and non-controllable factors respectively. In a system design study, it is of great interest to find an optimal and robust solution where variation is considered. An advantage in the use of experimental design is the possibility to set proper values to the controllable factors in order to minimize or eliminate the influence of non-controllable or disturbing factors, and thereby obtain a more robust design.

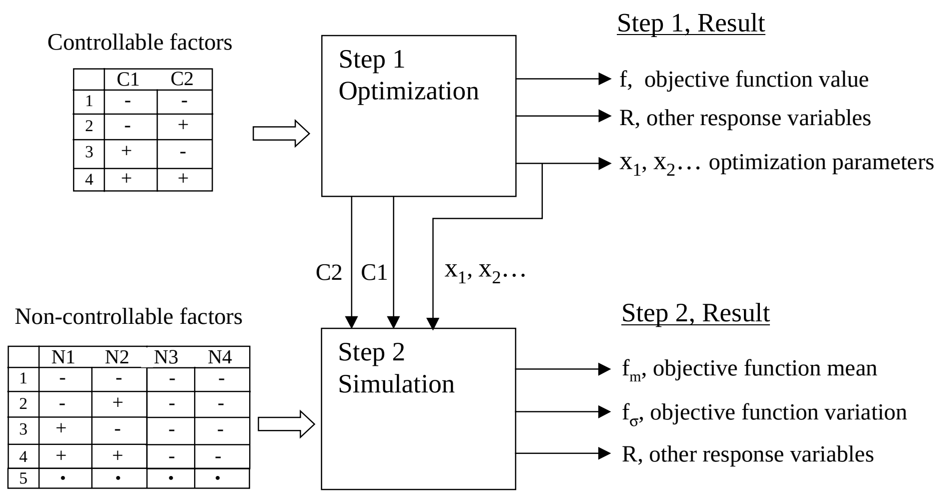


Figure 3. The calculation procedure.

The purpose of using orthogonal arrays is to facilitate a statistical analysis, where the effects of different factors can be estimated independently of each other. It is recommended that a preliminary investigation is performed to check that the chosen factor levels represent the space around the optimum. A two-level array is usually preferred because many factors can be included and a minimum of trials is required. A center point is added to check the function linearity of the parameter space. Since a non-linear objective function behavior can be expected close to optimum, an analysis including second order terms should be chosen for the further analysis. The effect of factor variation is obtained by use of regression analysis with the method of least squares that gives a prediction model for each of the response variables. In this study, the response variables are the objective function value in Step 1 and the mean and the standard deviation of the objective function in Step 2. Any system parameter and any number of parameters can be chosen as response variables in the investigation.

The controllable factors are given levels that lie within the region of operability and interest based on knowledge about the system. The levels for the non-controllable factors can be determined as plus/minus the standard deviation corresponding to an assumed real-life distribution around a mean value.

Table 3. A two-level orthogonal array for two factors with a center point - Step 1 Optimization.

No.	Controllable factors	
	C1 Cylinder area	C2 Tank pressure
1	-	-
2	+	-
3	-	+
4	+	+
5	0	0

Table 4. A two-level orthogonal array for five factors with a center point - Step 2 Simulation.

No.	Non-controllable factors				
	N1 Pump friction	N2 Cyl. friction	N3 Pump leakage	N4 Cyl. leakage	N5 Load seq.
1	-	-	-	-	+
2	+	-	-	-	-
3	-	+	-	-	-
4	+	+	-	-	+
5	-	-	+	-	-
6	+	-	+	-	+
7	-	+	+	-	+
8	+	+	+	-	-
9	-	-	-	+	-
10	+	-	-	+	+
11	-	+	-	+	+
12	+	+	-	+	-
13	-	-	+	+	+
14	+	-	+	+	-
15	-	+	+	+	-
16	+	+	+	+	+
17	0	0	0	0	0

The experimental design can follow two routes of procedure. A dual response approach can be considered where averages and sample standard deviations of the response variables are calculated. These two measures are then analyzed to identify the controllable factors that affect the mean of the data and the non-controllable factors that affect the spread.

The other approach involves crossed arrays and can be considered as one experiment or a single response case, where all factors will affect the mean of the data. The fundamental difference between the two approaches is that what appears as effects on the spread of the data in the first case will appear as interaction effects between the controllable and the non-controllable factors in the single response approach. This interaction effect can not be found when common sensitivity analyses are used and it implies that the controllable factors can be given values that will reduce or eliminate the impact of non-controllable or disturbing factors.

The two experimental design arrays that have been used in the study of the two actuator systems are shown in the Tables 3 and 4. The plus/minus signs indicate an upper and a lower factor level, whereas the zero indicates the center point with a mean level value.

The first part of the calculation procedure involves optimization of each row of Table 3. The result of these calculations will be the objective function value and the value of the optimization parameters, which also constitute input to the next calculation step. Each optimization row is then followed by the whole set of simulations shown in Table 4, where the non-controllable factors are varied. This variation will affect the system performance to some extent, which may be noticed in the mean value and the spread of the objective function value of Step 2 in accordance with the dual response approach. Interaction effects between the controllable and the non-controllable factors may be found by use of the single response approach, where the two calculation arrays are crossed.

6 Results

The result is presented in the Tables 5 and 6 as the objective function value of Step 1 and its mean value and standard deviation of Step 2. The regression analysis gives information about the impact of the controllable factors and the effect of disturbances caused by the non-controllable factors.

The studied system is optimized prior to performing the calculation procedure described in Figure 3. The controllable factors are chosen from the optimization variables and varied around their optimal values. The first calculation step gives a measure of how narrow the optimum is in both directions, without the influence of the non-controllable factors. Thus, the best result of Step 1 is given by the center point, row 5 in the Tables 5 and 6. This also indicates that an optimum really has been found. In the second step, the shape of the objective function around the optimum can be studied under the influence of disturbance caused by the non-controllable factors. A result of the statistical evaluation is obtained as polynomials, showing how the response variables are affected by variation of the controllable and the non- controllable factors. It is also possible to determine how the levels of the controllable factors can be chosen to minimize the influence of the non-controllable factors. System performance or optimality is then traded for system robustness with respect to parameters that are hard to control.

The statistical evaluation shows that in the EHA-FP system, the tank pressure has no significant influence neither on the level of the objective function nor the standard deviation. The cylinder area, on the other hand, has a significant influence on the result and affects both level and deviation. A large cylinder area reduces the effect of the disturbing factors. Thus, one can trade system performance for increased system robustness by choosing a larger cylinder area. These results are also valid in the EHA-VP system, with the reservation that the same clear influence of the cylinder area on the level of the mean objective function value can not be found.

In the EHA-FP system, the disturbing factors renders the objective function steeper in one direction from the optimum. This is not as obvious in the EHA-VP system.

The tank pressure may have an influence on system performance in the EHA-VP case, something that can not be statistically confirmed without more observations.

By examining the simulation runs of Step 2, it can be seen that the load sequence and the pump leakage has the largest influence on the level as well as the deviation of the objective function value in both systems.

Table 5. Calculation result, EHA-FP.

No.	Result Step 1	Result Step 2	
	f Obj.f.value	f_m Mean	f_σ Std.dev.
1	7.87	8.06	0.58
2	7.85	7.92	0.40
3	7.99	8.13	0.62
4	7.81	7.88	0.38
5	7.57	7.65	0.42

Table 6. Calculation results, EHA-VP.

No.	Result Step 1	Result Step 2	
	f Obj.f.value	f_m Mean	f_σ Std.dev.
1	5.45	5.48	0.28
2	5.13	5.14	0.16
3	4.93	4.96	0.28
4	5.11	5.11	0.16
5	4.86	4.87	0.20

When crossing the two calculation matrices, the influence of both controllable and non-controllable factors can be estimated. In this study, the obvious relation between a large cylinder area and a reduced sensitivity to load disturbances has been found.

The reason for lower levels of the objective function values in the case of the EHA-VP system is that it has a faster response. For a more detailed view of the objective function formulation and the behavior of the two studies systems, see [1].

Efforts are being made to also include the thermal aspects in the EHA models. These aspects will probably affect the results referred to here, but have not been included in this study.

7 Discussion and conclusion

In this paper, a method for the design of complex and heterogeneous systems is presented. In order to find reliable and robust solutions, it is often necessary to combine several tools. The employment of simulation and optimization techniques may seem straight forward, but the complexity of the design problem requires a large number of aspects to be considered. The method involves the use of a simulation tool combined with a non-gradient optimization method in an iterative calculation procedure together with a statistical analysis based on experimental design.

The method presented here also includes means to facilitate the understanding of the design process and the formulation of the objective function. A Design Structure Matrix is employed in order to visualize the information flows in the design process, thus revealing the coupling between different system parameters and thereby also different engineering domains. This knowledge supports the designer in choosing optimization parameters. With the help of the House of Quality method it can be determined how specific system parameters affect the system characteristics, which will facilitate the formulation of the objective function. Once the design problem is formulated as an optimization problem, it is time to find a strategy to perform the necessary calculations. At this point, the aspect of robustness can be regarded by using experimental design to formulate a calculation procedure where variation among system parameters can be introduced in the optimization study. It is possible to distinguish between controllable and non-controllable (disturbing) factors and to find their impact on system performance as well as possibilities to reduce the effect of the disturbing factors.

The strategy of calculation involves a first step, where the location of the optimum and the size of the studied solution area is confirmed, and a second step, where the system is examined around its optimum with consideration taken to the influence of disturbing factors. The basic idea is to evaluate the behavior of the response variable around the optimum with a polynomial including the controllable and non- controllable factors. The coefficients of the polynomial are estimated using statistical methods. By examining the coefficient values, the influence of the corresponding factors can be found.

Two concepts of EHA systems have been studied. By applying the method presented here, it has been shown that the optimal solution is influenced by the disturbing factors and that system performance can be traded for increased system robustness. The presented method also gave valuable insight into which of the disturbing factors had the greatest influence on system performance and how the controllable factors could be tuned in order to minimize the impact of the disturbing factors.

In the development of aircraft general systems, possibilities to reach the overall goals and to find effective system solutions depend on the ability to integrate and co-develop the different subsystems. One step in this direction is to find connections between the systems and different surrounding engineering domains and include these aspects in the system development work, as is described here. A further step would be modeling and optimization on a higher aircraft level, where the subsystems can be included. When several systems are integrated in the modeling effort, the demand for effective tools and methods will be even more pronounced. Increased ability to handle the complexity of heterogeneous systems will require tools for optimization and robust design.

8 References

[1] Andersson J., Krus P. and Nilsson K., Optimization as a support for selection and design of aircraft actuation systems, Seventh AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, Missouri, September 2-4, 1998.

[2] Bildstein M., Application of electrohydrostatic actuators (EHA) for future aircraft primary flight controls, 1. Internationales Fluidtechnisches Kolloqium, Aachen, Germany, March 17-18, 1998.

[3] Box M.J., A new method for constraint optimization and a comparison with other methods, Computer Journal, 8:42-52, 1965.

[4] Clough B., Lessons Learned: Flight Control System Electrical Actuator Development, SAE Aerospace Power Systems Conference, Williamsburg, Virginia, USA, April 21-23, 1998.

[5] Goldberg D.E., Genetic Algorithms in Search, Optimization & Machine Learning, Addison-Wesley, Reading, Massachusetts, 1989.

[6] Hopsan, Hopsan, A Simulation Package, User’s Guide, Technical Report LITH-IKP-R-704, Div. of Fluid and Mech. Eng. Systems, Dept. of Mech. Eng., Linköping University, 1991.

[7] Johns P.B., Brien M.A., Use of the transmission line modelling (t.l.m.) method to solve lumped networks, The Radio Electron and Engineer, 50:59-70, 1980.

[8] Krus P., Jansson A., Palmberg J.-O., Weddfeldt K., Distributed simulation of hydromechanical systems, Third Bath International Fluid Power Workshop, Bath, UK,1990.

[9] Navarro R., Performance of an Electro-Hydrostatic Actuator on the F-18 Systems Research Aircraft, NASA/TM-97-206224, 1997.

[10] Nilsson K., Åkerlund O. and Hynén A., Using optimization and statistical methods to analyse the robustness of integrated hardware/software systems, The Fifth Scandinavian International Conference on Fluid Power, Linköping, May 28-30, 1997.

[11] Roach J.M., FLASH Electrohydrostatic Actuation Modeling, Analysis, and Test Results, Aerospace Power Systems Conference, Williamsburg, Virginia, USA, April 9-11, 1997.

[12] Su J., Renaud J.E., Automatic Differentiation in Robust Optimization, 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Bellevue, WA, September 4-6, 1996.

[13] Taguchi G., Wu Y., Introduction to Off-Line Quality Control, Central Japan Quality Control Association, Nagoya, Japan, 1980.

[14] Wild R.H., Pignatiello Jr. J.J., An experimental design strategy for designing robust systems using discrete- event simulation, Simulation, 57:6, 358-368, 1991.

Paper IV

Pareto Optimization using the Struggle Genetic Crowding Algorithm

Johan Andersson and David Wallace

Pareto Optimization using the Struggle Genetic Crowding Algorithm

Johan Andersson[*] and David Wallace[**]

[*]Department of Mechanical Engineering
Linköping University
SE-581 83 Linköping, Sweden

[**]Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02142, USA

Abstract

Many real-world engineering design problems involve the simultaneous optimization of several conflicting objectives. In this paper, a method combining the struggle genetic crowding algorithm with Pareto-based population ranking is proposed to elicit tradeoff frontiers. The new method has been tested on a variety of published problems, reliably locating both discontinuous Pareto frontiers as well as multiple Pareto frontiers in multi-modal search spaces. Other published multi-objective genetic algorithms are less robust in locating both global and local Pareto frontiers in a single optimization. For example, in a multi-modal test problem a previously published non-dominated sorting GA (NSGA) located the global Pareto frontier in 41% of optimizations, while the proposed method located both global and local frontiers in all test runs. Additionally, the algorithm requires little problem specific tuning of parameters.

1 Introduction

Design engineers are often interested in identifying a Pareto optimal set of alternatives when exploring a design space. Vilfredo Pareto [14] defined Pareto-optimality as a set where every element is a problem solution for which no other solutions can be better in all design attributes. A solution in a Pareto-optimal set cannot be deemed superior to the others in the set without including preference information to rank competing attributes.

This paper develops a Pareto optimization method for use in multi-objective, multi-modal design spaces, where simulations from many designers create a mix of continuous numerical simulations and discrete catalog choices. Thus, a Pareto optimization technique suitable for mixed continuous and discrete problems is needed. Genetic algorithms are well suited for such applications—they have been shown to be effective in optimizing mixed variable problems [13, 16] and multi-modal search spaces [8].

This paper first defines a general multi-objective optimization problem and reviews related work on multi-objective genetic algorithms. Then, a new method is proposed and validated using a suite of test problems gathered from the literature. The initial algorithm's performance is analyzed, and then the algorithm is improved and re-evaluated.

1.1 Background

1.2 General multi-objective optimization problem

A general multi-objective design problem is expressed by equations (1) and (2).

$$\min \mathbf{F}(\mathbf{x}) = \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})\right)^T \qquad s.t.\ \mathbf{x} \in S \tag{1}$$

$$\mathbf{x} = \left(x_1, x_2, \ldots, x_n\right)^T \tag{2}$$

where $f_1(x), f_2(x), \ldots, f_k(x)$ are the k objectives functions, $\left(x_1, x_2, \ldots, x_n\right)$ are the n optimization parameters, and $S \in R^n$ is the solution or *parameter* space. Obtainable objective vectors, $\left\{\mathbf{F}(\mathbf{x}) \middle| x \in S\right\}$ are denoted by Y, so $\mathbf{F} : S \mapsto Y$, S is mapped by $\mathbf{F}$ onto Y.

$Y \in R^k$ is usually referred to as the *attribute* space, where ∂Y is the boundary of Y. For a general design problem, $\mathbf{F}$ is non-linear and multi-modal, and S might be defined by non-linear constraints containing both continuous and discrete member variables.

$f_1^*, f_2^*, \ldots, f_k^*$ will be used to denote the individual minima of each respective objective function, and the utopian solution is defined as $\mathbf{F}^* = \left(f_1^*, f_2^*, \ldots, f_k^*\right)^T$. As $\mathbf{F}^*$ simultaneously minimizes all objectives, it is an ideal solution that is rarely feasible.

The Pareto subset of ∂Y is of particular interest to the rational decision-maker. The Pareto set is defined by equation (3). Considering a minimization problem and two solution vectors **x**, **y**$\in S$. **x** is said to dominate **y**, denoted $\mathbf{x} \succ \mathbf{y}$, if:

$$\forall i \in \{1,2,...,k\}: f_i(\mathbf{x}) \leq f_i(\mathbf{y}) \quad \textit{and} \quad \exists j \in \{1,2,...,k\}: f_j(\mathbf{x}) < f_j(\mathbf{y}) \tag{3}$$

The space in R^k formed by the objective vectors of Pareto optimal solutions is known as the Pareto optimal frontier, $\mathcal{P}$. It is clear that any final design solution should preferably be a member of the Pareto optimal set. Pareto optimal solutions are also known as non-dominated or efficient solutions. Figure 1 provides a visualization of this nomenclature.

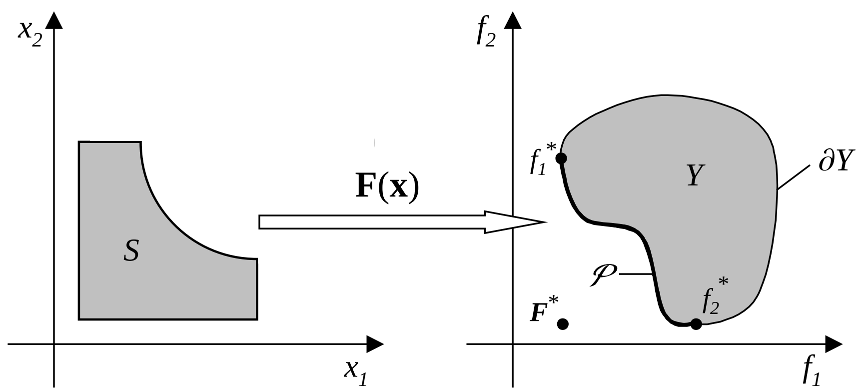


Figure 1. Parameter/solution and attribute space nomenclature for a two dimensional problem with two objectives.

1.3 Multi-objective genetic algorithms

Genetic algorithms are modeled after mechanisms of natural selection. Each optimization parameter (x_n) is encoded by a gene using an appropriate representation, such as a real number or a string of bits. The corresponding genes for all parameters $x_1,..x_n$ form a chromosome capable of describing an individual design solution. A set of chromosomes representing several individual design solutions comprise a population where the most fit are selected to reproduce. Mating is performed using crossover to combine genes from different parents to produce children. The children are inserted into the population and the procedure starts over again, thus creating an artificial Darwinian environment. For a general introduction to genetic algorithms, see work by Goldberg [7].

Additionally, there are many different types of multi-objective genetic algorithms. For a review of genetic algorithms applied to multi-objective optimization readers are referred to work by Fonseca and Fleming [3]. Literature surveys and comparative studies on multi-objective genetic algorithms are also provided by several other authors [1, 10, 18, 20]. Multi-objective genetic algorithms are typically divided into non-Pareto and Pareto based approaches.

1.3.1 Non-Pareto based multi-objective approaches

The first multi-objective genetic algorithm was VEGA (Vector Evaluating Genetic Algorithm) developed by Schaffer [15]. VEGA uses the selection mechanism of the GA to produce non-dominated individuals. Each individual objective is designated as the selection metric for a portion of the population. However, it is reported that the method tends to crowd results at extremes of the solution space, often yielding poor coverage of the Pareto frontier.

Fourman [6] presents a genetic algorithm using binary tournaments, randomly choosing one objective to decide each tournament. Kurasawe [12] further developed this scheme by allowing the objective selection to be random, fixed by the user, or to evolve with the optimization process. He also added crowding techniques, dominance, and diploidy to maintain diversity in the population.

All of these Non-Pareto techniques tend to converge to a subset of the Pareto-optimal frontier, leaving a large part of the Pareto set unexplored. Preferably, one wants to maintain diversity so that the entire Pareto frontier is elicited. Additionally, maintaining diversity will tend to improve robustness in multi-objective problems by ensuring that there is a genetic variety for mating mechanisms to operate upon [8, 9].

1.3.2 Pareto based multi-objective approaches

Goldberg [7] introduced non-dominated sorting to rank a search population according to Pareto optimality. First, non-dominated individuals in the population are identified. They are given the rank 1 and are removed from the population. Then the non-dominated individuals in the reduced population are identified, given the rank 2, and then they are also removed from the population. This procedure of identifying non-dominated sets of individuals is repeated until the whole population has been ranked, as depicted in Figure 2. Goldberg also discusses using niching methods and speciation to promote diversity so that the entire Pareto frontier is covered.

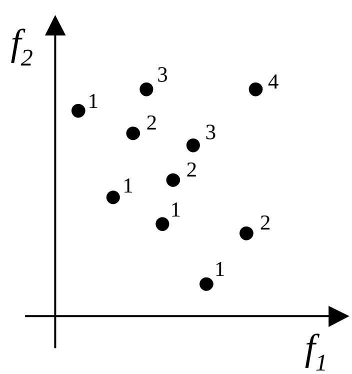


Figure 2. Population ranking based upon non-dominated sorting.

The non-dominated sorting GA (NSGA) of Srinivas and Deb [17] implements Goldberg's thoughts about the application of niching methods. In NSGA, non-dominated individuals in the population are identified, given a high initial individual score and are then removed from the population. These individuals are considered to be of the same rank. The score is then reduced using sharing techniques between individuals with the same ranking. Thereafter, the non-dominated individuals in the remaining population are identified and scored lower than the lowest one of the previously ranked individuals. Sharing is then applied to this second set of non-dominated individuals and the procedure continues until the whole population is ranked.

Sharing is performed in the parameter space rather than in the attribute space. This means that the score of an individual is reduced according to how many individuals there are with similar parameters, regardless of how different or similar they might be based on objective attributes.

In the multi-objective GA(MOGA) presented by Foseca and Fleming [4, 5] each individual is ranked according to their degree of dominance. The more population members that dominate an individual, the higher ranking the individual is given. An individual's ranking equals the number of individuals that it is dominated by plus one (see Figure 3). Individuals on the Pareto front have a rank of 1 as they are non-dominated. The rankings are then scaled to score individuals in the population. In MOGA both sharing and mating restrictions are employed in order to maintain population diversity. Fonseca and Fleming also include preference information and goal levels to reduce the Pareto solution set to those that simultaneously meet certain attribute values.

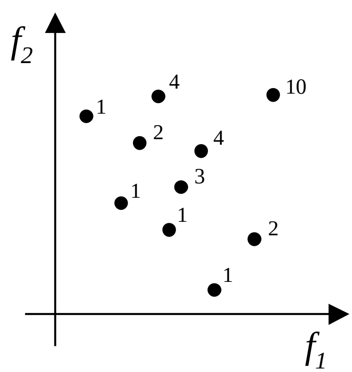


Figure 3. Population ranking according to Fonseca and Fleming.

The niched Pareto GA (NPGA) by Horn et al. [11] is Pareto-based but does not use ranking methods. Rather, Pareto domination tournaments are used to select individuals for the next generation. For binary tournaments, a subset of the population is used as a basis to assess the dominance of the two contestants. If one of the contestants is dominated by a member in the subset but the other is not, the non-dominated one is selected to survive. If both or neither are dominated, selection is based on the niche count of similar individuals in the attribute space. An individual with a low niche count is preferred to an individual with a high count to help maintain population diversity.

Zitzler and Thiele [20] developed a multi-objective genetic algorithm called the strengthen Pareto evolutionary algorithm (SPEA). SPEA uses two populations, *P* and

P'. Throughout the process copies of all non-dominated individuals are stored in P'. Each individual is given a fitness value, f_i, based on Pareto dominance. The fitness of the members of *P'* is calculated as a function of how many individuals in *P* they dominate (3).

$$f_i = \frac{number\ of\ individuals\ dominated\ by\ i}{size(P)+1} \qquad (4)$$

The individuals in *P* are assigned their fitness according to the sum of the fitness values for each individual in *P'* that dominate them plus one (see Figure 4). Lower scores are better and ensure that the individual spawns a larger number of offspring in the next generation. Selection is performed using binary tournaments from both populations until the mating pool is filled. In this algorithm, fitness assignment has a built-in sharing mechanism. The fitness formulation ensures that non-dominated individuals always get the best fitness values and that fitness reflects the crowdedness of the surroundings.

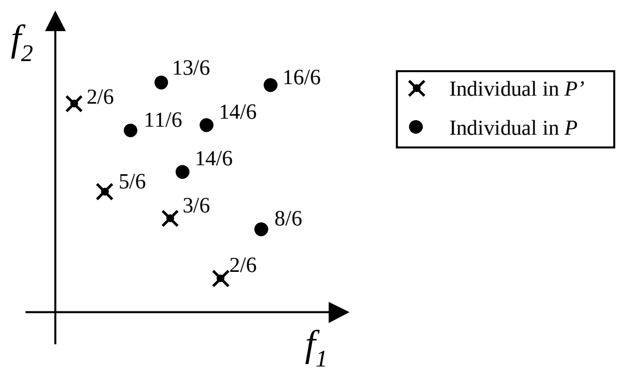


Figure 4. Population ranking according to Zitzler and Thiele.

Although there is a substantial body of research on Pareto multi-objective genetic algorithms, there are still important issues that current methods address with only partial success. The methods typically require extensive genetic algorithm parameter tuning on a problem-by-problem basis in order for the algorithm to perform well. However, in a real-world problem there is little knowledge about the shape of attribute space, which makes it difficult to assess problem specific parameters. Additionally, existing methods do not handle consistently the location of multiple Parteo frontiers in multi-modal problem spaces. The work in this paper attempts to develop a reliable algorithm that distributes solutions evenly across Pareto frontiers in a variety of multi-modal problems without problem specific tuning.

2 The multi-objective struggle genetic algorithm

The multi-objective struggle genetic algorithm (MOSGA) developed in this work combines the struggle crowding genetic algorithm presented by Grueninger and Wallace [8] with Pareto-based ranking as devised by Fonseca and Fleming [4].

In the struggle algorithm, a variation of restricted tournament selection, two parents are chosen from the population, and crossover/mutation are performed to create a child. The child replaces the most similar individual in the entire population, but only if it has a better fitness. This replacement strategy counteracts genetic drift that can spoil population diversity. The struggle genetic algorithm has been demonstrated to perform well in multi-modal function landscapes where it successfully identifies and maintains multiple peaks. The struggle genetic algorithm is defined by the following steps.

Step 1: Initialize the population.

Step 2: Select individuals uniformly from population.

Step 3: Perform crossover and mutation.

Step 4: Find the most similar individual in parameter space, and replace it if the new individual has a higher objective score.

Step 5: Perform steps 2-5 according to the population size.

Step 6: If the stop criterion is not met go to step 2 and start a new generation.

As there is no single objective function to determine the fitness of the different individuals in a Pareto optimization, the ranking scheme presented by Fonseca and Fleming is employed, and the “degree of dominance” in attribute space is used to rank the population. Each individual is given a rank based on the number of individuals in the population that are preferred to it, i.e. for each individual the algorithm loops through the whole population counting the number of preferred individuals. “Preferred to” is implemented in a strict Pareto sense, but one could also combine Pareto optimality with the satisfaction of objective goal levels.

The principle of the MOSGA algorithm is outlined below.

Step 1: Initialize the population.

Step 2: Select individuals uniformly from population.

Step 3: Perform crossover and mutation to create a child.

Step 4: Calculate the rank of the new child, and a new ranking of the population that considers the presence of the child.

Step 5: Find the most similar individual in *attribute* space, and replace it with the new child if the child’s ranking is better.

Step 6: Update the ranking of the population if the child has been inserted.

Step 7: Perform steps 2-6 according to the population size.

Step 8: If the stop criterion is not met go to step 2 and start a new generation.

3 Test Problems

To investigate the performance of the new algorithm benchmark test problems from the literature were employed. Unless otherwise noted, the variables are real coded, BLX crossover is used to produce offspring, and the likeness of two individuals is measured using the Euclidean distance in attribute space. The result from a BLX crossover between two real numbers A and B is randomly selected from a uniform distribution centered at the mean of A and B, with a width corresponding to twice the difference between A and B.

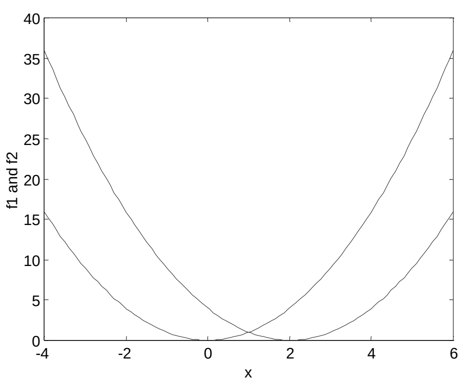
3.1 Single variable test problem

The first test problem is the simple f2 function from Schaffer [15]. This is a function in one variable.

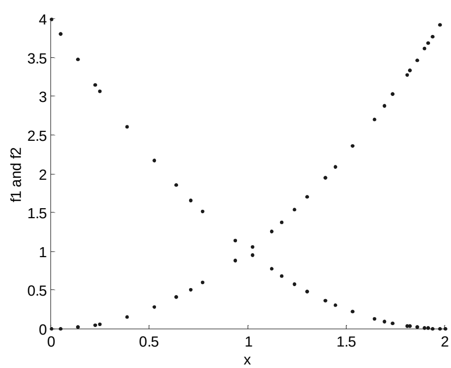
$$\text{Minimize } f_1(x) = x^2 \tag{5}$$

$$\text{Minimize } f_2(x) = (x-2)^2 \tag{6}$$

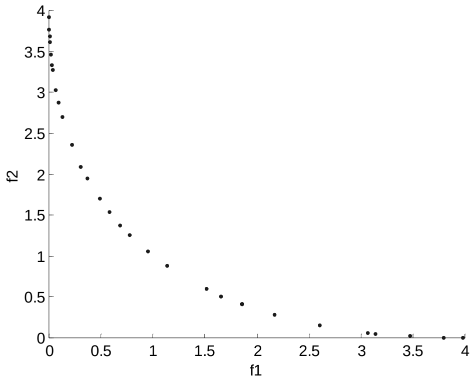
Figure 5(a) plots f_1 and f_2 over the interval [-4,6], whereas (b) and (c) show the optimization results after 100 generations with a population of 30 individuals.



(a)



(b)



(c)

Figure 5. (a) shows Schaffers f_1 and f_2 as function of x. In (b) the optimization results is shown. (c) shows optimization result as a f_1-f_2 plot, showing how the population spreads evenly on the Pareto-optimal front.

Like many of the published Pareto techniques, MOSGA maintains diversity to provide solutions evenly distributed over the frontier. VEGA, however, tends to converge to the extremes of the individual objectives as the number of generation increases.

3.2 Multi-variable test problems

A second set of test problems from Deb [2] was also explored. Deb developed a set of problems to highlight difficulties that multi-objective genetic algorithms may encounter. For visualization reasons, the focus is on two-dimensional problems defined generally by equations (7) and (8).

$$\text{minimize} \quad f_1(x_1, x_2) = x_1 \tag{7}$$

$$\text{minimize} \quad f_2(x_1, x_2) = \frac{g(x_2)}{x_1}, \quad g(x_2) > 0, \quad x_1 > 0 \tag{8}$$

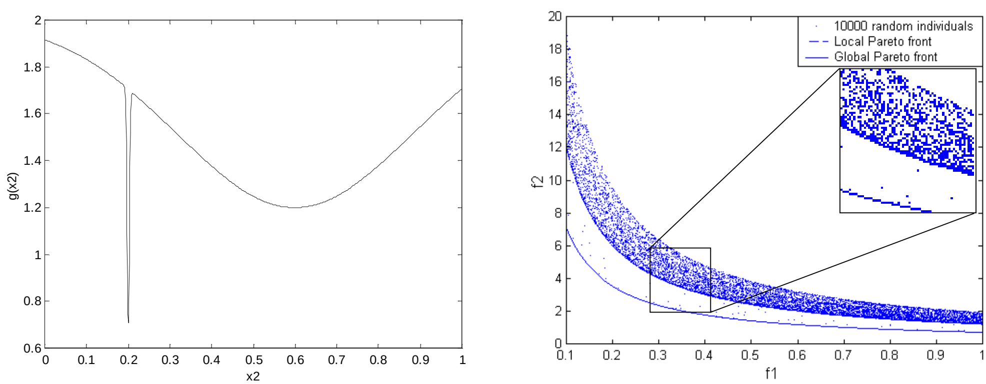
This formulation is used to construct both a multi-modal multi-objective problem and a problem with a discrete Pareto-optimal front.

3.3 Multi-modal problem formulation

If the function g is multi-modal, the corresponding multi-objective problem will have global and local Pareto-optimal frontiers. The multi-modal g function is defined in equation (9).

$$g(x_2) = 2 - \exp\left\{-\left(\frac{x_2 - 0.2}{0.004}\right)^2\right\} - 0.8\exp\left\{-\left(\frac{x_2 - 0.6}{0.4}\right)^2\right\} \tag{9}$$

Figure 6(a) shows the g function for $0 \leq x_2 \leq 1$ with the global optima located at $x_2=0.2$ and a local optima at $x_2=0.6$. Figure 6(b) shows a plot of f_1 and f_2 in the attribute space with the global and local Pareto optimal solutions. 10000 randomly chosen solutions are generated and plotted to illustrate that the problem is biased—the solution density is higher towards the local Pareto-optimal front.

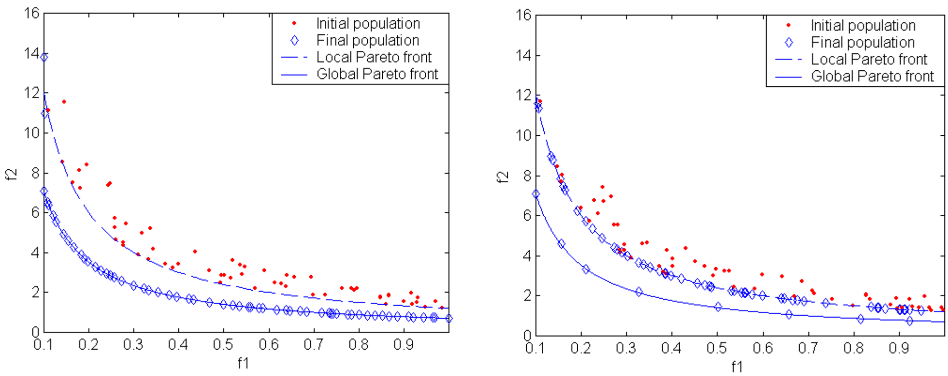


(a) (b)

Figure 6. (a): The function $g(x_2)$ with the global optima situated at $x_2=0.2$ and the local optima at $x_2=0.6$. In (b) a random set of 10.000 solutions is shown on a f_1-f_2 plot. Notice the low solution density at the global Pareto-optimal front

Deb reported that the NSGA was trapped in the local Pareto front in 59 out of 100 runs. The proposed MOSGA converges to local Pareto solution in only 7 of 100 runs, finding the preferred global Pareto optimal front in 86 out of 100 runs. In 7 runs it converges to both frontiers. Figure 7 shows optimization results after 200 generations for the cases when the global Pareto optimal front is located, and when both are found in the same optimization. The algorithm spreads the population evenly on the Pareto-optimal front.

This initial MOSGA appears to be more robust in locating the global frontier, but performance still could be better. After evaluating performance in a discontinuous discrete frontier in the following section, the MOSGA algorithm will be modified so that it more reliably locates both Pareto solutions during a single optimization.



(a) (b)

Figure 7. Optimization results using a population of 60 individuals after 200 generations. In (a) the population has converged on the global Pareto-optimal front. In (b) the population has converged on both the local and global fronts.

3.4 Discrete Pareto optimal frontier

Deb also constructed a test to produce a discontinuous Pareto optimal frontier, as defined in equations (10) through (13).

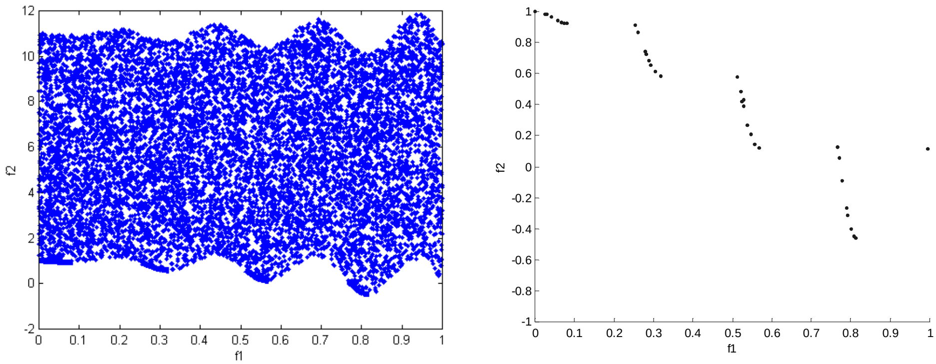
$$\text{Minimize} \quad f_1(x_1, x_2) = x_1 \tag{10}$$

$$\text{Minimize} \quad f_2(x_1, x_2) = g(x_2) \cdot h(f_1, g) \tag{11}$$

$$g(x_2) = 1 + 10x_2 \tag{12}$$

$$h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^2 - \frac{f_1}{g}\sin(2\pi q f_1) \tag{13}$$

The parameter q determines the number of discrete Pareto optimal fronts. We have chosen to use $q=4$. Figure 8(a) shows 10.000 random solutions for x_1 and x_2 in the interval [0,1]. Figure 8(b) shows the optimization result of the MOSGA algorithm after 200 generations.



(a) (b)

Figure 8. (a) 10.000 random solutions shown in a f_1-f_2 plot, with the Pareto-optimal front indicate in the lower part. (b) the population of 60 individuals at generation 200. The algorithm has found all four regions and spreads evenly among them.

Compared to NSGA, MOSGA spreads the population more evenly on the different Pareto frontiers, and covers the whole Pareto set with a considerably smaller population, resulting in much fewer function evaluations. Deb uses a population size of 200 and runs the algorithm for 300 generations, resulting in five times as many function evaluations as used for MOSGA. As a real-world problem often involve CAE tools to evaluate each new solution, there might be a heavy computation burden associated with each function evaluation.

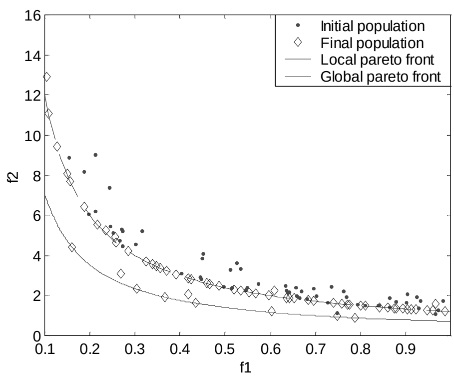
Similar tests were completed on the remainder of Deb's test suites, including a deceptive binary encoded problem using hamming distance as the similarity measure, with similar results. These results suggest the MOSGA algorithm can reliably find global frontiers in the multi-objective test problems without problem specific tuning of the GA, other than population size. In general, the goal was to reduce computation by choosing small populations that, when spread across the frontiers, were sufficient to define their contours. In general, the MOSGA required significantly smaller populations and function evaluations to locate frontiers.

4 Improvements to the MOSGA algorithm

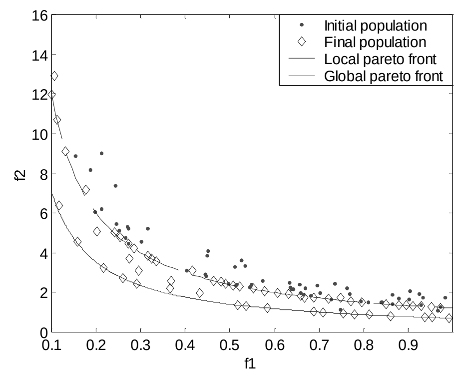
Although more robust in finding global frontiers, the MOSGA, like other approaches, showed poor performance in locating multiple frontiers in a single optimization. For the multi-model test problem presented, it stably converged to both the local and global frontier in only 7% of the optimizations. In this section, properties of the algorithm are explored and improvements are made to reliably converging to multiple frontiers. The multi-modal test problem is used as the basis for the exploration.

4.1 Robust location of multiple frontiers

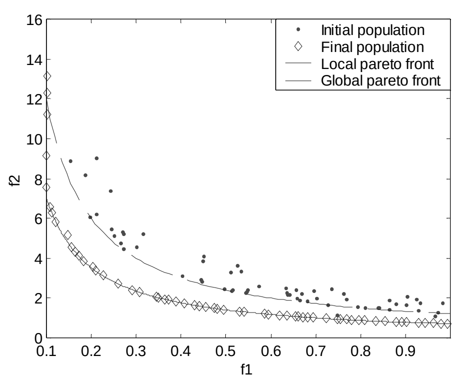
The MOSGA algorithm identifies points on both the local and global Pareto frontiers in early stages of the optimization process. A typical population after 40 generations is shown in Figure 9(a). As the optimization progresses, individuals move towards the global front and the individuals on the local front are eliminated, Figure 9(b) and (c). Ideally, the algorithm should be capable of identifying both the global and local frontiers in a stable manner.



(a)



(b)

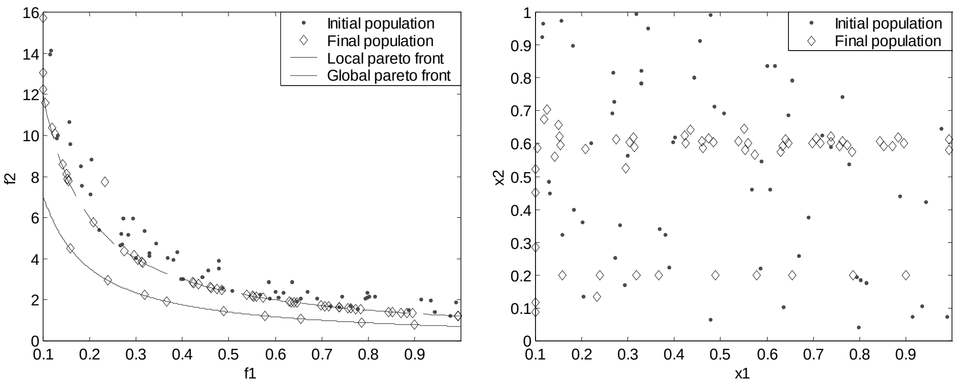


(c)

Figure 9. The population after (a) 40 generations, (b) 80 generations and (c) after 200 generations.

The MOSGA algorithm progresses by permitting new offspring to replace the most similar individual in attribute space if their rank is better. Similarity is determined with a function that calculates the distance between two individuals. For example, Euclidian distance is used for comparing attribute values. While this mechanism effectively preserves diversity on the global Pareto optimal front, it can still eliminate local Pareto frontiers. A new individual in the surrounding of the global Pareto optimal set may be closest to an individual at the local frontier in attribute space, and thus would replace it even though it might be quite different in parameter space.

To validate this belief, a set of tests was conducted on the original multi-modal *g* function in equation (9) using a parameter based similarity measure. The resulting distribution of population in the attribute space supports our observations, but there is poor design convergence in parameter space (see Figure 10).

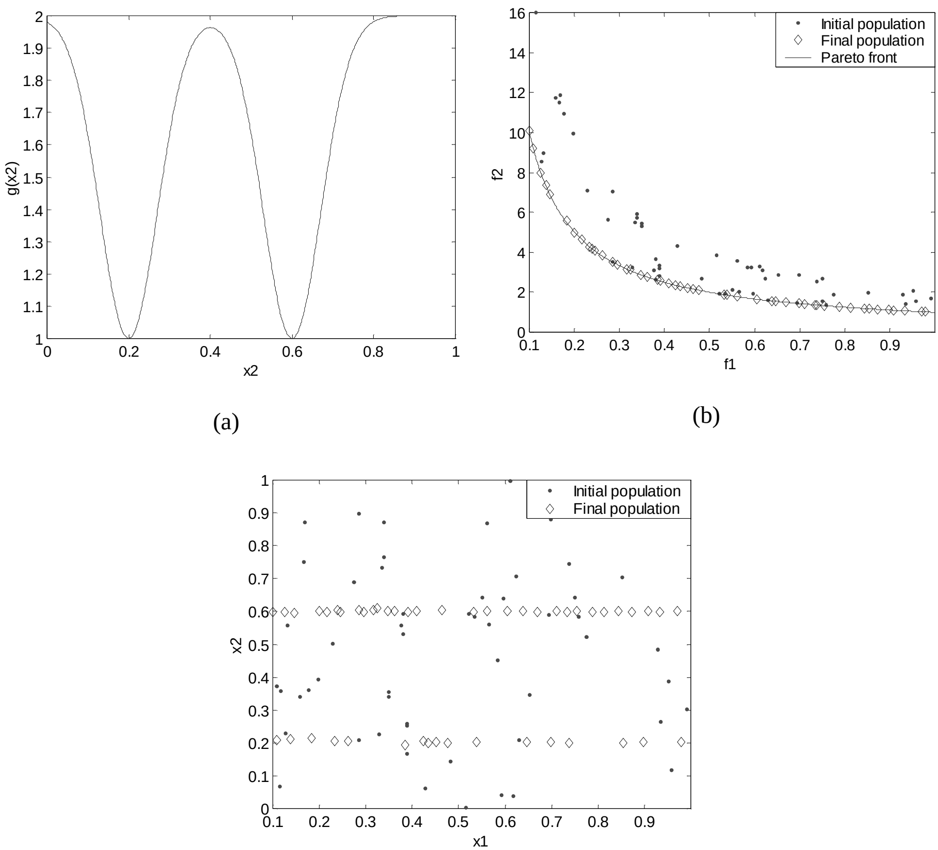


(a) (b)

Figure 10. The population after 400 generations using comparison in the parameter space, (a) shows the result in attribute space whereas (b) shows the result in parameter space.

In this case, many non-Pareto solutions survive because they are not dominated by the individuals in their close surroundings. The poor performance in parameter space is partly due to the fact that the local Pareto front is very broad for this problem. Individuals not exactly on the local front will not be dominated by the individuals in their neighborhood and therefore continue to survive.

In order to test this hypothesis, a *g*-function with two equal global optima corresponding to Figure 11(a) In this problem comparison in parameter space is preferred, see Figure 11(b) and (c). Note that both global fronts have the same values in the attribute space so a comparison in attribute space cannot distinguish between the optima. When employing a distance function in the attribute space only one front is identified in a given optimization run.

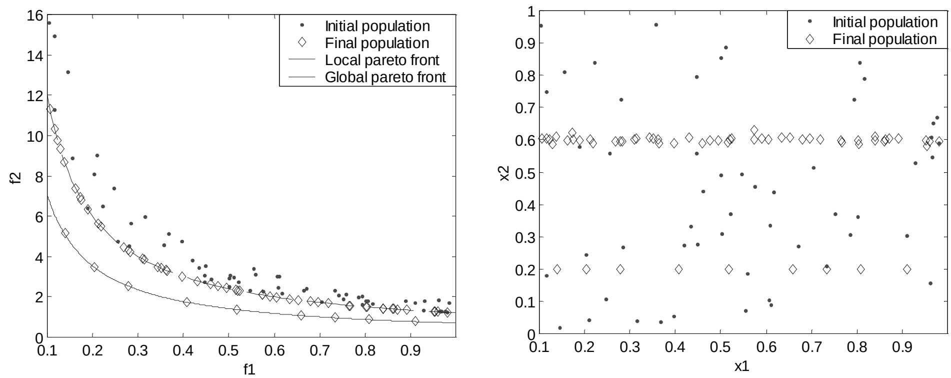


(a) (b)

(c)

Figure 11. (a) a modified g-function with two equal optima. (b) shows the final population in attribute space and (c) in parameter space. Notice that both Pareto fronts have the same values.

In order to take advantage of the strengths of both a parameter based and attribute based similarity measures, they were combined into one equally weighted distance measure. This mixed distance function was implemented and tests on the original *g*-function in equation (9) are shown in Figure 12. The new mixed distance measure maintains stable populations on both Pareto frontiers.



(a) (b)

Figure 12. The final population after 200 generation with the mixed distance measure using the original g function. (a) shows the result in attribute space and (b) in parameter space.

The mixed distance measure yields a better convergence in fewer generations than the parameter based distance measure. Both the parameter based and the mixed similarity measure managed to locate both the global and the local frontiers in all 100 test runs. The population is evenly spread on the frontiers with in average 9 individuals at the narrow global frontier and the remaining 51 at the broader local frontier.

Thus, we conclude that the original an attribute based similarity measure yielded rapid and precise convergence but is only capable of identifying one Pareto frontier in each run. On the other hand, with a parameter based similarity measure the algorithm was able to locate multiple Pareto frontiers but the convergence was slower and not as precise. By combining both distance measures to one mixed distance measure the advantages of both approaches are realized.

The refined MOSGA algorithm is thus defined below.

Step 1: Initialize the population.

Step 2: Select individuals uniformly from the population.

Step 3: Perform crossover and mutation to create a child.

Step 4: Calculate the rank of the new child, and a new ranking of the population that considers the presence of the child.

Step 5: Find the most similar individual, equally combining differences in both the parameter space and attribute space. Replace this individual if the new child's rank is better.

Step 6: Update the ranking of the population if the child has been inserted.

Step 7: Perform steps 2-6 according to the population size.

Step 8: If the stop criterion is not met go to step 2 and start a new generation.

4.2 Reduction of optimization time

In order to speed up the optimization process, alterations to the replacement strategy might be considered. For example, rather than replacing the most similar individual with the newly generated better ranking individual, one might also replace the worst individual if it is dominated by *ndom* individuals. This modified replacement strategy could drastically increase the convergence speed of the algorithm. However, robustness might be compromised by the heuristic if *ndom* is a low number—population diversity will be reduced rapidly. In the original method, poor individuals have to be gradually moved towards more promising areas. With the modified strategy, very poor individuals are instantly replaced with better performing individuals to speed convergence.

This strategy has been applied to the discrete problem in equation (10)-(13) with *ndom=3*, and the results are shown in Figure 13. Note that the whole population has converged on the different frontiers after only 40 generations, and compare with the population after 40 generations using the normal replacement strategy. With the normal replacement strategy 200 generations were required for the population to converge (see Figure 8).

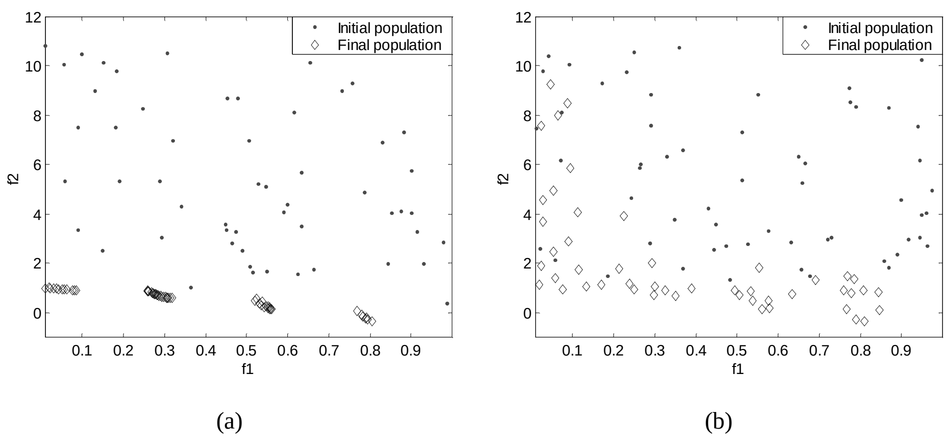


Figure 13. (a) shows the optimization results after 40 generations using the modified replacement strategy, whereas (b) shows the results after 40 generations using the original replacement strategy.

For specific problems, where the MOSGA algorithm shows slow convergence this might be a useful technique. However, the MOSGA algorithm with combined parameter and attribute similarity measure works reliably without problem specific adjustment—problem specific fine-tuning with *ndom* might reduce computation time but may also result in premature convergence if used inappropriately.

5 Conclusions

Many real-world engineering problems are characterized by the presence of multiple conflicting objectives. A Pareto optimization algorithm that can handle mixed continuous and discrete problems in multi-modal solution spaces is needed. In this paper, a new Pareto multiple objective struggle genetic algorithm (MOSGA) was developed for this purpose.

The proposed method combines Pareto based ranking with the struggle genetic crowding algorithm. Each individual is ranked based on how many members of the population that is preferred to it in a Pareto-optimal sense. Then parents are selected uniformly from the population and crossover and mutation are employed in order to produce a child. The child is inserted into the population and replaces the individual most similar to itself, but only if it has a better ranking. The similarity between two individuals is measured with a function considering differences in both parameter and attribute space.

Observations based on a suite of test problems shows that MOSGA identifies and maintains multiple Pareto optimal frontiers in multi-modal landscapes more reliably than previously published GA-based methods. Another strength of the algorithm is its ability to address a wide variety of problems without tuning the genetic optimization parameters and to operate successfully with relatively small populations. This is an important issue for real world applications where there may be little knowledge about the shape of the attribute space.

6 Acknowledgments

The software for this work used the GAlib genetic algorithm package, written by Matthew Wall [19] at the Massachusetts Institute of Technology. The authors wish to thank researchers in the MIT CADlab for their assistance.

7 References

[1] C. Coello, *An empirical study of evolutionary techniques for multiobjective optimization in engineering design*, Dissertation, Department of Computer Science, Tulane University, 1996.

[2] K. Deb, “Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems,” *Evolutionary Computation*, vol. 7, pp. 205-230, 1999.

[3] C. Fonseca and P. Fleming, “An overview of evolutionary algorithms in multiobjective optimization,” *Evolutionary Computation*, vol. 3, pp. 1-18, 1995.

[4] C. M. Fonseca and P. J. Fleming, “Multiobjective optimization and multiple constraint handling with evolutionary algorithms - Part I: a unified formulation,” *IEEE Transactions on Systems, Man, & Cybernetics Part A: Systems & Humans*, vol. 28, pp. 26-37, 1998.

[5] C. M. Fonseca and P. J. Fleming, "Multiobjective genetic algorithms made easy: Selection, sharing and mating restriction," presented at 1st IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems, Sheffield, England, 1995.

[6] M. P. Fourman, "Compaction of symbolic layout using genetic algorithms," presented at 1st Int. Conference on Genetic Algorithms, Pittsburgh, 1985.

[7] D. Goldberg, *Genetic Algorithms in Search and Machine Learning*, Reading, Addison Wesley, 1989.

[8] T. Grueninger and D. Wallace, "Multi-modal optimization using genetic algorithms", Technical Report 96.02, CADlab, Massachusetts Institute of Technology, Cambridge, 1996.

[9] G. Harik, "Finding multimodal solutions using restricted tournament selection," presented at Sixth International Conference on Genetic ALgorithms, 1995.

[10] J. Horn, "Multicriterion decision making," in *Handbook of evolutionary computation*, T. Bäck, D. Fogel, and Z. Michalewicz, Eds., IOP Publishing Ltd and Oxford University Press, 1997.

[11] J. Horn and N. Nafpliotis, "Multiobjective Optimization Using the Niched Pareto Genetic Algorithm", Technical Report 93005, Illinois Genetic Algorithm Laboratory, Dept. of General Engineering, University of Illinois at Urbana-Champaign, Urbana, USA, 1993.

[12] F. Kurasawe, "A variant of evolution strategies for vector optimization," in *Parallel Problem Solving from Nature*, *Lecture Notes in Computer Science 496*, Berlin, Springer Verlag, pp. 193-197, 1991.

[13] F. Pahng, N. Senin, and D. Wallace, "Distributed object-based modeling and evaluation of design problems," *Computer-aided Design*, vol. 30, pp. 411-423, 1998.

[14] V. Pareto, *Cours d'Économie Politique*, Lausanne, Rouge, 1896.

[15] J. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," presented at 1st Int. Conf. on Genetic Algorithms, Pittsburgh, 1985.

[16] N. Senin, D. R. Wallace, and N. Borland, "Mixed continuous and discrete catalog-based design modeling and optimization," presented at 1999 CIRP International Design Seminar, University of Twente, Enschede, The Netherlands, 1999.

[17] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, pp. 221-248, 1995.

[18] H. Tamaki, H. Kita, and S. Kobayashi, "Multi-objective optimization by genetic algorithms: a review," presented at 1996 IEEE International Conference on Evolutionary Computation, ICEC'96, Nagoya, Japan, 1996.

[19] M. Wall, "Matthew's GAlib: A C++ library of genetic algorithm components," http://www.lancet.mit.edu/ga/, 1996.

[20] E. Zitzler and L. Thiele, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," *IEEE Transaction on evolutionary computation*, vol. 3, pp. 257-271, 1999.

Paper V

Multiobjective Optimization of Hydraulic Actuation Systems

Johan Andersson, Petter Krus and David Wallace

This paper has been published as:
ANDERSSON J., KRUS P., Wallace D., “Multiobjective Optimization of Hydraulic Actuation Systems”, in *Proceedings of ASME Design Automation Conference*, Baltimore, USA, September 11-13, 2000.

Multiobjective Optimization of Hydraulic Actuation Systems

Johan Andersson[*], Petter Krus[*] and David Wallace[**]

[*]Department of Mechanical Engineering
Linköping University
SE-581 83 Linköping, Sweden

[**]Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02142, USA

Abstract

Many real world engineering problems are characterized by the presence of several conflicting design objectives. In this paper, a new multi-objective genetic algorithm is employed to optimize two different concepts of hydraulic actuation systems. The different concepts have been modeled in a simulation environment to which the optimization strategy has been coupled.

The outcome from the proposed optimization strategy is a set of Pareto optimal solutions elucidating the tradeoffs between competing objectives. By comparing Pareto frontiers for competing concepts, valuable insights about the properties of the different concepts can be gained. Depending on how the decision-maker values the different objectives, different design solutions are more appropriate. This is exemplified in the hydraulic actuation systems, where the acceptance of a larger control error results in a design with low energy consumption.

1 Introduction

Many real-world engineering design problems involve simultaneous optimization of several conflicting objectives. In many cases, multiple objective problems are aggregated into one single overall objective function. Optimization is then conducted with one optimal design as the result. This paper presents a method where the solution space is searched for a set of Pareto optimal solutions, from which the decision-maker may choose the final design. Vilfredo Pareto, [16] defined Pareto-optimality as a set where every element is a problem solution for which no other solutions can be better in all design attributes. A solution in the Pareto optimal set cannot be deemed superior to the others in the set without including preference information to rank competing attributes. For the two-dimensional case, the Pareto front is a curve that clearly elucidates the tradeoff between the objectives. By comparing such Pareto frontiers for different competing concepts, valuable support for concept selection could be gained.

The paper starts with presenting a nomenclature for the multi-objective design problem. Thereafter, existing multi-objective genetic algorithms are reviewed. Then the proposed method is presented and tested on a benchmark problem. Finally, a design problem consisting of two different hydraulic actuation concepts is studied with the help of simulation models and the proposed optimization strategy. It is shown that the new method performs well and that Pareto optimization could be a powerful tool for concept selection.

2 Background

2.1 The multi-objective design problem

A general multi-objective design problem could be expressed by equations (1) and (2).

$$\min \mathbf{F}(\mathbf{x}) = \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})\right)^T \qquad (1)$$

$$s.t.\ \mathbf{x} \in S$$

$$\mathbf{x} = \left(x_1, x_2, \ldots, x_n\right)^T \qquad (2)$$

where $f_1(x), f_2(x), \ldots, f_k(x)$ are the k objectives functions, $(x_1, x_2, \ldots, x_n)$ are the n optimization parameters, and $S \in R^n$ is the solution or parameter space. Obtainable objective vectors, $\{\mathbf{F}(\mathbf{x}) | x \in S\}$ are denoted by *Y*, so $\mathbf{F} : S \mapsto Y$, *S* is mapped by **F** onto *Y*. $Y \in R^k$ is usually referred to as the attribute space, where ∂Y is the boundary of *Y*.

For a general design problem, **F** is non-linear and multi-modal, and S might be defined by non-linear constraints containing both continuous and discrete member variables.

The Pareto subset of ∂Y is of particular interest to the rational decision-maker. The Pareto set is defined by equation (3). Considering a minimization problem and two solution vectors **x**, **y**$\in S$. **x** is said to dominate **y**, denoted $\mathbf{x} \succ \mathbf{y}$, if:

$$\forall i \in \{1,2,...,k\}: f_i(\mathbf{x}) \leq f_i(\mathbf{y}) \quad and \quad \exists j \in \{1,2,...,k\}: f_j(\mathbf{x}) < f_j(\mathbf{y}) \tag{3}$$

The space in R^k formed by the objective vectors of Pareto optimal solutions is known as the Pareto optimal front, $\mathcal{P}$. It is clear that any final design solution should preferably be a member of the Pareto optimal set. Pareto optimal solutions are also known as non-dominated or efficient solutions.

2.2 Related work

Genetic algorithms are modeled after mechanisms of natural selection. Each optimization parameter (x_n) is encoded by a gene using an appropriate representation, such as a real number or a string of bits. The corresponding genes for all parameters $x_1,..x_n$ form a chromosome capable of describing an individual design solution. A set of chromosomes representing several individual design solutions comprise a population where the most fit are selected to reproduce. Mating is performed using crossover to combine genes from different parents to produce children. The children are inserted into the population and the procedure starts over again, thus creating an artificial Darwinian environment. For a general introduction to genetic algorithms, see Goldberg [9].

Additionally, there are many different types of multi-objective genetic algorithms. For a review of genetic algorithms applied to multi-objective optimization, readers are referred to work by Fonseca and Fleming [7]. Literature surveys and comparative studies on multi-objective genetic algorithms are also provided by several other authors, see for example [2], [13], [19] and [21].

The first multi-objective genetic algorithm was VEGA (Vector Evaluating Genetic Algorithm) developed by Schaffer [17]. VEGA uses the selection mechanism of the GA to produce non-dominated individuals. Each individual objective is designated as the selection metric for a portion of the population.

Fourman [8] presents a genetic algorithm using binary tournaments, randomly choosing one objective to decide each tournament. Kurasawe [15] further developed this scheme by allowing the objective selection to be random, fixed by the user, or to evolve with the optimization process.

These early techniques tend to converge to a subset of the Pareto-optimal frontier, leaving a large part of the Pareto set unexplored. Preferably, one wants to maintain diversity so that the entire Pareto frontier is elicited. Additionally, maintaining diversity will tend to improve robustness in multi-objective problems by ensuring that there is a genetic variety for mating mechanisms to operate upon [10], [11].

Goldberg [9] introduced non-dominated sorting to rank a search population according to Pareto optimality. First, non-dominated individuals in the population are identified. They are given the rank 1 and are removed from the population. Then the non-dominated individuals in the reduced population are identified, given the rank 2, and then they are also removed from the population. This procedure of identifying non-dominated sets of individuals is repeated until the whole population has been ranked, see Figure 1.

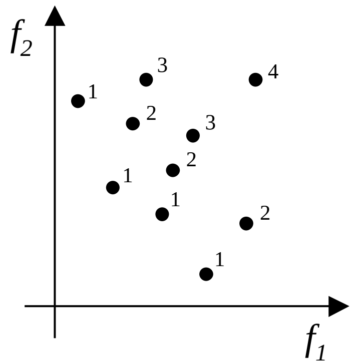


Figure 1. Population ranking based upon non-dominated sorting.

The non-dominated sorting GA (NSGA) of Srinivas and Deb [18] implements Goldberg's algorithm together with niching techniques to maintain diversity in the population.

In the multi-objective GA (MOGA) presented by Foseca and Fleming [6] and [7] each individual is ranked according to their degree of dominance. The more population members that dominate an individual, the higher ranking the individual is given. An individual's ranking equals the number of individuals that it is dominated by plus one, see Figure 2. Individuals on the Pareto front have a rank of 1 as they are non-dominated. The rankings are then scaled to score individuals in the population. In MOGA both sharing and mating restrictions are employed in order to maintain population diversity. Fonseca and Fleming also include preference information and goal levels to reduce the Pareto set to those that simultaneously meet certain attribute values.

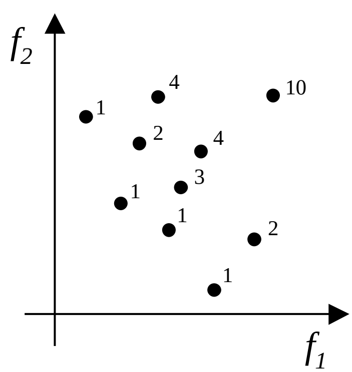


Figure 2. Population ranking according to Fonseca and Fleming.

The niched Pareto GA (NPGA) by Horn and Nafpliotis [14] does not use ranking methods. Rather, Pareto domination tournaments are used to select individuals for the next generation. For binary tournaments, a subset of the population is used as a basis to assess the dominance of the two contestants. If one of the contestants is dominated by a member in the subset but the other is not, the non-dominated one is selected to survive. If both or neither are dominated, selection is based on the niche count of similar individuals in the attribute space. An individual with a low niche count is preferred to an individual with a high count to help maintain population diversity.

Zitzler and Thiele [21] developed a multi-objective genetic algorithm called the strengthen Pareto evolutionary algorithm (SPEA). SPEA uses two populations, P and P'. Throughout the process copies of all non-dominated individuals are stored in P'. Each individual is given a fitness value based on Pareto dominance. The fitness of the members of P' is calculated as a function of how many individuals in P they dominate. The individuals in P are assigned their fitness according to the sum of the fitness values for each individual in P' that dominates them. Selection is performed using binary tournaments from both populations until the mating pool is filled. In this algorithm, fitness assignment has a built-in sharing mechanism. The fitness formulation ensures that non-dominated individuals always get the best fitness values and that the fitness reflects the crowdedness of the surroundings.

Donne et al. [4] used a multi-objective parallel genetic algorithm to optimize different hydraulic circuits. Their method employs a number of subpopulations and migration between populations in order to maintain diversity. Within each subpopulation the Pareto optimal individuals are identified and given a high ranking. All Pareto optimal solutions found are stored in an array, which is updated each generation in order to include new Pareto optimal individuals.

Although there is a substantial body of research on Pareto multi-objective genetic algorithms, there are still important issues that current methods address with only partial success. The methods typically require extensive genetic algorithm parameter tuning on a problem-by-problem basis in order for the algorithm to perform well. However, in a real-world problem there is little knowledge about the shape of attribute space, which makes it difficult to assess problem specific parameters. Additionally, existing methods do not handle consistently the location of multiple Pareto frontiers in multi-modal problem spaces. The method presented in this paper is capable of identifying multiple frontiers without any problem specific parameter tuning.

3 The proposed method

The multi-objective struggle genetic algorithm (MOSGA) [1] combines the struggle crowding genetic algorithm presented by Grueniger and Wallace in [10] with Pareto-based ranking as devised by Fonseca and Fleming in [6].

In the struggle algorithm, a variation of restricted tournament selection, two parents are chosen from the population, and crossover/mutation are performed to create a child. The child replaces the most similar individual in the entire population, but only if it has a better fitness. This replacement strategy counteracts genetic drift that can spoil population diversity. The struggle genetic algorithm has been demonstrated to perform well in multi-modal function landscapes where it successfully identifies and maintains multiple peaks.

As there is no single objective function to determine the fitness of the different individuals in a Pareto optimization, the ranking scheme presented by Fonseca and Fleming is employed, and the “degree of dominance” in attribute space is used to rank the population. Each individual is given a rank based on the number of individuals in the population that are preferred to it, i.e. for each individual the algorithm loops through the whole population counting the number of preferred individuals. "Preferred to" could be implemented in a strict Pareto optimal sense or extended to include goal levels on the objectives in order to limit the frontier.

The principle of the MOSGA algorithm is outlined below.

Step 1: Initialize the population.

Step 2: Select individuals uniformly from population.

Step 3: Perform crossover and mutation to create a child.

Step 4: Calculate the rank of the new child, and a new ranking of the population that considers the presence of the child.

Step 5: Find the most similar individual, and replace it with the new child if the child's ranking is better.

Step 6: Update the ranking of the population if the child has been inserted.

Step 7: Perform steps 2-6 until the mating pool is filled.

Step 8: If the stop criterion is not met go to step 2 and start a new generation.

The likeness of two individuals is measured using a distance function. The method has been tested with distance functions based upon the Euclidean distance in both the attribute as well as parameter space. A mixed distance function combining both the attribute and parameter distance has been evaluated as well.

3.1 Test function

In order to assess the performance of the algorithm a set of test problems from Deb [3] was explored. Deb developed a set of problems to highlight difficulties that multi-objective genetic algorithms may encounter. For visualization reasons, the focus is on two-dimensional problems defined generally by equations (4) and (5).

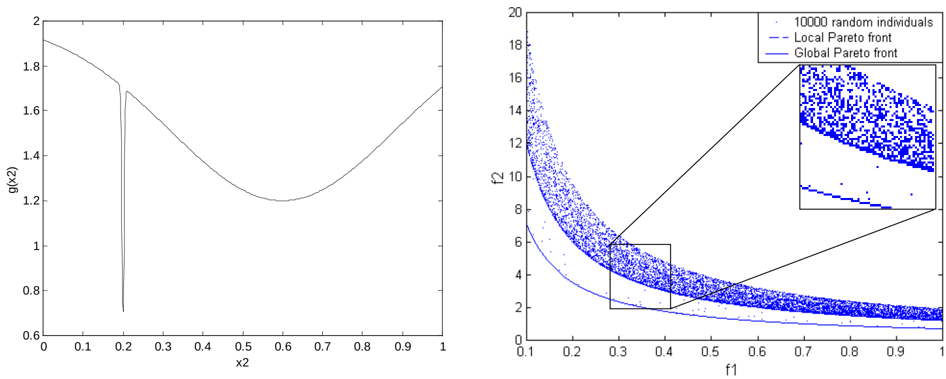
$$f_1(x_1, x_2) = x_1 \tag{4}$$

$$f_2(x_1, x_2) = \frac{g(x_2)}{x_1}, \quad g(x_2) > 0, \quad x_1 > 0 \tag{5}$$

If the function g is multi-modal, the corresponding multi-objective problem will have global and local Pareto-optimal frontiers. The multi-modal g function is defined in equation (9).

$$g(x_2) = 2 - \exp\left\{-\left(\frac{x_2 - 0.2}{0.004}\right)^2\right\} - 0.8\exp\left\{-\left(\frac{x_2 - 0.6}{0.4}\right)^2\right\} \quad (6)$$

Figure 3 (a) shows the g function for $0 \leq x_2 \leq 1$ with the global optima located at $x_2=0.2$ and a local optima at $x_2=0.6$. Figure 3(b) shows a plot of f_1 and f_2 in the attribute space with the global and local Pareto optimal solutions. 10,000 randomly chosen solutions are generated and plotted in Figure 3 (b) to illustrate that the problem is biased—the solution density is higher towards the local Pareto-optimal front.



(a) (b)

Figure 3. In (a) the function g(x2) is depicted. The global optimum is situated at x2=0.2 and the local optimum at x2=0.6. Figure (b) shows a random set of 10.000 solutions on a f1-f2 plot. Notice the low solution density at the global Pareto optimal front.

The optimization was conducted with a population size of 60 individuals and ran for 200 generations. The variables are real encoded, and BLX crossover is employed to produce offspring. The result from a BLX crossover between two real numbers A and B is randomly selected from a uniform distribution centered at the mean of A and B, with a width corresponding to twice the difference between A and B.

Deb reported that the NSGA was trapped in the local Pareto front in 59 out of 100 runs.

The original MOSGA algorithm used an attribute based distance function resulting in the algorithm converging to the local Pareto frontier in only 7% of 100 optimizations. The algorithm found the preferred global Pareto optimal front in 86% of the optimizations. In 7% of the optimizations it converged to both frontiers. Thus, the MOSGA seems more robust in locating the global Pareto optimal frontier.

However, one whishes that the algorithm should be capable of identifying both frontiers in every optimization run. By changing to a parameter based distance function this was achieved. However, the parameter based distance function was slower and less exact in its convergence to the frontier.

By combining equally weighted attribute-based and a parameter-based distance functions to form a mixed distance measure, the advantages of fast convergence and the ability of finding multiple frontiers were realized. Figure 4 shows how the algorithm spreads the population evenly on both frontiers when using the mixed distance function. A more detailed discussion about the properties of the algorithm is given in [1].

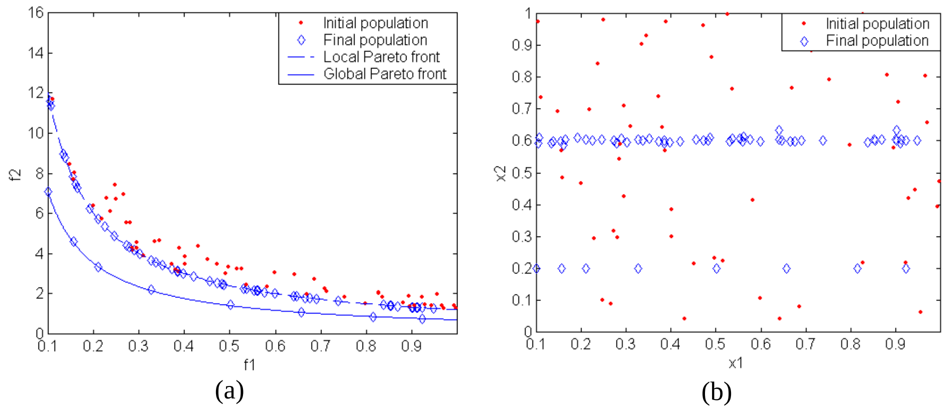


Figure 4. The final population after 200 generation using the mixed distance measure. (a) shows the population in attribute space whereas (b) shows the population in parameter space.

Thus, the method is capable of reliably identifying multiple Pareto frontiers in a single optimization run, outperforming other techniques. Another advantage is that the method does not require problem specific parameter settings. The only GA parameters that have to be determined are population size, number of generations and the distance function. The distance function is dependent on how the optimization parameters are encoded. As long as they are real encoded, a mixed distance function based upon Euclidean distance is preferred. If a problem requires for instance binary encoding, other distance functions such as the hamming distance might be considered.

The method has been successfully tested on several benchmark problems proposed by Deb [3].

4 The design problem

The objects of study for the design problem are two different concepts of hydraulic actuation systems. Both systems consist of a hydraulic cylinder that is connected to a mass of 1000 kilograms. The objective is to follow a pulse in the position command with a small control error and simultaneously obtain a low energy consumption. Naturally, these two objectives are in conflict with each other. The problem is thus to minimize both the control error and the energy consumption from a Pareto optimal perspective.

Two different ways of controlling the cylinder are studied. In the first more conventional system, the cylinder is controlled by a servo valve, which is powered from a constant pressure system. In the second concept, the cylinder is controlled by a servo pump. Thus, the systems have different properties. The valve concept has all that is required for a low control error, as the valve has a very high bandwidth. On the other hand, the valve system associated with higher losses, as the valve constantly throttles fluid to the tank.

The different concepts have been modeled in the simulation package Hopsan [12]. The models of each component consist of a set of algebraic and differential equations taking aspects such as friction, leakage and non-linearities into account. The system models are depicted in Figure 5 and Figure 6 respectively.

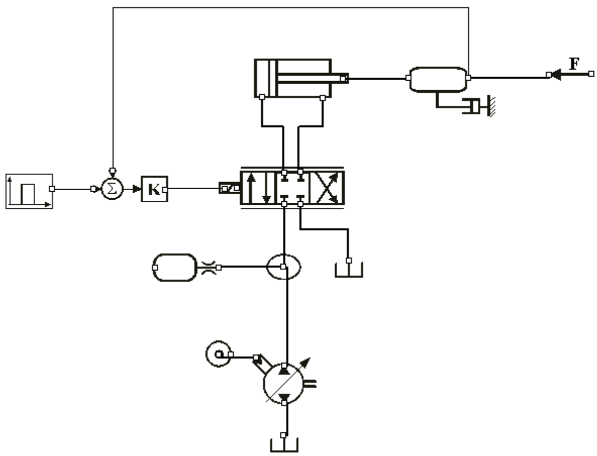


Figure 5. The servo valve concept for hydraulic actuation.

The servo valve system consists of the mass and the hydraulic cylinder, the servo valve and a p-controller that is controlling the motion. The servo valve is powered by a constant pressure pump and a accumulator, which keeps the system pressure at a constant level. The optimization parameters are the sizes of the cylinder, valve and the pump, the pressure lever, the feedback gain and a leakage parameter that is necessary to dampen the system. Thus, this problem consists of six optimization parameters and two objectives.

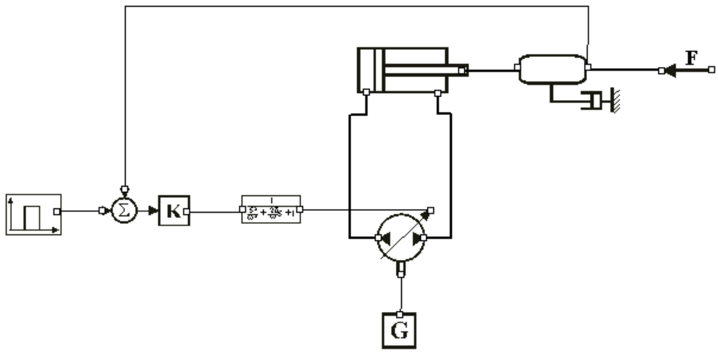


Figure 6. The servo pump concept of hydraulic actuation.

The servo pump concept contains fewer components, the cylinder and the mass, the controller and the pump. A second order low-pass filter is added in order to model the dynamics of the pump. The servo pump system consists of only four optimization parameters. The performance of a relatively fast servo pump system is depicted in Figure 7.

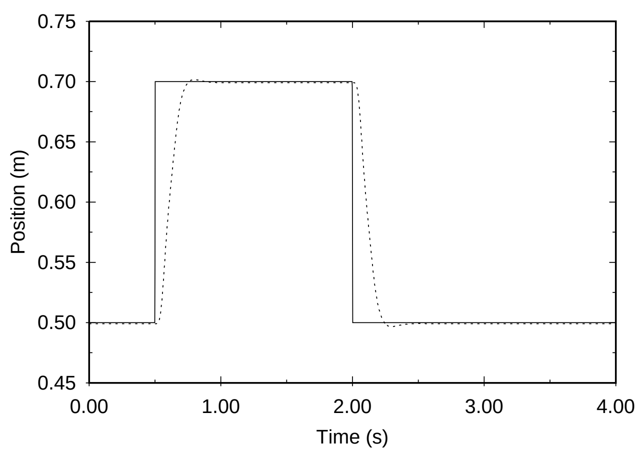


Figure 7. The pulse response for a relatively fast servo pump system, control error = 0.05 ms.

4.1 Optimization

The optimization is based on component size selection rather then component design, i.e. it is assumed that each component is a predefined entity. As a consequence of this assumption most component parameters are expressed as a function of the component size.

Both systems where optimized in order to simultaneously minimize the control error f1 and the energy consumption f2. The control error is obtained by integrating the absolute value of the control error and adding a penalty for overshoots, see equation (7). The energy consumption is calculated by integrating the hydraulic power, expressed as the pressure times the flow, see equation (8).

$$f_1 = \int_0^4 \left| x_{ref} - x \right| dt + \alpha \left(\int_0^2 \left(x > x_{ref} \right) dt + \int_2^4 \left(x < x_{ref} \right) dt \right) \tag{7}$$

$$f_2 = \int_0^4 \left(q_{pump} \cdot p_{pump} \right) dt \tag{8}$$

The optimization is conducted with a population size of 30 individuals over 200 generations. The parameters are real encoded and BLX crossover is used to produce new offspring. Euclidean distance measures were used.

As a Pareto optimization searches for all non-dominated individuals, the final population will contain individuals with a very high control error, as they have low energy consumption. It is possible to obtain an energy consumption close to zero, if the cylinder does not move at all. However, these solutions are not of interest, as we want the system to follow the pulse. Therefore, a goal level on the control error is introduced. The optimization strategy is modified so that solutions, which are bellow the goal level on the control error are always preferred to solutions that are above it regardless of their energy consumption. In this manner, the population is focused on the relevant part of the Pareto front.

The obtained Pareto optimal front for the servo valve system is depicted in Figure 8 (a), and for the servo pump system in Figure 8 (b).

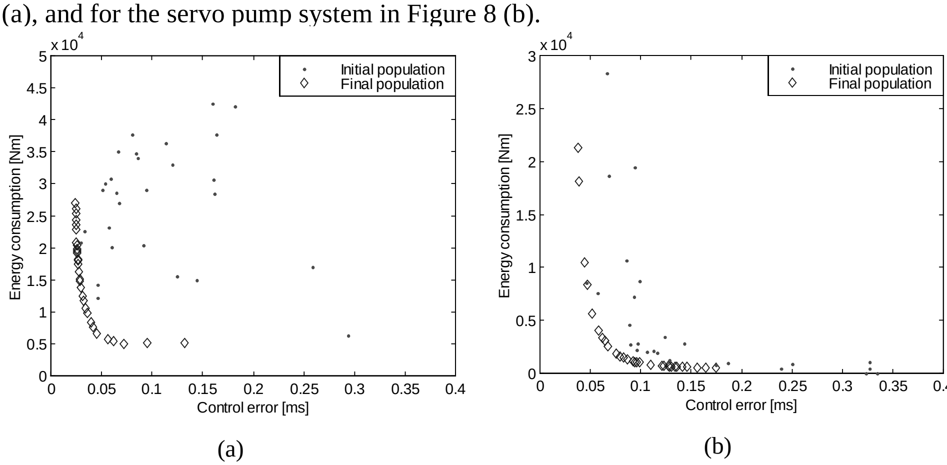


Figure 8. The resulting Pareto optimal front, in (a) for the servo valve concept, and in (b) for the servo pump concept.

In order to achieve fast systems, and thereby low control errors, large pumps and valves are chosen by the optimization strategy. A large pump delivers more fluid, which enables a higher speed of the cylinder. However, bigger components consume more energy, which explains the shape of the Pareto frontiers.

If the Pareto fronts for both concepts are displayed within the same graph, the properties of both systems are clearly elucidated, see Figure 9.

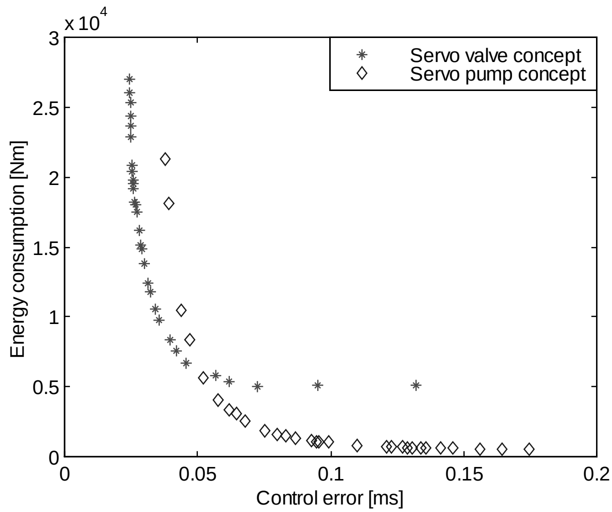


Figure 9. The Pareto frontiers for both concepts.

It is evident that the final design should preferably be on the overall Pareto frontier, which elucidate when to change between concept. The servo pump system consumes less energy, and is preferred if a control error larger then 0.05ms is acceptable. The servo valve system is fast but consumes more energy. If lower control error then 0.05ms is desired, the final design should preferably be a servo valve system.

In order to choose the final design, the decision-maker has to select concept and then study the tradeoff between the control error and the energy consumption and select a solution point on the Pareto frontier.

Naturally there are other criteria that has to be taken into account as well, for instance system weight and cost, flexibility, reliability and robustness. Some of these properties are hard to assess, and has to be left outside the optimization. New criteria could be introduced either as new objectives or aggregated with one of the other objectives to a new composite objective. There are no fundamental limitations as to how many objectives one could optimize, but visualization becomes harder as the number of objectives increases.

5 Conclusion

A multi-objective Pareto genetic algorithm has been proposed and shown to outperform previously published methods in its capability to identify multiple Pareto frontiers in a single optimization run. Another advantage of the method is that it does not require extensive parameter tuning on a problem by problem basis.

The method has been tested on both mathematical functions and design problems where simulation models have been employed to predict the performance of different design solutions. The outcome of the optimization is a set of Pareto optimal designs, where the tradeoff of the conflicting objectives is clearly elucidated. By comparing Pareto frontiers for different design concepts, valuable insight on the properties of the different concepts could be gained. Thus, Pareto optimization could be a valuable support for concept selection.

The method has been applied to two concepts of hydraulic actuation systems. The resulting Pareto optimal frontiers elucidate the advantages of the different concepts and, advice the decision-maker which concept to choose depending on his or her preferences.

6 Acknowledgment

The software for this work builds upon the GAlib genetic algorithm package, written by Matthew Wall [20] at the Massachusetts Institute of Technology.

7 References

[1] Andersson J. and Wallace D., 2000, “Pareto optimization using the struggle genetic crowding algorithm,” Submitted for review to ASME Journal of Mechanical Design.

[2] Coello C., 1996, An empirical study of evolutionary techniques for multiobjective optimization in engineering design, Dissertation, Department of Computer Science, Tulane University.

[3] Deb K., 1999, “Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems,” Evolutionary Computation, vol. 7, pp. 205-230.

[4] Donne M., Tilley D. and Richards W., 1995, "The use of multi-objective parallel genetic algorithms to aid fluid power design", Proc. Instn Mech. Engrs, Part I, Journal of Systems and Control Engineering, vol. 209, No 1, pp. 53-61.

[5] Fonseca C. and Fleming P., 1995, “An overview of evolutionary algorithms in multiobjective optimization,” Evolutionary Computation, vol. 3, pp. 1-18.

[6] Fonseca C. M. and Fleming P. J., 1998, “Multiobjective optimization and multiple constraint handling with evolutionary algorithms - Part I: a unified formulation,” IEEE Transactions on Systems, Man, & Cybernetics Part A: Systems & Humans, vol. 28, pp. 26-37.

[7] Fonseca C. M. and Fleming P. J., 1995, “Multiobjective genetic algorithms made easy: Selection, sharing and mating restriction,” presented at 1st IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems, Sheffield, England.

[8] Fourman M. P., 1985, “Compaction of symbolic layout using genetic algorithms,” presented at 1st Int. Conference on Genetic Algorithms, Pittsburgh.

[9] Goldberg D., 1989, "Genetic Algorithms in Search and Machine Learning. Reading, Addison Wesley.

[10] Grueninger T. and Wallace D., 1996, “Multi-modal optimization using genetic algorithms”, Technical Report 96.02, CADlab, Massachusetts Institute of Technology, Cambridge.

[11] Harik G., 1995, “Finding multimodal solutions using restricted tournament selection,” presented at Sixth International Conference on Genetic Algorithms.

[12] Hopsan, 1991, “Hopsan, a simulation package - User's guide”, Technical report LiTH-IKP-R-704, Department of Mechanical Engineering, Linköping University, Linköping, Sweden.

[13] Horn J., 1997, “Multicriterion decision making,” in Handbook of evolutionary computation, T. Bäck, D. Fogel, and Z. Michalewicz, Eds., IOP Publishing Ltd and Oxford University Press.

[14] Horn J. and Nafpliotis N., 1993, “Multiobjective Optimization Using the Niched Pareto Genetic Algorithm”, Technical Report 93005, Illinois Genetic Algorithm Laboratory, Dept. of General Engineering, University of Illinois at Urbana-Champaign, Urbana, USA.

[15] Kurasawe F., 1991, “A variant of evolution strategies for vector optimization,” in Parallel Problem Solving from Nature, Lecture Notes in Computer Science 496, Berlin, Springer Verlag, pp. 193-7.

[16] Pareto V., 1896, "Cours d'Économie Politique. Lausanne, Rouge.

[17] Schaffer J., 1985, “Multiple objective optimization with vector evaluated genetic algorithms,” presented at 1st Int. Conf. on Genetic Algorithms, Pittsburgh.

[18] Srinivas N. and Deb K., 1995, “Multiobjective optimization using nondominated sorting in genetic algorithms,” Evolutionary Computation, vol. 2, pp. 221-248.

[19] Tamaki H., Kita H., and Kobayashi S., 1996, “Multi-objective optimization by genetic algorithms: a review,” presented at 1996 IEEE International Conference on Evolutionary Computation, ICEC'96, Nagoya, Japan.

[20] Wall M., 1996, “Matthew's GAlib: A C++ library of genetic algorithm components,” http://www.lancet.mit.edu/ga/.

[21] Zitzler E. and Thiele L., 1999, “Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach,” IEEE Transaction on evolutionary computation, vol. 3, pp. 257-271.

Paper VI

Multiobjective Optimization of Mixed Variable Design Problems

Johan Andersson and Petter Krus

This paper has been published as:
ANDERSSON J., KRUS P., “Multiobjective Optimization of Mixed Variable Design Problems”, in *Proc. of 1st International Conference on Evolutionary Multi-criterion optimization* (Eds. E. Zitzler et al.), Lecture Notes in Computer Science (LNCS) Vol. 1993, Springer-Verlag,. Zurich, Switzerland, March 7-9, 2001

Multiobjective Optimization of Mixed Variable Design Problems

Johan Andersson and Petter Krus

Department of Mechanical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Abstract

In this paper, a new multiobjective genetic algorithm is employed to support the design of a hydraulic actuation system. First, the proposed method is tested using benchmark problems gathered from the literature. The method performs well and it is capable of identifying multiple Pareto frontiers in multi-modal function spaces. Secondly, the method is applied to a mixed variable design problem where a hydraulic actuation system is analyzed using simulation models. The design problem consists of a mixture of determining continuous variables and selecting components from catalogs. The multiobjective optimization results in a discrete Pareto front, which illustrates the trade-off between system cost and system performance.

1 Introduction

Most engineering design problems consist of several, often conflicting, objectives. In many cases, the multiple objectives are aggregated into one single overall objective function. Optimization is then conducted with one optimal design as the result. The result is then strongly dependent on how the objectives are aggregated. To avoid this difficulty and in order to explore a broader set of optimal solutions, the concept of Pareto optimality is employed. Valuable insights about the trade-off between the objectives could be gained by investigating the set Pareto optimal solutions. Vilfredo Pareto defined Pareto optimality as the set where every element is a problem solution for which no other solutions can be better in all design attributes. A solution in a Pareto optimal set cannot be deemed superior to the others in the set without including preference information to rank competing attributes.

This paper develops a Pareto optimization method for use in multiobjective, multimodal design spaces. For a general design problem, the design space consists of continuous variables as well as selection of individual components from catalogs or databases. Furthermore, numerical simulations and other CAE tools are often employed to evaluate design solutions; i.e. simulation is employed to transform solutions from the design space to the attribute space. As the attributes or objectives are calculated using numerical simulations, there is no simple way of obtaining derivatives of the objective functions. Therefore genetic algorithms are-well suited for such applications—they do not need derivatives of the objective functions and they have shown to be effective in optimizing mixed variable problems [11] in multi-modal search spaces [7].

The paper first defines a general multiobjective optimization problem and reviews related work on multiobjective genetic algorithms. Then, a new method is proposed and validated using a problem gathered from the literature. Later the method is applied to a real design problem containing a mixture of continuous design variables and discrete selections of components from catalogs. The problem is solved by connecting the optimization strategy to a simulation program.

1.1 The multiobjective design problem

A general multiobjective design problem could be expressed by equations (1) and (2)

$$\min \mathbf{F}(\mathbf{x}) = \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})\right)^T \qquad s.t.\ \mathbf{x} \in S \tag{1}$$

$$\mathbf{x} = \left(x_1, x_2, \ldots, x_n\right)^T \tag{2}$$

where $f_1(x), f_2(x), \ldots, f_k(x)$ are the k objective functions, $(x_1, x_2, \ldots, x_n)$ are the n optimization parameters, and $S \in R^n$ is the solution or parameter space. Obtainable objective vectors, $\{\mathbf{F}(\mathbf{x}) | x \in S\}$ are denoted by *Y*. $Y \in R^k$ is usually referred to as the attribute space.

The Pareto set consists of solutions that are not dominated by any other solutions. Considering a minimization problem and two solution vectors **x**, **y**∈*S*. **x** is said to dominate **y**, denoted $\mathbf{x} \succ \mathbf{y}$, if:

$$\forall i \in \{1,2,...,k\}: f_i(\mathbf{x}) \leq f_i(\mathbf{y}) \quad and \quad \exists j \in \{1,2,...,k\}: f_j(\mathbf{x}) < f_j(\mathbf{y}) \tag{3}$$

The space in R^k formed by the objective vectors of Pareto optimal solutions is known as the Pareto optimal front.

1.2 Multiobjective genetic algorithms

Genetic algorithms are modeled after mechanisms of natural selection. Each optimization parameter (x_n) is encoded by a gene using an appropriate representation, such as a real number or a string of bits. The corresponding genes for all parameters $x_1,..x_n$ form a chromosome capable of describing an individual design solution. A set of chromosomes representing several individual design solutions comprises a population where the fittest are selected to reproduce. Mating is performed using crossover to combine genes from different parents to produce children. The children are inserted into the population and the procedure starts over again, thus creating an artificial Darwinian environment. For a general introduction to genetic algorithms, see [6].

Additionally, there are many different types of multiobjective genetic algorithms. Literature surveys and comparative studies on multiobjective genetic algorithms can be found in for example [3], [10] and [12].

Most multiobjective genetic algorithms use either the selection mechanism or some sort of Pareto-based ranking to produce non-dominated solutions. In the proposed method, the ranking scheme presented by Fonseca and Fleming [5] is employed.

In the multiobjective GA (MOGA) [5] each individual is ranked according to its degree of dominance. The more population members that dominate an individual, the higher the ranking of the individual. Here an individual's ranking equals the number of individuals that it is dominated by plus one, see Figure 1. Individuals on the current Pareto front will have a rank of 1, as they are non-dominated. The rankings are then scaled to score individuals in the population. In MOGA, both sharing and mating restrictions are employed in order to maintain population diversity. Fonseca and Fleming also introduce preference information and goal levels to reduce the Pareto set to those that simultaneously meet certain attribute values.

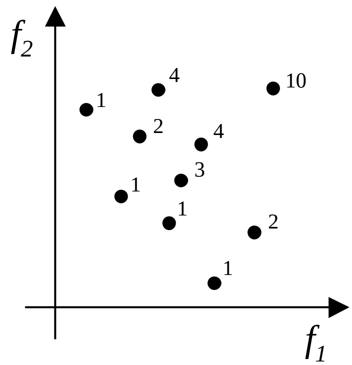


Figure 1. Population ranking according to Fonseca and Fleming.

Although there is a substantial body of research on multiobjective genetic algorithms, there are still important issues that current methods address with only partial success. The methods typically require extensive genetic algorithm parameter tuning on a problem-by-problem basis in order for the algorithm to perform well. However, in a real-world problem there is little knowledge about the shape of the attribute space, which makes it difficult to assess problem-specific parameters. Additionally, existing methods do not handle the location of multiple Pareto frontiers in multi-modal problem spaces consistently. The method presented in this paper is capable of identifying multiple frontiers without any problem-specific parameter tuning.

2 The Proposed Method

The multiobjective struggle genetic algorithm (MOSGA) [1] combines the struggle crowding genetic algorithm [7] with Pareto-based ranking as devised in [5].

In the struggle algorithm, a variation of restricted tournament selection [8], two parents are chosen randomly from the population, and crossover/mutation is performed to create a child. The child then has to compete with the most similar individual in the entire population, and replaces it if the child has a better fitness. This replacement strategy counteracts genetic drift that can spoil population diversity. The struggle genetic algorithm has been demonstrated to perform well in multi-modal function landscapes where it successfully identifies and maintains multiple peaks.

There is no single objective function to determine the fitness of the different individuals in a Pareto optimization. Therefore, the ranking scheme presented by Fonseca and Fleming is employed, and the “degree of dominance” is used to rank the population. Each individual is given a rank based on the number of individuals in the population that are preferred to it, i.e. for each individual the algorithm loops through the population counting the number of preferred individuals. “Preferred to” can be implemented in a strict Pareto optimal sense or extended to include goal levels on the objectives in order to limit the frontier.

The principle of the MOSGA algorithm is outlined below.

Step 1: Initialize the population.
Step 2: Select parents randomly from the population.
Step 3: Perform crossover and mutation to create a child.
Step 4: Calculate the rank of the child, and a new ranking of the population that considers the presence of the child.
Step 5: Find the most similar individual, and replace it with the new child if the child's ranking is better.
Step 6: Update the ranking of the population if the child has been inserted.
Step 7: Perform steps 2-6 until the mating pool is filled.
Step 8: If the stop criterion is not met go to step 2 and start a new generation.

The similarity between two individuals is measured using a distance function. The method was tested with distance functions based upon the Euclidean distance in both the attribute as well as the parameter space. A mixed distance function combining both the attribute and the parameter distance was evaluated as well.

2.1 Genome Representation

The genome encodes design variables in a form suitable for the GA to operate upon. Design variables may be values of parameters (real or integer) or represent individual components selected from catalogs or databases. Thus, the genome is a hybrid list of real numbers (for continuous parameters), integers and references to catalog selections, see Figure 2.

A catalog could be either a straight list of elements, or the elements could be arranged in a hierarchy. Each element of a catalog represents an individual component. The characteristics of catalogs will be discussed further on and exemplified by the design example.

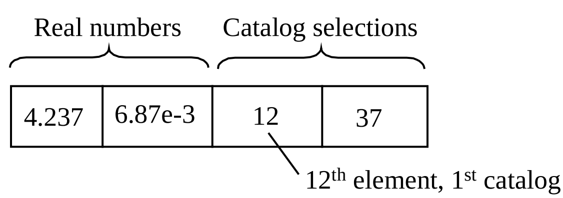


Figure 2. Example of the genome encoding. The first two elements represent real variables and the last two elements catalog selections.

2.2 Similarity Measures

Speciating GAs require a measure of likeness between individuals, a so-called similarity measure. Here the similarity measure is based on a distance function calculating the distance between two genomes. The similarity could be based on the distance in either the attribute space (between the objectives), the phenotype space (between the design parameters) or the genotype space (in the genome encoding). As direct encoding is used (not a conversion to a string of bits), a phenotype and a genotype distance function would yield the same result. It is shown that the choice between an attribute-based and a parameter based distance function might have a great influence on the outcome of the optimization.

2.2.1 Attribute Based Distance Function

One way of comparing two individual designs is to calculate their distance in attribute space. As we want the population to spread evenly on the Pareto front (in attribute space) it seems to be a good idea to use an attribute-based distance measure. The distance between two solutions (genomes) in attribute space is calculated using the normalized Euclidean distance (4).

$$\text{Distance}(a,b) = \sqrt{\sum_{i=1}^{k}\left(\frac{f_{ia} - f_{ib}}{f_{i\max} - f_{i\min}}\right)^2 \frac{1}{k}} \tag{4}$$

Here, f_{ia} and f_{ib} are the objective values for the i:th objective for a and b respectively. f_{imax} and f_{imin} are the maximum and the minimum of the i:th objective in the current population, and k is the number of objectives. Thus, the distance function will vary between 0, indicating that the individuals are identical, and 1 for the very extremes.

2.2.2 Phenotype Based Distance Function

Another way of calculating the distance between solutions is to use the distance in parameter (phenotype) space. As the genome is a hybrid mixture of real numbers and catalog selections, we have to define different distance functions to work on different types of elements. The methods described here are founded on the framework presented by Senin et al. [11]. In order to obtain the similarity between two individuals, the distance between each search variable is calculated. The overall similarity is then obtained by summing up the distances for each search variable.

Real Number Distance

A natural distance measure between two real numbers is the normalized Euclidean distance, see equation (5).

$$\text{Distance}(a,b) = \sqrt{\left(\frac{a-b}{\text{max distance}}\right)^2} \tag{5}$$

Here, a and b are the values for the two real numbers and max distance is the maximum possible distance between the two values (i.e. the search boundaries).

Catalog Selection Distance

Distance between two catalog selections could be measured through relative position in a catalog or a catalog hierarchy. The relative position is only meaningful if the catalog is ordered, see Figure 3.

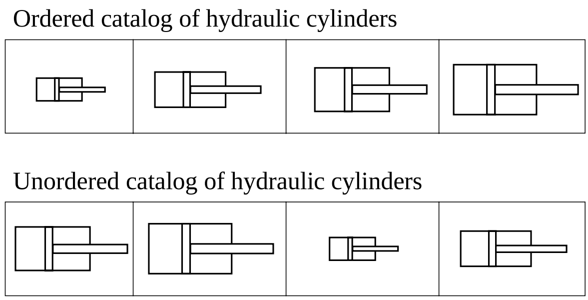


Figure 3. Examples of ordered and unordered catalogs.

The dimensionless distance between two elements within the same catalog is expressed by equation (6) and exemplified in Figure 4.

$$\text{Distance}(a,b) = \frac{pos(a) - pos(b)}{\text{max distance}} \tag{6}$$

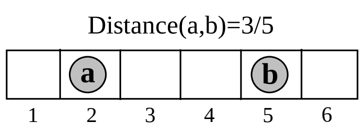


Figure 4. Distance evaluation for two elements of an ordered catalog.

For catalog hierarchies equation (6) has to be generalized. For elements belonging to the same sub-catalog, the distance is evaluated using the relative position within that sub-catalog. Otherwise, the maximum length of the path connecting the different sub-catalogs is used. This implies that for two given sub-catalogs an element in one catalog is equally distant from every element in the other catalog. The length of the path is calculated as the maximal distance within the smallest common hierarchy. In both cases, the distance is normalized by dividing with the maximum distance (i.e. the catalog size).

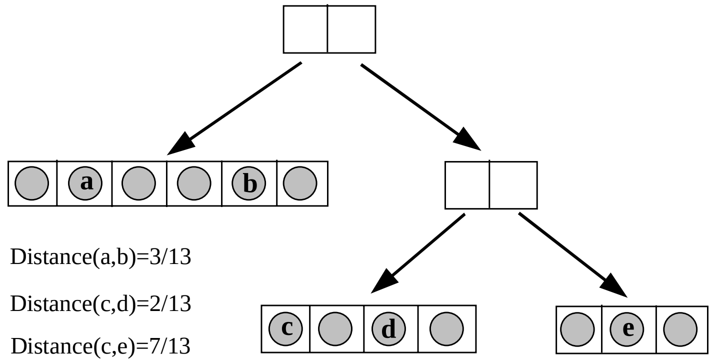


Figure 5. Exemplification of distances between different catalog elements in a hierarchical catalog.

Overall Distance

So far, distance measures for individual design variables have been developed. An overall distance measure for comparing two genomes is obtained by aggregating the distances for the individual design variables, see equation (7).

$$\text{Distance}(a,b) = \sum_{i=1}^{n} \frac{\text{Distance}(\text{DV}_i)}{n} \tag{7}$$

Where a and b are the two designs being compared, and n is the number of design variables (DV) encoded by the genome. Thus, the phenotype distance between two individual designs is calculated by summing up the individual distances for each element of the genome.

Mixed Distance Function

As we will show later, different distance functions have different properties. By combining an attribute based distance function with a parameter-based one, the strengths from both methods were taken advantage of. As each distance function is normalized, the mixed distance function is simply calculated according to equation (8).

$$\text{Mixdistance}(a,b) = \frac{\text{AttributeDistance}(a,b) + \text{PhenotypeDistance}(a,b)}{2} \tag{8}$$

2.3 Genetic operators

As the genome is a hybrid mix of continuous variables and catalog selections, we define different operators to work on different types of elements. Uniform crossover is used, which implies that each element of the father's genome is crossed with the corresponding element from the mother's genome.
For real numbers, BLX crossover is used, see exemplification in Figure 6.

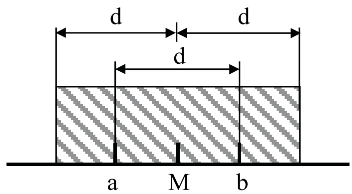


Figure 6. The outcome of a BLX crossover between two real numbers a and b is randomly selected from an interval of width 2d centered on the average M.

For catalog selections, an analog crossover scheme is employed as illustrated in Figure 7.

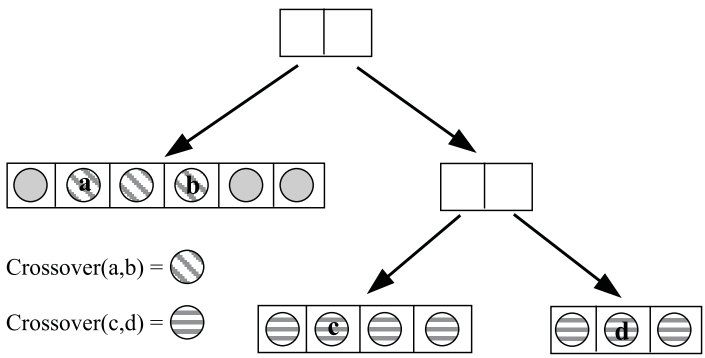


Figure 7. An exemplification of the catalog crossover. The outcome of a crossover of individuals within the same catalog (a and b) are randomly selected from the interval between them. For individuals from different sub-catalogs (c and d) the outcome is randomly selected within the smallest common hierarchy.

2.4 Test Function

In order to assess the performance of the algorithm, a set of test problems from Deb [19] was explored. Deb developed a set of problems to highlight difficulties that multiobjective genetic algorithms may encounter. For visualization purposes, the focus is on two-dimensional problems defined generally by equations (9) and (10).

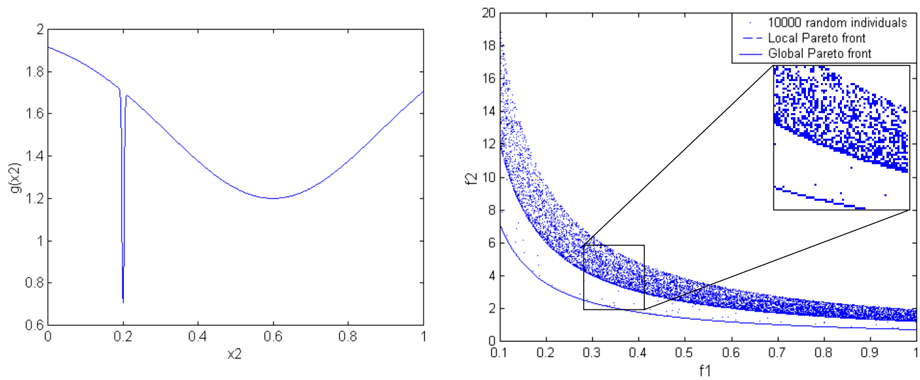
$$f_1(x_1, x_2) = x_1 \tag{9}$$

$$f_2(x_1, x_2) = \frac{g(x_2)}{x_1}, \quad g(x_2) > 0, \quad x_1 > 0 \tag{10}$$

If the function g is multi-modal, the corresponding multiobjective problem will have global and local Pareto optimal frontiers. A multi-modal g function is defined in equation (11).

$$g(x_2) = 2 - \exp\left\{-\left(\frac{x_2 - 0.2}{0.004}\right)^2\right\} - 0.8\exp\left\{-\left(\frac{x_2 - 0.6}{0.4}\right)^2\right\} \tag{11}$$

Figure 8(a) shows the g function for $0 \leq x_2 \leq 1$ with the global optimum located at $x_2=0.2$ and a local optimum at $x_2=0.6$. Figure 8(b) shows a plot of f_1 and f_2 in the attribute space with the global and local Pareto optimal solutions. 10,000 randomly chosen solutions are generated and plotted in Figure 8(b) to illustrate that the problem is biased—the solution density is higher towards the local Pareto optimal front.



(a)

(b)

Figure 8. Figure (a) shows the multi-modal function $g(x_2)$, where the global optimum is situated at $x_2=0.2$ and the local optimum at $x_2=0.6$. For the multiobjective problem, a f_1-f_2 plot for 10,000 random solutions is shown in (b). Note the low solution density at the global Pareto optimal front.

The optimization was conducted with a population size of 60 individuals and ran for 200 generations. The variables are real-encoded, and BLX crossover is employed to produce offspring. Deb reported that the NSGA was trapped in the local Pareto front in 59 out of 100 runs.

The original MOSGA algorithm used an attribute based distance function resulting in the algorithm converging to the local Pareto frontier in only 7% of 100 optimizations. The algorithm found the preferred global Pareto optimal front in 86% of the optimizations, as shown in figure 9 (a) and (b). In 7% of the optimizations, it converged to both frontiers. Thus, the MOSGA seems more robust in locating the global Pareto optimal frontier.

However, the algorithm should ideally be capable of identifying both fronts in every optimization run. By changing to a parameter based distance function this can be achieved. However, the parameter based distance function was slower and less exact in its convergence to the frontier.

In the MOSGA, the new child has to compete with the individual most similar to itself. When the comparison is done in parameter space, a portion of the population will find and maintain local optima, where solutions close in the parameter space are all dominated. When using an attribute based distance function, solutions at local optima might have to compete with solutions at the global optima, as they might be close in attribute space. Therefore, local optima would not be maintained.

By combining equally weighted attribute based and a parameter based distance functions to form a mixed distance measure, the advantages of fast convergence and the ability to find multiple frontiers were realized. Figure 9 shows how the algorithm spreads the population evenly on both frontiers when using the mixed distance function. To summarize, the attribute distance function performs well on problems with one Pareto frontier. For problems with multiple frontiers, a mixed distance function is preferred. A more detailed discussion about the properties of the algorithm is given in [1] and [2].

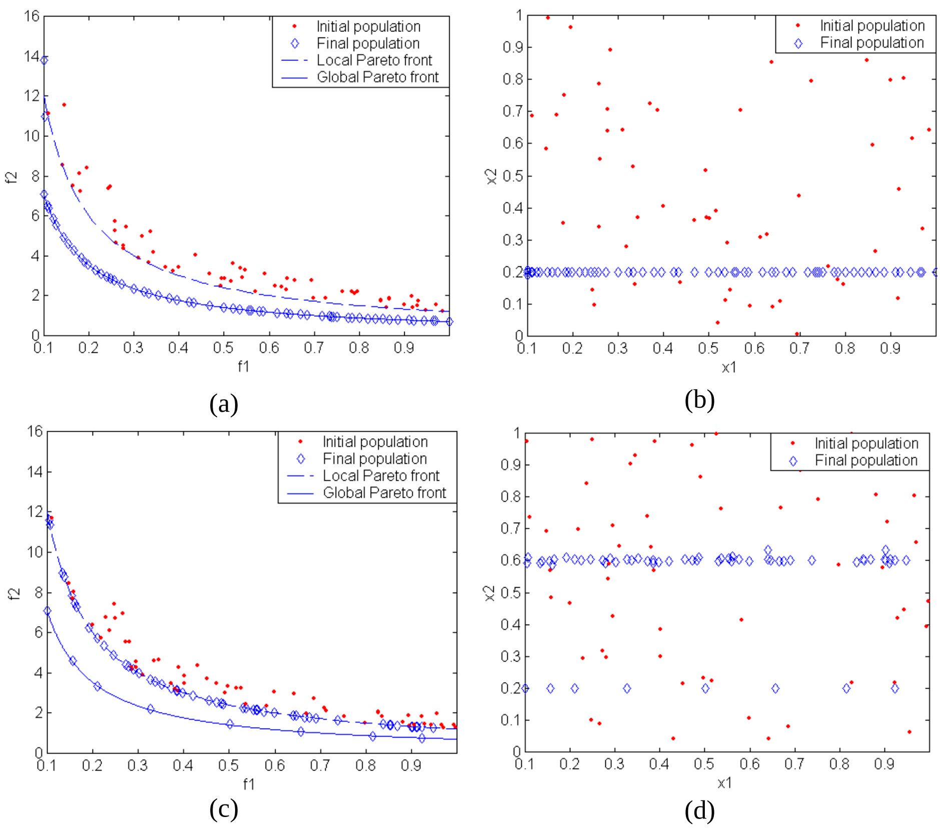


Figure 9. Optimization results using different distance functions. In (a) and (b) an attribute based distance function is used and the population has converged to the global Pareto front. In (c) and (d) the mixed distance function is used and the population converges to both the global and the local frontier. (a) and (c) show the result in attribute space, whereas (b) and (d) show the result in parameter space.

Thus, the method is capable of reliably identifying multiple Pareto frontiers in a single optimization run, thus outperforming other techniques. Another advantage is that the method does not require problem-specific parameter settings. The only GA parameters to be determined are population size, number of generations and the distance function. The method was successfully tested on several benchmark problems proposed by Deb, see [1].

3 Design Example

The object of study for the design example is a hydraulic actuation system. The system consists of a hydraulic cylinder connected to a mass. The motion of the mass is controlled by a directional valve, which in turn is controlled by a proportional controller. The system is powered from a constant pressure hydraulic supply system.

In order to investigate the properties of different designs, the system was modeled in the simulation package Hopsan [9]. For every new genome, the optimization strategy calls the simulation program to evaluate that particular design. Each component in the simulation model consists of a set of algebraic and differential equations taking aspects such as friction, leakage and non-linearities into account. A graphical representation of the system model is depicted in figure 10.

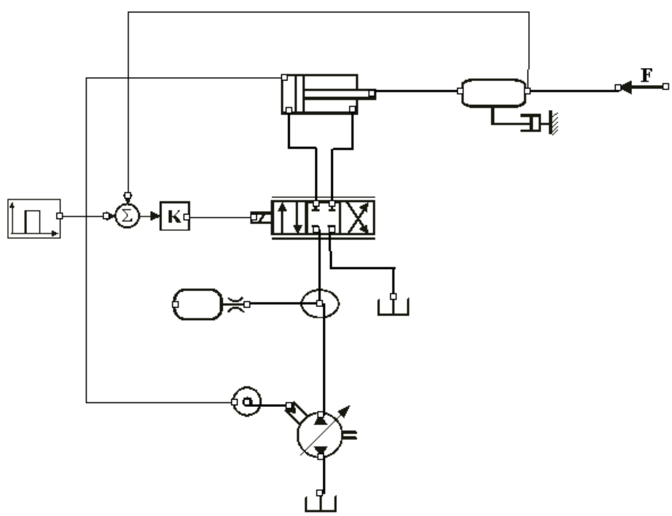


Figure 10. The simulation model of the hydraulic actuation system. The main components are (from the upper left): cylinder, mass, pulse generator, p-controller, directional valve, accumulator and constant pressure pump.

The objective of the study is to design a system with good controllability to a low cost. Naturally, these two objectives are in conflict with each other. To achieve good controllability we can choose a fast servo valve, which is more expensive than a slower proportional valve. Therefore, there is a trade-off between cost and controllability. The cost for a particular design is composed of the cost for the individual components as well as the cost induced by the energy consumption.

The system was studied for a pulse in the position command. The control error and the energy consumption are calculated based on the simulation result.

When designing, the system cylinders and valves are selected from a catalog of existing components. Other parameters such as the control parameter, a leakage coefficient and the maximal flow of the supply system have to be determined as well. Thus the problem is multiobjective with two objectives and five optimization variables, of which two are discrete catalog selections and three are continuous variables.

3.1 Component Catalogs

For the catalog selections, catalogs of valves and cylinders were created. For the directional valve, the choice is between a slow but inexpensive proportional valve or an expensive and fast servo valve. Valves from different suppliers were arranged in two ordered sub-catalogs as depicted in figure 11. The same structure applies to the cylinders as they are divided into sub-catalogs based on their maximal pressure level. The pressure in the system has to be controlled so that the maximum pressure for the cylinder is not exceeded. A low-pressure system is more economical but shows inferior performance compared to a high-pressure system.

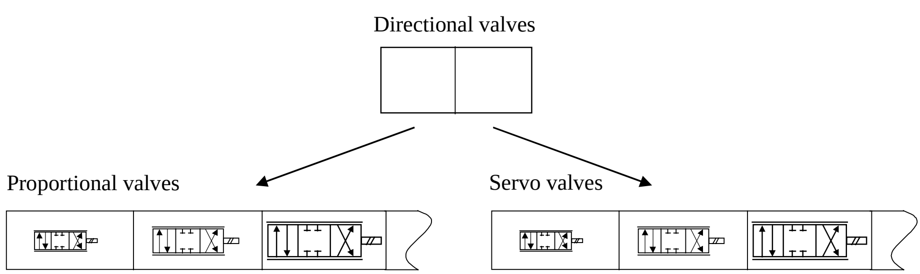


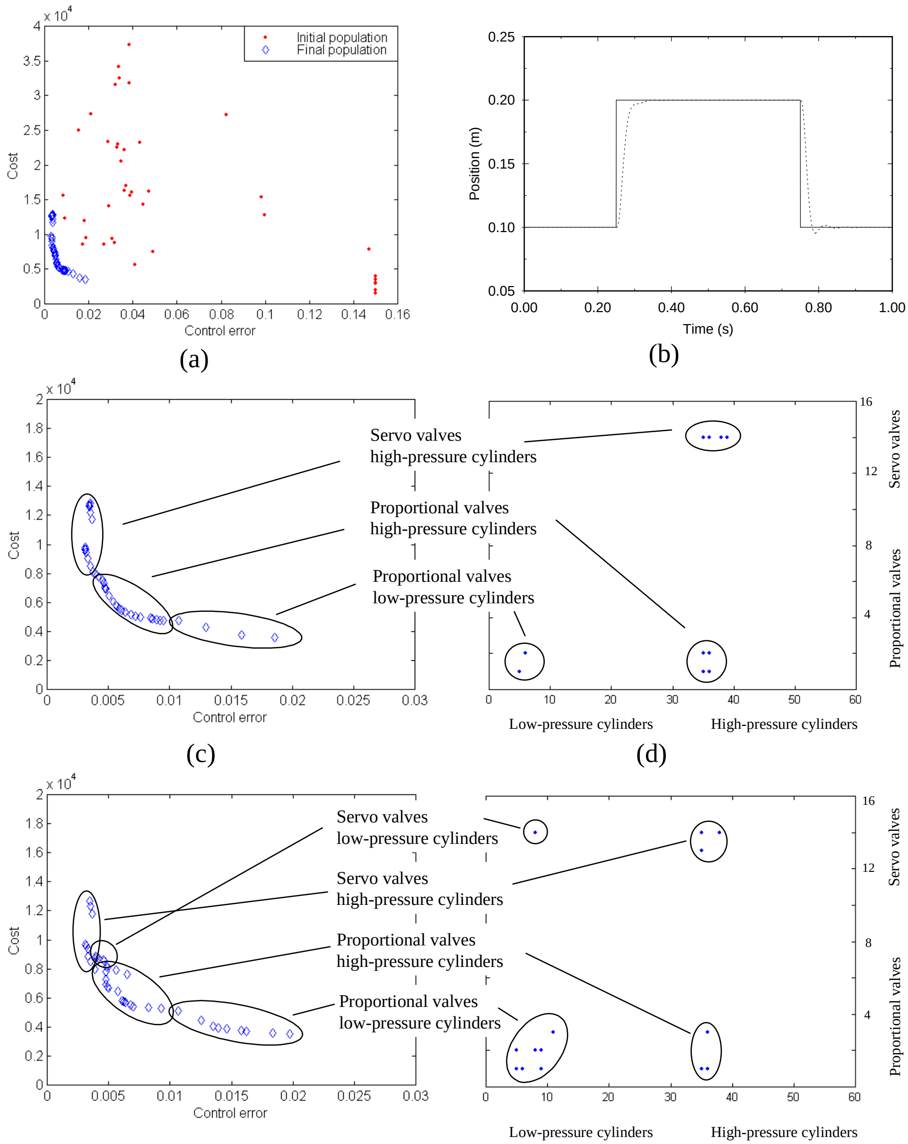
Figure 11. The catalog of directional valves is divided into proportional valves and servo valves. Each sub-catalog is ordered based on the valve size. For each component, a set of parameters describing the component is stored together with information on price and weight.

Naturally, the component catalog is connected to the simulation program. The optimization strategy however needs information about the topology of the catalog in order for the genetic operators to work.

3.2 Optimization Results

The system was optimized using a population of 40 individuals and 400 generations. In order to limit the Pareto frontier, a goal level on the control error was introduced. The goal level corresponds to the highest acceptable control error. Without such a goal level, the result would include very inexpensive designs that do not follow the position command at all. The introduction of goal levels therefore focuses the population on the most interesting parts of the Pareto frontier.

The result could be divided into four distinct regions depending on valve type and pressure level, see figure 12.



(a) (b) (c) (d) (e) (f)

Figure 12. Optimization results. In (a) the initial and final population of the optimization is shown. In (b) the simulated pulse response for a reasonably fast solution is depicted. Figure (c) shows an enlargement of the Pareto front where different regions were identified based on valve and cylinder selections, as shown in (d). The graphs (c) and (d) are obtained using an attribute based distance function, whereas (e) and (f) are the corresponding graphs obtained using the mixed distance function.

As can be seen from figure 12, there is a trade-off between system performance (control error) and system cost. By accepting a higher cost, better performance could be achieved. The most economical designs consist of small proportional valves and low-pressure cylinders. By choosing larger proportional valves and high-pressure cylinders, the performance could be increased at the expense of higher cost. If an even better performance is desired, a servo valve has to be chosen, which is more expensive but has better dynamics.

The continuous parameters, such as the control parameter, tend to smoothen out the Pareto front. For a given valve and cylinder, different settings on the continuous parameters affect the pulse response. A faster response results in a lower control but also a higher energy consumption and thereby higher cost. Therefore, there is a local trade-off between cost and performance for each catalog selection.

4 Discussion

In the proposed method, new solutions have to compete with the most similar individual before they are inserted into the population. Therefore, the similarity measure has a great influence on the optimization result. When using the attribute based distance function as a similarity measure, the true Pareto optimal front is identified, as shown in figure 12 (c) and (d). When using the mixed distance function, some dominated solutions survive, for example servo valves with low-pressure cylinders, see 12 (e) and (f). These solutions represent local optima, as they dominate the solutions close in parameter space.

The obtained results are in accordance with the results from the mathematical test functions. An attribute based distance function gives fast convergence to the Pareto optimal front, whereas a mixed distance function is a little slower in convergence but is capable of finding and maintaining multiple Pareto frontiers, see figure 9.

For an engineering problem, the optimization formulation is often a simplification of the real world problem, which in part requires human or inquantifiable judgment. When deciding upon the final design there are usually more criteria to consider than just the optimization objectives. Therefore, knowledge of the existence of local Pareto optimal solutions is very valuable. For example, aspects such as robustness, product portfolio, maintenance and quality might be important but hard to include in the optimization. A local Pareto optimal solution might therefore be preferred to a solution at the global Pareto optimal front. Hence, a method to identify and maintain local Pareto optimal solutions is valuable from an engineering perspective.

5 Conclusions

In this paper, a new multiobjective genetic algorithm was presented and applied to solve a mathematical test problem as well as a mixed variable design problem. The method is capable of finding and maintaining multiple Pareto optimal fronts with a minimum of problem-specific parameter settings. For the design problem, a hydraulic actuation system was studied with the help of a simulation program. The optimization parameters were divided into continuous parameters and discrete catalog selections. For the catalog selections, hierarchical catalogs of valves and cylinders were created using existing components. The optimization results in a set of Pareto optimal designs, elucidating the trade-off between system cost and system performance. Among the optimal solutions, distinct regions representing different catalog choices could be distinguished

In future work, comparisons between MOSGA and other multiobjective genetic algorithms should be performed. We will also develop methods to assess the robustness of individual solutions and the importance of different design parameters. Such methods would facilitate the use of multiobjective optimization in engineering design.

6 Acknowledgement

The software for this work used the GAlib genetic algorithm package, written by Matthew Wall at the Massachusetts Institute of Technology.

7 References

[1] Andersson J. and Wallace D., 2000, “Pareto optimization using the struggle genetic crowding algorithm,” Submitted for review to ASME Journal of Mechanical Design.

[2] Andersson J., Krus P. and Wallace D., 2000 “Multi-objective optimization of hydraulic actuation systems,” Proceedings of the ASME Design Automation Conferences, DETC2000/DAC-14512, Baltimore, MD, USA.

[3] Coello C., 1996, An empirical study of evolutionary techniques for multiobjective optimization in engineering design, Diss., Department of Computer Science, Tulane University.

[4] Deb K., 1999, “Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems,” Evolutionary Computation, vol. 7, pp. 205-230.

[5] Fonseca C. M. and Fleming P. J., 1998, “Multiobjective optimization and multiple constraint handling with evolutionary algorithms - Part I: a unified formulation,” IEEE Transactions on Systems, Man, & Cybernetics Part A: Systems & Humans, vol. 28, pp. 26-37.

[6] Goldberg D., 1989, “Genetic Algorithms in Search and Machine Learning.” Reading, Addison Wesley.

[7] Grueninger T. and Wallace D., 1996, “Multi-modal optimization using genetic algorithms,” MIT CADlab - Technical Report 96.02, http://cadlab.mit.edu/ publications/.

[8] Harik G., Finding multiple solutions in problems of bounded difficulty, IlliGAL Technical report No. 94002, http://www-illigal.ge.uiuc.edu/techreps.php3.

[9] Hopsan, 1991, “Hopsan, a simulation package - User's guide,” Technical report LiTH-IKP-R-704, Department of Mechanical Engineering, Linköping University, Linköping, Sweden.

[10] Horn J., 1997, “Multicriterion decision making,” in Handbook of evolutionary computation,” T. Bäck, D. Fogel, and Z. Michalewicz, Eds., IOP Publishing Ltd.

[11] Senin N., Wallace D., Borland N. and Jakiela M., 1999, “Distributed modeling and optimization of mixed variable design problems,” MIT CADlab - Technical Report: 99.01, http://cadlab.mit.edu/publications/.

[12] Zitzler E. and Thiele L., 1999, “Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach,” IEEE Transaction on evolutionary computation, vol. 3, pp. 257-271.

Paper VII

Metamodel Representations for Robustness Assessment in Multiobjective Optimization

Johan Andersson and Petter Krus

Intended for publication as:
ANDERSSON J., KRUS P., "Metamodel Representations for Robustness Assessment in Multiobjective Optimization," in *Proceedings of the 13th International Conference on Engineering Design ICED 2001*, Glasgow, UK, August 21-23, 2001

Metamodel Representations for Robustness Assessment in Multiobjective Optimization

Johan Andersson and Petter Krus

Department of Mechanical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Abstract

Engineering design problems are often characterized by the presence of several conflicting objectives. When using optimization to support engineering design, these objectives are usually aggregated to one overall objective function. Optimization is then conducted with one optimal design as the result. Another way of handling the problem of multiple objectives is to introduce the concept of Pareto optimality. The outcome from a Pareto optimization is a set of Pareto optimal solutions, which visualizes the trade-off between the objectives.

However, we want the final design to be not only optimal, but also robust. In this paper, a multiobjective genetic algorithm is applied to identify the Pareto optimal front. Metamodels are then introduced in order to assess the robustness of the Pareto optimal solutions. The metamodel is a second order polynomial that represents a response surface around the Pareto front. The coefficients of the metamodel are determined with the help of a recursive least squares (RLS) method. As the population of the genetic algorithm evolves, the RLS method extracts information about the shape of the Pareto front. By investigating the coefficients of the metamodel, the robustness of individual solutions was assessed.
.

1 Introduction

Most engineering design problems consist of several, often conflicting, objectives. Commonly the different objectives are aggregated to one overall objective function. Optimization is then conducted with one optimal design as the result. The result is strongly dependent on how the objectives are aggregated. To avoid this difficulty, the concept of Pareto optimality is employed. In his paper, a multiobjective genetic algorithm is used to identify the Pareto optimal front, which elucidates the trade-off between the conflicting objectives. The final design solution is then chosen from the set of Pareto optimal solutions. An important aspect in optimization is to assess the robustness of optimal solutions. Here, the use of metamodels is introduced to extract information and visualize the properties of different Pareto optimal solutions. By utilizing information from the evolving population of a genetic algorithm, a response surface representing the Pareto optimal front is obtained. The second order terms of such a response surface give a good estimate of the robustness of the Pareto front.

The paper starts by introducing a nomenclature for the multiobjective design problem. Thereafter, the proposed multiobjective genetic algorithm is presented. Then the concept of metamodels is introduced together with the Recursive Least Square (RLS) method, which estimates the coefficients of these models. The method is then applied to a benchmark problem, which illustrates the strengths of the proposed method.

2 Multiobjective optimization

A general multiobjective design problem could be expressed by equations (1) and (2).

$$\begin{gathered} \min \mathbf{F}(\mathbf{x}) = \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x}) \right)^T \\ s.t. \;\; \mathbf{x} \in S \end{gathered} \tag{1}$$

$$\mathbf{x} = \left(x_1, x_2, \ldots, x_n \right)^T \tag{2}$$

Here, $f_1(x), f_2(x), \ldots, f_k(x)$ are the k objective functions, $(x_1, x_2, \ldots, x_n)$ are the n optimization parameters, and $S \in R^n$ is the solution or parameter space. The obtainable objective vectors, $\{\mathbf{F}(\mathbf{x}) | x \in S\}$, are usually referred to as the attribute or objective space. The Pareto set consists of solutions that are not dominated by any other solutions. A solution **x** is said to dominate **y** if **x** is better or equal to **y** in all attributes, and strictly better in at least one attribute. Considering a minimization problem and two solution vectors **x**, **y**$\in S$. **x** is said to dominate **y**, denoted $\mathbf{x} \succ \mathbf{y}$, if:

$$\forall i \in \{1,2,\ldots,k\}: f_i(\mathbf{x}) \leq f_i(\mathbf{y}) \quad and \quad \exists j \in \{1,2,\ldots,k\}: f_j(\mathbf{x}) < f_j(\mathbf{y}) \tag{3}$$

The space in R^k formed by the objective vectors of Pareto optimal solutions is known as the Pareto optimal front.

2.1 Genetic Algorithms

Genetic algorithms are modeled after mechanisms of natural selection. Each optimization parameter (x_i) is encoded by a gene using an appropriate representation, such as a real number or a string of bits. The corresponding genes for all parameters $x_1,..x_n$ form a chromosome capable of describing an individual design solution. A set of chromosomes representing several individual design solutions comprises a population where the fittest are selected to reproduce. Mating is performed using crossover to combine genes from different parents to produce children. The children are inserted into the population and the procedure starts over again, thus creating an artificial Darwinian environment.

There are many different types of multiobjective genetic algorithms. Literature surveys and comparative studies on multiobjective genetic algorithms could be found in for example [7]. In the proposed method, the ranking scheme of the multiobjective GA (MOGA) presented by Fonseca and Fleming, [3] is employed. In MOGA, each individual is ranked according to its degree of dominance. The more population members that dominate an individual, the higher the ranking for the individual. Here an individual's ranking equals the number of individuals it is dominated by plus one. Hence, the individuals on the Pareto front would have the rank 1.

3 The proposed method

The multiobjective struggle genetic algorithm (MOSGA) [1] combines the struggle crowding genetic algorithm [7] with Pareto-based ranking as devised in [3]. In the struggle algorithm, two parents are chosen randomly from the population, and crossover/mutation is performed to create a child. The child then has to compete with the most similar individual in the entire population, and replaces it if the child has a better ranking. This replacement strategy counteracts genetic drift that can spoil population diversity.

The principle of the MOSGA algorithm is outlined below.

Step 1: Initialize the population.

Step 2: Select parents randomly from the population.

Step 3: Perform crossover and mutation to create a child.

Step 4: Calculate the rank of the child, and a new ranking of the population that considers the presence of the child.

Step 5: Find the most similar individual, and replace it with the new child if the child's ranking is better.

Step 6: Update the ranking of the population if the child has been inserted.

Step 7: Perform steps 2-6 until the mating pool is filled.

Step 8: If the stop criterion is not met go to step 2 and start a new generation.

The algorithm proved to perform well on a wide range of problems, where it successfully identifies and maintains multiple Pareto fronts, see [1].

3.1 Metamodel representations and response surface modeling

For the general design problem, simulation models and other CAE tools are usually employed to predict the properties of design proposals. Thus, the relation between design parameters and system characteristics is not exactly known. Metamodels are therefore introduced in order to assess the properties of optimal designs by creating a response surface at the Pareto optimal front. The metamodel is typically a second order polynomial describing a system characteristic in terms of the design parameters, see equation (4). In order to keep the number of parameters in the model low, the cross-product terms are omitted. Although this is simplification, the robustness could still be assessed. Had the cross-product terms not been omitted, the required number of calculations for the estimation would have been larger than the number of calculations needed for the actual optimization.

$$y = \theta_1 + \theta_2 x_1 + \theta_3 {x_1}^2 + \theta_4 x_2 + \theta_5 {x_2}^2 + \ldots. \tag{4}$$

In order to estimate the coefficients, θ_i, a Recursive Least Square (RLS) scheme is employed. As the GA evolves a population of individuals, there are a large number of evaluations that could be utilized to estimate the model coefficients. The RLS method continuously estimates the coefficients of the metamodel and is ready to present the estimate when the optimization has converged. A similar approach is presented by Krus in [5], where the RLS method is applied together with the Complex optimization method to solve a single objective problem. The RLS method could be described according to equation (5) see for example Ljung [6].

$$\hat{\theta}(t) \quad = \quad \hat{\theta}(t-1) \quad + \quad L(t) \quad \left[\; y(t) \quad - \quad \varphi^T(t)\hat{\theta}(t-1)\right]$$

$$\begin{matrix}\text{new}\\ \text{estimate}\end{matrix} = \begin{matrix}\text{old}\\ \text{estimate}\end{matrix} + \begin{matrix}\text{correcting}\\ \text{vector}\end{matrix} \begin{bmatrix}\text{new} & - & \text{Estimated y}\\ \text{y-value} & & \text{using old } \hat{\theta}\end{bmatrix}$$, where

$$L(t) = \frac{P(t-1)\varphi(t)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)} \quad \text{and} \tag{5}$$

$$P(t) = \frac{1}{\lambda(t)}\left[P(t-1) - L(t)\varphi^T(t)P(t-1)\right]$$

λ is the forgetting factor and $\varphi^T = \begin{bmatrix} 1 & x_1 & x_1^2 & x_2 & x_2^2 & .... \end{bmatrix}$ is the data vector. To use the recursive algorithm, initial values for the start-up are required. Here we use $\hat{\theta}(0) = 0$ and $P(0) = \alpha\,\mathrm{I}$, where α is a large number. If the design parameters are of different magnitudes, the value of α has to be scaled, so that $P(0) = diag(\alpha_1, ..., \alpha_{2n+1})$.

The MOSGA has been modified to incorporate the RLS estimation. For each genome the current estimate, $\hat{\theta}(t)$, and the $P(t)$ matrix are stored. The RLS estimation is then added to **Step 5** of the algorithm, so that the child is used to update the estimate of the most similar individual. If the child has a better ranking than the old individual, the child replaces it. $\hat{\theta}(t)$ and $P(t)$ are then transferred to the child. However, we have to make sure that the two individuals are similar enough for the estimate to improve. Different techniques could be employed to assure this. Here we simply state that the two individuals have to be within a certain distance $d_{\max}$ from each other. There is a trade-off between the accuracy of the estimation and the number of optimization runs required that has to considered when determining $d_{\max}$.

4 Mathematical test function

In order to assess the performance of the algorithm, a test problem from Deb [2] was explored. Deb developed a set of problems to highlight difficulties that multiobjective genetic algorithms may encounter. For visualization reasons, the focus is on two-dimensional problems, defined generally by equations (6) and (7).

$$f_1(x_1, x_2) = x_1 \tag{6}$$

$$f_2(x_1, x_2) = \frac{g(x_2)}{x_1}, \quad g(x_2) > 0, \quad x_1 > 0 \tag{7}$$

If the function *g* is multi-modal, the corresponding multiobjective problem will have global and local Pareto optimal frontiers, see Figure 3. A multi-modal *g* function is defined in equation (8).

$$g(x_2) = 2 - \exp\left\{ -\left(\frac{x_2 - 0.2}{0.08} \right)^2 \right\} - 0.8 \exp\left\{ -\left(\frac{x_2 - 0.6}{0.4} \right)^2 \right\} \tag{8}$$

Figure 1 shows the *g* function for $0 \leq x_2 \leq 1$ with the global optimum located at $x_2=0.2$ and a local optimum at $x_2=0.6$. As the global optimum is narrower than the local one is, one can argue that optimality could be traded for robustness by choosing a design on the local Pareto front. In Figure 2 $f_2(x_1, x_2)$ is shown for the entire solution space, i.e. $0.1 < x_1 < 1,\ 0 < x_2 < 1$.

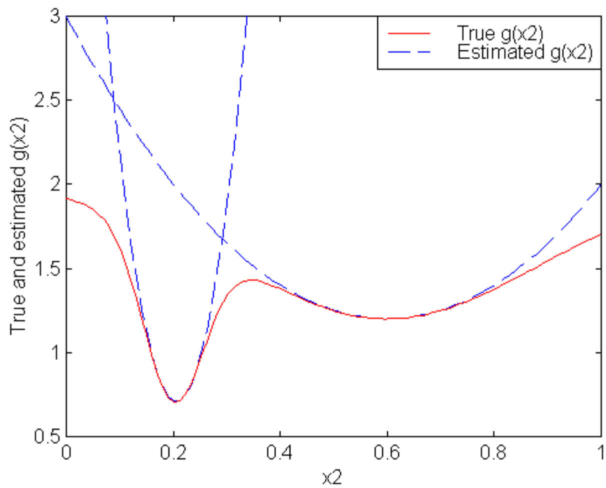


Figure 1. A multimodal g-function (solid line) and the model estimation at the global and local optima (dashed lines).

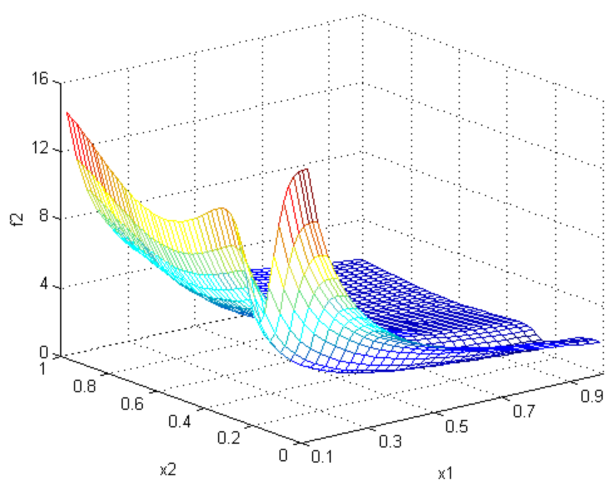
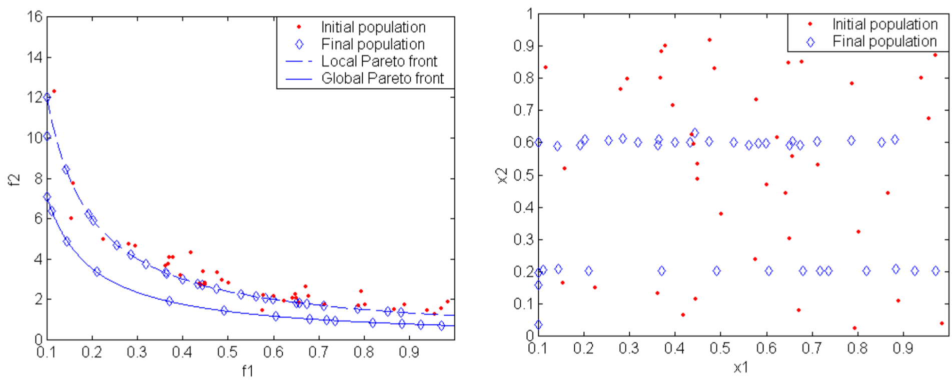


Figure 2. A plot of $f_2(x_1, x_2) = \dfrac{g(x_2)}{x_1}$.

The optimization was conducted with a population size of 40 individuals and ran for 400 generations. The optimization results are shown in Figure 3(a) and (b). The method identifies both the global and the local Pareto optimal fronts, and spreads the population evenly on them.



(a) (b)

Figure 3. (a) Shows the optimization result in attribute space, whereas (b) shows the result in parameter space. Note how the population is evenly spread on both the global and the local Pareto optimal fronts.

The first test of the RLS method was to estimate a second order polynomial to fit $f_2 \cdot x_1$, i.e. $g(x_2)$. The solutions on the global Pareto front had a $\hat{\theta}$ vector according to equation (9) and the solutions on the local front according to (10). These two functions are plotted in Figure 1, and it can be seen that the estimated functions match the true ones around each optimum.

$$g_{global}(x_2) = 7 - 62x_2 + 152x_2^{\ 2} \tag{9}$$

$$g_{local}(x_2) = 3 - 6x_2 + 5x_2^{\ 2} \tag{10}$$

Now consider the estimation of $f_2(x_1, x_2)$. As can be seen in Figure 2, the shape of the surface of f_2 is strongly dependent on the x_1, x_2 location. Hence, the $\hat{\theta}$-vector will show great variations for different optimal solutions. In Table 1 and 2 the estimated $\hat{\theta}$-vectors are shown together with the values of f_2, x_1 and x_2 for a set of points on both the global and the local Pareto front.

Table 1. Estimated $\hat{\theta}$-vectors at the global Pareto optimal front.

f_2	x_1	x_2	θ_1	θ_2	θ_3	θ_4	θ_5
6.40	0.11	0.20	69.27	-146.25	399.81	-503.28	1226.52
3.36	0.21	0.20	39.03	-46.68	72.63	-284.10	693.82
1.43	0.49	0.20	16.13	-6.75	3.87	-119.96	292.18
1.03	0.68	0.20	11.47	-3.67	1.58	-84.66	206.53
0.95	0.74	0.20	11.40	-4.65	2.28	-80.54	196.45
0.80	0.88	0.21	9.39	-2.86	1.11	-67.67	165.21
0.76	0.93	0.20	8.28	-1.58	0.41	-62.34	151.72

Table 2. Estimated $\hat{\theta}$-vectors at the local Pareto optimal front.

f_2	x_1	x_2	θ_1	θ_2	θ_3	θ_4	θ_5
8.45	0.14	0.59	37.73	-179.50	421.64	-40.96	34.13
4.71	0.25	0.61	21.01	-57.55	76.56	-22.06	18.41
3.31	0.36	0.59	14.94	-28.14	26.19	-16.24	13.52
2.53	0.47	0.60	11.25	-15.65	10.87	-12.49	10.41
2.06	0.58	0.60	9.27	-10.67	6.12	-10.24	8.54
1.83	0.66	0.60	8.24	-8.59	4.42	-8.91	7.43
1.41	0.85	0.60	6.33	-4.98	1.95	-6.97	5.81

Plots of these estimates show good agreement with the surface in Figure 2. Thus, the method seems capable of capturing the shape of complex functions as well. However, as the magnitude of the design parameters may vary, normalization of the estimates would be useful. The estimates are normalized and equation (4) is reformulated according to (11).

$$\Delta y = \tilde{\theta}_1 + \tilde{\theta}_2 \Delta x_1 + \tilde{\theta}_3 \Delta x_1^2 + \tilde{\theta}_4 \Delta x_2 + \tilde{\theta}_5 \Delta x_2^2 + ... \tag{11}$$

where $\tilde{\theta}_1 = \dfrac{\theta_1}{y_0}, \tilde{\theta}_2 = \dfrac{\theta_2 x_{10}}{y_0}, \tilde{\theta}_3 = \dfrac{\theta_3 x_{10}^2}{y_0}, ...$ and $\Delta x_1 = \dfrac{x_1}{x_{10}}, \Delta x_2 = \dfrac{x_2}{x_{20}}, ...$

Here, x_{10}, x_{20} ... are the values of the design parameters at each optimal point. The normalized formulation in equation (11) is very useful in order to answer questions of the type: What effect on y has a 10 percent increase in x_1? Normalized θ-values for the global and local fronts are shown in Table 3 and 4 respectively. As the identified mathematical function is the same at all points on the respective fronts, the normalized θ-values are also very similar.

Table 3. Global front normalized estimates

θ_1	θ_2	θ_3	θ_4	θ_5
10.82	-2.51	0.75	-16.09	8.03
11.61	-2.91	0.95	-17.05	8.40
11.24	-2.31	0.65	-16.90	8.31
11.11	-2.42	0.71	-16.62	8.22
11.94	-3.58	1.29	-17.16	8.51
11.79	-3.17	1.08	-17.60	8.90
10.90	-1.93	0.46	-16.70	8.27

Table 4. Local front normalized estimates

θ_1	θ_2	θ_3	θ_4	θ_5
4.46	-3.02	1.01	-4.85	1.40
4.46	-3.11	1.05	-4.68	1.44
4.51	-3.08	1.04	-4.90	1.44
4.45	-2.94	0.97	-4.94	1.50
4.50	-3.01	1.01	-4.97	1.48
4.51	-3.09	1.04	-4.88	1.48
4.50	-3.02	1.01	-4.96	1.50

Another useful reformulation is to express y as an orthogonal polynomial, so that

$$y = \widehat{\theta}_1 + \widehat{\theta}_2 x_1 + \widehat{\theta}_3 \left(x_1 - x_{10}\right)^2 + \widehat{\theta}_4 x_2 + \widehat{\theta}_5 \left(x_2 - x_{20}\right)^2 + ... \tag{12}$$

$x_{10}, x_{20}, ...$ are again the parameter values of the optimal point. $\widehat{\theta}_1, \widehat{\theta}_2, ...$ are determined by identification, so that $(4) \equiv (12)$. In this formulation the first order term indicates the slope of the surface at the optimal point and the second order term the curvature. The θ-values for the orthogonal polynomials are shown in Table 5 and 6 respectively. In this case, when the optimum is situated at $x_2 = 0.2$ and $x_2 = 0.6$ respectively, the corresponding first order term for x_2, θ_4, equals 0. This could be seen as an extra check to prove that the optimum is truly found. The first order x_1 term is negative and diminishing, which intuitively is in accordance with Figure 2 as the slope of the curve is negative and declining in the x_1 direction.

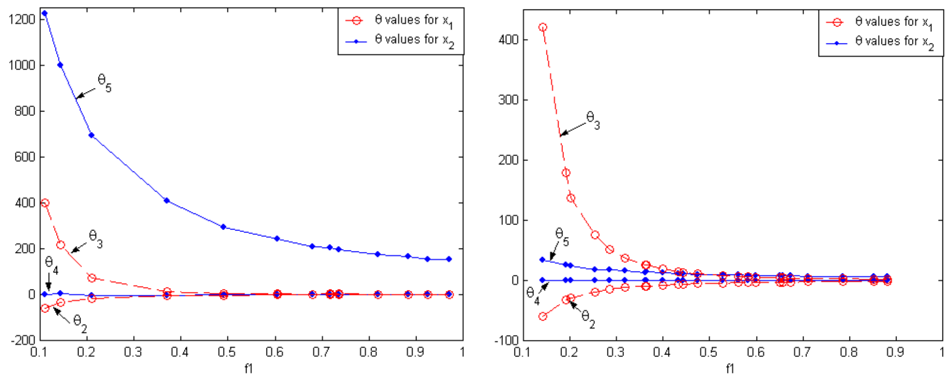
Table 5. Global front orthogonal polynomials

θ_1	θ_2	θ_3	θ_4	θ_5
13.08	-58.54	399.81	-1.21	1226.52
7.63	-16.26	72.63	-4.29	693.82
3.27	-2.96	3.87	-1.92	292.18
2.25	-1.52	1.58	-0.93	206.53
2.03	-1.30	2.28	-0.62	196.45
1.44	-0.91	1.11	0.75	165.21
1.65	-0.82	0.41	-0.63	151.72

Table 6. Local front orthogonal polynomials

θ_1	θ_2	θ_3	θ_4	θ_5
17.41	-59.74	421.64	-0.80	34.13
9.26	-18.56	76.56	0.29	18.41
6.74	-9.15	26.19	-0.18	13.52
5.02	-5.33	10.87	0.06	10.41
4.15	-3.54	6.12	-0.05	8.54
3.63	-2.65	3.77	-0.12	7.29
2.66	-1.54	1.75	0.10	5.67

Another way of visualizing the result is to plot the θ-values versus one of the objective functions. In Figure 4(a) and (b) the orthogonal θ-values for the global and local front are plotted against f_1. For each point at the global Pareto front in Figure 3, the corresponding set of θ-values is plotted in Figure 4(a), whereas the points in Figure 4(b) represent the corresponding θ-values on the local front. Both graphs show how the sensitivities are diminishing as we move against higher f_1 values.



(a) (b)

Figure 4. Sensitivity graphs showing the orthogonal $\hat{\theta}$ values for solutions on the global Pareto front in (a) and on the local Pareto front in (b).

5 Discussion

The sensitivity graphs clearly elucidate how the different design parameters influence the objective function at different points on the respective Pareto front. For the decision-maker, these types of graphs could be of great support in selecting the final design. One part of choosing the final design is to trade the objectives against one another by analyzing the Pareto front. Another part is to avoid non-robust solutions by studying the sensitivity graphs. As we now know which parameters have the greatest influence on the design, we can focus our efforts accordingly to ensure an optimal performance. Furthermore, by examining the sensitivities, profound insights into the properties of the optimization problem could be gained. For a complex design problem, the relations between the design parameters and different system characteristics are not exactly known, therefore the use of metamodels would extend our knowledge about the system.

A strength of the proposed method is that it is capable of identifying multiple Pareto frontiers in a single optimization run, something that no other methods are able to. For an engineering problem, the optimization formulation is often a simplification of the real world problem, which in part requires human or inquantifiable judgement. When deciding on the final design, there are usually more criteria to consider than just the optimization objectives, e.g. the robustness of the system. Therefore, knowledge of the existence of local Pareto optimal solutions and how they differ from the global ones is very valuable.

Not only the objectives, but also other system characteristics could be assessed with the help of metamodels. For example, the system could be optimized with respect to performance, and at the same time metamodels were established which represent other quantities such as cost and quality. Here the focus was on a multiobjective problem with two objectives. The method is of course also applicable to optimization problems with more objectives as well as problems with just one single objective.

6 Conclusions

This paper presents a method where the recursive least square method is applied to extract information from the evolving population of a multiobjective genetic algorithm. The method proved to be capable of establishing a metamodel representing a response surface around Pareto optimal solutions. By studying the identified coefficients, the robustness of individual solutions could be assessed. When analyzing the result it could be seen which design parameters contributed most to the objective functions at different locations on the Pareto front. When multiple Pareto fronts were identified, they could not only be evaluated based on objective function value, but also according to their robustness. There might be occasions where a local optimum is preferred to the global optimum because it is more robust.

The methods developed in the paper constitute a good support for multiobjective optimization in engineering design. In future work, authentic engineering design problems as well as problems with more design parameters will be studied. Methods will also be developed that facilitate presentation and analysis of the results.

7 References

[1] Andersson J., Krus P. and Wallace D., "Multi-objective optimization of hydraulic actuation systems," ASME Design Automation Conference, Baltimore, September 11-13, 2000.

[2] Deb K., 1999, "Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems," Evolutionary Computation, vol. 7, pp. 205-230.

[3] Fonseca C. and Fleming P., "Multiobjective optimization and multiple constraint handling with evolutionary algorithms - Part I: A unified formulation," IEEE Trans. on systems, man, and cybernetics, vol. 28, no. 1, pp. 26-37, 1998.

[4] Grueninger T. and Wallace D., 1996, "Multi-modal optimization using genetic algorithms", MIT CADlab - Technical Report 96.02, http://cadlab.mit.edu/ publications/.

[5] Krus P., "Post optimal system analysis using aggregated design impact matrix," ASME Design Automation Conference, Baltimore, September 11-13, 2000.

[6] Ljung L., "System Identification: Theory for the user, 2nd ed., Prentice Hall, 1999.

[7] Zitzler E. and Thiele L., 1999, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," IEEE Transaction on evolutionary computation, vol. 3, pp. 257-271.