

**Robust optimization for computationally expensive systems
With applications to integrated photonics**

Rehman, Samee

DOI

[10.4233/uuid:b963b8c4-49c9-446b-8128-358a301d12e3](https://doi.org/10.4233/uuid:b963b8c4-49c9-446b-8128-358a301d12e3)

Publication date

2016

Document Version

Final published version

Citation (APA)

Rehman, S. (2016). *Robust optimization for computationally expensive systems: With applications to integrated photonics*. [Dissertation (TU Delft), Delft University of Technology].
<https://doi.org/10.4233/uuid:b963b8c4-49c9-446b-8128-358a301d12e3>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

**ROBUST OPTIMIZATION FOR COMPUTATIONALLY
EXPENSIVE SYSTEMS**

WITH APPLICATIONS TO INTEGRATED PHOTONICS

ROBUST OPTIMIZATION FOR COMPUTATIONALLY EXPENSIVE SYSTEMS

WITH APPLICATIONS TO INTEGRATED PHOTONICS

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K. C. A. M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op donderdag 21 januari 2016 om 10:00 uur

door

Samee ur REHMAN

Master of Science in Electrical Engineering
Jacobs University Bremen, Germany
geboren te Lahore, Pakistan.

This dissertation has been approved by the promotor:

Prof. dr. ir. F. van Keulen

Copromotor: Dr. ir. M. Langelaar

Composition of the doctoral committee:

Rector Magnificus,

Prof. dr. ir. F. van Keulen, TU Delft, promotor

Dr. ir. M. Langelaar, TU Delft, copromotor

Independent members:

Prof. dr. ir. W. Bogaerts, Ghent University

Dr. L.F.P. Etman, TU Eindhoven

Prof. dr. ir. D. den Hertog, Tilburg University

Prof. dr. V.V. Toropov, Queen Mary University of London

Prof. dr. H.P. Urbach, TNW, TU Delft

Prof. dr. G.C.A.M. Janssen, 3ME, TU Delft, reserve member



Keywords: Robust optimization, metamodel, Kriging, efficient global optimization, expected improvement, system optimization, multidisciplinary design optimization, integrated photonics, ring resonators.

Front & Back: The three-dimensional plot on the left shows a convex-concave problem whose saddle point (white dot) is the robust optimum. The illustration on the right shows a vortex which abstractly represents light propagation in an integrated optical circuit. The picture on the back shows a floral array of optical fibers.

This research is supported by the Dutch Technology Foundation STW, which is part of the Netherlands Organisation for Scientific Research (NWO) and partly funded by the Ministry of Economic Affairs (project number 11363)

Copyright © 2015 by S.U. Rehman

ISBN 978-94-6186-595-3

An electronic version of this dissertation is available at

<http://repository.tudelft.nl/>.

CONTENTS

1	Introduction	1
1.1	Motivation	1
1.2	Optimization under uncertainty	4
1.3	Device level approach.	5
1.4	System level approach	6
1.5	Thesis structure.	8
1.5.1	Component level robust optimization	8
1.5.2	System level deterministic and robust optimization	10
1.5.3	Integrated photonics.	11
	References	11
I	Computational Methods - Component level robust optimization	15
2	Robust optimization under implementation error	17
2.1	Introduction	17
2.2	Robust optimization under implementation error	19
2.3	Kriging and Efficient Global Optimization (EGO)	19
2.3.1	Kriging.	19
2.3.2	Efficient Global Optimization	20
2.4	Robust optimization using expected improvement	22
2.4.1	Concept and central idea	22
2.4.2	Algorithm	22
2.4.3	Discussion	25
2.5	Numerical performance evaluation	26
2.5.1	Formulation of test problems	26
2.5.2	One-dimensional problem.	26
2.5.3	Three-dimensional problem	27
2.5.4	Eight-dimensional and ten-dimensional problems	28
2.5.5	Comparison with other techniques	28
2.6	Results	29
2.6.1	One-dimensional problem.	29
2.6.2	Three-dimensional non-convex problem	29
2.6.3	Eight-dimensional and ten-dimensional problems	33
2.6.4	Comparison with available techniques.	34
2.7	Conclusion	35
	References	35

3	Robust optimization under parametric uncertainties	39
3.1	Introduction	39
3.2	Robust optimization of unconstrained problems affected by parametric uncertainties	41
3.3	Kriging and Efficient Global Optimization	43
3.3.1	Kriging	43
3.3.2	Efficient Global Optimization	44
3.4	Efficient global robust optimization of parametric uncertainties affected problems	45
3.4.1	Main concept	45
3.4.2	Algorithm	46
3.4.3	Optimal sampling location in \mathbb{X}_c	47
3.4.4	Optimal sampling location in \mathbb{X}_e	48
3.4.5	Algorithm choices and rationale	50
3.5	Results	51
3.5.1	Test problems and evaluation methodology	51
3.5.2	Illustrative example	52
3.5.3	Numerical performance evaluation	55
3.5.4	Comparison with other approaches	57
3.5.5	Engineering case study: Robust optimization of an optical filter	58
3.6	Conclusion	61
	References	64
4	Robust optimization of unconstrained problems	67
4.1	Introduction	67
4.2	Robust optimization of problems involving parametric uncertainties and implementation error	69
4.3	Robust optimization of unconstrained expensive simulation-based problems	70
4.3.1	Preliminary - Kriging and EGO	70
4.3.2	Problems affected by implementation error only	72
4.3.3	Problems affected by parametric uncertainties only	72
4.3.4	Numerical case study: Treating implementation error problems as parametric uncertainty problems	73
4.4	Efficient global robust optimization under parametric uncertainties and implementation error	75
4.4.1	Identifying control variable infill location \mathbf{x}_c^{new}	77
4.4.2	Identifying environment variable infill location \mathbf{x}_e^{new}	78
4.4.3	Discussion	79
4.5	Results	80
4.5.1	Test problems	80
4.5.2	Algorithm performance evaluation	81
4.5.3	Engineering problem: Robust optimization of an optical filter	83
4.6	Conclusion	86
	References	88

5	Robust optimization of constrained problems	91
5.1	Introduction	91
5.2	Robust optimization of problems with constraints	93
5.3	Kriging-based deterministic optimization of constrained problems.	95
5.3.1	Kriging.	95
5.3.2	Deterministic unconstrained optimization	95
5.3.3	Deterministic constrained optimization	96
5.4	Efficient global robust optimization of constrained problems.	97
5.4.1	Optimal sampling location in \mathbb{X}_c	100
5.4.2	Optimal sampling location in \mathbb{X}_e	101
5.4.3	Implementation aspects	103
5.5	Results	104
5.5.1	Testing methodology.	104
5.5.2	Numerical performance evaluation	105
5.5.3	Engineering case study: Robust optimization of an optical filter	111
5.6	Conclusion	114
	References	116
II	Computational Methods - System level deterministic and robust optimization	119
6	Deterministic optimization of systems with independent components	121
6.1	Introduction	121
6.2	Optimization of systems with independent components	123
6.3	Infill sampling criterion for system level optimization	124
6.3.1	Kriging and Efficient Global Optimization	124
6.3.2	System level EI criterion	126
6.3.3	Constrained optimization of systems	130
6.4	Results	130
6.4.1	Numerical performance evaluation	130
6.4.2	Engineering problem: Deterministic optimization of an optical filter	138
6.5	Conclusion	142
	References	143
7	Robust optimization of systems with independent components	145
7.1	Introduction	145
7.2	Robust optimization of systems with independent components	147
7.3	Background.	149
7.3.1	Kriging.	149
7.3.2	Component level Efficient Global Optimization	150
7.3.3	Component level Efficient Global Robust Optimization	151
7.3.4	Bilevel Efficient Global Optimization (BEGO)	152
7.4	Bilevel Efficient Global Robust Optimization (BEGRO)	153
7.4.1	Estimation of \mathbf{x}_d^{new}	153
7.4.2	Estimation of \mathbf{x}_e^{new}	155
7.4.3	Infill sampling criteria for constrained systems	156

7.5	Results	158
7.5.1	Numerical test problem	158
7.5.2	Engineering problem: Robust optimization of an optical filter	160
7.6	Conclusion	166
	References	167
III	Integrated photonics	171
8	Robust optimization for integrated photonic systems	173
8.1	Introduction	173
8.2	Application: Serial ring resonators	175
8.3	System robust optimization.	177
8.4	Adaptive improvement of approximate system	178
8.4.1	Component metamodels: Kriging	178
8.4.2	System level robust expected improvement	178
8.5	Results	180
8.5.1	Second order serial ring resonator	181
8.5.2	Third order serial ring resonator	182
8.6	Conclusion	185
	References	185
9	Conclusion and future outlook	189
9.1	Recap	189
9.2	Conclusion	191
9.3	Future outlook	192
	References	194
	Summary	197
	Samenvatting	199
	Acknowledgements	201
	List of Publications	203
	Curriculum Vitae	205

1

INTRODUCTION

1.1. MOTIVATION

Improving engineering components and systems such that they perform optimally is an established research subject with a rich background and vast amount of literature [1, 2]. Within this realm, the design of engineering problems such that they perform as well as possible even when the manufacturing process is imperfect has also received considerable attention [3]. This particular subject is of great importance due to the relevance of Design for Manufacturing strategies for many different types of engineering problems affected by uncertainties.

Integrated photonics is one of the engineering disciplines that can greatly benefit from improved approaches and methods for Design for Manufacturing [4, 5]. Integrated photonic circuits enable the propagation and manipulation of light inside a higher refractive index waveguide that is surrounded by relatively lower refractive index material(s). Different types of devices and systems such as optical filters [6], multiplexers and interferometers [7] can be integrated into relatively compact spaces using this technology. The primary applications of integrated photonics include optical communication and bio-sensing amongst others. Integrated photonic devices and systems are prone to manufacturing uncertainties due to the extremely high precision required for fabricating devices having a particular geometry and material properties.

Improving fabrication facilities and investing in higher precision equipment for manufacturing may mitigate the uncertainties involved but this would typically come at high cost. Furthermore, some variations and uncertainties will always remain. Therefore, unless the design process takes into account the effect of these variations, economically feasible high volume production of integrated photonic devices and systems is expected to be difficult to achieve. On the other hand, if design engineers would have the means to account for the effects of the uncertainties involved in the fabrication process, not only could this lead to higher yield but this could also leap forward the field towards standardization [8].

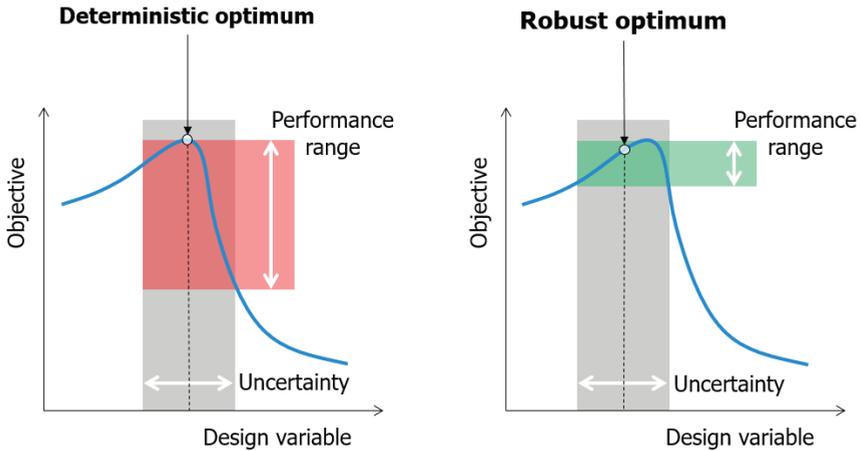


Figure 1.1: Illustration of a deterministic and a robust design for a constructed one-dimensional example problem.

In order to design higher yield devices and systems, the design framework needs to depart from the traditional approach of trial-and-error based design and fabrication. Instead, methods and algorithms need to be developed that not only incorporate the uncertainty in the process but also guarantee a certain minimum performance level. In addition, the methods need to be efficient in terms of providing a robust design, even if the underlying simulation involves high computational and time costs.

Figure 1.1 shows a comparison of a deterministic and robust optimal design for a one-dimensional problem affected by uncertainties in the design variable. The deterministic solution is found without taking the uncertainty in the design variable into account. On the other hand, to find the robust design the uncertainty is explicitly included in the optimization definition. It can be observed that the robust design is nominally suboptimal but has a relatively much smaller performance variation range within the uncertainty set.

Many engineering and non-engineering problems are simultaneously affected by uncertainties and require expensive simulations in order to provide high fidelity output [9]. The field of integrated photonics forms a typical example of disciplines that can benefit from an efficient approach for robust optimization. Methods that are implemented in such a way that they can be applied across disciplines and across engineering and non-engineering fields could be the most attractive options due to their generality. In order to maintain this general definition, robust optimization has to be performed on a black-box problem.

Even though robust optimization has been an active field of research in the recent past, the focus has been limited to solving convex problems, for example, linear, convex-quadratic or semi-definite problems [10]. On the other hand, robust optimization of computationally expensive problems, for which no structure can be assumed, has not received as much attention [11–13]. Many practical problems are neither convex nor cheap to simulate. Therefore, there is a strong need for effective strategies to be devel-

oped that can tackle such problems.

In this work, we strive to develop generic, scalable and efficient methods for robust optimization of black-box devices and systems that are based on expensive simulations [14]. The primary application of the techniques is the robust optimization of integrated photonic components and systems. However, the developed computational approaches can be applied to many other problems under uncertainty as well.

Even though the computational power of the state-of-the-art computing machines is improving with time, many engineering problems cannot be simulated reasonably well in a short period of time. Computational facilities are constantly improving in order to address this shortfall. However, even while considering the presence of super computers, large cluster resources, parallelization and ever improving GPU processing, there remains a gap between the pace at which the complexity of the engineering problem is increasing and the corresponding ability of computing resources to simulate these complex problems in a reasonably short time. The issue is exacerbated by the fact that, as computers become faster, users tend to simulate even more demanding problems.

Global, or even local, optimization of expensive simulations often requires many evaluations of the design with different design parameters and disparate settings. Applying optimization directly on the simulation is often prohibitively expensive. When simulating a single design requires, for example, one hour of computation, then serially evaluating the design at only 4 locations per dimension for a problem with 10 dimensions would require $4^{10} = 1.05$ million hours of computation. Such an extensive timeline for obtaining a solution is not practical.

Different approaches can potentially be taken to efficiently optimize computationally expensive problems [15]. Cheaper computational models can be developed. These simulations could be based on physical models that have, for instance, a smaller mesh size. Or a combination of cheap and more expensive physical models could be employed [16]. Alternatively, a completely non-physical model that is based purely on mathematical equations could also be used [17]. The ultimate goal is to replace the expensive simulation with these cheaper models so that the optimization process could become more efficient.

This replacement of the expensive simulation with a cheaper version, however, brings with it a plethora of difficulties. The fidelity of the reference simulation is usually much higher than it is for the cheap model [18]. This basically means that any optimal result found on the cheap model can often not be trusted. Instead of reducing the amount of error in the problem definition, in effect, by using cheap models, we increase the number of parameters with respect to which the problem must be robust. In such a scenario, methods have to be devised that can adaptively improve a cheap model in regions that are relevant for robust optimization. The ultimate aim is to employ relatively few iterative improvements to obtain a solution that matches the robust optimum that would have been found on the expensive simulation. The efficiency of such an iterative optimization strategy heavily depends on the location at which the expensive simulation is adaptively sampled. For this purpose, **infill sampling criteria** have to be developed that enable the robust solution to be found using only a few expensive simulations. Equivalent strategies exist for adaptive sampling of expensive simulations in order to reach a deterministic optimum [19]. However, extension of these approaches to non-deterministic

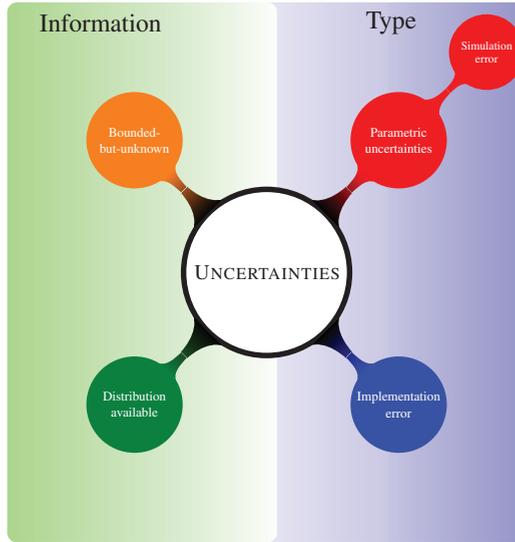


Figure 1.2: Classification of uncertainties based on their type and the information available concerning the uncertainty set.

problems requires further attention.

An important aspect of engineering problems is that they can often be decomposed into several components [2]. Some of these components may be expensive to simulate while others could be orders of magnitude cheaper. For these engineering problems, it often makes sense to treat the problem at system level. Cheaper models can then be constructed for the expensive parts of the problem, while the original complexity can be retained for other parts that are cheap to compute [20]. Such a system level response can often have higher fidelity. However, the transformation from component to system level can complicate the derivation of a reasonable criterion for adaptively sampling the component metamodels in regions of interest for system optimization [21].

1.2. OPTIMIZATION UNDER UNCERTAINTY

For devices and systems under uncertainty, robustness can be included in several ways. The different approaches are largely governed by the amount of information that is available concerning the uncertainties involved in the problem. Figure 1.2 classifies uncertainties in terms of their type and information availability. Broadly, uncertainties may be categorized as either parametric uncertainties or implementation errors. Parametric uncertainties are uncertainties that impact the parameters of the problem [22]. In the context of this work, simulation error is treated as a subset of parametric uncertainties. Implementation errors are variations that directly have an effect on the design variables of the problem [14]. In general, implementation error can be written in terms of parameter uncertainties. However, rewriting the problem in such a way increases the dimensions of the problem. This detrimentally impacts the efficiency of a metamodel based

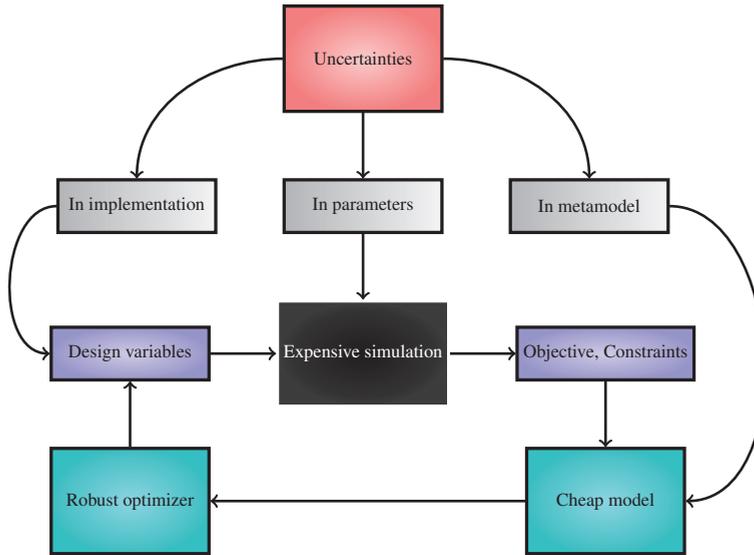


Figure 1.3: Generic illustration of methodology employed for optimization under uncertainty of expensive to evaluate device level problems.

approach since more samples are then needed to obtain a high fidelity metamodel response.

As shown by Figure 1.2, the information about the uncertainties may be limited to the bounds of the uncertainty set or the complete probability distribution within the uncertainty set may be available. If the probability distribution of all uncertainties is completely known then different moments of the distribution can be used to find a reliable and robust design. Stochastic optimization [23] and Reliability Based Design Optimization (RBDO) [24, 25] are the fields that explore this aspect. However, often such detailed information about the engineering problems is either not available or is classified. In such situations, the extent of the information concerning the uncertainties may only be limited to the bounds of the respective uncertainty sets. Such uncertainties are sometimes referred to as bounded-but-unknown [26]. In this case, a robust optimum can often only be found by computing the so called best worst-case cost of the problem [27].

1.3. DEVICE LEVEL APPROACH

Figure 1.3 shows the approach taken in this work for robust optimization at device level. In addition to the two types of uncertainties, implementation error and parameter uncertainties, the uncertainty in the cheap model also has to be taken into account. This is due to the fact that we apply robust optimization on a cheap model instead of on the expensive simulation.

In this work, Kriging [27] is used as the mathematical modeling technique to replace the expensive simulation. Kriging has a statistical basis that enables the error in its in-

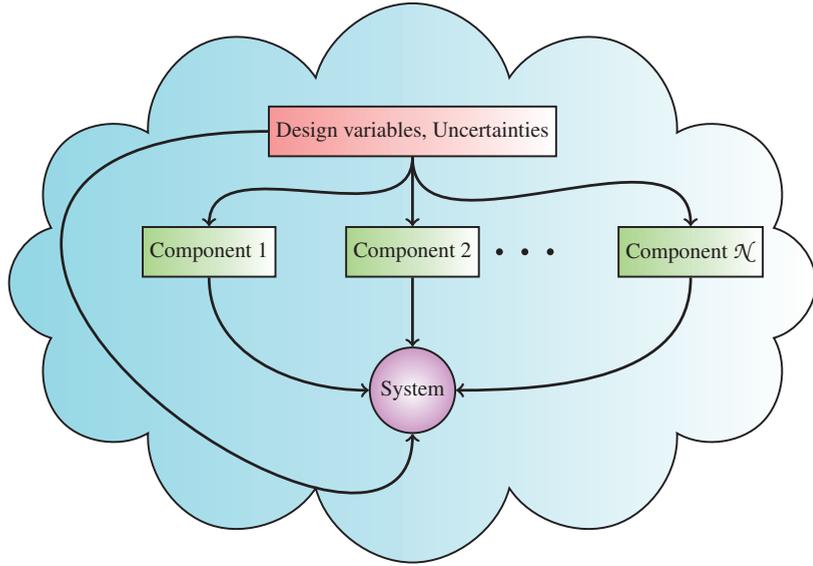


Figure 1.4: The bi-level problem, consisting of a set of components at one level and a system response at the next level, is shown here.

terpolation to be estimated. This interpolation error estimate is of primary importance in order to update and improve the cheap model in regions of interest. Jones *et al.* used the interpolation error estimator together with the Kriging prediction in order to derive several useful infill sampling criteria [28, 29]. Amongst these, the expected improvement criterion was the most sophisticated and powerful in terms of global optimization of unconstrained problems. Methods for handling constrained problems were first suggested by Schonlau [30]. The approach suggested by Schonlau involved computing the probability of feasibility of each constraint based on the constraint metamodel response and interpolation error. The method was recently further refined by Parr *et al.* [31, 32].

The expected improvement criterion is extended to different robust optimization settings at device level. Separate infill sampling criteria are proposed for unconstrained and constrained problems. Some infill sampling criteria are focused on problems affected by implementation error only or parametric uncertainties only, while others can handle problems affected by a combination of implementation error and parametric uncertainties. We emphasize here that the methods developed in this work are not tied to Kriging. In fact the developed techniques are applicable to any metamodeling approach that provides a local error estimate.

1.4. SYSTEM LEVEL APPROACH

At the system level, we consider robust optimization of systems with a hierarchical structure where components do not exchange any coupling variables. Figure 1.4 visualizes the bi-level problem. All the components of the system affect the system response but

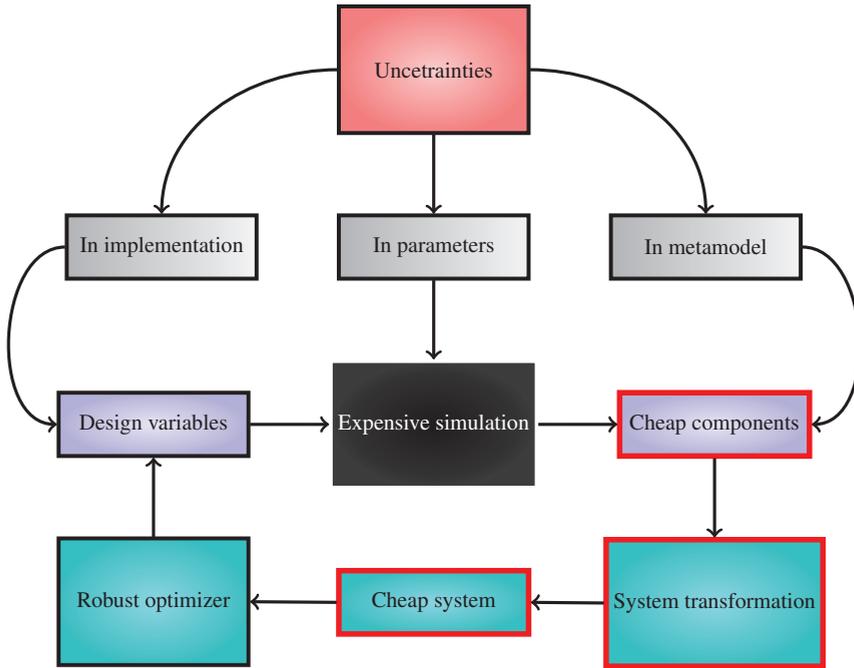


Figure 1.5: Generic illustration of methodology employed for optimization under uncertainty of system response based on expensive to evaluate components. The blocks with red boundary indicate the items that need to be included in order to apply system robust optimization instead of component robust optimization.

there is no interaction between components. The design variables and uncertainties can however be shared across components. As shown by Figure 1.4, some design variables and uncertainties may operate directly at system level. The setting shown in Figure 1.4 is only applicable to a subset of system level problems since we assume that the components are independent. We consider this particular class of problems since integrated photonic systems usually have such a hierarchical structure where the component responses can be computed independently and the system response is cheap. It is shown that an efficient approach can be derived for solving this type of system. For system level optimization of problems involving components that interact with one another, we refer readers to the extensive literature in the field of Multidisciplinary Design Optimization (MDO) [2, 33].

Figure 1.5 illustrates the methodology employed at system level to efficiently estimate the robust optimum. The system will ordinarily consist of several components. Metamodels are constructed for each expensive component. The metamodel error in each component has to be included in the uncertainties of the system problem. The component responses and component errors undergo a system transformation. The robust optimizer operates on the cheap system response.

Kriging is again used to construct the component metamodels. A system level error estimator is derived based on the component metamodels. A new infill sampling

criterion based on expected improvement is proposed for both system deterministic optimization and system robust optimization.

1.5. THESIS STRUCTURE

Figure 1.6 illustrates the structure of this thesis. This work is divided into three main parts. Part I and Part II concern computational methods at component and system level respectively. Part III is focused on the application of the methods on integrated photonic problems.

1.5.1. COMPONENT LEVEL ROBUST OPTIMIZATION

In **Chapter 2** robust optimization is applied on unconstrained problems that are affected by implementation error only. Kriging is used for the construction of the metamodel. A novel expected improvement criterion that is adapted to enable robust optimization instead of deterministic optimization is proposed. The special structure of the problem, where uncertainties reside in the same dimension as the design variables, is harnessed to devise the infill sampling criterion. The iterative optimization strategy is applied on several numerical problems for which the method shows consistent convergence.

Chapter 3 focuses on expensive to simulate unconstrained problems affected by parametric uncertainties. An efficient global robust optimization strategy is developed using the Kriging framework. The primary novelty of this work is the formulation of infill sampling criteria for the design variable space and the parametric uncertainty space. A separate criterion is needed for the parametric uncertainties since they belong to different dimensions of the design space than the design variables. The new sampling location is searched for in two stages, firstly, in the design variables space and, secondly, in the parametric uncertainties space. In the design variables space, the maximization of the expected improvement criterion provides the location that is likely to give the highest improvement over the best worst-case cost on the metamodel. On the other hand, maximization of the criterion in the parametric uncertainties space results in a new sampling location that is likely to give the greatest deterioration in the worst-case cost.

The algorithm is tested on several benchmark test problems and its performance is also compared with other techniques. In addition, robust optimization is performed on a TripleX [34] based ring resonator affected by geometrical variations. A robust Full Width at Half Maximum (FWHM) is found that deteriorates much less than the deterministic solution in the presence of uncertainties. The application of this particular chapter is more general compared to the implementation error method of Chapter 2, since all problems affected by bounded-but-unknown uncertainties can be written as problems affected by parametric uncertainties.

In **Chapter 4**, the scenario in which the expensive to simulate problem is affected by both uncertainty types, i.e. implementation error and parametric uncertainties, is addressed. The algorithms described in Chapter 2 and Chapter 3 are combined in order to address this problem. A novel infill sampling criterion is proposed that can explicitly take into account the two distinct uncertainty types. It is shown that making a distinction between implementation errors and parametric uncertainties enables the proposed algorithm to be more effective than other strategies for robust optimization of problems

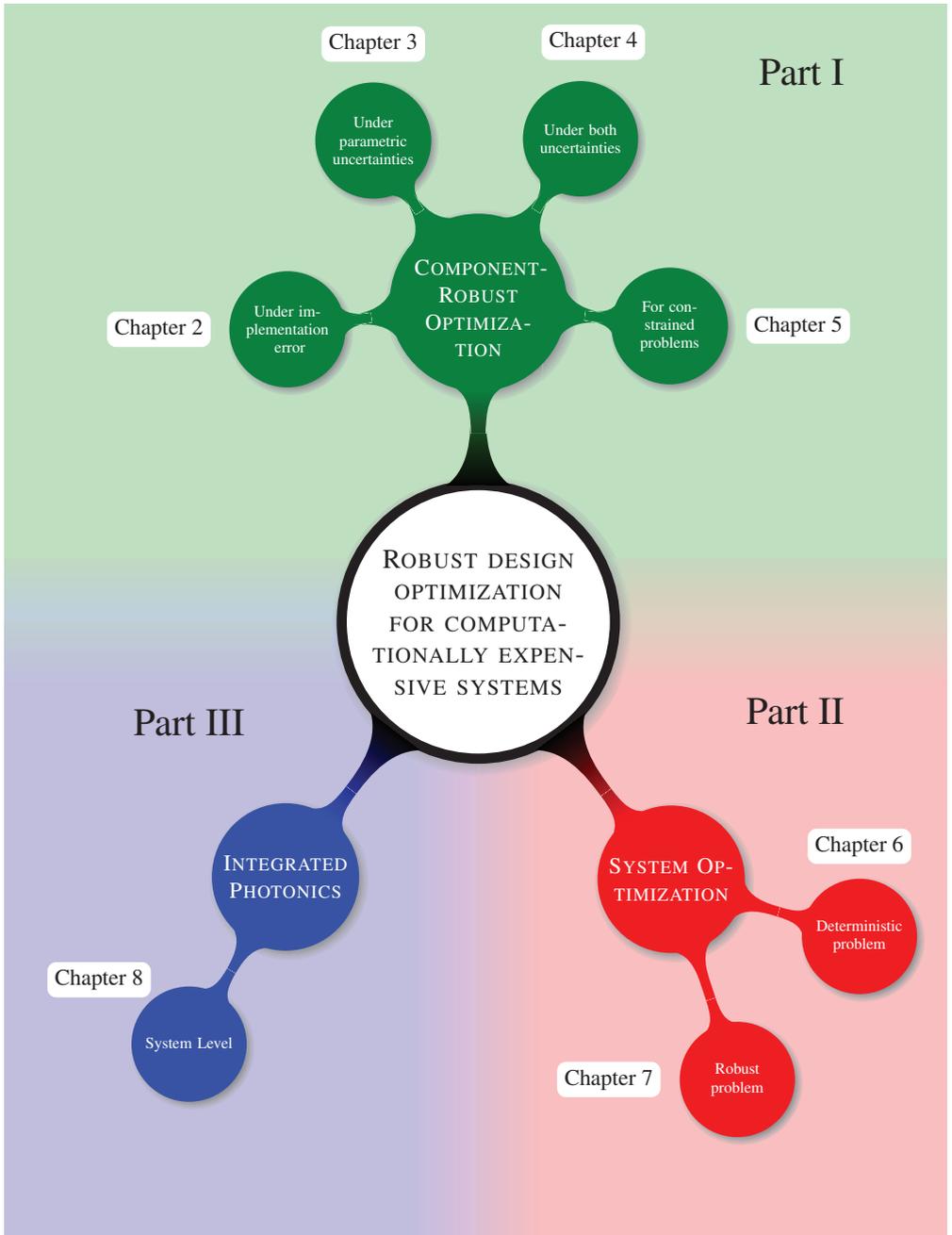


Figure 1.6: The structure of this thesis is visualized here. This work is divided into three main sections. In the first part, we discuss component level robust optimization. Part II deals with techniques for system level optimization. Part III is devoted to the application of the methods to integrated photonic systems.

affected by both uncertainties.

The performance of the approach is showcased on several numerical problems as well as on a TripleX based ring resonator. The waveguide width variation is considered as an implementation error since the width is used as a design variable. On the other hand, the uncertainty in the waveguide thickness remains a parametric uncertainty since the layer thickness cannot be a design variable. The effectiveness of the strategy is confirmed on the engineering example by showing that the robust optimum for the FWHM is determined more efficiently than for the example shown in Chapter 3.

Robust optimization of constrained problems is discussed in **Chapter 5**. Given that a majority of engineering problems are constrained, the algorithm proposed in this chapter is the most important for robust optimization of expensive to simulate devices. Since it is assumed that the constraints are also based on expensive simulations, Kriging metamodels are constructed for each constraint. The probability of feasibility criterion for constraints suggested by Schonlau *et al.* [30] is adapted to enable robust constrained optimization instead of nominal constrained optimization. The method is applied on five benchmark problems and on the TripleX single ring resonator example. Maximizing the FWHM still remains the objective. Constraints are placed on the minimum extinction ratio and the maximum insertion loss. The algorithm is shown to exhibit steady convergence towards the robust optimum on all test problems.

1.5.2. SYSTEM LEVEL DETERMINISTIC AND ROBUST OPTIMIZATION

Efficient global optimization at system level is discussed in **Chapter 6**. The method addresses optimization of systems consisting of expensive components that do not interact with one another. Uncertainties are not considered in the problem and only deterministic optimization is performed. Kriging metamodels are constructed for the expensive components. A system level error estimator is found based on a linear transformation of the component level metamodel errors. A system level expected improvement criterion is then proposed using the combination of the system response and system error estimate. Expensive to evaluate constraints are incorporated in the problem by deriving a system level probability of feasibility criterion. The algorithm is tested on several numerical problems and is found to outperform a space filling based metamodel construction and optimization strategy. TripleX based serial ring resonators are used as the engineering example. We choose a bandpass filter response at the through port as the objective and construct component metamodels for the directional coupler section. It is shown that the global optimum can efficiently be found for the serial ring resonators.

In **Chapter 7** we propose efficient infill sampling criteria for system level global robust optimization. The method builds on the algorithm developed in Chapter 6. The primary novelty is the induction of uncertainties in the problem. The system level expected improvement criterion found in Chapter 6 is updated to suggest locations that could lead to the global robust optimum instead of the deterministic solution. In addition, a system level expected deterioration criterion is developed for the parametric uncertainties domain. The probability of feasibility expression derived in Chapter 6 is also adapted to address problems with uncertain constraints. The algorithm is applied on the TripleX based serial ring resonator problem and the results are compared against the system deterministic optimum found in Chapter 6.

1.5.3. INTEGRATED PHOTONICS

System level robust design optimization of integrated photonic systems is discussed in **Chapter 8**. The algorithm developed in Chapter 7 is applied on TripleX based second and third order serial ring resonators in order to obtain a bandpass filter response. The ease of applicability and efficiency of the system level approach is emphasized in the context of integrated photonic problems. We show that the proposed system level robust optimization method is generic, scalable and efficient. Furthermore, emphasis is placed on the fact that the serial ring resonator example is merely used for demonstration of the developed robust optimization methodology and that the component level and system level algorithms proposed in this work can be applied to a varied set of problems within and outside integrated photonics.

The conclusions derived from this thesis are presented in **Chapter 9**. Suggestions for future work and possible improvements to the proposed techniques in this work are also discussed. We analyze the strengths and weaknesses of the methods and place them within the wider context of approaches developed for optimization under uncertainty at device and system level.

REFERENCES

- [1] S. Boyd and L. Vandenberghe, *Convex Optimization* (Cambridge University Press, New York, NY, USA, 2004).
- [2] J. R. R. A. Martins and A. B. Lambe, *Multidisciplinary Design Optimization: A Survey of Architectures*, *AIAA Journal* **51**, 2049 (2013).
- [3] H.-G. Beyer and B. Sendhoff, *Robust optimization – A comprehensive survey*, *Computer Methods in Applied Mechanics and Engineering* **196**, 3190 (2007).
- [4] G. Lifante, *Integrated photonics: fundamentals* (J. Wiley, 2003).
- [5] R. R. A. Syms and J. R. Cozens, *Optical guided waves and devices* (McGraw-Hill, 1992).
- [6] A. Melloni and M. Martinelli, *Synthesis of direct-coupled-resonators bandpass filters for WDM systems*, *Lightwave Technology, Journal of* **20**, 296 (2002).
- [7] N. S. Lagali, M. R. Paiam, R. I. MacDonald, K. Worhoff, and A. Driessen, *Analysis of generalized Mach-Zehnder interferometers for variable-ratio power splitting and optimized switching*, *Lightwave Technology, Journal of* **17**, 2542 (1999).
- [8] A. Melloni, G. Cusmai, and F. B. Morichetti, *Design on tolerances in integrated optics, in ECIO'08 Eindhoven - Proceedings of the 14th European Conference on Integrated Optics and Technical Exhibition, Contributed and Invited Papers* (2008) pp. 205–208.
- [9] R. Jin, X. Du, and W. Chen, *The use of metamodeling techniques for optimization under uncertainty*, *Structural and Multidisciplinary Optimization* **25**, 99 (2003).
- [10] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust optimization* (Princeton University Press, 2009).

- [11] D. Bertsimas, O. Nohadani, and K. M. Teo, *Nonconvex Robust Optimization for Problems with Constraints*, INFORMS Journal on Computing **22**, 44 (2010).
- [12] D. Bertsimas and O. Nohadani, *Robust optimization with simulated annealing*, Journal of Global Optimization **48**, 323 (2010).
- [13] J. Marzat, E. Walter, and H. Piet-Lahanier, *Worst-case global optimization of black-box functions through Kriging and relaxation*, Journal of Global Optimization , 1 (2012).
- [14] E. Stinstra and D. den Hertog, *Robust optimization using computer experiments*, European Journal of Operational Research **191**, 816 (2008).
- [15] A. Forrester, S. Andras, and A. J. Keane, *Engineering design via surrogate modelling : a practical guide* (J. Wiley, 2008).
- [16] J. W. Bandler, Q. S. Cheng, S. A. Dakroury, A. S. Mohamed, M. H. Bakr, K. Madsen, and J. Sondergaard, *Space mapping: the state of the art*, Microwave Theory and Techniques, IEEE Transactions on **52**, 337 (2004).
- [17] G. G. Wang and S. Shan, *Review of Metamodeling Techniques in Support of Engineering Design Optimization*, Journal of Mechanical Design **129**, 370 (2007).
- [18] D. W. Apley, J. Liu, and W. Chen, *Understanding the Effects of Model Uncertainty in Robust Design With Computer Experiments*, Journal of Mechanical Design **128**, 945 (2006).
- [19] A. I. J. Forrester and A. J. Keane, *Recent advances in surrogate-based optimization*, Progress in Aerospace Sciences **45**, 50 (2009).
- [20] F. A. C. Viana, T. W. Simpson, V. Balabanov, and V. Toropov, *Metamodeling in Multi-disciplinary Design Optimization: How Far Have We Really Come?* AIAA Journal **52**, 670 (2014).
- [21] M. Kokkolaras, Z. P. Mourelatos, and P. Y. Papalambros, *Design Optimization of Hierarchically Decomposed Multilevel Systems Under Uncertainty*, Journal of Mechanical Design **128**, 503 (2005).
- [22] D. Bertsimas, O. Nohadani, and K. M. Teo, *Nonconvex Robust Optimization for Problems with Constraints*, INFORMS Journal on Computing **22**, 44 (2010).
- [23] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming* (Springer Verlag, New York, NY, USA, 1997).
- [24] B. D. Youn and K. K. Choi, *A new response surface methodology for reliability-based design optimization*, Computers & Structures **82**, 241 (2004).
- [25] K. Deb, S. Gupta, D. Daum, J. Branke, A. K. Mall, and D. Padmanabhan, *Reliability-Based Optimization Using Evolutionary Algorithms*, Evolutionary Computation, IEEE Transactions on **13**, 1054 (2009).

- [26] S. P. Gurav, J. F. L. Goosen, and F. VanKeulen, *Bounded-But-Unknown uncertainty optimization using design sensitivities and parallel computing: Application to MEMS*, *Computers & Structures* **83**, 1134 (2005).
- [27] A. Ben-Tal and A. Nemirovski, *Selected topics in robust convex optimization*, *Mathematical Programming* **112**, 125 (2008).
- [28] D. R. Jones, *A Taxonomy of Global Optimization Methods Based on Response Surfaces*, *Journal of Global Optimization* **21**, 345 (2001).
- [29] D. R. Jones, M. Schonlau, and W. J. Welch, *Efficient Global Optimization of Expensive Black-Box Functions*, *Journal of Global Optimization* **13**, 455 (1998).
- [30] M. Schonlau, *Computer experiments and global optimization*, Ph.D. thesis, University of Waterloo (1997).
- [31] J. Parr, A. Forrester, A. J. Keane, and C. M. Holden, *Enhancing infill sampling criteria for surrogate-based constrained optimization*, *Journal of Computational Methods in Sciences and Engineering* **12**, 25 (2012).
- [32] J. M. Parr, A. J. Keane, A. I. Forrester, and C. M. Holden, *Infill sampling criteria for surrogate-based optimization with constraint handling*, *Engineering Optimization* **44**, 1147 (2012).
- [33] M. Li and S. Azarm, *Multiobjective Collaborative Robust Optimization With Interval Uncertainty and Interdisciplinary Uncertainty Propagation*, *Journal of Mechanical Design* **130**, 081402 (2008).
- [34] R. Heideman, M. Hoekman, and E. Schreuder, *TriPleX-Based Integrated Optical Ring Resonators for Lab-on-a-Chip and Environmental Detection*, *Selected Topics in Quantum Electronics*, *IEEE Journal of* **18**, 1583 (2012).

I

COMPUTATIONAL METHODS - COMPONENT LEVEL ROBUST OPTIMIZATION

2

ROBUST OPTIMIZATION UNDER IMPLEMENTATION ERROR

2.1. INTRODUCTION

Many engineering and structural problems involve uncertainties. If these uncertainties are not taken into account during optimization, undesirable phenomena such as high variations in performance are observed. Ben-Tal *et al.* [1] showed this by analyzing the performance of an antenna array when subjected to slight perturbation in the nominal data. The array was nominally optimized to attenuate sidelobes, but even small implementation errors in the design variables cause the radiation pattern of the antenna to worsen dramatically in the region of interest.

Applying optimization directly on an expensive to evaluate computer simulation is prohibitively costly. To avoid this problem, an approximate fast mathematical model of the simulation can be constructed and optimization can be applied on the cheaper model. There is a plethora of choices available in terms of building a cheap response surface of an expensive simulation. These include, but are not limited to, polynomial approximation, regression models as well as interpolation techniques such as radial basis functions, splines and Kriging.

The statistical framework of Kriging [2, 3] provides an estimator of the variance of the Kriging interpolator; this variance is useful for performing adaptive sampling. Using this potential error, different metrics have been proposed to efficiently find the deterministic optimum of unconstrained problems [4]. Amongst these metrics, the expected improvement measure and the Efficient Global Optimization (EGO) algorithm are adept at finding the nominal optimum of unconstrained problems using only a small number of expensive computer simulations [5, 6].

Efficient global optimization has been extensively applied on deterministic optimization problems [7]. However, the strategy has not been widely used to solve optimization problems that involve uncertainties. When the probability distributions of the uncertainties are not available, the resulting robust optimization problem is a nested min-max

optimization problem. Here the term nested optimization means that the objective to be optimized is itself a result of an optimization. Formally speaking, robust optimization is the minimization of the maximum possible realization of the objective with respect to the uncertainty set, subject to the non-violation of the worst-case constraints. In this work, we consider robust optimization of unconstrained problems. Given the importance of design under uncertainty, research in the field of robust optimization has seen a steady increase in the past decades [8]. Application examples include, but are not limited to, structural design [9], portfolio selection [10], electromagnetic scattering [11] and truss topology design [12].

With respect to the uncertainties involved in these problems, a clear distinction can be made between implementation errors and parametric uncertainties. Implementation errors are those uncertainties that directly affect the design variables. Parametric uncertainties, on the other hand, are defined as variations in the problem data or parameters.

Since robust optimization is a nested optimization problem, computing the robust optimum requires a much greater amount of computational resources than the resources needed to find the deterministic optimum. In this context, surrogate based optimization techniques have the potential to drastically reduce the computational budget required to solve such problems. In general, surrogates have often been used in conjunction with evolutionary algorithms to find the robust optimum of unconstrained problems [13, 14]. To the best of our knowledge, the use of efficient global optimization to find the robust optimum of unconstrained problems has previously only been explored by Marzat *et al.* [15]. These algorithms, however, were all directed towards problems affected by parametric uncertainties.

Implementation errors can also be treated as parametric uncertainties. However, there are certain inherent disadvantages in performing this conversion. When implementation errors are included as parametric uncertainties, then the total number of dimensions of the problem increases. Parametric uncertainties act independently of the design variables and therefore belong to separate dimensions. Moving from a nominal optimization problem to a problem affected by implementation errors, on the other hand, does not result in an increase in the total number of dimensions. This is because implementation errors reside in the same dimensions as the design variables. Therefore the pragmatic approach would be to treat uncertainties in the design variables as implementation errors. Based on this observation, we propose a modified adaptive Kriging-based approach to this class of problems, and demonstrate its effectiveness.

In this work, robust optimization using a modified expected improvement criterion is applied to unconstrained problems affected by implementation errors. Such unconstrained problems are often encountered in the field of integrated photonics [16], where device analysis involves expensive computer simulations. Designers of integrated photonic devices typically do not have access to the probability distribution of the uncertainties involved in the fabrication process. However, the bounds on the uncertainty set are known. Therefore, in this work, it is assumed that no probability distribution information is available concerning the uncertainty set and that the set is bounded-but-unknown. In order to find the robust optimum, we have to compute the best worst-case solution. This is evaluated by minimizing the maximum realizable value of the objective

with respect to the uncertainty set.

The cost of the proposed algorithm is compared against other techniques such as the method of Marzat *et al.* [15]. EGO was used by Marzat *et al.* for min-max optimization of problems with parametric uncertainties. The focus in this work, on the other hand, is on problems under implementation error. It is shown that, for problems affected by implementation error, the proposed technique is much more efficient as it exploits the particular structure of this class of problems. In addition, the algorithm is compared with the classical approach of using space-filling sampling [17, 18] and applying robust optimization on the resulting surrogate.

This chapter is organized as follows. Robust optimization of an unconstrained problem under implementation error is introduced in Section 2.2. We briefly discuss Kriging and efficient global optimization in Section 2.3. In Section 2.4 we introduce the proposed algorithm for the application of robust optimization to unconstrained problems affected by implementation errors. Section 2.5 contains description of the test cases on which the algorithm is applied while Section Bch2-sec:res and 2.7 consist of the results and conclusions, respectively.

2.2. ROBUST OPTIMIZATION UNDER IMPLEMENTATION ERROR

Deterministic optimization [19] of an unconstrained problem can be stated as

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad (2.1)$$

where $f(\mathbf{x}) \in \mathbb{R}$ is the objective function and $\mathbf{x} \in \mathbb{R}^n$ is the set of design variables. If the problem is affected by uncertainties the problem definition changes. Let us assume that due to an error in implementation $\Delta \in \mathcal{U}$, where \mathcal{U} is the uncertainty set, \mathbf{x} deviates from its nominal value to a new value $\mathbf{x} + \Delta$. Assuming that the information of the probability distribution of the set \mathcal{U} is not available, the robust optimum can only be found by computing the best worst-case solution. In other words, the maximum possible realization of the objective function $f(\mathbf{x} + \Delta)$ has to be minimized, instead of the nominal function $f(\mathbf{x})$. The problem is defined as

$$\min_{\mathbf{x}} g(\mathbf{x}) \quad (2.2)$$

where

$$g(\mathbf{x}) = \max_{\Delta \in \mathcal{U}} f(\mathbf{x} + \Delta). \quad (2.3)$$

2.3. KRIGING AND EFFICIENT GLOBAL OPTIMIZATION (EGO)

2.3.1. KRIGING

Kriging is an interpolation technique based on statistics. In this work, a brief overview of Kriging and efficient global optimization is provided, for a more in-depth study please refer to [3, 6]. There are several types of Kriging formulations, but we assume the simplest and most popular type; namely, Kriging with a constant mean and a Gaussian multiplicative correlation function.

Kriging assumes that the function response at a position \mathbf{x} can be described as a normally distributed random variable $Y(\mathbf{x})$ with mean μ and variance σ^2 . Furthermore, the

function is assumed to be smooth and continuous. A parameterized Gaussian correlation function is used to model the correlation between the responses at any two points \mathbf{x}_i and \mathbf{x}_j ,

$$\text{Corr}[Y(\mathbf{x}_i), Y(\mathbf{x}_j)] = \exp\left(-\sum_{q=1}^k \theta_q |x_{iq} - x_{jq}|^{p_q}\right). \quad (2.4)$$

k represents the total number of dimensions of the problem. θ_q is a measure of how influential the q^{th} dimension is with respect to the design domain. Higher values of θ_q denote greater influence. The parameter p_q regulates the relative smoothness of the function. Values of p_q close to 2 represent smooth, differentiable functions. Values of p_q close to 0 model non-smooth, non-differentiable functions. A constant value of 2 is assigned to p_q in this work.

The parameters θ_q , μ and σ^2 are estimated by choosing them such that the likelihood of the observed data is maximized. Once the unknown parameters have been determined, Kriging is used to predict how the function behaves in between the sample points. This is performed by maximizing the combined likelihood of the observed data and the predicted value. The Kriging prediction \hat{y} is defined as

$$\hat{y}(\mathbf{x}) = \hat{\mu} + \hat{\mathbf{r}}^T \hat{\mathbf{R}}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}) \quad (2.5)$$

where $\hat{\mu}$ is the maximum likelihood estimate for the mean μ . $\hat{\mathbf{R}}$ is the $N \times N$ correlation matrix between the N sample points, $\hat{\mathbf{r}}$ is the vector of correlations between the observed data and the new prediction, while \mathbf{y} is the observed response. Both the correlation vector $\hat{\mathbf{r}}$ and the correlation matrix $\hat{\mathbf{R}}$ are computed using Equation (2.4).

The statistical basis of Kriging gives an estimate of the error in the predicted response. This mean squared error (MSE) is given by

$$s^2(\mathbf{x}) = \hat{\sigma}^2 \left[\mathbf{1} - \hat{\mathbf{r}}^T \hat{\mathbf{R}}^{-1} \mathbf{r} + \frac{\mathbf{1} - \mathbf{1}^T \hat{\mathbf{R}}^{-1} \hat{\mathbf{r}}}{\mathbf{1}^T \hat{\mathbf{R}}^{-1} \mathbf{1}} \right]. \quad (2.6)$$

where $\hat{\sigma}^2$ is the maximum likelihood estimate for the variance σ^2 . It should be noted that the mean squared error, given by Equation (2.6), is only an approximate since we ignore the fact that the Kriging parameters are estimated [20]. The MSE is zero at the sample points since the true response of the function is known at these locations.

2.3.2. EFFICIENT GLOBAL OPTIMIZATION

The adaptive sampling strategy based on Expected Improvement (EI) in EGO can be used to estimate the global minimum of a deterministic unconstrained problem. The EI metric is constructed by assuming that the uncertainty in the Kriging prediction at any position \mathbf{x} can be described in terms of a normally distributed random variable $Y(\mathbf{x})$. The mean of $Y(\mathbf{x})$ is given by the predicted value $\hat{y}(\mathbf{x})$ while the variance is given by the MSE $s^2(\mathbf{x})$. Let y_{\min} represent the minimum objective value in the sample data. The next sampling point should be added at the location most likely to lead to the highest improvement over the current minimum y_{\min} . The improvement I over y_{\min} is defined as $I = \max(y_{\min} - Y, 0)$. The expected value of this improvement I determines the amount of improvement that can be expected. The expected improvement is thus defined as

$$E[I(\mathbf{x})] = (y_{\min} - \hat{y})\Phi\left(\frac{y_{\min} - \hat{y}}{s}\right) + s\phi\left(\frac{y_{\min} - \hat{y}}{s}\right) \quad (2.7)$$

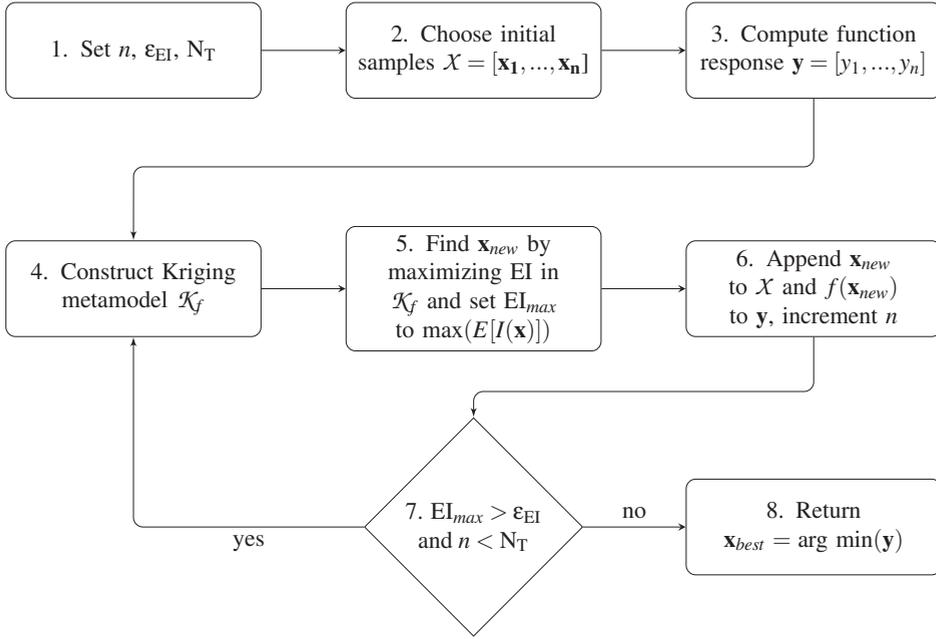


Figure 2.1: Flowchart shows the Efficient Global Optimization (EGO) algorithm [6]. EGO uses expected improvement to find the deterministic optimum of an unconstrained problem using relatively few evaluations of an expensive computer simulation.

where $\Phi(\cdot)$ is the normal cumulative distribution function and $\phi(\cdot)$ is the normal probability density function. The full derivation of EI can be found in [21]. Using a global optimization algorithm, we can find the location at which the expected improvement is maximum in the whole design domain. Since this is the location most likely to lead to the highest improvement over the current minimum y_{\min} , a new sampling point is added at this location.

The flowchart in Figure 2.1 shows how expected improvement is used to find the global nominal optimum of an unconstrained problem using relatively few samples. The algorithm, known as Efficient Global Optimization (EGO), is initialized with n samples, through a space-filling strategy, e.g., Latin hypercube sampling (LHS) [22]. At each iteration the Kriging metamodel of the objective, represented by \mathcal{K}_f , is constructed. Thereafter, the expected improvement metric is applied to the design domain. A new sample is added at the location where the maximum EI is found. The algorithm terminates when the total number of samples N_T is exhausted or when EI_{max} falls below the threshold ϵ_{EI} . At this stage, \mathbf{x}_{best} , the argument that yields the minimum value amongst the sampled data is returned.

2.4. ROBUST OPTIMIZATION USING EXPECTED IMPROVEMENT

2.4.1. CONCEPT AND CENTRAL IDEA

Thus far, we have described the algorithm for efficient global optimization that is used to find the nominal optimum of an unconstrained problem. In the following discussion, a novel algorithm is proposed for finding the robust optimum of an unconstrained problem. We demonstrate how a modified version of efficient global optimization can be used to find this best worst-case cost.

The algorithm begins with the construction of a metamodel based on an initial set of samples and responses. First, a reference robust optimum is computed on the metamodel. A modified EI criterion is then applied. For each location at which the modified EI needs to be computed, a worst-case Kriging prediction, with respect to the uncertainty set, is evaluated on the metamodel. We sample the location with the highest expectation of improvement for the worst-case Kriging prediction over the reference robust optimum. At the following iterations, the process of finding the reference robust optimum, evaluating the worst-case Kriging prediction on the metamodel and applying EI is repeated until convergence.

2.4.2. ALGORITHM

This section is devoted to a detailed explanation of the proposed algorithm. The method will from now on be referred to as Efficient Global Robust Optimization under Implementation Error (EGRO-IE).

The flowchart in Figure 2.2 illustrates the steps EGRO-IE follows in order to find the global robust optimum of an unconstrained problem affected by implementation error. Apart from three steps, highlighted by a thicker boundary for the flowchart boxes in Figure 2.2, the flowchart for EGRO-IE is the same as the one for EGO in Figure 2.1.

After initialization in Step 1, n initial sampling locations are generated using a design of experiments (e.g. Latin hypercube sampling). The response at these sample locations is computed on the expensive to evaluate function in Step 3. Based on the set of samples and responses, a Kriging metamodel \mathcal{K}_f of $f(\mathbf{x})$ is constructed in Step 4. At this point EGRO-IE deviates from the efficient global optimization algorithm. Step 5 is divided into two sub-steps. In Step 5a, a reference robust optimum $r_{\mathcal{K}}$ is computed on the constructed metamodel \mathcal{K}_f ,

$$r_{\mathcal{K}} = \min_{\mathbf{x} \in \mathbb{R}^n} \max_{\Delta \in \mathcal{U}} \mathcal{K}_f(\mathbf{x} + \Delta). \quad (2.8)$$

$r_{\mathcal{K}}$ represents the best worst-case cost on the metamodel.

In Step 5b, the algorithm identifies the next sampling location \mathbf{x}_{new} . This location is found using a modified expected improvement criterion. The worst-case Kriging prediction, with respect to the uncertainty set, at any position \mathbf{x} in the design variable space is

$$\hat{y}_{\max}(\mathbf{x}) = \max_{\Delta \in \mathcal{U}} \mathcal{K}_f(\mathbf{x} + \Delta). \quad (2.9)$$

The corresponding location within the range $\mathbf{x} + \Delta$, where \hat{y}_{\max} is found, is

$$\mathbf{x}_{\max} = \mathbf{x} + \arg \max_{\Delta \in \mathcal{U}} \mathcal{K}_f(\mathbf{x} + \Delta). \quad (2.10)$$

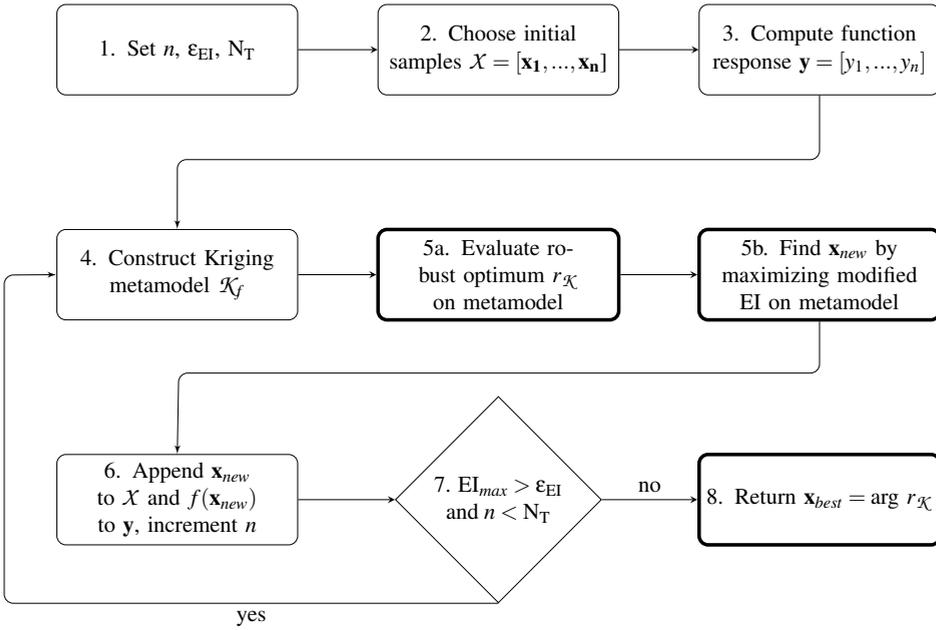


Figure 2.2: Flowchart shows the modifications applied to the Efficient Global Optimization (EGO) algorithm [6] in order to efficiently find the robust optimum of an unconstrained problem affected by implementation error. The modifications are highlighted by a bold boundary for the flowchart boxes.

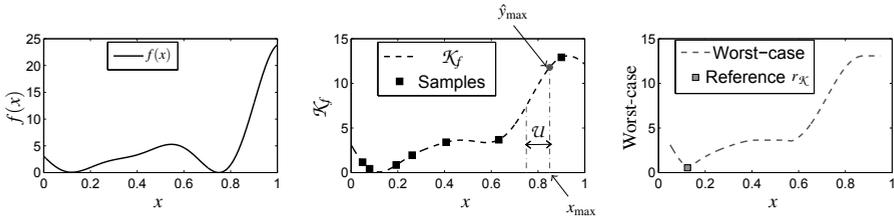


Figure 2.3: A one-dimensional function is shown on the left. A Kriging metamodel of the function is plotted in the centre. For $\mathcal{U} \in [-0.05, 0.05]$, the worst-case Kriging prediction \hat{y}_{\max} at $x = 0.8$ is indicated along with the location x_{\max} where it is found. On the right, the worst-case Kriging prediction with respect to \mathcal{U} is plotted. The reference robust optimum $r_{\mathcal{X}}$ is the minimum value obtained on this plot.

When constructing \mathcal{K}_f , the design domain is scaled to a range of $[0, 1]$ across each dimension. It is assumed that \mathcal{K}_f is only valid in the domain of \mathbf{x} and we do not consider extrapolation outside this domain. Let Δ_{\max} be the maximum realized $\Delta \in \mathcal{U}$. Furthermore, assume that the range of \mathcal{U} is symmetric around zero. Then it can be noted from Equation (2.9) that we may not be able to compute $\hat{y}_{\max}(\mathbf{x})$ if the sample is located at $\mathbf{x} < \Delta_{\max}$ or $\mathbf{x} > 1 - \Delta_{\max}$. This is because in this range there would be no nominal metamodel \mathcal{K}_f to refer to. Therefore, the domain within which the worst-case Kriging

prediction \hat{y}_{\max} is allowed to be computed is reduced to $[\Delta_{\max}, 1 - \Delta_{\max}]$ across each dimension.

Figure 2.3 illustrates some of the concepts that have been introduced so far. A one-dimensional function $f(x)$ is shown on the left. A Kriging metamodel \mathcal{K}_f of the function is constructed based on a sample set. Let the size of the implementation error be given by $\mathcal{U} \in [-0.05, 0.05]$ for the scaled design domain $x \in [0, 1]$. The bounds on the uncertainty set \mathcal{U} at $x = 0.8$ are displayed in the central plot. It can be seen that for $x = 0.8$ the worst-case Kriging prediction \hat{y}_{\max} is found at the positive bound of \mathcal{U} , i.e. at $x_{\max} = 0.85$. Note that in general $\mathbf{x}_{\max} \neq \mathbf{x}$, i.e. the worst-case generally occurs at a finite implementation uncertainty. The plot on the right shows the worst-case Kriging prediction as a function of x . The reference robust optimum $r_{\mathcal{K}}$ is the minimum value found on this plot.

As in deterministic EGO, it is assumed that the uncertainty in the value of the worst-case Kriging prediction, \hat{y}_{\max} , at any point \mathbf{x}_{\max} can be modelled using a normally distributed random variable $Y_{\max}(\mathbf{x}_{\max})$ with mean \hat{y}_{\max} and variance $s^2(\mathbf{x}_{\max})$. The term $s^2(\mathbf{x}_{\max})$ represents the Kriging mean squared error at \mathbf{x}_{\max} ; see Equation (2.6). We emphasize that no distribution is assumed for the uncertainty set \mathcal{U} and the implementation error remains bounded-but-unknown. The normal distribution introduced here expresses the uncertainty in the worst-case Kriging prediction \hat{y}_{\max} .

The next sampling point should be added at the location most likely to lead to the highest improvement over the current robust optimum $r_{\mathcal{K}}$. This improvement over $r_{\mathcal{K}}$ is defined as $I_w = \max(r_{\mathcal{K}} - Y_{\max}, 0)$. I_w represents the improvement of the worst-case Kriging prediction over $r_{\mathcal{K}}$. The expected improvement is found by computing the expected value of the improvement I_w under the normal distribution setting,

$$\underbrace{E[I_w(\mathbf{x}_{\max})]}_{\text{EI}_w} = \int_{I_w=0}^{I_w=\infty} I_w \left\{ \frac{1}{\sqrt{2\pi}s(\mathbf{x}_{\max})} \exp \left[-\frac{(r_{\mathcal{K}} - I_w - \hat{y}_{\max})^2}{2s^2(\mathbf{x}_{\max})} \right] dI_w \right\}. \quad (2.11)$$

Let

$$t = \frac{r_{\mathcal{K}} - I_w - \hat{y}_{\max}}{s(\mathbf{x}_{\max})}. \quad (2.12)$$

Also take into consideration that the standard normal probability density function is defined as

$$\phi(z) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right). \quad (2.13)$$

Then Equation (2.11) can be written as

$$E[I_w(\mathbf{x}_{\max})] = (r_{\mathcal{K}} - \hat{y}_{\max}) \int_{t=-\infty}^{t=\frac{r_{\mathcal{K}} - \hat{y}_{\max}}{s}} \phi(t) dt - s \int_{t=-\infty}^{t=\frac{r_{\mathcal{K}} - \hat{y}_{\max}}{s}} t \phi(t) dt. \quad (2.14)$$

The first integral in (2.14) is just the standard normal cumulative distribution function $\Phi\left(\frac{r_{\mathcal{K}} - \hat{y}_{\max}}{s}\right)$. The second integral in (2.14) can be solved by using the substitution $z = \frac{-t^2}{2}$. The final expression for EI_w is given by,

$$E[I_w(\mathbf{x}_{\max})] = (r_{\mathcal{K}} - \hat{y}_{\max}) \Phi\left(\frac{r_{\mathcal{K}} - \hat{y}_{\max}}{s}\right) + s \phi\left(\frac{r_{\mathcal{K}} - \hat{y}_{\max}}{s}\right). \quad (2.15)$$

To find the next sampling location, the global maximum of EI_w in Equation (2.15) needs to be found. For this, the design landscape in the range $\mathbf{x} \in [\Delta_{max}, 1 - \Delta_{max}]$ is scanned using a global optimizer. For each location \mathbf{x} requested by the optimizer, the worst-case Kriging prediction \hat{y}_{max} and the corresponding location where it is found, i.e. \mathbf{x}_{max} , are computed using (2.9) and (2.10) respectively. Once these quantities are found, EI_w is evaluated using Equation (2.15). The location \mathbf{x}_{max} for which EI_w is maximum is chosen as the new sampling location \mathbf{x}_{new} , where the expensive function is sampled. Comparing (2.7) with (2.15) we note that the structure of EI_w is similar to that of the deterministic EI. However, EI_w is a function of \mathbf{x}_{max} instead of \mathbf{x} . In the deterministic EI, the minimum observed sample y_{min} was the reference solution over which an improvement was sought. In the case of EI_w , the expected improvement is computed over the global robust optimum on the metamodel, $r_{\mathcal{X}}$. Secondly, the reference solution is now compared to the worst-case Kriging prediction \hat{y}_{max} instead of to the nominal Kriging prediction \hat{y} .

Using this adaptive sampling strategy, the algorithm steadily progresses towards the robust optimum. As illustrated by Figure 2.2, EGRO-IE terminates when EI_{max} falls below the threshold ϵ_{EI} or when the total number of samples is exhausted. At this stage, the algorithm returns the location, \mathbf{x}_{best} , where the global robust optimum $r_{\mathcal{X}}$ was found on the metamodel in the last iteration, which was built using all the collected information.

2.4.3. DISCUSSION

As can be noted in the procedure outlined above, the worst-case Kriging prediction \hat{y}_{max} was computed as a deterministic quantity on the metamodel. As in the conventional EGO procedure, it was then assumed that the uncertainty in the value of the worst-case Kriging prediction can be modelled as a normally distributed random variable. The expectation of improvement of the worst-case cost over the reference robust optimum was then computed using an expression similar to the deterministic EI (2.7).

The proposed approach to use the deterministic value for the worst-case cost on the metamodel with respect to the uncertainty set is not entirely rigorous. Instead, the distribution of the maximum [23] should have been found given the assumption that the Kriging surface is a Gaussian field. For this, additional assumptions would be necessary on the joint distribution of all points. The expectation of improvement should have been computed for this, generally non-Gaussian, distribution of the maximum. However, the computational costs of finding the distribution of the maximum numerous times at each iteration for the whole Kriging surface are prohibitively high. Furthermore, numerically evaluating the integral for the expectation of improvement of a non-Gaussian distribution across the design domain at each iteration is also expensive. Balancing the value of an expected improvement indicator function with its computational costs, we propose to focus only on improvement of the most relevant point in each uncertainty interval, i.e. the worst-case design. After all, EGO is only a heuristic, because it uses a sequence of Kriging metamodels that are only approximations of the true function defined in Equation (2.1).

2.5. NUMERICAL PERFORMANCE EVALUATION

2.5.1. FORMULATION OF TEST PROBLEMS

In order to evaluate the performance of the proposed algorithm, we apply it on a set of test problems. The aim is to test various aspects of the algorithm. The most important aspects to be tested are the ability of the algorithm to reach the global robust optimum for a given test problem and the convergence speed. The performance of EGRO-IE is compared to other methods that could potentially be used. Comparisons are made against sampling using a design of experiments strategy such as Latin hypercube sampling as well as against the MiMaReK algorithm proposed by Marzat *et al.* [15].

Due to the non-deterministic nature of the initial sampling, the results can vary from one run to another. The robust optimum found by EGRO-IE should be reproducible and consistent regardless of the random samples used to initialize the algorithm. In order to guarantee convergence for different initial sample sets, all the test cases are run 100 times and the statistical output is analysed. In this section, we introduce the test problems, while numerical results are presented in Section 2.6.

2.5.2. ONE-DIMENSIONAL PROBLEM

In order to illustrate the choices made by EGRO-IE at each individual step as it proceeds towards the robust optimum, it is applied on a one-dimensional problem. A modified version of the one-variable test function proposed by Forrester *et al.* [7] is used for this purpose. This function is defined as

$$f(x) = (6x - 2)^2 \sin(12x - 4) + 8x, \quad x \in [0, 1] \quad (2.16)$$

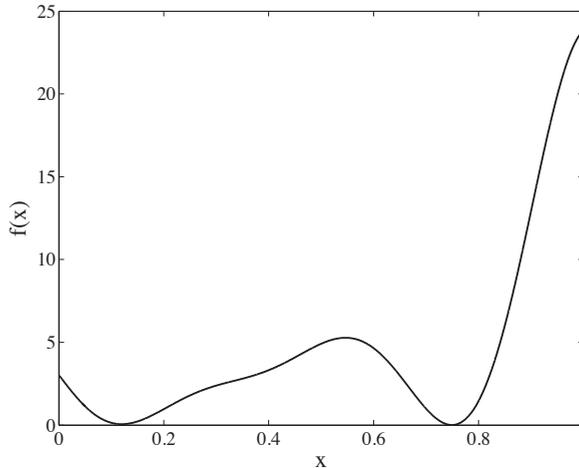


Figure 2.4: One-dimensional test problem for EGRO-IE, modified from [7]. The test function is used to illustrate the choices made by EGRO-IE at each individual step and to show how the algorithm proceeds toward the robust solution.

From Figure 2.4 it can be noticed that the function is non-convex and has two minima whose objective values are very close to each other. The nominal global optimum of the problem lies at 0.75. However, it can be seen that the local minimum that lies at 0.12 has a lower curvature than the nominal global minimum, so this local optimum is the global robust optimum.

To test the ability of the algorithm to identify a nominally suboptimal minimum as the global robust solution, an uncertainty range \mathcal{U} that ensures that the robust optimum lies at the lower curvature minimum at 0.12 is intentionally chosen. As mentioned previously, the one-variable problem also allows us to visually follow the evolution of the algorithm as it adds new samples at each iteration and gradually identifies the global robust solution. A range of $[-0.05, 0.05]$ is appropriate for the uncertainty set \mathcal{U} in order to test the relevant aspects of the algorithm.

2.5.3. THREE-DIMENSIONAL PROBLEM

The one-dimensional problem enables us to observe the progress of the algorithm at each iteration. However, the problem is too small in number of dimensions to judge the ability of the algorithm to tackle higher-dimensional, multi-modal problems. In order to evaluate the ability of the algorithm to converge to the global robust optimum of multi-dimensional, non-convex problems it is tested on the following three-dimensional test function,

$$f(\mathbf{x}) = \left(x_2 - \frac{5.1}{4\pi} x_2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left(\left(1 - \frac{1}{8\pi} \right) \cos(x_1) + 1 \right) + (6x_3 - 2)^2 \sin(12x_3 - 4) + 8x_3, \\ x_1 \in [-5, 10], x_2 \in [0, 15], x_3 \in [0, 1]. \quad (2.17)$$

This test function is a combination of the two-dimensional Branin function [24] and the one-dimensional test problem that has been described previously. The one-dimensional test problem is introduced into the test function through the third design variable. The Branin function is a multimodal, non-convex function with three nominal minima, all with the same objective value. Therefore, all three minima are nominally global. In the third dimension the function has one local and one global nominal minimum.

The main aspect that needs to be tested using this function is the algorithm's ability to find the true global robust optimum of a multimodal, non-convex problem. The problem is very flat in certain parts of the domain but changes value dramatically in other parts. In the third dimension, the function is relatively much flatter than in the other two dimensions of the problem. Due to its flatness in some regions and its high sensitivity to change in the design variables in other locations, the function is a challenging test problem for EGRO-IE. The non-convexity of the problem enables us to evaluate the ability of the algorithm to find the true global robust minimum rather than a robust local minimum.

Before the metamodel building process, the three design variables are normalized so that all of them lie in the range of $[0, 1]$. The non-convex problem is tested with an uncertainty set \mathcal{U} which has the same number of dimensions as the non-convex problem. The

size of the set is taken as $[-0.125, 0.125]$ across each dimension. This represents a ± 12.5 percent maximum change in the design variables due to the implementation error.

2.5.4. EIGHT-DIMENSIONAL AND TEN-DIMENSIONAL PROBLEMS

The three-dimensional problem should sufficiently test the ability of the algorithm to locate the global robust optimum of a non-convex function. However, to gauge the scalability of the algorithm with respect to increasing number of design variables, the benchmark should be a higher dimensional problem. The following test function is proposed for this criterion,

$$f(\mathbf{x}) = \sum_{i=1}^{n_t} \exp(0.3x_i)x_i^2, \quad (2.18)$$

$$x_i \in [-1, 1] \quad \forall i.$$

where n_t is the total number of design variables. The nominal optimum of this convex function would be at $x_i = 0 \quad \forall i$. We test EGRO-IE using an eight-variable and a ten-variable version of the problem in Equation (2.18). For the eight-dimensional problem, an eight-dimensional uncertainty set \mathcal{U} is assumed. For the ten-dimensional problem, we assume a one-dimensional uncertainty set \mathcal{U} that affects all the design variables equally. For both test-cases, the design variables are normalized so that they lie in the range of $[0, 1]$. Furthermore, the range of \mathcal{U} is $[-0.125, 0.125]$ for both the eight-dimensional and the ten-dimensional problem.

Given the range of the uncertainty set \mathcal{U} , the robust optimum does not lie at the same location as the nominal optimum for either test-case. The presence of the exponential means that the function is not symmetric around the nominal optimum on the origin. The resulting non-symmetric curvature of the problem results in a different optimum once robust optimization is performed given the implementation error. Therefore, with these functions it can be checked whether the algorithm can converge to a robust optimum, that is different from the nominal optimum, of large multi-dimensional problems.

Since the test problem is convex, the worst-case cost is always found on the bounds of the uncertainty set. On the test function, the reference robust optimum location can be computed by finding the design variables \mathbf{x} for which the lower bound and upper bound of \mathcal{U} return the same objective value. Shifting slightly from this location would mean that the worst-case cost at one of the bounds would become higher than the worst-case of the other bound. It can be shown that the robust optimum for this convex test function is found at $x_i = -0.0094 \quad \forall i$.

2.5.5. COMPARISON WITH OTHER TECHNIQUES

The results produced via EGRO-IE are compared against other techniques that could be employed to find the robust optimum of a problem affected by implementation error. The MiMaReK algorithm proposed by Marzat *et al.* [15] can be applied to the test problems. However, instead of using the property of implementation error that the uncertainties reside in the same dimension as the design domain, the uncertainties would be represented by environmental variables. Environmental variables reside in a dimension

that is separate from the design domain. The problem is posed as

$$\min_{\mathbf{x}} g(\mathbf{x}, \mathbf{x}_e) \quad (2.19)$$

where

$$g(\mathbf{x}, \mathbf{x}_e) = \max_{\mathbf{x}_e \in \mathcal{U}} f(\mathbf{x}, \mathbf{x}_e). \quad (2.20)$$

The environment variables are denoted by \mathbf{x}_e . As an example, the one-dimensional problem, in this format, can be written as

$$f(x, x_e) = (6(x + x_e) - 2)^2 \sin(12(x + x_e) - 4) + 8(x + x_e), \quad x \in [0, 1], x_e \in \mathcal{U} \quad (2.21)$$

EGRO-IE is also compared against the simple method of building \mathcal{K}_f using a design of experiments strategy such as Latin hypercube sampling and applying robust optimization on the resulting surrogate. This comparison is made for the non-convex three-dimensional problem. \mathcal{U} is assumed to have the same number of dimensions as the number of design variables.

2.6. RESULTS

2.6.1. ONE-DIMENSIONAL PROBLEM

The algorithm is applied on the one-dimensional problem introduced in Section 2.5.2. Fig. 2.5 shows snapshots of the progress of EGRO-IE as it tries to find the robust optimum of the one-dimensional test problem. The parameters are set to $\epsilon_{EI} = 10^{-6}$, $N_T = 12$, $n = 2$, $\mathcal{U} \in [-0.05, 0.05]$. The column of plots on the far left represent the nominal function $f(x)$ and its Kriging metamodel \mathcal{K}_f . The true worst-case cost $g(x)$, computed using (2.3), is shown in the middle column plots. The worst-case Kriging prediction is also plotted on the central column. On the right column, EI_w is plotted as a function of the design variable x . The numbering on the far left denotes the current iteration of the algorithm.

In the first iteration, it is observed that \mathcal{K}_f is a relatively poor approximate of the true nominal function, $f(x)$. At this stage EI_w is also quite high apart from at the sample point locations. From $x = 0.6$ to $x = 1$, EI_w is almost flat. This takes place because the Kriging RMSE remains relatively constant for this region. It can be noted that in the following iterations, points are added in the range of $x = 0$ to $x = 0.2$. By the third iteration, both the Kriging metamodel and the worst-case Kriging prediction approximate the true functions $f(x)$ and $g(x)$ fairly well in this range. At the 7th iteration, EI_w suggests that points should be added at the other local minimum at 0.75. By the 9th iteration, the Kriging metamodel and the worst-case Kriging prediction look visually indistinguishable from $f(x)$ and $g(x)$. EI_w has also dropped below the threshold ϵ_{EI} at this stage. The last robust optimum location found, at $x = 0.124$, is returned as the final solution.

2.6.2. THREE-DIMENSIONAL NON-CONVEX PROBLEM

Next, EGRO-IE is tested on the three-dimensional non-convex problem. The dimensions of the uncertainty set \mathcal{U} are equal to the number of design variables. \mathcal{U} varies in a range of $[-0.125, 0.125]$ across each dimension. The total number of function evaluation available is assumed to be $N_t = 125$. This represents 5 function evaluations per dimension.

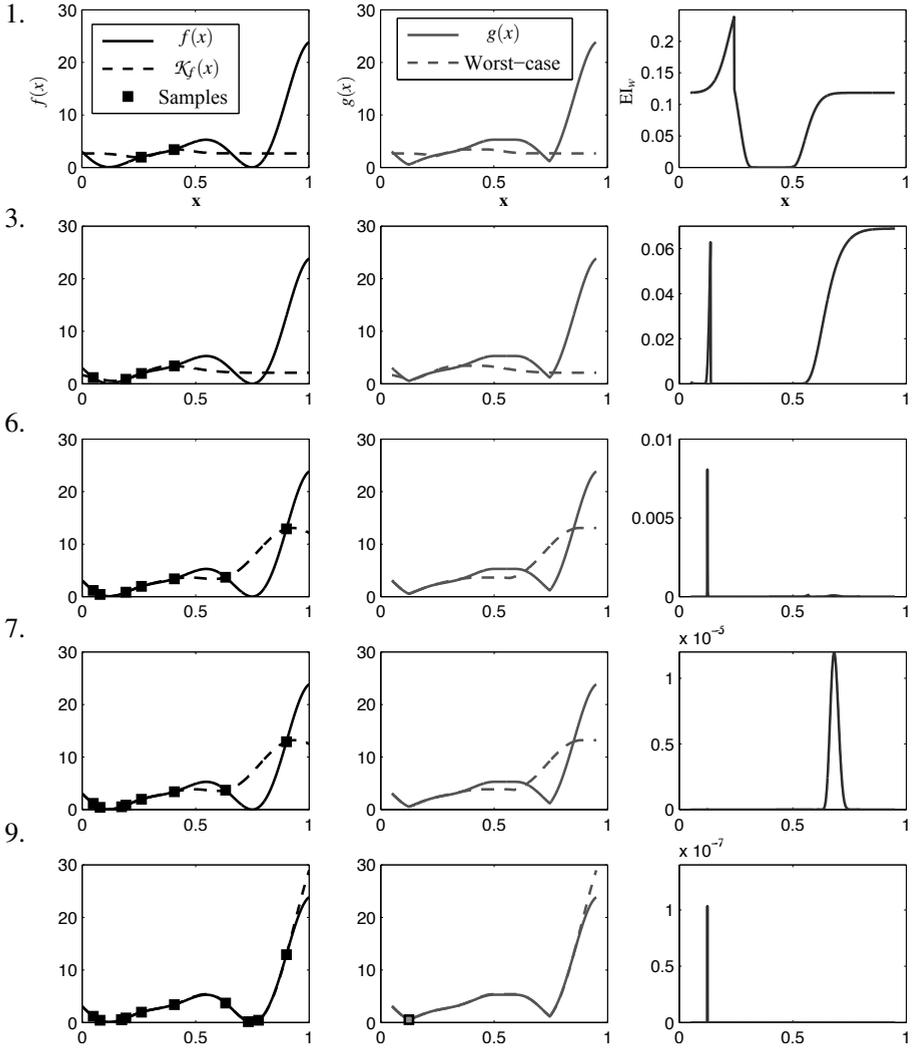


Figure 2.5: Snapshots of the progress of the algorithm are shown when it is applied to the one-dimensional test problem. By the 9th iteration EI_w has gone below the threshold ϵ_{EI} and the robust optimum has been found.

The algorithm is initialized with $n = 20$ samples chosen using Latin hypercube sampling. EGRO-IE is run 100 times with different initial sampling. Figure 2.6 shows the result. The triangular markers in the figure represent the mean, over 100 runs, of the robust optimum $r_{\mathcal{N}}$ found at each iteration. The error bars denote the first standard deviation around the mean value. The solid black line represents the true robust optimum directly evaluated on the three-dimensional non-convex function given the three-dimensional uncertainty set \mathcal{U} with range $[-0.125, 0.125]$.

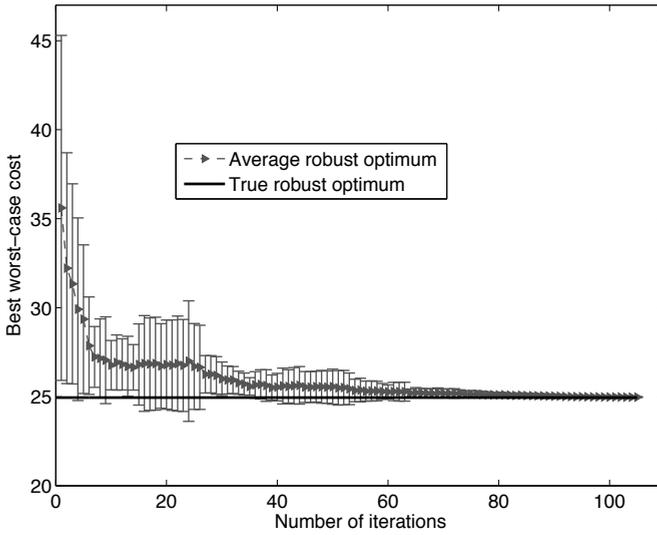


Figure 2.6: Robust optimum for three-dimensional non-convex problem for $\mathcal{U} \in [-0.125, 0.125]$. The plots show the mean and standard deviation of the robust optimum achieved at each iteration of EGRO-IE.

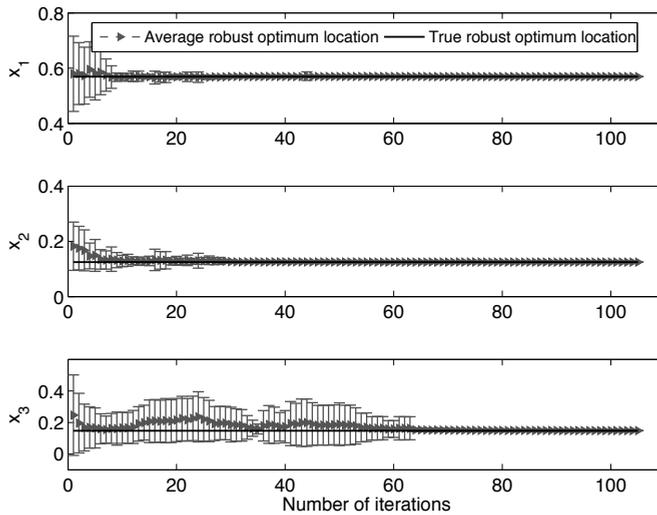


Figure 2.7: Robust optimum location for three-dimensional problem across each design variable when $\mathcal{U} \in [-0.125, 0.125]$. The plot shows mean and standard deviation of the best robust optimum location achieved for each design variable for all iterations of EGRO-IE.

Observing Figure 2.6 it can be seen that the average robust optimum value drops very fast from iteration 1 to 10. The corresponding standard deviation around this average also goes down fairly quickly at this stage. On the other hand, from the 11th to the 25th iteration, the average robust optimum value does not improve further and in

fact the standard deviation around the mean increases. This may be due to the fact that the algorithm is exploring the general landscape at this point. The exploration does not immediately improve the average robust optimum, but results in an overall more accurate Kriging metamodel. From the 26th to the 40th iteration, the mean and standard deviation of the robust optimum steadily go down again. This is followed by another period, from the 41st to the 55th iteration, when there is little improvement. Again this can be attributed to EI_w being high in regions that have not been explored yet and that could be potential robust optimum candidates. By the 80th iteration of EGRO-IE, the average robust optimum is almost exactly the same as the true robust optimum value. Furthermore, the standard deviation is also extremely small at this stage. Once the total computation budget of $N_t = 125$ samples is exhausted, the average robust optimum is visually indistinguishable from the true robust optimum. Similarly, at least visually, the standard deviation seems to have become almost 0.

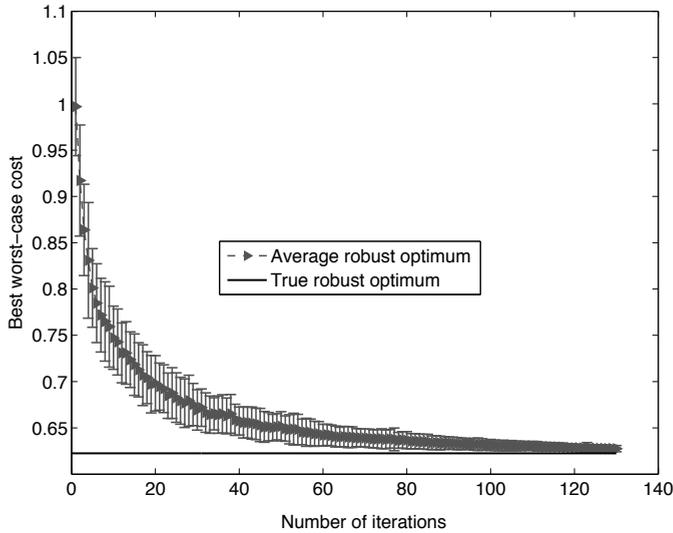


Figure 2.8: Robust optimum for ten-dimensional problem given a one-dimensional uncertainty set \mathcal{U} with range $[-0.125, 0.125]$. The plot shows mean and standard deviation of the best robust optimum achieved at each iteration of EGRO-IE.

It is quite instructive to observe how the average robust optimum location, with respect to each design variable, evolves with increasing number of iterations. Figure 2.7 plots the mean and standard deviations of the robust optimum location for each design variable. Observing the figure, it can be noted that by the 25th iteration, the average robust optimum location for x_1 and x_2 has converged to the true robust optimum. The standard deviation has also become quite small. However, the average robust optimum location for x_3 only converges to the exact location after 65 iterations. As mentioned in Section 2.5.3, the function is relatively flat in the third dimension. Therefore, the algorithm needs more samples to identify the true robust optimum in the x_3 dimension.

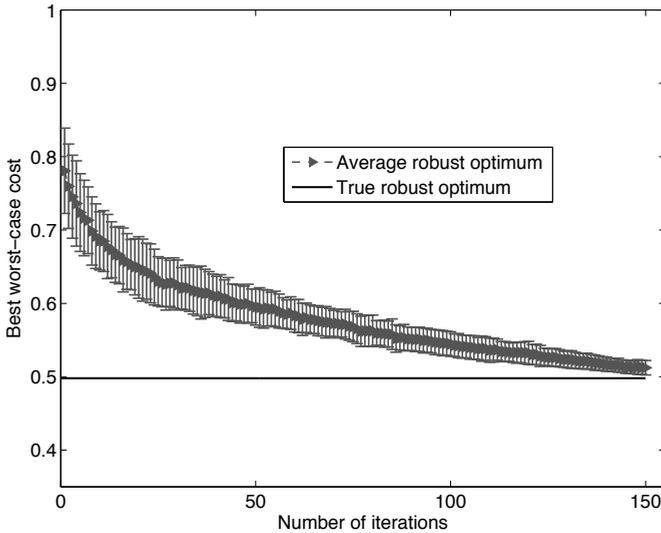


Figure 2.9: Robust optimum for eight-dimensional problem given an eight-dimensional uncertainty set \mathcal{U} with range $[-0.125, 0.125]$. The plot shows mean and standard deviation of the best robust optimum at each iteration of EGRO-IE.

2.6.3. EIGHT-DIMENSIONAL AND TEN-DIMENSIONAL PROBLEMS

Thus far we have illustrated the evolution of EGRO-IE when it is applied on a one-dimensional problem. We have also demonstrated that the algorithm can find the global robust optimum of a challenging non-convex problem. In the following, the scalability of the algorithm is tested on an eight-dimensional and a ten-dimensional problem.

Figure 2.8 shows the result of applying the algorithm on the ten-dimensional problem (2.18) given a one-dimensional uncertainty set \mathcal{U} with range $[-0.125, 0.125]$. EGRO-IE is initialized with $n = 100$ samples [25]. The algorithm uses a total of 230 function evaluations. The figure shows that the average robust optimum falls quite quickly with increasing number of iterations. The standard deviation around the average robust optimum value also goes down steadily. When the computational budget of 230 samples is exhausted, the average robust optimum is quite close to the true robust optimum. Additionally, the standard deviation is also quite small at this stage. EGRO-IE uses only 230 function evaluations to reach the robust solution. This number represents less than a quarter of the number of samples needed to have 2 samples per dimension.

Finally, the algorithm is applied on the eight-dimensional problem (2.18) given an eight-dimensional uncertainty set \mathcal{U} with range $[-0.125, 0.125]$. We initialize EGRO-IE with $n = 80$ samples. Once again, the algorithm is allowed to use a total of 230 function evaluations. Figure 2.9 shows the result. The mean and standard deviation of the robust optimum go down steadily with increasing number of iterations. However, compared to the ten-dimensional problem, the convergence is relatively slow. This is to be expected, since the algorithm now has to search for the worst-case cost in an eight-dimensional uncertainty set instead of a one-dimensional uncertainty set. By the 150th iteration, the

Table 2.1: Comparison of statistics based on 100 runs of EGRO-IE and LHS on three dimensional non-convex problem assuming a three-dimensional uncertainty set \mathcal{U} of range $[-0.125, 0.125]$

Robust optimum	Reference	EGRO-IE	LHS	Function evaluations
Mean	24.95	25.10	27.27	100
Standard deviation	-	0.129	1.595	100
Mean	24.95	24.98	25.96	120
Standard deviation	-	0.027	0.9094	120

average robust optimum is close to the true solution and the standard deviation has also decreased to a low value. For an eight-dimensional problem, filling the design landscape with only 2 samples per dimension would require $2^8 = 256$ total samples. EGRO-IE uses less than 2 samples per dimension to locate an average robust optimum value that is quite close to the true robust optimum.

2.6.4. COMPARISON WITH AVAILABLE TECHNIQUES

EGRO-IE was compared against other techniques such as Latin hypercube sampling and the MiMaReK algorithm of Marzat *et al.* [15] for robust optimization of a problem affected by implementation error. The MiMaReK algorithm is a more widely applicable robust optimization algorithm in that it can also be applied to problems with parametric uncertainties while EGRO-IE is specifically meant for problems with implementation error.

MiMaReK is used to find the robust optimum of the one-dimensional function (2.16). Including uncertainties, the problem takes the form of Equation (2.21). We assume $\mathcal{U} \in [0.05, -0.05]$ which is the same size as the problem that was solved by EGRO-IE. Based on an average of 100 runs of EGRO-IE on the one-dimensional function, the robust optimum was found after 9 iterations of EGRO-IE. Since the problem was initialized with $n = 2$, the total number of function evaluations required to find the robust solution was 11. When MiMaReK is applied on the function to find the robust optimum of the one-dimensional function, an average of more than 200 function calls was required to reach the robust optimum. This is more than an order of magnitude of the number of function evaluations required by EGRO-IE. The comparison of EGRO-IE with MiMaReK of higher dimensional problems was therefore not necessary since EGRO-IE used much less than 200 expensive function evaluations to find the global robust optimum of the three-dimensional problem. Furthermore, the algorithm almost reached the robust optimum of the eight-dimensional and ten-dimensional problem using only 230 function evaluations.

Next, EGRO-IE is compared with the application of a design of experiments strategy such as Latin hypercube sampling to come up with a nominal metamodel on which robust optimization is performed. Table 2.1 compares EGRO-IE and LHS for robust optimization of the three-dimensional non-convex problem assuming a three-dimensional uncertainty set \mathcal{U} of range $[-0.125, 0.125]$. Both methods are run 100 times and the mean and standard deviation of the robust optimum after 100 and 120 expensive function eval-

uations are computed. The reference global robust optimum, evaluated directly on the function, has an objective value of 24.95. Comparing the case when there are 100 function evaluations available, it can be noted that EGRO-IE has a much closer mean robust optimum value to the reference robust optimum and the standard deviation around this average is also an order of magnitude lower. Application of LHS and EGRO-IE based on 120 function evaluations also shows that EGRO-IE performs much better than LHS. At this stage, the average robust optimum for EGRO-IE is very close to the true robust optimum and the standard deviation is more than 30 times smaller than the corresponding standard deviation for LHS.

2.7. CONCLUSION

In this chapter, we presented a novel method for finding the robust optimum of unconstrained problems involving implementation error. The algorithm employs Kriging-based optimization, specifically expected improvement and efficient global optimization, to locate the robust optimum in a small number of function evaluations. The special structure of the problem, where uncertainties reside in the same dimension as the design variables, was employed to devise an efficient method that uses a limited number of function evaluations to reach the robust solution.

The algorithm was applied on four proposed problems under different settings of the size and range of the uncertainty set for the implementation error. It was shown that EGRO-IE can find the robust optimum of these problems, for all the settings that were tested, in only a limited number of function evaluations. The results of the algorithm were confirmed statistically by running it 100 times on the functions and averaging the result. Other available techniques were also compared against the method. We observed that the performance of the proposed method was superior in terms of finding the robust solution in the lowest number of expensive function evaluations on the considered test problems.

ERGO-IE is expected to be very efficient in finding the robust optimum of engineering problems affected by implementation error such as the expensive to simulate integrated photonic devices [16]. We plan to apply EGRO-IE on these practical problems in the near future. The algorithm also has to be extended in order to perform robust optimization of problems affected by both implementation error and parametric uncertainties. This provides a new challenge since parametric uncertainties operate in a different dimension than the design variables of the problem.

Finding the global robust optimum of higher-dimensional problems is difficult since it involves a nested global optimization. Although EGRO-IE demonstrates a clear efficiency improvement, dealing with higher-dimensional robust optimization problems remains a challenge.

REFERENCES

- [1] A. Ben-Tal and A. Nemirovski, *Robust optimization – methodology and applications*, Mathematical Programming **92**, 453 (2002).
- [2] D. G. Krige, *A statistical approach to some basic mine valuation problems on the*

- Witwatersrand*, Journal of the Chemical, Metallurgical and Mining Society of South Africa **52**, 119 (1951).
- [3] H. P. Sacks, J., Welch, W., Mitchell, T. J., and Wynn, *Design and Analysis of Computer Experiments*, Statistical Science **4**, 409 (1989).
- [4] D. R. Jones, *A Taxonomy of Global Optimization Methods Based on Response Surfaces*, Journal of Global Optimization **21**, 345 (2001).
- [5] M. Franey, P. Ranjan, and H. Chipman, *Branch and bound algorithms for maximizing expected improvement functions*, Journal of Statistical Planning and Inference **141**, 42 (2011).
- [6] D. R. Jones, M. Schonlau, and W. J. Welch, *Efficient Global Optimization of Expensive Black-Box Functions*, Journal of Global Optimization **13**, 455 (1998).
- [7] A. Forrester, S. Andras, and A. J. Keane, *Engineering design via surrogate modelling : a practical guide* (J. Wiley, 2008).
- [8] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust optimization* (Princeton University Press, 2009).
- [9] M. Lombardi and R. T. Haftka, *Anti-optimization technique for structural design under load uncertainties*, Computer Methods in Applied Mechanics and Engineering **157**, 19 (1998).
- [10] A. Ben-Tal, T. Margalit, and A. Nemirovski, *Robust modeling of multi-stage portfolio problems*, in *High performance optimization* (Kluwer Acad. Publ., Dordrecht, 2000) pp. 303–328.
- [11] D. Bertsimas, O. Nohadani, and K. M. Teo, *Robust optimization in electromagnetic scattering problems*, **101**, 74507 (2007).
- [12] A. Ben-Tal and A. Nemirovski, *Robust Convex Optimization*, Mathematics of Operations Research **23**, 769 (1998).
- [13] Y. Jin, *Surrogate-assisted evolutionary computation: Recent advances and future challenges*, Swarm and Evolutionary Computation **1**, 61 (2011).
- [14] Y.-S. Ong, P. B. Nair, and K. Y. Lum, *Max-min surrogate-assisted evolutionary algorithm for robust design*, Evolutionary Computation, IEEE Transactions on **10**, 392 (2006).
- [15] J. Marzat, E. Walter, and H. Piet-Lahanier, *Worst-case global optimization of black-box functions through Kriging and relaxation*, Journal of Global Optimization , **1** (2012).
- [16] S. Ur Rehman, M. Langelaar, and F. Van Keulen, *Robust optimization of 2x2 multi-mode interference couplers with fabrication uncertainties*, , 862711 (2013).

- [17] G. Dellino, J. P. C. Kleijnen, and C. Meloni, *Robust optimization in simulation: Taguchi and Response Surface Methodology*, International Journal of Production Economics **125**, 52 (2010).
- [18] M. Morris and T. J. Mitchell, *Exploratory designs for computational experiments*, Journal of Statistical Planning and Inference **43**, 381 (1995).
- [19] S. Boyd and L. Vandenberghe, *Convex Optimization* (Cambridge University Press, New York, NY, USA, 2004).
- [20] D. den Hertog, J. P. C. Kleijnen, and A. Y. D. Siem, *The Correct Kriging Variance Estimated by Bootstrapping*, The Journal of the Operational Research Society **57**, pp. 400 (2006).
- [21] M. Schonlau and W. Welch, *Global optimization with nonparametric function fitting*, in *Proceedings of the Section on Physical and Engineering Sciences* (American Statistical Association, 1996) pp. 183–186.
- [22] M. D. McKay, R. J. Beckman, and W. J. Conover, *A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code*, Technometrics **21**, pp. 239 (1979).
- [23] R. B. Arellano-Valle and M. G. Genton, *On the exact distribution of the maximum of absolutely continuous dependent random variables*, Statistics & Probability Letters **78**, 27 (2008).
- [24] G. Dixon, L. Szego, *The global optimization problem: an introduction*, in *Towards Global Optimisation 2* (North-Holland, Amsterdam, 1978) pp. 1–15.
- [25] J. L. Loeppky, J. Sacks, and W. J. Welch, *Choosing the Sample Size of a Computer Experiment: A Practical Guide*, Technometrics **51**, 366 (2009).

3

ROBUST OPTIMIZATION UNDER PARAMETRIC UNCERTAINTIES

3.1. INTRODUCTION

Most practical design problems have some degree of uncertainty associated with them. Problems that are sensitive to even slight perturbations may give rise to suboptimal or even infeasible solutions when optimized without incorporating uncertainties. The uncertainties may be present in the parameters or in the design variables. Uncertainty in the parameters is known as parametric uncertainty while uncertainty in the design variables is referred to as implementation error. Often the uncertainties are bounded-but-unknown [1] and the probability distribution of the uncertainties is not available. In such a scenario, robust optimization [2, 3] has to be applied to find a robust solution.

The basic idea of robust optimization is the minimization of the maximum realizable value of the objective with respect to the uncertainty set, subject to the non-violation of the worst-case constraints. Other terms such as best worst-case optimization or min-max optimization are used to describe the same concept.

Due to its wide ranging applications, robust optimization has been a topic of intense research in several different fields. Tackling robustness has been a fairly established concept in robust control, please refer to [4] and the references therein for more detail. From a purely engineering perspective, the pioneering paper was written by Taguchi [5]. Considerable progress in robust optimization has been made in the field of mathematical programming in recent years [2]. However, the focus has been limited to solving convex problems of varying complexity. Min-max optimization has also received a lot of attention in decision and game theory [6]. Work in this field has shown that for certain convex-concave functions, the global robust optimum can be found by searching for a saddle point solution [7]. Therefore, saddle point optimization has been used extensively to find the robust optimum of such unconstrained min-max problems [7].

However, many practical robust optimization problems are not convex. Additionally, the underlying simulation of these unconstrained continuous min-max problems

could be expensive to evaluate. There is considerably less work on finding the global robust optimum of such non-convex unconstrained problems. Unconstrained continuous min-max problems have many applications including, but not limited to, aerodynamic shape design [8], finance [7], game theory [9], fault detection [10] and signal processing [11]. Evolutionary algorithms have been used to find the global solution of such unconstrained continuous min-max problems for which no special structure has been assumed [8, 12–15]. However, using evolutionary algorithms to find the min-max solution of such problems leads to extremely high computational costs since these algorithms usually require very high number of function evaluations.

Since robust optimization involves solving a nested min-max problem, applying robust optimization directly on a problem based on an expensive computer simulation usually leads to prohibitively high computational costs. Instead, one can sample the expensive function at carefully chosen points and build a response surface or surrogate [16, 17]. However, the best sample placement depends on the characteristics of the underlying function, which makes the sampling strategy itself a challenging problem. The ultimate goal is to reduce the number of expensive function evaluations required to find the optimum. Many different kinds of response surfaces can be used to build an approximate model. These include non-interpolating methodologies such as polynomial and regression models as well as interpolating approaches, e.g. radial basis functions, Kriging and splines.

Amongst the different techniques for response surface building, Kriging [18, 19] holds distinctive appeal due to its statistical framework which enables an estimation for the error in the interpolation between the sample points. This provides the groundwork for the development of expected improvement [20] and Efficient Global Optimization (EGO) [21, 22] procedures, which allow the adaptive placement of sample points at locations most likely to lead to the global optimum. EGO has successfully been applied in deterministic unconstrained optimization, e.g. [23], while the convergence properties of expected improvement have been established in [24]. A drawback of Kriging is that the correlation matrix that contains the underlying basis functions may tend to suffer from ill-conditioning [21]. Furthermore, it has also been shown that Kriging underestimates the potential error in the interpolation [25].

In recent work [26, 27], expected improvement has also been applied to perform unconstrained continuous min-max optimization of black-box functions. The robust solution is found by transforming the min-max problem into a nominal constrained optimization problem with constraint relaxation. Although the method uses less expensive function evaluations compared to evolutionary algorithms [15], the number of expensive function calls is still quite high.

This work also deals with global robust optimization of unconstrained continuous min-max problems, where the uncertainties are considered to be bounded-but-unknown. The black box function is assumed to be continuous and to be based on an expensive simulation. To solve the min-max problem, we propose a technique that extends the established Efficient Global Optimization (EGO) algorithm for deterministic optimization to the non-deterministic case. At each iteration of the proposed algorithm, the problem is divided into a separate search for the next control variables sample location and the next parametric uncertainties sample location. The EI criteria are suitably adapted

to find the robust optimum instead of the nominal optimum using a small number of expensive function evaluations. We compare the efficiency of our approach with other methods using a set of standard test problems. Since the proposed algorithm is based on surrogate construction, it is not suitable for solving very high-dimensional problems. For a more in-depth discussion on challenges involved in applying surrogate-based optimization on high-dimensional problems we refer readers to [28].

This chapter is organized as follows. We introduce the problem in Section 3.2 and provide the mathematical background of robust optimization for problems involving parametric uncertainties. Section 3.3 contains a brief description of Kriging and Efficient Global Optimization. In Section 3.4 a detailed description of the robust optimization algorithm based on EGO is provided. Finally, Section 3.5 and 3.6 contain the results and conclusions, respectively.

3.2. ROBUST OPTIMIZATION OF UNCONSTRAINED PROBLEMS AFFECTED BY PARAMETRIC UNCERTAINTIES

The considered problem may formally be stated as

$$\min_{\mathbf{x}_c \in \mathbb{X}_c} \max_{\mathbf{x}_e \in \mathbb{X}_e} f(\mathbf{x}_c, \mathbf{x}_e), \quad (3.1)$$

where \mathbf{x}_c is the set of control variables while \mathbf{x}_e is the set of parametric uncertainties or environment variables. No special structure, such as convexity or monotonicity, *etc.* is assumed for the function $f(\mathbf{x}_c, \mathbf{x}_e)$. However, $f(\mathbf{x}_c, \mathbf{x}_e)$ is assumed to be continuous. We seek to find the global min-max solution of $f(\mathbf{x}_c, \mathbf{x}_e)$. The function is assumed to be expensive to evaluate and under this constraint, our aim is to develop a method that finds the location of the global best worst-case cost using a small number of function evaluations. This is done by estimating the robust optimum location on a relatively much cheaper surrogate model built using Kriging. The problem may be written as,

$$\min_{\mathbf{x}_c \in \mathbb{X}_c} \max_{\mathbf{x}_e \in \mathbb{X}_e} \mathcal{K}_f(\mathbf{x}_c, \mathbf{x}_e), \quad (3.2)$$

where \mathcal{K}_f is the continuously differentiable Kriging model of the expensive to evaluate function $f(\mathbf{x}_c, \mathbf{x}_e)$. The primary contribution in this work is to provide an adaptive sampling scheme, dedicated to the robust optimization setting, such that the Kriging function models the behaviour of the reference function $f(\mathbf{x}_c, \mathbf{x}_e)$ very accurately in regions of interest, i.e. potential robust optimum locations, using relatively few expensive function calls of $f(\mathbf{x}_c, \mathbf{x}_e)$.

In principle, any global optimization approach can be used to perform the min-max optimization in Equation (3.2). However, since the Kriging model is available and the Jacobian, Hessian information can be obtained cheaply as well, optimizers that leverage this information would be preferable. In this context there are several promising algorithms that could be used to perform robust optimization on the surrogate model. It should be noted that solving Equation (3.2) is equivalent to solving a semi-infinite optimization problem [29] since Equation (3.2) can be written as a constrained minimization problem over an infinite number of constraints. For a survey of the state of the art methods that solve Equation (3.2) via a semi-infinite programming approach please refer to

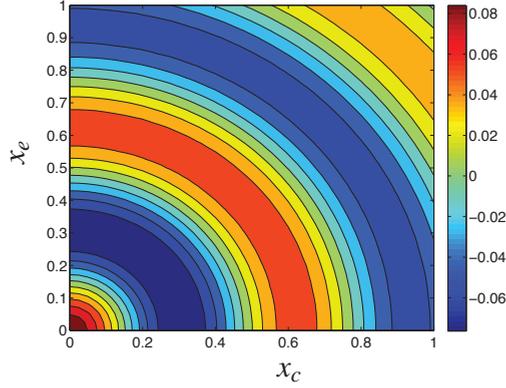


Figure 3.1: Contour lines of the reference function $f_{11}(\mathbf{x}_c, \mathbf{x}_e)$. Used to illustrate the choices made by the proposed algorithm at different stages.

[30]. Many of the algorithms reviewed in [30] require first and second order derivative information for the specified function, which can easily be provided for the continuously differentiable Kriging function.

A strategy that does not require gradient information is that of redefining Equation (3.2) as a constrained minimization problem [31] and using constraint relaxation to reduce the continuous set \mathbb{X}_e to a finite discrete set. This method was used for min-max optimization in combination with Kriging by [27].

In this work, our focus is on the adaptive sampling strategy instead of on the particular method chosen to solve the optimization problems on the resulting surrogate model. Nevertheless, in the optimization process we have opted to make full use of the availability of the Jacobian and Hessian information. Details on the applied optimizers are given in Appendix B.

Throughout this work, the 11th test problem from Appendix A will be used to illustrate the steps taken by the proposed algorithm,

$$f_{11}(\mathbf{x}_c, \mathbf{x}_e) = \frac{\cos(\sqrt{x_{c1}^2 + x_{e1}^2})}{\sqrt{x_{c1}^2 + x_{e1}^2 + 10}}. \quad (3.3)$$

The problem is a damped cosine wave in two dimensions with $\mathbb{X}_c \in [0, 10]$ and $\mathbb{X}_e \in [0, 10]$. As shown by Fig. 3.1, the function is non-convex and multimodal. Both x_c and x_e have been rescaled such that the design domain is in the range $[0, 1]$. The oscillatory behaviour of the function in both dimensions suggests that it is a non-trivial problem even for nominal global optimization. Therefore, finding the global min-max solution is also a challenging exercise. The problem will be used in Section 3.4 to visualize the steps involved in the algorithm. In Section 3.5 it will be employed to demonstrate the evolution of the algorithm as it searches for the global robust optimum.

3.3. KRIGING AND EFFICIENT GLOBAL OPTIMIZATION

3.3.1. KRIGING

Kriging is an interpolation method that uses a stochastic process approach to construct a cheap model of the expensive function. In this work, Kriging is basically used as a fitting technique for the deterministic simulation data. A detailed derivation of Kriging surrogate construction and prediction can be found in [19]. In this section, a concise description of Kriging and EGO is provided, while omitting details of the derivation.

The function response at any position \mathbf{x} in the domain is assumed to be a normally distributed random variable $Y(\mathbf{x})$ with mean μ and variance σ^2 . Each random variable in this stochastic process is assumed to be correlated to other random variables via the following parameterized Gaussian correlation function,

$$\text{Corr}[Y(\mathbf{x}_i), Y(\mathbf{x}_j)] = \exp\left(-\sum_{q=1}^k \theta_q |x_{iq} - x_{jq}|^p\right) \quad (3.4)$$

where \mathbf{x}_i and \mathbf{x}_j are any two locations in the domain and k represents the total number of dimensions of the problem. We set p to a constant value of 2. The correlation is therefore governed by the parameter θ_q and the distance between the points. The correlation is 1 when $\mathbf{x}_i = \mathbf{x}_j$ and it drops as the distance between the points increases. θ_q determines how active the q^{th} dimension is in shaping the cheap response. A higher value of θ_q means that the correlation will fall relatively quickly with respect to the q^{th} dimension as the distance between the points increases. A lower value of θ_q for the q^{th} dimension, on the other hand, will result in a relatively slower fall in correlation with increasing distance. This would also lead to a comparatively flatter response with respect to the q^{th} dimension.

The maximum likelihood estimator for a normal distribution is used to find values for θ_q , μ and σ^2 such that the likelihood of the observed data is maximized.

Once these model parameters are found, Kriging estimates the interpolation between the sample points that is most consistent with the observed data. The value of the response at these locations is found by maximizing the combined likelihood of the observed data and the Kriging prediction. The Maximum Likelihood Estimate (MLE) for the prediction \hat{y} is given by

$$\hat{y}(\mathbf{x}) = \hat{\mu} + \mathbf{r}^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1} \hat{\mu}), \quad (3.5)$$

where $\hat{\mu}$ is the estimated value for the mean, \mathbf{R} is the $N \times N$ correlation matrix between the N sample points, \mathbf{r} is the vector of correlations between the observed data and the new prediction, while \mathbf{y} is the observed response. We find the values for the correlation vector \mathbf{r} and the correlation matrix \mathbf{R} via Equation (3.4). It can be noted from Equation (3.5) that the Kriging predictor is simply a linear combination of basis functions and a constant.

One of the advantages of the stochastic process assumption in Kriging is that the error in the predicted response $\hat{y}(\mathbf{x})$ can be estimated. The mean squared error (MSE) in the prediction is given by

$$s^2(\mathbf{x}) = \hat{\sigma}^2 \left[1 - \mathbf{r}^T \mathbf{R}^{-1} \mathbf{r} + \frac{1 - \mathbf{1}^T \mathbf{R}^{-1} \mathbf{r}}{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}} \right], \quad (3.6)$$

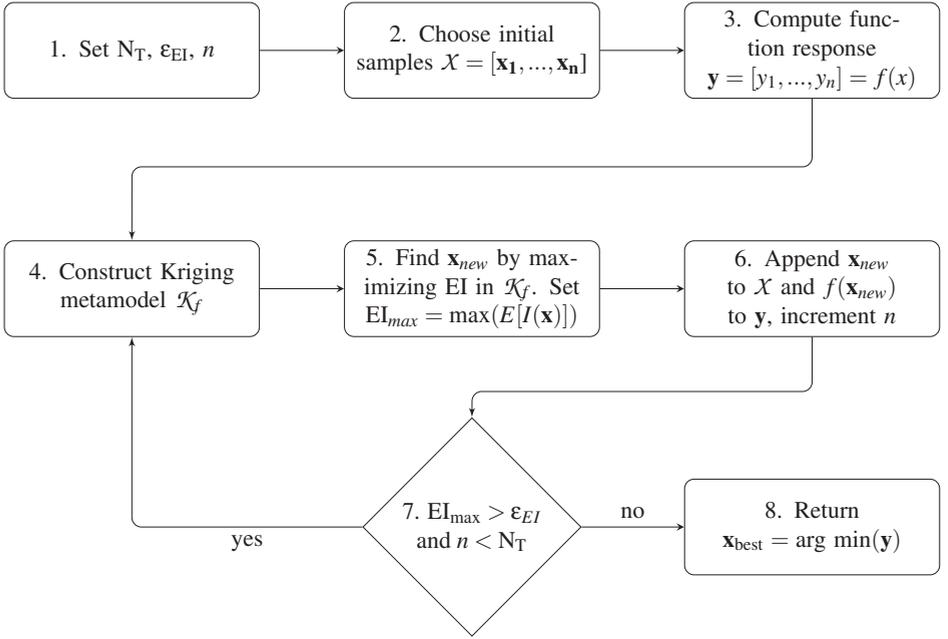


Figure 3.2: Flowchart of the Efficient Global Optimization algorithm [21]. The algorithm uses Kriging and expected improvement to find the nominal optimum of an unconstrained problem with relatively few function calls of an expensive to evaluate function.

where $\hat{\sigma}^2$ is the maximum likelihood estimate of the variance σ^2 . The fraction term in Equation (3.6) represents the estimated error in using a maximum likelihood estimate $\hat{\mu}$ for the mean to compute the error instead of using the true mean. An intuitive explanation of the mean squared error in terms of the combined log-likelihood between the observed data and the prediction can be found in [21], while the full derivation of Equation (3.6) is available in [19].

3.3.2. EFFICIENT GLOBAL OPTIMIZATION

The estimated MSE enables adaptive sampling of the constructed surrogate. Therefore, it proves useful in order to efficiently reach the global deterministic optimum of an expensive to evaluate function. The adaptive sampling strategy of expected improvement is also based on the MSE [21]. To formulate the expected improvement metric, entirely new assumptions have to be made. In this formulation, the uncertainty in the predicted value $\hat{y}(\mathbf{x})$ at any position \mathbf{x} in the domain is described as a random variable $Y(\mathbf{x})$ having a normal distribution. The mean and the variance of this distribution are assumed to be given by the Kriging prediction $\hat{y}(\mathbf{x})$ and the mean squared error $\hat{s}^2(\mathbf{x})$ respectively. Let y_{\min} denote the minimum objective value in the observed responses. We may be able to improve on y_{\min} at a position \mathbf{x} if a part of the distribution $Y(\mathbf{x})$ lies below the current

minimum. We find the expectation of this improvement I by computing the expectation $E[I(\mathbf{x})] = E[\max(y_{\min} - Y, 0)]$. Using integration by parts, the expected improvement can be written as

$$E[I(\mathbf{x})] = (y_{\min} - \hat{y})\Phi\left(\frac{y_{\min} - \hat{y}}{s}\right) + s\phi\left(\frac{y_{\min} - \hat{y}}{s}\right) \quad (3.7)$$

where $\Phi(\cdot)$ is the normal cumulative distribution function and $\phi(\cdot)$ is the normal probability density function. By finding the position in the design domain where EI is maximum we get an indication of where adding a new point would be most beneficial. In this way, the Kriging prediction \hat{y} and the error estimate \hat{s}^2 enables the development of the methodology of expected improvement (EI) and efficient global optimization (EGO) [21]. Based on the Kriging prediction and MSE, EI gives an indication of the best location to sample in order to improve on y_{\min} . Using the EI strategy, EGO is able to converge to the global optimum of the problem by adding new data points until the maximum EI is sufficiently low.

The flowchart in Fig. 3.2 shows how EGO makes use of EI to find the global optimum. N_T is the total number of sample points available. This is determined by the simulation time and cost. The initial number of samples, n , is chosen through uniform sampling for single design variable problems or through a space-filling strategy, e.g., Latin hypercube sampling (LHS) [32] for two or more design variables. The Kriging metamodel of $f(\mathbf{x})$ is denoted by \mathcal{K}_f . The algorithm terminates when either EI_{\max} falls below the threshold ϵ_{EI} or the total number of samples N_T is consumed.

3.4. EFFICIENT GLOBAL ROBUST OPTIMIZATION OF PARAMETRIC UNCERTAINTIES AFFECTED PROBLEMS

3.4.1. MAIN CONCEPT

We propose an efficient global robust optimization algorithm for finding the best worst-case cost of unconstrained problems affected by parametric uncertainties. The primary goal of this method is to find the global robust solution using a relatively small number of expensive function evaluations. Before expounding the algorithm, the main concept behind the technique will be discussed.

The proposed algorithm is similar to the efficient global optimization technique in that both methods employ an initialization phase followed by an iterative surrogate-based optimization approach, where the optimal sampling location at each iteration is chosen via adaptive sampling. We refer to the proposed method as Efficient Global Robust Optimization (EGRO).

During the initialization phase, the underlying expensive simulation is sampled in the control and environment variable space, $(\mathbb{X}_c, \mathbb{X}_e)$, using an appropriate design of experiments strategy. Based on the sampled set, an approximate model that encompasses the combined space $(\mathbb{X}_c, \mathbb{X}_e)$ is constructed using Kriging.

This is followed by the iteration phase, in which the expensive function is adaptively sampled by evaluating it at the location that is most expected to improve the current robust optimum. The sampling location is searched for in two stages, firstly, in the control variable space (Section 3.4.3) and secondly in the environment variable space (Section 3.4.4). For both stages, the control variables and environment variables location is found

by using modified versions of the expected improvement criterion. Thereafter, the expensive function is evaluated at the new location. The Kriging metamodel is then rebuilt with the augmented set of sampling locations and responses. The process of finding the new sampling location is repeated until convergence.

3.4.2. ALGORITHM

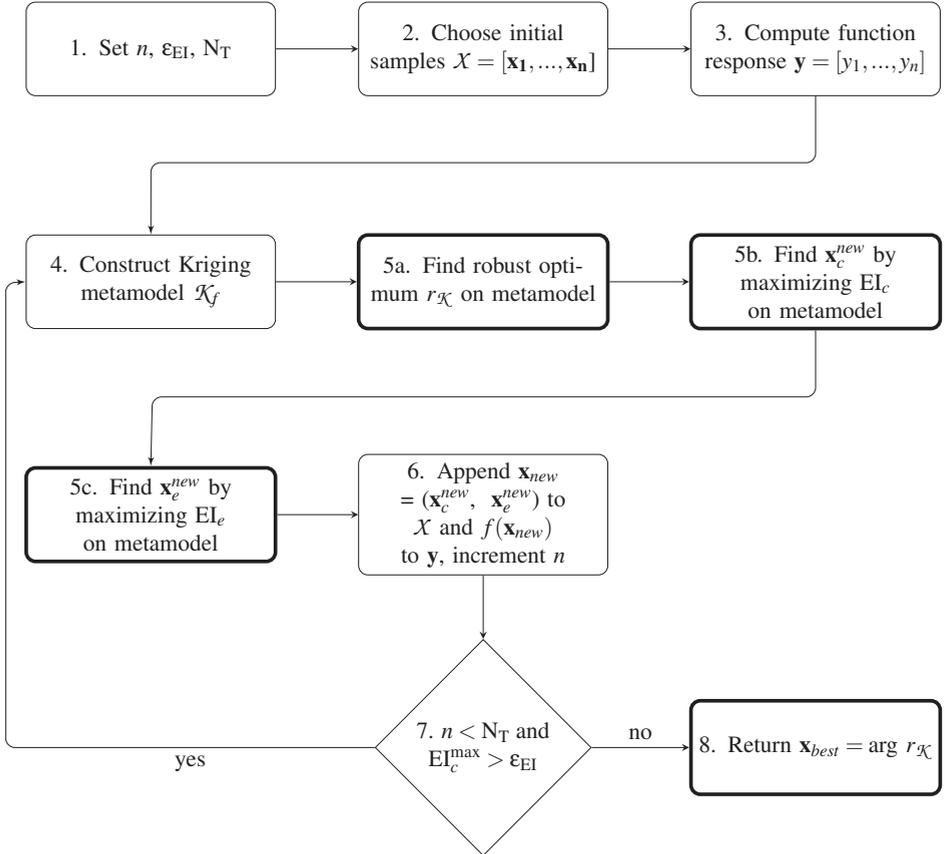


Figure 3.3: Flowchart shows the algorithm for efficient global robust optimization of parametric uncertainties affected problems (EGRO). The steps with the bold borders represent the changes that have been made to the EGO algorithm in Fig. 3.2 in order to incorporate robust optimization in the presence of parametric uncertainties.

Fig. 3.3 shows a flowchart that helps to visualize the main steps involved in the proposed algorithm. Comparing EGO (Fig. 3.2) and EGRO (Fig. 3.3), we observe that four steps, highlighted by a bold border for the flowchart boxes, have been added.

After initialization in Step 1, n initial sampling locations are chosen in the combined control and environment variable space through a space-filling technique in Step 2. The

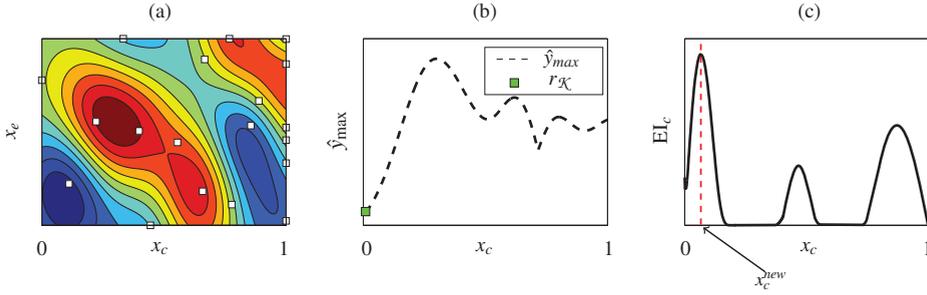


Figure 3.4: The Kriging metamodel of a two-dimensional function is shown in (a). The worst-case Kriging prediction, \hat{y}_{\max} , is plotted in (b). The reference robust optimum $r_{\mathcal{K}}$ is the minimum value obtained on \hat{y}_{\max} . EI_c is plotted in (c). \mathbf{x}_c^{new} is the global maximizer of EI_c .

response at these sampling locations is evaluated on the expensive function in Step 3. Thereafter, a Kriging metamodel \mathcal{K}_f of $f(\mathbf{x})$ is constructed using the samples and responses. Step 5 is divided into three sub-steps. In Step 5a, a reference robust optimum solution $r_{\mathcal{K}}$ is searched for on the constructed metamodel \mathcal{K}_f ,

$$r_{\mathcal{K}} = \min_{\mathbf{x}_c \in \mathbb{X}_c} \max_{\mathbf{x}_e \in \mathbb{X}_e} \mathcal{K}_f(\mathbf{x}_c, \mathbf{x}_e). \quad (3.8)$$

In Step 5b and 5c, the algorithm identifies the next location \mathbf{x}_{new} at which the expensive function should be evaluated. The location corresponding to the highest expectation of improvement over the current robust optimum $r_{\mathcal{K}}$ is chosen as \mathbf{x}_{new} . The search for \mathbf{x}_{new} is performed by consecutively finding the control variables \mathbf{x}_c^{new} and the environment variables \mathbf{x}_e^{new} . The key steps are treated in the next subsections.

3.4.3. OPTIMAL SAMPLING LOCATION IN \mathbb{X}_c

In order to find \mathbf{x}_c^{new} we limit the search space to the control variable space \mathbb{X}_c only. \mathbf{x}_c^{new} should be the location in the control variable space that is expected to give the highest improvement over the current robust optimum $r_{\mathcal{K}}$. To compute the expectation of improvement we need to know the worst-case Kriging prediction across the whole control variable space:

$$\hat{y}_{\max}(\mathbf{x}_c) = \max_{\mathbf{x}_e \in \mathbb{X}_e} \mathcal{K}_f(\mathbf{x}_c, \mathbf{x}_e) \quad (3.9)$$

and the corresponding maximizer in the environment variable space \mathbb{X}_e is denoted by \mathbf{x}_e^{max} .

Fig. 3.4 visualizes the steps described so far. Fig. 3.4a shows a Kriging metamodel of the damped cosine function of one control variable and one environment variable plotted in Fig. 3.1. The worst-case Kriging prediction \hat{y}_{\max} is plotted in Fig. 3.4b. The robust optimum $r_{\mathcal{K}}$ is the minimum value on this curve. $r_{\mathcal{K}}$ is determined in Step 5a.

As in deterministic expected improvement, we assume that the uncertainty in the value of the worst-case Kriging prediction, \hat{y}_{\max} , at any point $(\mathbf{x}_c, \mathbf{x}_e^{max})$ can be modelled using a normally distributed random variable Y_{max} with mean \hat{y}_{\max} and variance $s^2(\mathbf{x}_c, \mathbf{x}_e^{max})$. The term $s^2(\mathbf{x}_c, \mathbf{x}_e^{max})$ represents the Kriging mean squared error at $(\mathbf{x}_c, \mathbf{x}_e^{max})$.

An improvement of the worst-case Kriging prediction \hat{y}_{\max} over the current robust optimum $r_{\mathcal{X}}$ occurs when $Y_{\max} < r_{\mathcal{X}}$. The expected improvement is found by computing the expected value of the improvement $I_c = \max(r_{\mathcal{X}} - Y_{\max}, 0)$ under the normal distribution setting,

$$\underbrace{E[I_c(\mathbf{x}_c)]}_{\text{EI}_c} = \int_{I_c=0}^{I_c=\infty} I_c \frac{\exp\left(-\frac{t_c^2}{2}\right)}{\sqrt{2\pi}s} dt_c, \quad (3.10)$$

with

$$t_c = \frac{r_{\mathcal{X}} - I_c - \hat{y}_{\max}}{s}, \quad s = s(\mathbf{x}_c, \mathbf{x}_e^{\max}) \quad (3.11)$$

Since the standard normal probability density function is defined as

$$\phi(z) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right), \quad (3.12)$$

the modified expected improvement criterion EI_c can be simplified as

$$\begin{aligned} E[I_c(\mathbf{x}_c)] &= (r_{\mathcal{X}} - \hat{y}_{\max}) \int_{t_c=-\infty}^{t_c=\frac{r_{\mathcal{X}} - \hat{y}_{\max}}{s}} \phi(t_c) dt_c \\ &- s \int_{t_c=-\infty}^{t_c=\frac{r_{\mathcal{X}} - \hat{y}_{\max}}{s}} t_c \phi(t_c) dt_c. \end{aligned} \quad (3.13)$$

We recognize the first integral in (3.13) as the normal cumulative distribution function $\Phi\left(\frac{r_{\mathcal{X}} - \hat{y}_{\max}}{s}\right)$. The second integral in (3.13) can be solved by using the substitution $z = \frac{-t_c}{2}$. The final expression for EI_c is

$$\begin{aligned} E[I_c(\mathbf{x}_c)] &= (r_{\mathcal{X}} - \hat{y}_{\max}) \Phi\left(\frac{r_{\mathcal{X}} - \hat{y}_{\max}}{s}\right) \\ &+ s \phi\left(\frac{r_{\mathcal{X}} - \hat{y}_{\max}}{s}\right). \end{aligned} \quad (3.14)$$

The global maximizer of Equation (3.14) is the new control variable location $\mathbf{x}_c^{\text{new}}$ at which the expensive function should be evaluated since this is the control variables location that gives the highest expectation of improvement. The flowchart in Fig. 3.3 shows that we search for this global maximizer in Step 5b. Fig. 3.4c shows the expected improvement EI_c as a function of x_c , along with the location of the maximizer x_c^{new} . Computing $r_{\mathcal{X}}$ and EI_c could be computationally expensive since they both involve estimation of \hat{y}_{\max} . For a more in-depth discussion on calculation of $r_{\mathcal{X}}$ and EI_c please refer to Appendix B.

3.4.4. OPTIMAL SAMPLING LOCATION IN \mathbb{X}_e

Once $\mathbf{x}_c^{\text{new}}$ has been identified, we need to find the location \mathbf{x}_e , in the environment variable space \mathbb{X}_e , at which the expensive function should be evaluated. In this space, the

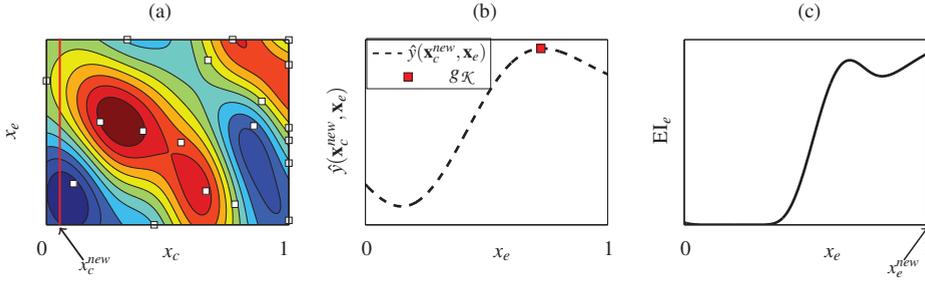


Figure 3.5: The Kriging metamodel of a two-dimensional function is shown in (a) along with the location of \mathbf{x}_c^{new} . The Kriging prediction at \mathbf{x}_c^{new} , corresponding to the red line on the left plot, is plotted as a one-dimensional function with respect to \mathbf{x}_e in (b). EI_e is plotted in (c). \mathbf{x}_e^{new} is the global maximizer of EI_e .

worst-case cost (maximum) is of interest. Let the deterministic maximum on the meta-model, with respect to \mathbf{x}_e , at the new control variable location \mathbf{x}_c^{new} be given by

$$g_{\mathcal{K}}(\mathbf{x}_c^{new}, \mathbf{x}_e) = \max_{\mathbf{x}_e \in \mathcal{X}_e} \mathcal{K}_f(\mathbf{x}_c^{new}, \mathbf{x}_e). \quad (3.15)$$

Fig. 3.5a shows the same Kriging metamodel of the damped cosine problem as Fig. 3.4a. The new sampling location in the control variable space, \mathbf{x}_c^{new} , is indicated. The Kriging surface along the red line at \mathbf{x}_c^{new} is reproduced as a function of \mathbf{x}_e in Fig. 3.5b. The maximum on this slice of the Kriging surface gives the reference worst-case cost $g_{\mathcal{K}}$.

In order to find \mathbf{x}_e^{new} , a modified expected improvement criterion is employed once again. To formulate the EI measure, it is again assumed that the uncertainty in the value of the Kriging prediction $\hat{y}(\mathbf{x}_c^{new}, \mathbf{x}_e)$ at any point $(\mathbf{x}_c^{new}, \mathbf{x}_e)$ can be modelled using a normally distributed random variable Y with mean \hat{y} and variance $s^2(\mathbf{x}_c^{new}, \mathbf{x}_e)$.

Note that when computing an improvement in the control variable space we were searching for the global minimum of the worst-case function. In the environment variable space we are searching for the global maximum of the function at \mathbf{x}_c^{new} . An improvement is sought over the current maximum $g_{\mathcal{K}}$ and this occurs when $Y > g_{\mathcal{K}}$. We define an improvement I_e over $g_{\mathcal{K}}$ as $I_e = \max(Y - g_{\mathcal{K}}, 0)$. Given the normal distribution setting, the expected improvement can be written as,

$$\underbrace{E[I_e(\mathbf{x}_c^{new}, \mathbf{x}_e)]}_{EI_e} = \int_{I_e=0}^{I_e=\infty} I_e \frac{\exp\left(-\frac{t_e^2}{2}\right)}{\sqrt{2\pi}s} dt_e, \quad (3.16)$$

with

$$t_e = \frac{\hat{y} - I_e - g_{\mathcal{K}}}{s}, \quad s = s(\mathbf{x}_c^{new}, \mathbf{x}_e). \quad (3.17)$$

Using derivation similar to the one shown for EI_c , the expected improvement simplifies to

$$\underbrace{E[I_e(\mathbf{x}_c^{new}, \mathbf{x}_e)]}_{EI_e} = (\hat{y} - g_{\mathcal{K}}) \Phi\left(\frac{\hat{y} - g_{\mathcal{K}}}{s}\right) + s \phi\left(\frac{\hat{y} - g_{\mathcal{K}}}{s}\right). \quad (3.18)$$

The global maximizer of Equation (3.18) is the new environment variable location \mathbf{x}_e^{new} at which the expensive function should be evaluated. Step 5c in Fig. 3.3 involves the search for this global maximizer. Fig. 3.5c shows the expected improvement EI_e as a function of x_e , and the location of the maximizer x_e^{new} .

Next, in Step 6 of the flowchart, the expensive to evaluate function f is sampled at $(\mathbf{x}_c^{new}, \mathbf{x}_e^{new})$. Step 7 is a conditional statement. If the total number of samples available is not exhausted or EI_c^{max} is greater than the threshold ϵ_{EI} , the algorithm returns to Step 4 where the Kriging metamodel is reconstructed with the additional sample and the process of finding \mathbf{x}_{new} is repeated. Otherwise, the algorithm terminates in Step 8 and the argument of the last robust optimum found, $r_{\mathcal{X}}$, is returned as the robust optimum location, \mathbf{x}_{best} .

3.4.5. ALGORITHM CHOICES AND RATIONALE

Here we discuss two important choices made in the definition of the EGRO algorithm.

1. Use of EI_e

Instead of computing the expected improvement EI_e in the environment variable space, a simpler option could have been to sample the location of the deterministic maximum $g_{\mathcal{X}}$. However, sampling the location of the deterministic maximum at \mathbf{x}_c^{new} is not viable since this often leads to the algorithm resampling the same location in the next iterations. The algorithm would stall in such a situation and would fail to converge. Furthermore, not only would the resampling be a waste of expensive function evaluations, but it would also cause the Kriging correlation matrix \mathbf{R} to become ill-conditioned. The use of EI_e effectively avoids such problems because the uncertainty near sampled points is low, resulting in low EI values.

2. Formulation of EI_c

The expression for EI_c , Equation (3.10), is approximate since we lack vital information to find the exact expression. In our formulation of EI_c , we first evaluated the deterministic maximum \hat{y}_{max} , with respect to the environment variable space \mathbb{X}_e , for a fixed \mathbf{x}_c . This evaluation was repeated until the worst-case Kriging prediction was known for the complete \mathbb{X}_c space. Thereafter, we imposed the assumption that each point on the deterministically found worst-case Kriging prediction surface is actually a random variable, whose mean is given by \hat{y}_{max} and variance is given by the Kriging mean squared error. This assumption enabled us to find an analytical expression for EI_c (3.14) similar to the one for the nominal EI (3.7).

To find the exact expression for EI_c , we need to first retain the assumption from deterministic EGO that each point on the Kriging surface is a random variable, whose mean is given by the Kriging prediction and variance is given by the Kriging mean squared error. Now we are interested in finding the distribution of the maximum [33], with respect to the environment variables, for a fixed \mathbf{x}_c . However, to perform this operation, additional assumptions concerning the joint distribution of the random variables are required. In all likelihood, the distribution of the maximum would be non-Gaussian and the computation of expected improvement would require numerical integration [34]. The lack of joint distribution information and the high cost of performing numerical integration makes exact evaluation of EI_c less attractive. Therefore, a more pragmatic approach of computing the worst-case Kriging prediction as a deterministic quantity and imposing

a Gaussian distribution assumption on \hat{y}_{\max} was taken to find an analytical expression for EI_c . Further details on the rationale behind this approach may be found in [34]. To what extent this choice affects the effectiveness of EGRO is investigated using numerical benchmark tests.

3.5. RESULTS

3.5.1. TEST PROBLEMS AND EVALUATION METHODOLOGY

The algorithm is tested on a set of standard robust optimization problems from literature of varying dimensions and type. The 13 test problems are provided in the Appendix. Problems 1 to 7 have been applied in [7, 27] while problems 8 to 13 have been used in [8, 12, 13, 15, 26, 27].

To illustrate the practical value of the algorithm, an engineering case study has been added to the set of numerical examples. The engineering problem is that of an optical filter that can be affected by manufacturing uncertainties, leading to deterioration in parameters of interest.

The first seven problems, f_1 to f_7 , are convex in the control variable space \mathbf{x}_c and concave in the environment variables space \mathbf{x}_e . These problems have been optimized using saddle-point optimization in [7] while a Kriging-based optimization approach has been used in [27]. Problem f_1 to f_3 are 4 dimensional with 2 dimensions in \mathbb{X}_c and 2 dimensions in \mathbb{X}_e . The largest problem is f_7 which has 5 dimensions in \mathbb{X}_c and 5 dimensions in \mathbb{X}_e .

Test problems f_8 to f_{13} have been widely used by the evolutionary optimization community [12–14] to test evolutionary methods designed to optimize min-max problems. The same problems have also been tested by Marzat *et al.* [26, 27]. The problems have a relatively smaller size, with a maximum of 4 dimensions. Some of the test problems, such as f_{10} and f_{11} , are especially difficult to optimize due to the non-convex and highly multimodal behaviour of the functions.

All the test problems used as numerical examples in this work are established benchmark problems for robust optimization using surrogates. By using these benchmarks, the performance of the proposed algorithm can easily be compared against state of the art methods in literature in terms of number of expensive function calls needed to reach the global robust optimum. The reference results for function f_1 to f_7 have been solved to theoretically global robust optimality [7]. Amongst the algorithms in literature, the MiMareK 1 and MiMareK 2 algorithms [26, 27] converge to the global robust optimum in the lowest number of expensive function evaluations. In this work, we compare the converge speed of EGRO in terms of number of expensive function evaluations required to reach the global robust optimum against currently available methods.

When constructing the metamodel, we choose the initial sampling locations using a space-filling technique such as Latin hypercube sampling. Since this type of initial sampling is non-deterministic, the algorithm is run 100 times on each test case and the results are averaged. The repeatability of the algorithm for random initial sampling is thereby tested. The average results are compared against other techniques used for min-max optimization.

For all problems, the initial number of sampling locations are chosen as $n = 10 \times n_d$,

where n_d represents the number of dimensions. The threshold for termination for EI_c^{\max} is fixed at $\epsilon = 10^{-7}$. The maximum number of function evaluations available is different for each function. However, this number is not allowed to exceed $N_T = 35 \times n_d$ for any problem.

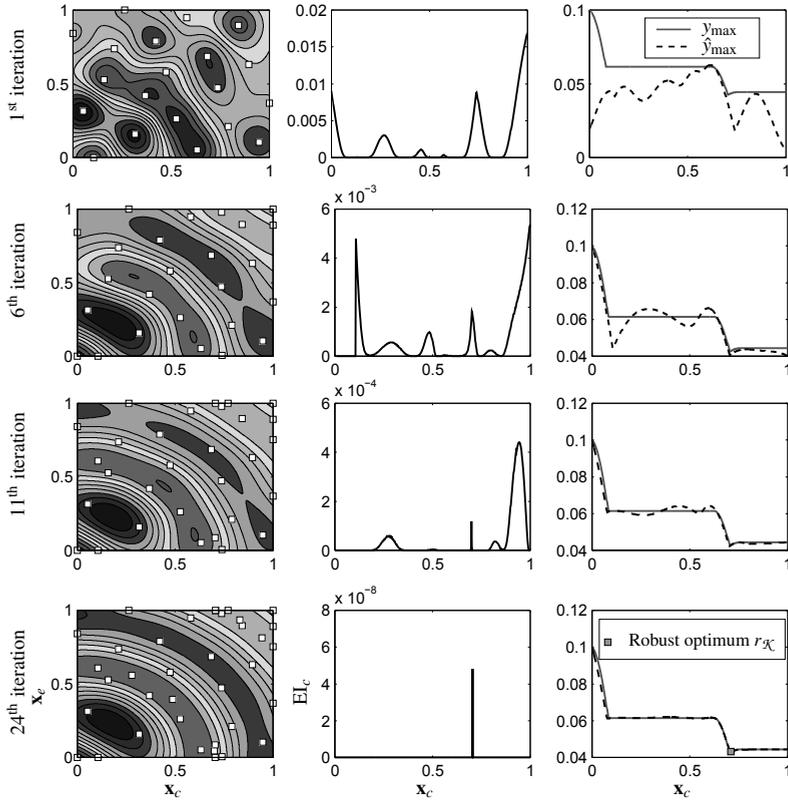


Figure 3.6: Snapshots of the progress of the algorithm applied to $f_{11}(\mathbf{x}_c, \mathbf{x}_e)$. The left column shows the Kriging metamodel \mathcal{K}_f of the function. The best worst-case expected improvement EI_c is plotted in the central column. The right column shows the deterministic worst-case cost \hat{y}_{\max} on \mathcal{K}_f and the true worst-case cost y_{\max} , which is provided as a reference. By the 24th iteration the robust optimum has been found.

3.5.2. ILLUSTRATIVE EXAMPLE

Before performing a detailed analysis of EGRO's numerical performance, we illustrate the evolution of the algorithm as it searches for the global robust optimum by applying it on a two-dimensional problem. The function, which is basically a damped cosine wave, is plotted in Fig. 3.1.

The problem is multimodal, non-convex and a function of one control variable and

one environment variable. Since the problem is only 2-dimensional, it is relatively easy to visualize the choices made by EGRO at each iteration of the algorithm as it searches for the global robust optimum. The initial variables of EGRO are set to $n = 20$, $\epsilon_{EI} = 10^{-7}$

Table 3.1: Reference results of all the test problems. The functions are listed in the Appendix. The reference results have been obtained from [7, 27].

Test function	\mathbb{X}_c	\mathbb{X}_e	\mathbf{x}_c	\mathbf{x}_e	$f_i(\mathbf{x}_c, \mathbf{x}_e)$	n_d
$f_1(\mathbf{x}_c, \mathbf{x}_e)$	$[-5, 5]^2$	$[-5, 5]^2$	-0.4833	0.0833	-1.6833	4
			-0.3167	-0.0833		
$f_2(\mathbf{x}_c, \mathbf{x}_e)$	$[-5, 5]^2$	$[-5, 5]^2$	1.6954	0.7186	1.4039	4
			-0.0032	-0.0001		
$f_3(\mathbf{x}_c, \mathbf{x}_e)$	$[-5, 5]^2$	$[-3, 3]^2$	-1.1807	2.0985	-2.4688	4
			0.9128	2.666		
$f_4(\mathbf{x}_c, \mathbf{x}_e)$	$[-5, 5]^2$	$[-3, 3]^3$	0.4181	0.709	-0.1348	5
			0.4181	1.0874		
				0.709		
$f_5(\mathbf{x}_c, \mathbf{x}_e)$	$[-5, 5]^3$	$[-1, 1]^3$	0.1111	0.4444	1.345	6
			0.1538	0.9231		
			0.2	0.4		
$f_6(\mathbf{x}_c, \mathbf{x}_e)$	$[-5, 5]^4$	$[-2, 2]^3$	-0.2316	0.6195	4.543	7
			0.2229	0.3535		
			-0.6755	1.478		
			-0.0838			
$f_7(\mathbf{x}_c, \mathbf{x}_e)$	$[-5, 5]^5$	$[-3, 3]^5$	1.4252	0.5156	-6.3509	10
			1.6612	0.8798		
			1.2585	0.2919		
			-0.9744	0.1198		
			-0.7348	-0.1198		
$f_8(\mathbf{x}_c, \mathbf{x}_e)$	$[0, 10]$	$[0, 10]$	5	5	0	2
$f_9(\mathbf{x}_c, \mathbf{x}_e)$	$[0, 10]$	$[0, 10]$	0	0	3	2
$f_{10}(\mathbf{x}_c, \mathbf{x}_e)$	$[0, 10]$	$[0, 10]$	10	2.1257	0.0978	2
$f_{11}(\mathbf{x}_c, \mathbf{x}_e)$	$[0, 10]$	$[0, 10]$	7.0441	10	0.0425	2
$f_{12}(\mathbf{x}_c, \mathbf{x}_e)$	$[-0.5, 0.5]$	$[0, 10]^2$	0.5	0	0.25	4
	$[0, 1]$		0.25	0		
$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	$[-1, 3]^2$	$[0, 10]^2$	1	Any	1	4
			1	Any		

and $N_T = 50$. We aim to find the robust optimum within 50 evaluations of $f_{11}(\mathbf{x}_c, \mathbf{x}_e)$. Latin hypercube sampling is used to choose $n = 20$ locations where the expensive function is evaluated. Before choosing these locations both the dimensions are scaled such

that they lie in the range $[0, 1]$. Given this range for the dimensions, the global robust optimum of the problem is located at $\mathbf{x} = (0.7044, 1)$.

The Kriging metamodel \mathcal{K}_f is built based on the sampling locations and responses. Fig. 3.6 shows how the algorithm evolves in its search for the robust optimum of the problem. The numbers on the far left represent the iteration of the algorithm. The column on the left shows the Kriging metamodel \mathcal{K}_f of the function. The expected improvement in the control variable space EI_c is plotted in the central column. The column on the right shows the deterministic worst-case cost \hat{y}_{\max} on \mathcal{K}_f and the true worst-case cost y_{\max} , computed on $f_{11}(\mathbf{x}_c, \mathbf{x}_e)$. The true worst-case cost y_{\max} is provided as a reference and the algorithm does not have access to it. It is interesting to note that y_{\max} , with its peaks and flat plateaus, looks slightly like a step or a staircase function. The worst-case cost experiences a very slight dip at $\mathbf{x}_c = 0.7044$. It is at this location that the worst-case cost is minimum and therefore this is the robust optimum location in \mathbb{X}_c .

Table 3.2: The average numerical performance of EGRO based on 100 runs is evaluated on the 13 test problems provided in the Appendix. The table shows the mean and standard deviation of the robust optimum along with the average robust optimum locations in \mathbb{X}_c and \mathbb{X}_e . The total number of evaluations per dimension on the expensive function is shown in the right-most column for each problem.

Test function	Average \mathbf{x}_c	Average \mathbf{x}_e	Average $f_i(\mathbf{x}_c, \mathbf{x}_e)$	Standard deviation f_i	Total evaluations n_f Number of dimensions n_d																																																																																																				
$f_1(\mathbf{x}_c, \mathbf{x}_e)$	-0.4830	0.0828	-1.6833	2.15×10^{-5}	24																																																																																																				
	-0.3168	-0.0828				$f_2(\mathbf{x}_c, \mathbf{x}_e)$	1.6953	0.7183	1.4039	1.5×10^{-3}	27	-0.0031	-0.0015	$f_3(\mathbf{x}_c, \mathbf{x}_e)$	-1.1908	2.1060	-2.4689	7.4×10^{-2}	32	0.9282	2.6966	$f_4(\mathbf{x}_c, \mathbf{x}_e)$	0.4181	0.7082	-0.1348	2.1685×10^{-4}	25	0.4156	1.0907	$f_5(\mathbf{x}_c, \mathbf{x}_e)$	0.1113	0.4463	1.3453	1.8286×10^{-4}	23	0.1544	0.9262	$f_6(\mathbf{x}_c, \mathbf{x}_e)$	0.2008	0.4022	4.543	3.1×10^{-3}	34	-0.2318	0.6180	$f_7(\mathbf{x}_c, \mathbf{x}_e)$	0.2233	0.3533	-6.3509	4.3×10^{-3}	29	-0.6747	1.4775	$f_8(\mathbf{x}_c, \mathbf{x}_e)$	-0.0840	0.5176	0	8.9×10^{-8}	11	1.4250	0.8783	$f_9(\mathbf{x}_c, \mathbf{x}_e)$	1.6608	0.2949	3	1.49×10^{-2}	18	1.2569	0.1201	$f_{10}(\mathbf{x}_c, \mathbf{x}_e)$	-0.9738	-0.1218	0.0978	3.47×10^{-4}	25	-0.7346	5	$f_{11}(\mathbf{x}_c, \mathbf{x}_e)$	5	0	0.0425	1.40×10^{-6}	30	0	0	$f_{12}(\mathbf{x}_c, \mathbf{x}_e)$	10	2.1181	0.251	2.7×10^{-3}	11	7.0496	10	$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.5	0	0.997	5.6×10^{-3}	16	0.25	0	$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.999	-	-
$f_2(\mathbf{x}_c, \mathbf{x}_e)$	1.6953	0.7183	1.4039	1.5×10^{-3}	27																																																																																																				
	-0.0031	-0.0015				$f_3(\mathbf{x}_c, \mathbf{x}_e)$	-1.1908	2.1060	-2.4689	7.4×10^{-2}	32	0.9282	2.6966	$f_4(\mathbf{x}_c, \mathbf{x}_e)$	0.4181	0.7082	-0.1348	2.1685×10^{-4}	25	0.4156	1.0907	$f_5(\mathbf{x}_c, \mathbf{x}_e)$	0.1113	0.4463	1.3453	1.8286×10^{-4}	23	0.1544	0.9262	$f_6(\mathbf{x}_c, \mathbf{x}_e)$	0.2008	0.4022	4.543	3.1×10^{-3}	34	-0.2318	0.6180	$f_7(\mathbf{x}_c, \mathbf{x}_e)$	0.2233	0.3533	-6.3509	4.3×10^{-3}	29	-0.6747	1.4775	$f_8(\mathbf{x}_c, \mathbf{x}_e)$	-0.0840	0.5176	0	8.9×10^{-8}	11	1.4250	0.8783	$f_9(\mathbf{x}_c, \mathbf{x}_e)$	1.6608	0.2949	3	1.49×10^{-2}	18	1.2569	0.1201	$f_{10}(\mathbf{x}_c, \mathbf{x}_e)$	-0.9738	-0.1218	0.0978	3.47×10^{-4}	25	-0.7346	5	$f_{11}(\mathbf{x}_c, \mathbf{x}_e)$	5	0	0.0425	1.40×10^{-6}	30	0	0	$f_{12}(\mathbf{x}_c, \mathbf{x}_e)$	10	2.1181	0.251	2.7×10^{-3}	11	7.0496	10	$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.5	0	0.997	5.6×10^{-3}	16	0.25	0	$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.999	-	-	-	-	0.999	-				
$f_3(\mathbf{x}_c, \mathbf{x}_e)$	-1.1908	2.1060	-2.4689	7.4×10^{-2}	32																																																																																																				
	0.9282	2.6966				$f_4(\mathbf{x}_c, \mathbf{x}_e)$	0.4181	0.7082	-0.1348	2.1685×10^{-4}	25	0.4156	1.0907	$f_5(\mathbf{x}_c, \mathbf{x}_e)$	0.1113	0.4463	1.3453	1.8286×10^{-4}	23	0.1544	0.9262	$f_6(\mathbf{x}_c, \mathbf{x}_e)$	0.2008	0.4022	4.543	3.1×10^{-3}	34	-0.2318	0.6180	$f_7(\mathbf{x}_c, \mathbf{x}_e)$	0.2233	0.3533	-6.3509	4.3×10^{-3}	29	-0.6747	1.4775	$f_8(\mathbf{x}_c, \mathbf{x}_e)$	-0.0840	0.5176	0	8.9×10^{-8}	11	1.4250	0.8783	$f_9(\mathbf{x}_c, \mathbf{x}_e)$	1.6608	0.2949	3	1.49×10^{-2}	18	1.2569	0.1201	$f_{10}(\mathbf{x}_c, \mathbf{x}_e)$	-0.9738	-0.1218	0.0978	3.47×10^{-4}	25	-0.7346	5	$f_{11}(\mathbf{x}_c, \mathbf{x}_e)$	5	0	0.0425	1.40×10^{-6}	30	0	0	$f_{12}(\mathbf{x}_c, \mathbf{x}_e)$	10	2.1181	0.251	2.7×10^{-3}	11	7.0496	10	$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.5	0	0.997	5.6×10^{-3}	16	0.25	0	$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.999	-	-	-	-	0.999	-												
$f_4(\mathbf{x}_c, \mathbf{x}_e)$	0.4181	0.7082	-0.1348	2.1685×10^{-4}	25																																																																																																				
	0.4156	1.0907				$f_5(\mathbf{x}_c, \mathbf{x}_e)$	0.1113	0.4463	1.3453	1.8286×10^{-4}	23	0.1544	0.9262	$f_6(\mathbf{x}_c, \mathbf{x}_e)$	0.2008	0.4022	4.543	3.1×10^{-3}	34	-0.2318	0.6180	$f_7(\mathbf{x}_c, \mathbf{x}_e)$	0.2233	0.3533	-6.3509	4.3×10^{-3}	29	-0.6747	1.4775	$f_8(\mathbf{x}_c, \mathbf{x}_e)$	-0.0840	0.5176	0	8.9×10^{-8}	11	1.4250	0.8783	$f_9(\mathbf{x}_c, \mathbf{x}_e)$	1.6608	0.2949	3	1.49×10^{-2}	18	1.2569	0.1201	$f_{10}(\mathbf{x}_c, \mathbf{x}_e)$	-0.9738	-0.1218	0.0978	3.47×10^{-4}	25	-0.7346	5	$f_{11}(\mathbf{x}_c, \mathbf{x}_e)$	5	0	0.0425	1.40×10^{-6}	30	0	0	$f_{12}(\mathbf{x}_c, \mathbf{x}_e)$	10	2.1181	0.251	2.7×10^{-3}	11	7.0496	10	$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.5	0	0.997	5.6×10^{-3}	16	0.25	0	$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.999	-	-	-	-	0.999	-																				
$f_5(\mathbf{x}_c, \mathbf{x}_e)$	0.1113	0.4463	1.3453	1.8286×10^{-4}	23																																																																																																				
	0.1544	0.9262				$f_6(\mathbf{x}_c, \mathbf{x}_e)$	0.2008	0.4022	4.543	3.1×10^{-3}	34	-0.2318	0.6180	$f_7(\mathbf{x}_c, \mathbf{x}_e)$	0.2233	0.3533	-6.3509	4.3×10^{-3}	29	-0.6747	1.4775	$f_8(\mathbf{x}_c, \mathbf{x}_e)$	-0.0840	0.5176	0	8.9×10^{-8}	11	1.4250	0.8783	$f_9(\mathbf{x}_c, \mathbf{x}_e)$	1.6608	0.2949	3	1.49×10^{-2}	18	1.2569	0.1201	$f_{10}(\mathbf{x}_c, \mathbf{x}_e)$	-0.9738	-0.1218	0.0978	3.47×10^{-4}	25	-0.7346	5	$f_{11}(\mathbf{x}_c, \mathbf{x}_e)$	5	0	0.0425	1.40×10^{-6}	30	0	0	$f_{12}(\mathbf{x}_c, \mathbf{x}_e)$	10	2.1181	0.251	2.7×10^{-3}	11	7.0496	10	$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.5	0	0.997	5.6×10^{-3}	16	0.25	0	$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.999	-	-	-	-	0.999	-																												
$f_6(\mathbf{x}_c, \mathbf{x}_e)$	0.2008	0.4022	4.543	3.1×10^{-3}	34																																																																																																				
	-0.2318	0.6180				$f_7(\mathbf{x}_c, \mathbf{x}_e)$	0.2233	0.3533	-6.3509	4.3×10^{-3}	29	-0.6747	1.4775	$f_8(\mathbf{x}_c, \mathbf{x}_e)$	-0.0840	0.5176	0	8.9×10^{-8}	11	1.4250	0.8783	$f_9(\mathbf{x}_c, \mathbf{x}_e)$	1.6608	0.2949	3	1.49×10^{-2}	18	1.2569	0.1201	$f_{10}(\mathbf{x}_c, \mathbf{x}_e)$	-0.9738	-0.1218	0.0978	3.47×10^{-4}	25	-0.7346	5	$f_{11}(\mathbf{x}_c, \mathbf{x}_e)$	5	0	0.0425	1.40×10^{-6}	30	0	0	$f_{12}(\mathbf{x}_c, \mathbf{x}_e)$	10	2.1181	0.251	2.7×10^{-3}	11	7.0496	10	$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.5	0	0.997	5.6×10^{-3}	16	0.25	0	$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.999	-	-	-	-	0.999	-																																				
$f_7(\mathbf{x}_c, \mathbf{x}_e)$	0.2233	0.3533	-6.3509	4.3×10^{-3}	29																																																																																																				
	-0.6747	1.4775				$f_8(\mathbf{x}_c, \mathbf{x}_e)$	-0.0840	0.5176	0	8.9×10^{-8}	11	1.4250	0.8783	$f_9(\mathbf{x}_c, \mathbf{x}_e)$	1.6608	0.2949	3	1.49×10^{-2}	18	1.2569	0.1201	$f_{10}(\mathbf{x}_c, \mathbf{x}_e)$	-0.9738	-0.1218	0.0978	3.47×10^{-4}	25	-0.7346	5	$f_{11}(\mathbf{x}_c, \mathbf{x}_e)$	5	0	0.0425	1.40×10^{-6}	30	0	0	$f_{12}(\mathbf{x}_c, \mathbf{x}_e)$	10	2.1181	0.251	2.7×10^{-3}	11	7.0496	10	$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.5	0	0.997	5.6×10^{-3}	16	0.25	0	$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.999	-	-	-	-	0.999	-																																												
$f_8(\mathbf{x}_c, \mathbf{x}_e)$	-0.0840	0.5176	0	8.9×10^{-8}	11																																																																																																				
	1.4250	0.8783				$f_9(\mathbf{x}_c, \mathbf{x}_e)$	1.6608	0.2949	3	1.49×10^{-2}	18	1.2569	0.1201	$f_{10}(\mathbf{x}_c, \mathbf{x}_e)$	-0.9738	-0.1218	0.0978	3.47×10^{-4}	25	-0.7346	5	$f_{11}(\mathbf{x}_c, \mathbf{x}_e)$	5	0	0.0425	1.40×10^{-6}	30	0	0	$f_{12}(\mathbf{x}_c, \mathbf{x}_e)$	10	2.1181	0.251	2.7×10^{-3}	11	7.0496	10	$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.5	0	0.997	5.6×10^{-3}	16	0.25	0	$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.999	-	-	-	-	0.999	-																																																				
$f_9(\mathbf{x}_c, \mathbf{x}_e)$	1.6608	0.2949	3	1.49×10^{-2}	18																																																																																																				
	1.2569	0.1201				$f_{10}(\mathbf{x}_c, \mathbf{x}_e)$	-0.9738	-0.1218	0.0978	3.47×10^{-4}	25	-0.7346	5	$f_{11}(\mathbf{x}_c, \mathbf{x}_e)$	5	0	0.0425	1.40×10^{-6}	30	0	0	$f_{12}(\mathbf{x}_c, \mathbf{x}_e)$	10	2.1181	0.251	2.7×10^{-3}	11	7.0496	10	$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.5	0	0.997	5.6×10^{-3}	16	0.25	0	$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.999	-	-	-	-	0.999	-																																																												
$f_{10}(\mathbf{x}_c, \mathbf{x}_e)$	-0.9738	-0.1218	0.0978	3.47×10^{-4}	25																																																																																																				
	-0.7346	5				$f_{11}(\mathbf{x}_c, \mathbf{x}_e)$	5	0	0.0425	1.40×10^{-6}	30	0	0	$f_{12}(\mathbf{x}_c, \mathbf{x}_e)$	10	2.1181	0.251	2.7×10^{-3}	11	7.0496	10	$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.5	0	0.997	5.6×10^{-3}	16	0.25	0	$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.999	-	-	-	-	0.999	-																																																																				
$f_{11}(\mathbf{x}_c, \mathbf{x}_e)$	5	0	0.0425	1.40×10^{-6}	30																																																																																																				
	0	0				$f_{12}(\mathbf{x}_c, \mathbf{x}_e)$	10	2.1181	0.251	2.7×10^{-3}	11	7.0496	10	$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.5	0	0.997	5.6×10^{-3}	16	0.25	0	$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.999	-	-	-	-	0.999	-																																																																												
$f_{12}(\mathbf{x}_c, \mathbf{x}_e)$	10	2.1181	0.251	2.7×10^{-3}	11																																																																																																				
	7.0496	10				$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.5	0	0.997	5.6×10^{-3}	16	0.25	0	$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.999	-	-	-	-	0.999	-																																																																																				
$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.5	0	0.997	5.6×10^{-3}	16																																																																																																				
	0.25	0				$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.999	-	-	-	-	0.999	-																																																																																												
$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.999	-	-	-	-																																																																																																				
	0.999	-																																																																																																							

At the first iteration, the Kriging metamodel, constructed using $n = 20$ initial samples, seems to have captured the general trend of the true function $f_{11}(\mathbf{x}_c, \mathbf{x}_e)$. However,

the worst-case cost \hat{y}_{\max} on the metamodel does not approximate the true worst-case cost y_{\max} too well at this stage. We note that for the first iteration El_c is maximum at $\mathbf{x}_c = 1$ and this is where a new sample is added in the \mathbb{X}_c domain. By the 6th iteration, there is significant improvement in the global accuracy of both the nominal Kriging metamodel \mathcal{K}_f as well as the worst-case cost \hat{y}_{\max} . However, locally, the plots show that there is significant room for improvement in some parts of the design landscape, such as between $\mathbf{x}_c = 0.1$ to $\mathbf{x}_c = 0.6$ in the worst-case cost plot. It is interesting to note that by the 6th iteration, \hat{y}_{\max} approximates y_{\max} very well at the location of the robust optimum at $\mathbf{x}_c = 0.7044$. The two plots are also quite close to each other in the general vicinity of this robust optimum at this stage.

By the 11th iteration of EGRO, \hat{y}_{\max} and y_{\max} , at least visually, seem to be almost the same for all values of \mathbf{x}_c . At this stage, the general location around $\mathbf{x}_c = 0.7044$ has also been sampled more than once. Shifting our focus to the central column of Fig. 3.6, we note that El_c has also fallen to quite a low value. By the 24th iteration, the Kriging metamodel \mathcal{K}_f seems to approximate the concentric circular contours of the true function in Fig. 3.1 quite well. Additionally, y_{\max} and \hat{y}_{\max} are now visually indistinguishable in the plots shown. The robust optimum, which is found by identifying the argument of $r_{\mathcal{K}}$, is returned at this point since El_c^{\max} is now smaller than the threshold ϵ_{EI} .

3.5.3. NUMERICAL PERFORMANCE EVALUATION

Now we turn our attention to analyzing the numerical performance of the algorithm based on the average performance of 100 runs on each test problem provided in the Appendix. Table 3.1 shows the reference results obtained from [7, 27]. The table shows the robust optimum locations for the control variable and environment variable and the robust optimum object value. The number of dimensions in \mathbb{X}_c and \mathbb{X}_e along with their size are also provided in the second column of Table 3.1 for each test problem. The right-most column shows the total number of dimensions n_d for each function. The numerical performance of EGRO on the test problems is summarized in Table 3.2. The first column lists the test functions. The mean robust optimum location in \mathbb{X}_c and \mathbb{X}_e based on 100 runs is provided in the second and third column respectively. The fourth and fifth column show the mean and standard deviation of the objective value $f_i(\mathbf{x}_c, \mathbf{x}_e)$ at the robust optimum, where i denotes the function number. Finally, the sixth column provides the total number of expensive function evaluations n_f scaled by the total number of dimensions n_d of the problem. For instance $f_7(\mathbf{x}_c, \mathbf{x}_e)$ required a total of $n_f = 288$ evaluations. The function has $n_d = 10$ total dimensions. The ratio of n_f to n_d is therefore $288/10$, which is rounded up to 29 in the table.

Fig. 3.7 shows the ratio of the average robust optimum (third column in Table 3.2) to the reference robust optimum (sixth column in Table 3.1) for all the test problems except f_8 . In case of f_8 , the ratio has not been plotted since the reference robust optimum is zero. Nevertheless, for f_8 there is no difference between the reference objective and the average robust optimum achieved by EGRO. The error bars in Fig. 3.7 indicate the standard deviation (fifth column in Table 3.2) around the average robust optimum. It is clear from the figure that the ratio of the average to the reference robust optimum is close to 1 for all the test problems. Furthermore, the standard deviation for the objective function is also low for most of the functions. For $f_3(\mathbf{x}_c, \mathbf{x}_e)$ the standard deviation is

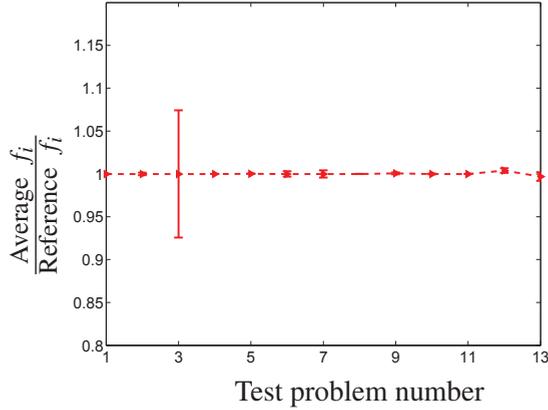


Figure 3.7: Figure shows the ratio of the average robust optimum to the reference robust optimum for all the test problems. The error bars indicate the standard deviation around the average robust optimum.

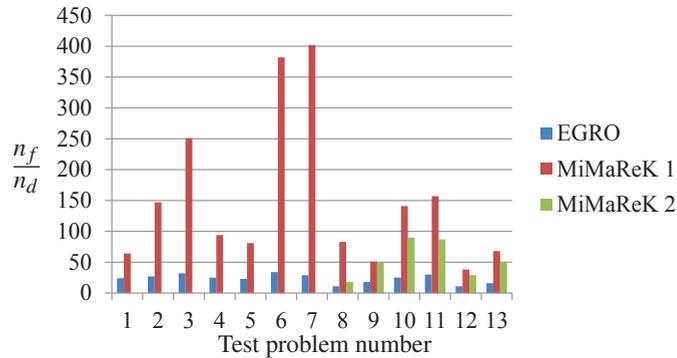


Figure 3.8: Figure shows the ratio of the total function evaluations to the total number of dimensions $\frac{n_f}{n_d}$ for EGRO, MiMaReK 1 and MiMaReK 2.

relatively higher, but the mean robust optimum objective is still quite accurate. The relatively higher standard deviation can be explained by the fact that the norm of the gradient at the robust optimum for $f_3(\mathbf{x}_c, \mathbf{x}_e)$ is more than two orders of magnitude larger than the corresponding value for the other functions. The overall results suggest that EGRO is consistently able to accurately find the robust optimum objective value, regardless of the function type or size.

We now turn our attention to the average robust optimum location in \mathbb{X}_c and \mathbb{X}_e (second and third column in Table 3.2) for the given test problems. The reference results are provided in the fourth and fifth column in Table 3.1. Again the numbers compare

quite favorably and differences between the reference results and the algorithm's results surface only in the second and third decimal place. It should be noted here that there is a greater scope for difference between the reference robust optimum location and those achieved by the algorithm because it may be the case that some functions are quite flat with respect to certain dimensions. Therefore, the robust optimum could be quite close to the reference result even if the robust optimum location is relatively not as close to the reference result. Nonetheless, we note that the average robust optimum location found by the algorithm compares very well against the reference results.

The most important aspect of the algorithm is its ability to reach the global robust optimum consistently in relatively few expensive function evaluations. Finding the global robust optimum is an especially challenging problem since it involves a nested global optimization. Therefore, a large amount of expensive function evaluations could be needed to solve even quite small problems compared to the amount needed for nominal optimization.

The sixth column in Table 3.2 shows the total number of expensive function evaluations n_f required for the corresponding average numerical performance scaled by the total number of dimensions n_d for each test problem. The combined results of this benchmark indicate that the ratio $\frac{n_f}{n_d}$ does not increase significantly with the number of dimensions n_d since all of the ratios remain in the same vicinity of 15-35 evaluations per dimension. Even for large problems such as f_6 (7 total dimensions) and f_7 (10 total dimensions) the ratio of the total evaluations to the number of dimensions remains steady. Similarly, non-convex and highly multimodal problems such as the damped sine problem, f_{10} , and the damped cosine problem, f_{11} , do not require especially high numbers of expensive function evaluations.

3.5.4. COMPARISON WITH OTHER APPROACHES

It is pertinent here to compare the efficiency of the proposed approach against contemporary techniques for min-max optimization. Problems f_8 to f_{13} have been used for min-max optimization using evolutionary approaches [12–14]. The algorithms in [12, 13] require a ratio of more than 10^4 total function evaluations to number of dimensions to reach the robust optimum consistently. The plots in [14] suggest that the robust optimum is found based on only 110 fitness function evaluations. However, the actual expensive function evaluations are hidden within each iteration of the surrogate assisted evolutionary algorithm, resulting in total function evaluation in the same order as [12, 13]. As shown by the last column in Table 3.2, EGRO is several orders faster than the evolutionary algorithms since it solves the problems f_8 to f_{13} using only a maximum ratio of 30 total function evaluations to number of dimensions.

We also compare EGRO against the recent Kriging-based min-max optimization approaches, known as MiMaReK 1 [27] and MiMaReK 2 [26]. Marzat *et al.* tested MiMaReK 1 on all the test problems, f_1 to f_{13} , while MiMaReK 2 was applied on the problems f_8 to f_{13} . Marzat *et al.* state that under most conditions MiMaReK 2 will converge faster than MiMaReK 1. Both these algorithms were shown to use much fewer expensive function evaluations compared to evolutionary techniques in order to reach the robust optimum.

A more complete comparison can be made against MiMaReK 1, since reference average results based on 100 runs are available for it in [27] for all the test problems. Fig.

Table 3.3: Comparison of the robust optimum and deterministic optimum for the filter bandwidth at the respective worst-case locations.

	W	g	L	ΔW	Δt	Nominal B	Worst-case B	Expensive simulations
Robust optimum	1	1.032	233.64	-0.1	0.003	254.1 GHz	23.3 GHz	150
Nominal optimum	1.071	1.045	300	-0.1	-0.003	405.8 GHz	19.5 GHz	150

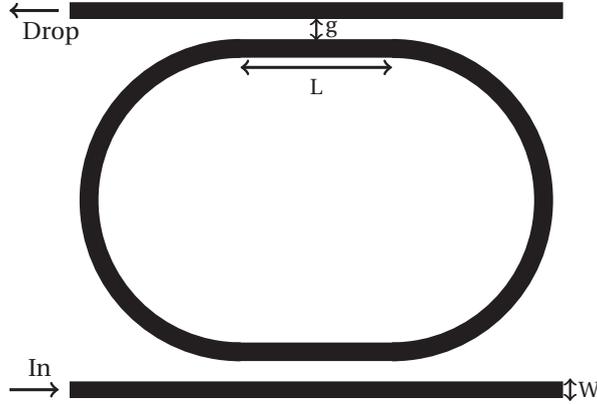


Figure 3.9: Top-view schematic of an optical ring resonator.

3.8 shows the ratio of the total function evaluations to the total number of dimensions $\frac{n_f}{n_d}$ for EGRO, MiMaReK 1 and MiMaReK 2. Comparing EGRO with MiMaReK 1 in Fig. 3.8, we note that EGRO uses fewer function evaluations to reach the global robust optimum for all functions. In some cases, EGRO is more than an order of magnitude faster than MiMaReK 1, such as for the two largest problems f_6 ($n_d = 10$) and f_7 ($n_d = 7$). As can be noted from Fig. 3.8, the difference between EGRO and MiMaReK 1 is not as dramatic for the smaller problems, but EGRO does considerably better on all functions. Fig. 3.8 also shows the comparison of EGRO with MiMaReK 2 for the test problems f_8 to f_{13} . MiMaReK 2 uses fewer function evaluations than MiMaReK 1, but it is still slower than EGRO.

EGRO achieves average results for the robust optimum that are relatively close to the reference in Table 3.1 and are comparable to those found for MiMaReK 1 and MiMaReK 2 [26, 27]. The standard deviation around the robust optimum for the 100 runs is also quite low for all problems.

3.5.5. ENGINEERING CASE STUDY: ROBUST OPTIMIZATION OF AN OPTICAL FILTER

In this work, we consider the robust optimization of an integrated photonic microdevice known as a ring resonator [35]. A top-view schematic of such a resonator is given in Fig. 3.9. The device consists of two straight optical paths separated by a ring-shaped optical path, all of which are integrated on a chip. These optical paths on the chip are known as waveguides. The chip consists of a single stripe of Silicon Nitride buried in Silicon Diox-

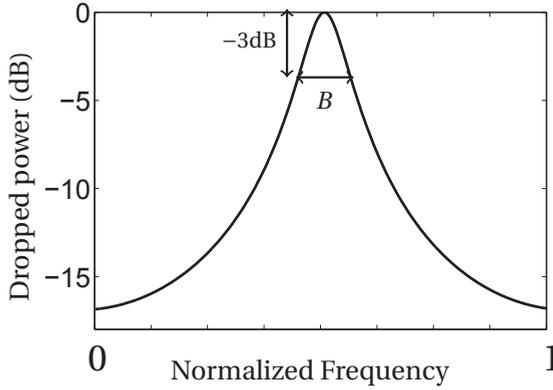


Figure 3.10: Spectral response at the drop port of a ring resonator. The bandwidth of the ring resonator, B , is also indicated on the plot.

ide and the geometry is referred to as TripleXTM [36]. When light at a certain frequency is inserted into one of the straight waveguides, it is transferred, via the ring section, into the other straight waveguide as a result of resonant coupling between the adjacent waveguides. The device operates as an optical filter that allows power to be transferred only at certain frequencies. This property motivates the use of ring resonators in many applications in optical communication and signal processing. The optical performance, however, is very sensitive to fabrication induced variations in geometry of the integrated circuit. For further details concerning the theory and application of ring resonators, please refer to [35].

Fig. 3.10 shows the spectral response at the drop port (output) of a ring resonator. In this case study, we are interested in maximizing the filter bandwidth, B , defined as the bandwidth of the response at -3 dB. The design variables are the gap, g , between the straight and the ring section, the length L of the straight coupling section in the ring and the width W of the waveguides. There are two parametric uncertainties, ΔW and Δt . ΔW represents uncertainty in the width due to fabrication variations while Δt represents the uncertainty in the out-of-plane thickness of the waveguide. The robust optimization problem is defined as,

$$\min_{w,g,L} \max_{\Delta W, \Delta t} -B, \quad (3.19)$$

where $w \in [1, 1.17]\mu\text{m}$, $g \in [1, 1.3]\mu\text{m}$ and $L \in [100, 300]\mu\text{m}$. The uncertainty $\Delta W \in [-0.1, 0.1]\mu\text{m}$ and $\Delta t \in [-3, 3]\text{nm}$. The nominal thickness of the waveguides is $t = 32\text{nm}$. The set of control variables is $\mathbf{x}_c \in [w \ g \ L]$ and the set of environment variables consists of $\mathbf{x}_e \in [\Delta W \ \Delta t]$.

The ring resonator is simulated using a commercial software package, [37]. Each simulation costs approximately 10 minutes. An initial Kriging metamodel is built using $10 \times n_d = 50$ samples chosen via Latin Hypercube sampling. The computational budget is set to $30 \times n_d = 150$ expensive simulations. EGRO is therefore allowed to run for 100 iterations. The robust optimum found via EGRO is compared to the nominal optimum

of the problem. This nominal optimum is estimated by using the Efficient Global Optimization algorithm described in Section 3.3. The same total computational budget of 150 simulations is allowed for the nominal optimization and the algorithm is initialized with $10 \times n_d = 30$ samples chosen via Latin Hypercube sampling.

Table 3.3 shows a comparison of the robust optimum estimated by EGRO with the nominal optimum found by EGO. The optimal locations for W , g and L are provided for both optima (column 2 to 4). The worst-case location in ΔW and Δt is also given for the two cases (column 5 and 6). Column 7 and 8 show the nominal and worst-case bandwidth found at the respective locations. It is important to note that the values given for both these quantities have been found on the expensive simulation via postprocessing after both algorithms terminate. The computational budget used by the two algorithms is given in the last column.

The performance of the robust optimum and the nominal optimum is first compared at the nominal location. The table shows that the nominal bandwidth B for the robust optimum is 254.0 GHz. On the other hand, the bandwidth at the nominal optimum is a much higher value, 405.8 GHz. However, this large difference, in reality, is not significant since even the slightest deviation from the ideal parameter values causes the bandwidth B to drop dramatically for both cases. The comparison of the nominal bandwidth to the worst-case bandwidth (second last column, Table 3.3) for the two optima also gives an indication of the sensitivity of the objective with respect to the parametric uncertainties. For both optima, the bandwidth falls by more than an order of magnitude when moving from the nominal to the worst-case solution.

It is therefore more instructive to compare the worst-case bandwidth found for the nominal and robust optimal locations. The worst-case bandwidth for the robust optimum is 3.8 GHz (19 percent) higher than the worst-case bandwidth for the nominal optimum. Although the robust optimum is nominally suboptimal, it will perform better than the nominal optimum in the worst-case scenario.

Interestingly, the worst-case location in ΔW and Δt for both the optima occurs at the bounds. This is purely coincidental since the objective function is, in general, non-convex with respect to ΔW and Δt for many choices of \mathbf{x}_c .

In addition to the final result, it is also informative to observe how the two methods, EGRO and EGO, evolve with each iteration. Fig. 3.11 shows the worst-case bandwidth found on the metamodel at each iteration of EGRO. Observing the evolution of the worst-case cost, we note that it fluctuates from the 1st to the 35th iteration. It stays relatively constant for the next 20 iterations but starts to shift again from iteration 55 to iteration 70. Thereafter, the value remains largely steady until the algorithm terminates at the 100th iteration. An interesting observation is that the worst-case cost found on the metamodel does not exactly match the worst-case cost found on the expensive simulation, Table 3.3, when the algorithm terminates at the 100th iteration. This indicates that more than 150 total simulations are needed in order to improve the metamodel fidelity in the local region of the robust optimum.

The response of the expensive simulation at each newly added sample point is plotted with respect to the number of iterations of EGO in Fig. 3.12. The figure also shows the nominal optimum found at each iteration. Studying the plot, we note that the nominal optimum at any iteration in EGO is given by the maximum simulated bandwidth found

till that iteration. The nominal optimum does not change after the 40th iteration since no better solution is found thereafter.

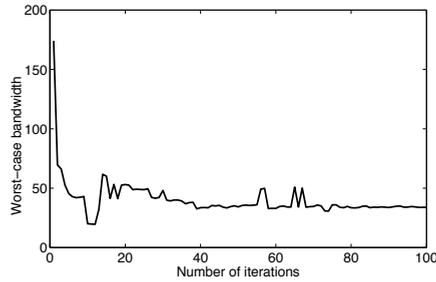


Figure 3.11: The worst-case bandwidth found on the metamodel at each iteration of EGRO is plotted for the total number of iteration.

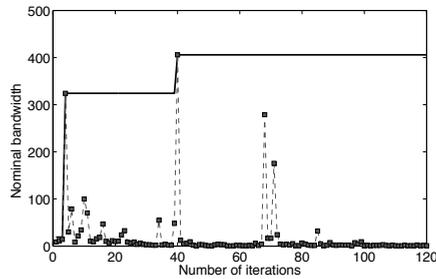


Figure 3.12: The nominal optimum at each iteration of EGO is plotted as a function of the number of iterations. The figure also shows the simulated response for the new sampling location chosen at each iteration.

3.6. CONCLUSION

Optimization problems involving bounded-but-unknown uncertainties and expensive simulations are often encountered in practice. In this work, we have proposed a novel method for efficient global robust optimization of such problems that are affected by parametric uncertainties. To avoid extensive use of expensive function evaluations, a surrogate-based robust optimization technique was formulated. The approach depended on constructing a Kriging metamodel and adaptively sampling the resulting surrogate. The sampling locations were found using expected improvement criteria that reflected the need for finding the best worst-case cost instead of the nominal cost.

The presented algorithm was tested on several test problems found in literature. We demonstrated that the proposed approach can consistently locate the global robust optimum of these functions using relatively much fewer expensive function evaluations than the amount reported in previous work. The reproducibility of the technique was tested by observing average performance based on 100 runs on each test problem. The

statistical comparison of the method against comparable approaches were also quite favorable.

In addition to the application of the algorithm on numerical problems, the technique was tested on an engineering problem as well. It was shown that the method can provide a fabrication tolerant integrated circuit design for an optical filter affected by fabrication uncertainties.

Tests on the benchmark problems showed that the proposed approach accurately found the global robust optimum. Furthermore, the algorithm scaled linearly in terms of expensive function evaluations with the number of dimensions and exhibited much faster convergence than existing techniques. Numerical results also suggest that the approximations made in the algorithm did not hinder convergence to the global robust optimum.

Appendix A: Test problems

The 13 test functions on which EGRO is applied are provided below. Functions 1 to 7 have been employed as test problems in [7, 27] while functions 8 to 13 have been used in [12–14, 26, 27].

$$f_1(\mathbf{x}_c, \mathbf{x}_e) = 5(x_{c1}^2 + x_{c2}^2) - (x_{e1}^2 + x_{e2}^2) + x_{c1}(-x_{e1} + x_{e2} + 5) + x_{c2}(x_{e1} - x_{e2} + 3). \quad (20)$$

$$f_2(\mathbf{x}_c, \mathbf{x}_e) = 4(x_{c1} - 2)^2 - 2x_{e1}^2 + x_{c1}^2 x_{e1} - x_{e2}^2 + 2x_{c2}^2 x_{e2}. \quad (21)$$

$$f_3(\mathbf{x}_c, \mathbf{x}_e) = x_{c1}^4 x_{e2} + 2x_{c1}^3 x_{e1} - x_{c2}^2 x_{e2}(x_{e2} - 3) - 2x_{c2}(x_{e1} - 3)^2 \quad (22)$$

$$f_4(\mathbf{x}_c, \mathbf{x}_e) = -\sum_{i=1}^3 (x_{ei} - 1)^2 + \sum_{i=1}^2 (x_{ci} - 1)^2 + x_{e3}(x_{c2} - 1) + x_{e1}(x_{c1} - 1) + x_{e2}x_{c1}x_{c2} \quad (23)$$

$$f_5(\mathbf{x}_c, \mathbf{x}_e) = -x_{e1}(x_{c1} - 1) - x_{e2}(x_{c2} - 2) - x_{e3}(x_{c3} - 1) + 2x_{c1}^2 + 3x_{c2}^2 + x_{c3}^2 - x_{e1}^2 - x_{e2}^2 - x_{e3}^2 \quad (24)$$

$$f_6(\mathbf{x}_c, \mathbf{x}_e) = x_{e1}(x_{c1}^2 - x_{c2} + x_{c3} - x_{c4} + 2) + x_{e2}(-x_{c1} + 2x_{c2}^2 - x_{c3}^2 + 2x_{c4} + 1) + x_{e3}(2x_{c1} - x_{c2} + 2x_{c3} - x_{c4}^2 + 5) + (5x_{c1}^2 + 4x_{c2}^2 + 3x_{c3}^2 + 2x_{c4}^2) - \sum_{i=1}^3 (x_{ei})^2 \quad (25)$$

$$f_7(\mathbf{x}_c, \mathbf{x}_e) = 2x_{c1}x_{c5} + 3x_{c4}x_{c2} + x_{c5}x_{c3} + 5x_{c4}^2 + 5x_{c5}^2 - x_{c4}(x_{e4} - x_{e5} - 5) + x_{c5}(x_{e4} - x_{e5} + 3) + \sum_{i=1}^3 x_{ei}(x_{ci}^2 - 1) - \sum_{i=1}^5 (x_{ei}^2) \quad (26)$$

$$f_8(\mathbf{x}_c, \mathbf{x}_e) = (x_{c1} - 5)^2 - (x_{e1} - 5)^2 \quad (27)$$

$$f_9(\mathbf{x}_c, \mathbf{x}_e) = \min(3 - 0.2x_{c1} + 0.3x_{e1}, 3 + 0.2x_{c1} - 0.1x_{e1}) \quad (28)$$

$$f_{10}(\mathbf{x}_c, \mathbf{x}_e) = \frac{\sin(x_{c1} - x_{e1})}{\sqrt{x_{c1}^2 + x_{e1}^2}} \quad (29)$$

$$f_{11}(\mathbf{x}_c, \mathbf{x}_e) = \frac{\cos(\sqrt{x_{c1}^2 + x_{e1}^2})}{\sqrt{x_{c1}^2 + x_{e1}^2} + 10} \quad (30)$$

$$f_{12}(\mathbf{x}_c, \mathbf{x}_e) = 100(x_{c2} - x_{c1}^2)^2 + (1 - x_{c1})^2 - x_{e1}(x_{c1} + x_{c2}^2) - x_{e2}(x_{c1}^2 + x_{c2}) \quad (31)$$

$$f_{13}(\mathbf{x}_c, \mathbf{x}_e) = (x_{c1} - 2)^2 + (x_{c2} - 1)^2 + x_{e1}(x_{c1}^2 - x_{c2}) + x_{e2}(x_{c1} + x_{c2} - 2). \quad (32)$$

Appendix B: Numerical choices and optimizers

The calculation of EI_c is more computationally demanding than calculating the deterministic EI, Equation (3.7), since \hat{y}_{\max} has to be computed for each evaluation of EI_c . In this work, Matlab implementation of the multistart framework for global optimization [38] is used to estimate \hat{y}_{\max} . The algorithm is a global optimization method that chooses the best solution after performing local optimization runs from multiple starting points. The Matlab implementation `fmincon`, of the interior point algorithm for non-linear programming [39], was used as the local solver in this work. Since \hat{y}_{\max} is computed on a known analytical function, i.e. the Kriging predictor, the Jacobian and Hessian of the Kriging predictor can very cheaply and easily be computed as well. In order to enhance the efficiency, the Jacobian and Hessian were also provided to the local solver.

Estimating the robust optimum $r_{\mathcal{N}}$ on the metamodel, Equation (3.8), is also a computationally intensive part of the algorithm since it requires computation of \hat{y}_{\max} as well. EI_c and $r_{\mathcal{N}}$ use the same strategy, described above, for estimating \hat{y}_{\max} .

It should be noted here, that any type of global optimizer can be used to estimate \hat{y}_{\max} . However, because \hat{y}_{\max} , Equation (3.9), has to be found each time EI_c needs to be evaluated, it is best to use an approach that can use analytically supplied Jacobian and Hessian of the Kriging predictor, since they can be computed very cheaply.

The global optimization of EI_c and the outer global minimization for estimating the optimal $r_{\mathcal{N}}$ in Equation (3.8) are performed via the Matlab implementation of the Simulated Annealing algorithm [40].

The computational cost of evaluating EI_e is much less than the calculation of EI_c , Equation (3.18), since there is no internal optimization that has to be performed each time EI_e is evaluated. However, EI_e requires a pre-computed value for the reference global worst-case cost $g_{\mathcal{N}}$. Matlab implementation of the Simulated Annealing algorithm [40] is again used for this global optimization.

REFERENCES

- [1] S. P. Gurav, J. F. L. Goosen, and F. van Keulen, *Bounded-But-Unknown uncertainty optimization using design sensitivities and parallel computing: Application to MEMS*, *Comp Struct* **83**, 1134 (2005).
- [2] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust optimization* (Princeton University Press, 2009).
- [3] H.-G. Beyer and B. Sendhoff, *Robust optimization – A comprehensive survey*, *Comput Method Appl M* **196**, 3190 (2007).
- [4] K. Zhou, J. C. Doyle, and K. Glover, *Robust and optimal control* (Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996).
- [5] G. Taguchi, *Quality Engineering Through Design Optimization*, Global Telecommunications Conference (1984).
- [6] M. Aghassi and D. Bertsimas, *Robust game theory*, *Math Program* **107**, 231 (2006).
- [7] B. Rustem and M. Howe, *Algorithms for Worst-Case Design and Applications to Risk Management* (Princeton University Press, 2002) p. 408.
- [8] Y.-S. Ong, P. B. Nair, and K. Y. Lum, *Max-min surrogate-assisted evolutionary algorithm for robust design*, *IEEE T Evolut Comput* **10**, 392 (2006).
- [9] S. Zuhe, A. Neumaier, and M. Eiermann, *Solving minimax problems by interval methods*, *BIT Numerical Math* **30**, 742 (1990).
- [10] P. M. Frank and X. Ding, *Survey of robust residual generation and evaluation methods in observer-based fault detection systems*, *J Proc Contr* **7**, 403 (1997).
- [11] S. Kassam and H. Poor, *Robust techniques for signal processing: A survey*, *P IEEE* **73** (1985).
- [12] A. M. Cramer, S. D. Sudhoff, and E. L. Zivi, *Evolutionary Algorithms for Minimax Problems in Robust Design*, *IEEE T Evolut Comput* **13**, 444 (2009).
- [13] Y. Shi and R. A. Krohling, *Co-evolutionary particle swarm optimization to solve minimax problems*, in *Proceedings of the Evolutionary Computation on 2002. CEC '02. Proceedings of the 2002 Congress - Volume 02*, CEC '02 (IEEE Computer Society, Washington, DC, USA, 2002) pp. 1682–1687.
- [14] A. Zhou and Q. Zhang, *A surrogate-assisted evolutionary algorithm for minimax optimization*, in *Evolutionary Computation (CEC), 2010 IEEE Congress on* (2010) pp. 1–7.
- [15] R. I. Lung and D. Dumitrescu, *A new evolutionary approach to minimax problems*, in *Evolutionary Computation (CEC), 2011 IEEE Congress on* (2011) pp. 1902–1905.

- [16] T. W. Simpson, J. D. Poplinski, P. N. Koch, and J. K. Allen, *Metamodels for Computer-based Engineering Design: Survey and recommendations*, Eng Comput **17**, 129 (2001).
- [17] A. I. J. Forrester and A. J. Keane, *Recent advances in surrogate-based optimization*, Prog Aerosp Sci **45**, 50 (2009).
- [18] D. G. Krige, *A statistical approach to some basic mine valuation problems on the Witwatersrand*, J Chem, Metall Min Soc South Africa **52**, 119 (1951).
- [19] J. Sacks, W. Welch, T. J. Mitchell, and H. P. Wynn, *Design and Analysis of Computer Experiments*, Stat Sci **4**, 409 (1989).
- [20] D. R. Jones, *A Taxonomy of Global Optimization Methods Based on Response Surfaces*, J Global Optim **21**, 345 (2001).
- [21] D. R. Jones, M. Schonlau, and W. J. Welch, *Efficient Global Optimization of Expensive Black-Box Functions*, J Global Optim **13**, 455 (1998).
- [22] M. Franey, P. Ranjan, and H. Chipman, *Branch and bound algorithms for maximizing expected improvement functions*, J Stat Plan Infer **141**, 42 (2011).
- [23] A. Forrester, S. Andras, and A. J. Keane, *Engineering design via surrogate modelling : a practical guide* (J. Wiley, 2008).
- [24] E. Vazquez and J. Bect, *Convergence properties of the expected improvement algorithm with fixed mean and covariance functions*, J Stat Plan Infer **140**, 3088 (2010).
- [25] D. den Hertog, J. P. C. Kleijnen, and A. Y. D. Siem, *The Correct Kriging Variance Estimated by Bootstrapping*, J Oper Res Soc **57**, pp. 400 (2006).
- [26] J. Marzat, E. Walter, and H. Piet-Lahanier, *A new strategy for worst-case design from costly numerical simulations*, in *American Control Conference (ACC), 2013* (2013) pp. 3991–3996.
- [27] J. Marzat, E. Walter, and H. Piet-Lahanier, *Worst-case global optimization of black-box functions through Kriging and relaxation*, J Global Optim , 1 (2012).
- [28] G. G. Wang and S. Shan, *Review of Metamodeling Techniques in Support of Engineering Design Optimization*, J Mech Design **129**, 370 (2007).
- [29] M. López and G. Still, *Semi-infinite programming*, Eur J Oper Res **180**, 491 (2007).
- [30] O. Stein, *How to solve a semi-infinite optimization problem*, Eur J Oper Res **223**, 312 (2012).
- [31] K. Shimizu, Y. Ishizuka, and J. Bard, *Nondifferentiable and two-level mathematical programming* (Kluwer Academic Publishers, 1997).
- [32] M. Morris and T. J. Mitchell, *Exploratory designs for computational experiments*, J Stat Plan Infer **43**, 381 (1995).

- [33] R. B. Arellano-Valle and M. G. Genton, *On the exact distribution of the maximum of absolutely continuous dependent random variables*, *Stat Probabil Lett* **78**, 27 (2008).
- [34] S. ur Rehman, M. Langelaar, and F. van Keulen, *Efficient Kriging-based robust optimization of unconstrained problems*, *Journal of Computational Science* **5**, 872 (2014).
- [35] W. Bogaerts, P. De Heyn, T. Van Vaerenbergh, K. De Vos, S. Kumar Selvaraja, T. Claes, P. Dumon, P. Bienstman, D. Van Thourhout, and R. Baets, *Silicon microring resonators*, *Laser Photonics Rev* **6**, 47 (2012).
- [36] R. Heideman, M. Hoekman, and E. Schreuder, *TriPleX-Based Integrated Optical Ring Resonators for Lab-on-a-Chip and Environmental Detection*, *IEEE J Quantum Electron* **18**, 1583 (2012).
- [37] PhoeniX Software, *PhoeniX Software 4.8.4*, (2014).
- [38] Z. Ugray, L. Lasdon, J. Plummer, F. Glover, J. Kelly, and R. Martí, *Scatter Search and Local NLP Solvers: A Multistart Framework for Global Optimization*, *INFORMS J Comput* **19**, 328 (2007).
- [39] R. Byrd, M. Hribar, and J. Nocedal, *An interior point algorithm for large-scale nonlinear programming*, *SIAM J Optimiz* **9**, 877 (1999).
- [40] L. Ingber, *Adaptive simulated annealing (ASA): Lessons learned*, *Control and cybernetics* (1996), arXiv:0001018 [arXiv:cs.MS] .

4

ROBUST OPTIMIZATION OF UNCONSTRAINED PROBLEMS

4.1. INTRODUCTION

Many engineering problems based on expensive computer simulations are affected by uncertainties. In addition, the distribution of the uncertainties of these problems is often not available and only the bounds of the uncertainty sets are known [1]. Robust optimization [2] has to be performed on such problems in order to find an insensitive solution. Unconstrained robust optimization involves min-max optimization where we seek to minimize the maximum objective value with respect to the uncertainties in the problem. The optimization is therefore nested in that the response to be minimized is itself a result of a maximization. The nested nature of the optimization automatically implies that the number of function calls required to find the robust optimum, even for small problems, can be quite high. Applying robust optimization directly on expensive simulation-based problems is therefore prohibitively costly.

Multiple classifications exist to distinguish uncertainties of different types. The available information separates probabilistic from bounded-but-unknown uncertainties [3]. In the former case, information on the probability distribution of uncertainties is available, while in the latter case distributions are unknown and only bounds on the uncertainties are specified. Another often used classification distinguishes aleatory and epistemic uncertainties, referring to their origin and (ir)reducibility [4]. In this chapter, the focus is on the way bounded-but-unknown uncertainties, of either aleatory or epistemic type, affect the involved simulation models: through a direct modification of the design variables (implementation error) or through a model parameter (parametric uncertainty) [5]. Although it is not strictly required, in this work all uncertainties are assumed to be independent.

Robust optimization has typically been applied to convex problems of varying complexity. The focus has generally been on addressing problems affected by parametric uncertainties, i.e. uncertainties that affect the problem data. On the other hand, problems

that involve bounded-but-unknown uncertainties directly affecting the design variables of the problem have received considerably less attention. Robust optimization of problems under bounded-but-unknown parametric uncertainty and implementation error has also not been analyzed in great detail insofar in literature. To the best of our knowledge, this combined problem has previously only been explored by Bertsimas *et al.* [6, 7]. In the same setting, there has been relatively more work on this subject for uncertainties whose probability distributions are known, see e.g. [8, 9].

The focus of this work is on efficient global robust optimization of expensive simulation-based problems affected by both parametric uncertainty and implementation error. In contrast to the typical scenario in robust optimization where the objective function is known explicitly and is assumed to be convex, we address the scenario in which there is a black-box function with no prior assumptions on the structure other than continuity. This setting matches many practical problems encountered in industrial practice, where simulation-based optimization is performed.

While deterministic optimization of expensive simulation-based problems has received a lot of attention in literature [10, 11], robust optimization of such problems has received less attention. Most recently Bertsimas *et al.* [7], Marzat *et al.* [12] and ur Rehman *et al.* [13], ur Rehman and Langelaar [14] have considered robust optimization of black-box functions. However, Bertsimas *et al.* do not consider the situation when the simulations are expensive to evaluate. Ur Rehman *et al.* solve problems that are either exclusively affected by implementation error only [13] or parametric uncertainties only [14]. Marzat *et al.* [12] also propose an algorithm that addresses problems affected by parametric uncertainties only.

To avoid the high computational costs involved in applying robust optimization directly on expensive simulation-based problems, cheap surrogate models of the simulation can be constructed using intelligently placed sampling points. There are many different techniques that can be used for creating a cheap response surface [15] such as polynomial models, radial basis functions, Kriging and regression models.

Kriging [16] is an interpolation technique that uses a parameterized Gaussian basis function. The statistical framework of Kriging provides an estimator of the variance of the Kriging interpolator. This variance has proven useful for adaptively sampling the expensive simulation to quickly reach the deterministic optimum of the problem [10]. Jones *et al.* [17] used the variance estimate to develop an Expected Improvement (EI) criterion and Efficient Global Optimization (EGO) procedure that adds infill samples such that the global deterministic optimum is found using only a few expensive function evaluations. Our approach can be seen as an extension of the EGO procedure to the robust optimization setting. There are, however, certain drawbacks in using Kriging. The computational efficiency of constructing Kriging metamodels scales badly as the number of samples and dimensions increase. Furthermore, the correlation matrix used in Kriging often tends to suffer from ill-conditioning.

In this work, we extend the Kriging-based robust optimization algorithms proposed by ur Rehman *et al.* [13], ur Rehman and Langelaar [14] to the case where the problem is affected by both implementation error *and* parametric uncertainties. It is shown how the expected improvement based adaptive sampling strategies proposed by the original algorithms can be further adapted to suggest new infill sampling locations for problems

affected by both types of uncertainties. We show that a clear performance improvement can be achieved by directly addressing the problem with both uncertainties rather than rewriting the problem as one affected by parametric uncertainties only. The proposed method provides a novel infill sampling criterion that enables efficient global robust optimum of problems involving both parametric uncertainties and implementation error.

This chapter is organized as follows. Robust optimization of problems involving parametric uncertainties and implementation error is introduced in Section 4.2. In Section 4.3, an introduction to Kriging is followed by a brief explanation of the algorithm for implementation error affected problems [13] and the algorithm for problems involving parametric uncertainties [14]. The proposed approach for efficient global robust optimization of problems affected by both parametric uncertainties and implementation error is introduced in Section 4.4. Finally, Sections 4.5 and 4.6 consist of the results and conclusions, respectively.

4.2. ROBUST OPTIMIZATION OF PROBLEMS INVOLVING PARAMETRIC UNCERTAINTIES AND IMPLEMENTATION ERROR

An unconstrained deterministic optimization problem may be defined as,

$$\min_{\mathbf{x}_c} f(\mathbf{x}_c) \tag{4.1}$$

where $f(\mathbf{x}_c)$ is the objective while \mathbf{x}_c is the set of control or design variables. As mentioned previously, uncertainties can be of two types, parametric uncertainties and implementation errors. Parametric uncertainties can be described as the uncertainties in the problem data. These uncertainties act in dimension(s) that are separate from the control variables. Implementation errors, on the other hand, directly affect the control variables.

We first define the robust optimization problem affected only by implementation error. Now, assume that due to an error in implementation $\Delta \in \mathcal{U}$ where \mathcal{U} is the uncertainty set, the design variables \mathbf{x}_c deviate to a new effective value $\mathbf{x}_c + \Delta$. The robust optimum is given by the minimization of the worst-case cost of the objective with respect to the uncertainty set \mathcal{U} . The problem can be expressed as,

$$\min_{\mathbf{x}_c \in \mathbb{X}_c} \max_{\Delta \in \mathcal{U}} f(\mathbf{x}_c + \Delta). \tag{4.2}$$

ur Rehman *et al.* [13] addressed the above problem by employing a Kriging-based optimization approach. To simplify the problem we make use of the fact that the implementation error resides in the same dimension as the design variables.

Next, the robust optimization problem affected only by parametric uncertainties is introduced. Let $\mathbf{x}_e \in \mathbb{X}_e$ be the set of parametric uncertainties or environment variables against which the problem should be robust. In order to find the robust optimum the worst-case cost of the objective with respect to the parametric uncertainties has to be minimized. Robust optimization of a problem affected by parametric uncertainties can formally be stated as

$$\min_{\mathbf{x}_c \in \mathbb{X}_c} \max_{\mathbf{x}_e \in \mathbb{X}_e} f(\mathbf{x}_c, \mathbf{x}_e), \tag{4.3}$$

where $f(\mathbf{x}_c, \mathbf{x}_e)$ is the objective, \mathbf{x}_c is the set of control variables while \mathbf{x}_e is the set of environment variables or parametric uncertainties. Robust optimization of black-box functions affected by parametric uncertainties has been dealt with by Marzat *et al.* [12], ur Rehman and Langelaar [14], Marzat *et al.* [18]. ur Rehman and Langelaar [14] again used a Kriging-based optimization approach to tackle this problem.

In order to be robust against both implementation error and parametric uncertainties, the worst-case cost of the objective with respect to both uncertainties has to be minimized. The problem may be written as,

$$\min_{\mathbf{x}_c \in \mathcal{X}_c} \max_{\mathbf{x}_e \in \mathcal{X}_e, \Delta \in \mathcal{Z}} f(\mathbf{x}_c + \Delta, \mathbf{x}_e). \quad (4.4)$$

It should be noted that the problem affected by both implementation error and parametric uncertainties in Equation (4.4) can be rewritten in terms of Equation (4.3) since an implementation error can also be treated as a parametric uncertainty. However, when implementation errors are included as parametric uncertainties, the total number of dimensions of the problem increases in response surface-based approaches [13]. This leads to a higher budget requirement for the number of expensive function evaluations needed to reach the robust optimum, because a higher dimensional space has to be sampled. Therefore, when a problem is affected by both types of uncertainties, it is preferable to solve Equation (4.4) instead of redefining the problem in terms of Equation (4.3).

4.3. ROBUST OPTIMIZATION OF UNCONSTRAINED EXPENSIVE SIMULATION-BASED PROBLEMS

4.3.1. PRELIMINARY - KRIGING AND EGO

Kriging is an interpolation technique based on a statistical framework where the function response is assumed to be a normally distributed random variable. Details and full derivation of Kriging model construction and prediction can be found in Sacks, J., Welch, W., Mitchell, T. J., and Wynn [16]. In this section, a short qualitative description is provided of those aspects of Kriging that are essential in order to explain the Efficient Global Optimization (EGO) algorithm. Some additional details and equations are provided in the Appendix.

In Kriging, a parameterized Gaussian correlation function, Equation (24), is used to describe the correlation between any two sample points. The parameters of the correlation function are chosen such that the likelihood of the observed data is maximized. Once the model parameters are obtained, Kriging estimates the interpolation between the sample points that is most consistent with the observed data. This is performed by maximizing the combined likelihood of the observed data and the predicted value. The Kriging prediction, Equation (25), is denoted by \hat{y} .

The Mean Squared Error (MSE) in the prediction, Equation (26), can readily be computed based on the underlying statistical assumptions. The term s^2 is used to denote this variance in the Kriging interpolator.

Jones *et al.* [17] proposed the Efficient Global Optimization (EGO) algorithm based on the Kriging prediction $\hat{y}(\mathbf{x})$ and variance $s^2(\mathbf{x})$. The idea is to reduce expensive function evaluations by adaptively adding new sample points in adequately chosen locations.

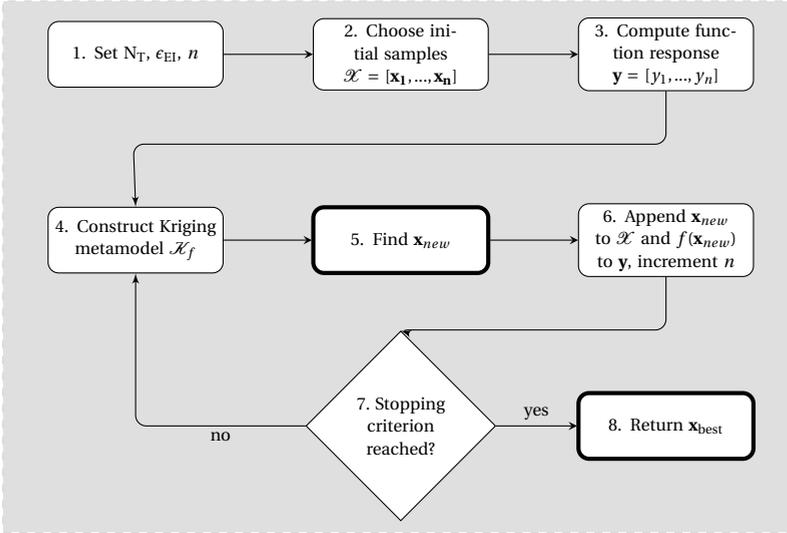


Figure 4.1: Flowchart shows the general steps involved in iterative Kriging-based optimization for deterministic optimization using EI (EGO) [17], robust optimization under implementation error [13] and robust optimization under parametric uncertainties [14].

In order to formulate EGO, the uncertainty in the predicted value $\hat{y}(\mathbf{x})$ at a position \mathbf{x} can be described in terms of a normally distributed random variable $Y(\mathbf{x})$ with mean $\hat{y}(\mathbf{x})$ and variance $s^2(\mathbf{x})$. Let y_{\min} represent the minimum objective value in the observed data. A lower value than y_{\min} may be found at a position \mathbf{x} if a part of the distribution $Y(\mathbf{x})$ lies below the current minimum. The expectation of this improvement I is found by computing the expectation $E[I(\mathbf{x})] = E[\max(y_{\min} - Y, 0)]$. It can be shown [19] that the expression for *expected improvement* is given by

$$E[I(\mathbf{x})] = (y_{\min} - \hat{y})\Phi\left(\frac{y_{\min} - \hat{y}}{s}\right) + s\phi\left(\frac{y_{\min} - \hat{y}}{s}\right) \quad (4.5)$$

where $\Phi(\cdot)$ is the normal cumulative distribution function and $\phi(\cdot)$ is the normal probability density function. The analytical expression in Equation (4.5) is computationally cheap to evaluate. A global optimizer can be used to estimate the global maximizer of EI. The expensive function is evaluated at the maximizer location and the metamodel is rebuilt with the augmented set of samples and responses.

Figure 4.1 shows a flowchart of the EGO algorithm. A metamodel \mathcal{K}_f is initially built based on the response $f(\mathbf{x})$ at n initial sampling locations chosen via an appropriate design of experiments strategy, e.g. space-filling. Step 5 involves the search for \mathbf{x}_{new} , i.e. the global maximizer of EI. The algorithm involves an iterative approach where the EI criterion is used to suggest new sampling locations at each iteration. The algorithm terminates when either EI_{\max} , the maximum EI found, falls below the threshold ϵ_{EI} or the total number of samples N_T is consumed. At this stage, \mathbf{x}_{best} , the sampling location that gives the minimum objective value, is returned as the final solution.

4.3.2. PROBLEMS AFFECTED BY IMPLEMENTATION ERROR ONLY

A robust optimization problem affected by implementation error was defined in Section 4.2 by Equation (4.2). This problem was tackled using a Kriging-based optimization approach by ur Rehman *et al.* [13]. For reference, a short description is provided here of the algorithm proposed by ur Rehman *et al.* [13]. The method is referred to as EGRO-IE.

Essentially, the method for solving the problem affected by implementation error is quite similar to the Efficient Global Optimization algorithm. The steps involved in the approach can again be described in terms of the flowchart in Figure 4.1. However, the method for identifying \mathbf{x}_{new} is different from the one employed for deterministic optimization. Step 5 is divided into two substeps, where in the first substep the global robust optimum is estimated on the metamodel,

$$r_{\mathcal{K}} = \min_{\mathbf{x}_c \in \mathbb{X}_c} \max_{\Delta \in \mathcal{U}} \mathcal{K}_f(\mathbf{x} + \Delta). \quad (4.6)$$

In the second substep, we search for \mathbf{x}_{new} by estimating the global maximizer of an *adapted* expected improvement criterion. This adapted expected improvement criterion is developed using $r_{\mathcal{K}}$ as a reference to improve on. For each location at which the adapted EI is applied, the worst-case Kriging prediction with respect to the uncertainty set \mathcal{U} is estimated on the metamodel. The global maximizer of the adapted EI, i.e. \mathbf{x}_{new} , is found by searching for the location with the highest expectation of improvement for the worst-case Kriging prediction over the reference robust optimum.

When one of the stopping criteria is met in Step 7, the algorithm returns \mathbf{x}_{best} , the location of the robust optimum found in the final iteration,

$$\mathbf{x}_{best} = \arg \min_{\mathbf{x}_c \in \mathbb{X}_c} \max_{\Delta \in \mathcal{U}} \mathcal{K}_f(\mathbf{x} + \Delta). \quad (4.7)$$

Ur Rehman *et al.* showed that the algorithm was significantly more efficient than comparable techniques in terms of the number of expensive function evaluations used to reach the robust optimum. A complete description of the method and the numerical results may be found in ur Rehman *et al.* [13].

4.3.3. PROBLEMS AFFECTED BY PARAMETRIC UNCERTAINTIES ONLY

Recently, ur Rehman and Langelaar [14] proposed a method known as EGRO for Efficient Global Robust Optimization of problems affected by parametric uncertainties only. The robust optimization problem is defined in Equation (4.3). Similar to EGO and to the robust optimization under implementation error approach [13], the method employs an iterative surrogate-based optimization approach. Again, the main concept behind the algorithm can be explained using the flowchart in Figure 4.1.

Once the Kriging metamodel is constructed in Step 4, the new sampling location \mathbf{x}_{new} is searched for in Step 5. To find \mathbf{x}_{new} , Ur Rehman *et al.* divided Step 5 into three substeps. In the first substep, the robust optimum was estimated on the metamodel,

$$r_{\mathcal{K}} = \min_{\mathbf{x}_c \in \mathbb{X}_c} \max_{\mathbf{x}_e \in \mathbb{X}_e} \mathcal{K}_f(\mathbf{x}_c, \mathbf{x}_e). \quad (4.8)$$

In the second substep, a new sampling location, \mathbf{x}_c^{new} , in the control variable space \mathbb{X}_c is found. Finally, in the third substep the new sampling location, \mathbf{x}_e^{new} , in the environment

variable space \mathbb{X}_e is identified. Adapted EI criteria are used to find the sampling location in both \mathbb{X}_c and \mathbb{X}_e . Details on the derivation of the adapted EI criteria may be found in ur Rehman and Langelaar [14]. The response at the new sampling location $\mathbf{x}_{new} = (\mathbf{x}_c^{new}, \mathbf{x}_e^{new})$ in the combined space $(\mathbb{X}_c, \mathbb{X}_e)$ is then computed in Step 6.

The algorithm iteratively adds new adaptively sampled locations until either EI_{max} , the maximum EI found in the control variable space, falls below the threshold ϵ_{EI} or the total number of samples N_T is consumed. In Step 8, the location of the robust optimum found on the metamodel in the final iteration is returned as the best solution,

$$\mathbf{x}_{best} = \arg \min_{\mathbf{x}_c \in \mathbb{X}_c} \max_{\mathbf{x}_e \in \mathbb{X}_e} \mathcal{K}_f(\mathbf{x}_c, \mathbf{x}_e). \quad (4.9)$$

The EGRO algorithm compared favorably with contemporary methods for robust optimization of expensive simulation-based problems affected by parametric uncertainties in terms of efficiency. A more in-depth description of the approach is provided in ur Rehman and Langelaar [14].

4.3.4. NUMERICAL CASE STUDY: TREATING IMPLEMENTATION ERROR PROBLEMS AS PARAMETRIC UNCERTAINTY PROBLEMS

A problem affected by implementation error, Equation (4.2) can be rewritten as a problem affected by parametric uncertainties, Equation (4.3). However, as mentioned in Section 4.2, this leads to an increase in the total number of dimensions of the problem. Due to the increase in dimensions, a relatively larger number of expensive simulations is needed to construct a high fidelity metamodel and to estimate the global robust solution. It is instructive to investigate how much impact such a change in problem definition can have on a problem affected by implementation error. The following three-dimensional non-convex problem, from the work by ur Rehman *et al.* [13], is used for this purpose,

$$\begin{aligned} f(\mathbf{x}_c) = & \left(x_{c2} - \frac{5.1}{4\pi} x_{c2} + \frac{5}{\pi} x_{c1} - 6 \right)^2 + 10 \left(\left(1 - \frac{1}{8\pi} \right) \cos(x_{c1}) + 1 \right) \\ & + (6x_{c3} - 2)^2 \sin(12x_{c3} - 4) + 8x_{c3}, \\ & x_{c1} \in [-5, 10], x_{c2} \in [0, 15], x_{c3} \in [0, 1]. \end{aligned} \quad (4.10)$$

ur Rehman *et al.* [13] assumed that each variable was independently affected by a ± 12.5 percent maximum change due to implementation error. The robust optimum was estimated by ur Rehman *et al.* [13] via the EGRO-IE algorithm described in Section 4.3.2. Latin Hypercube Sampling (LHS) was employed to initialize the algorithm with $10 \times n_d = 30$ expensive simulations. Since LHS is random, EGRO-IE was run 100 times and the average result was analyzed. The algorithm was given a total budget of 120 expensive simulations.

If the problem is expressed as one affected by parametric uncertainties, the number of dimensions doubles to six. The updated problem has three design variables \mathbf{x}_c and

Table 4.1: Comparison of statistics based on 100 runs of EGRO-IE and LHS on numerical test problem.

Robust optimum	Mean	Standard deviation	Expensive function evaluations
Reference	24.95	-	-
EGRO-IE	24.98	0.027	120
EGRO	28.14	1.825	300

three parametric uncertainties \mathbf{x}_e ,

$$\begin{aligned}
 f(\mathbf{x}_c) = & \left((x_{c2} + x_{e2}) - \frac{5.1}{4\pi}(x_{c2} + x_{e2}) + \frac{5}{\pi}(x_{c1} + x_{e1}) - 6 \right)^2 \\
 & + 10 \left(\left(1 - \frac{1}{8\pi} \right) \cos(x_{c1} + x_{e1}) + 1 \right) + (6(x_{c3} + x_{e3}) - 2)^2 \sin(12(x_{c3} + x_{e3}) - 4) + 8(x_{c3} + x_{e3}), \\
 & x_{c1} \in [-3.125, 8.125], x_{c2} \in [1.875, 13.125], x_{c3} \in [0.125, 0.875], \\
 & x_{e1} \in [-1.875, 1.875], x_{e2} \in [-1.875, 1.875], x_{e3} \in [-0.125, 0.125].
 \end{aligned} \tag{4.11}$$

We use EGRO, ([14]), introduced in Section 4.3.3 to estimate the robust optimum of the problem. LHS is again used to choose $10 \times n_d = 60$ initial sampling locations and EGRO is allowed a total budget of 300 expensive function evaluations.

Table 4.1 shows the average result for 100 runs of EGRO-IE and EGRO applied on the numerical test problem, Equation (4.10) and Equation 4.11 respectively. The reference robust optimum estimated on the actual function is also provided. Despite the fact that EGRO-IE uses only 120 expensive simulations compared to the 300 expensive simulations employed by EGRO, the robust optimum found by EGRO-IE is much closer to the reference optimum. The standard deviation over the 100 runs around the mean robust value is also much smaller for EGRO-IE compared to the standard deviation for EGRO.

The primary reason for the dramatic difference in performance can be attributed to the fact that EGRO-IE operates on a three dimensional function while EGRO is applied on a six dimensional problem. The number of expensive simulations needed to construct a high fidelity metamodel for the three dimensional problem, Equation (4.10), is much smaller compared to the number of expensive simulations required for the six dimensional problem, Equation (4.11). Therefore EGRO-IE is a significantly more effective strategy compared to EGRO for estimating the robust optimum of implementation error affected problems. Furthermore, the result suggests that redefining Equation (4.2) or Equation (4.4) in terms of Equation (4.4) can dramatically impact the efficiency of a metamodel based strategy in estimating the robust optimum. This motivates the method proposed in the present chapter, in which the strengths of both approaches are combined.

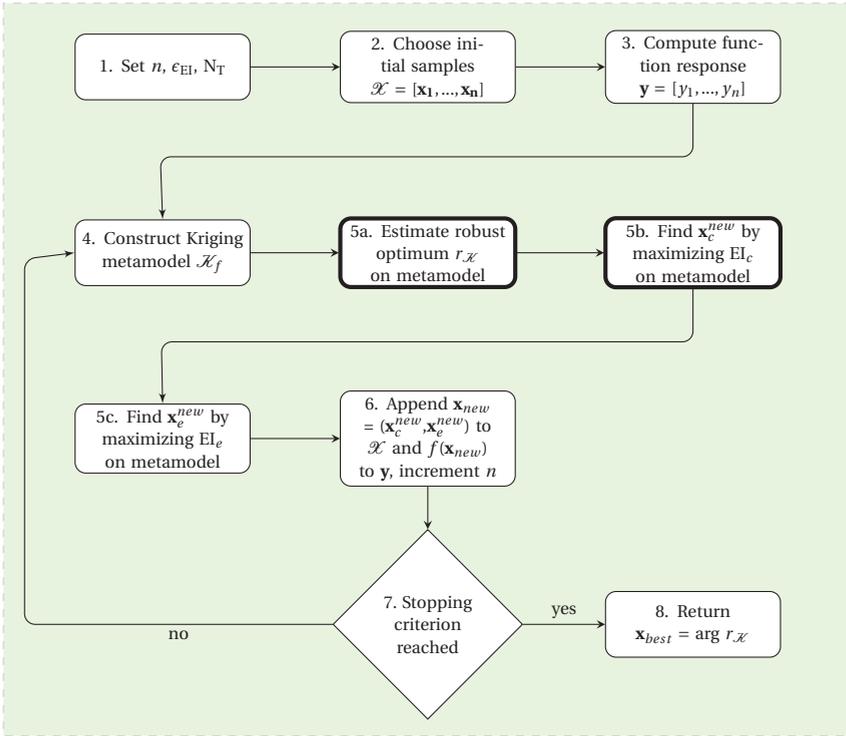


Figure 4.2: Flowchart shows the efficient global robust optimization algorithm for problems affected by implementation error and parametric uncertainties. The bold borders indicate that these steps are different from the EGRO algorithm for robust optimization under parametric uncertainties [14].

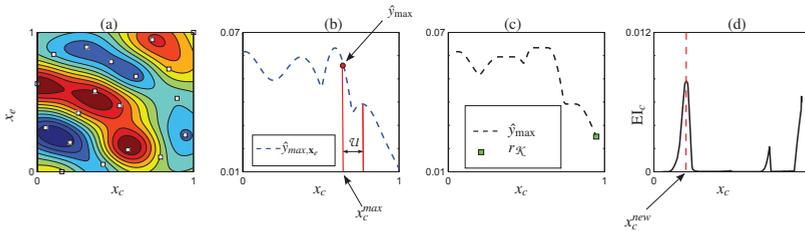


Figure 4.3: The Kriging metamodel of a two-dimensional function is shown in (a). The worst-case Kriging prediction with respect to the parametric uncertainty is plotted in (b). The overall worst-case cost with respect to both uncertainties is plotted in (c). Plot (d) shows EI_c as a function of \mathbf{x}_c . The new control variable location \mathbf{x}_c^{new} is also indicated.

4.4. EFFICIENT GLOBAL ROBUST OPTIMIZATION UNDER PARAMETRIC UNCERTAINTIES AND IMPLEMENTATION ERROR

In this work, we extend the discussed algorithms for robust optimization under implementation error [13] and robust optimization under parametric uncertainty [14] in order

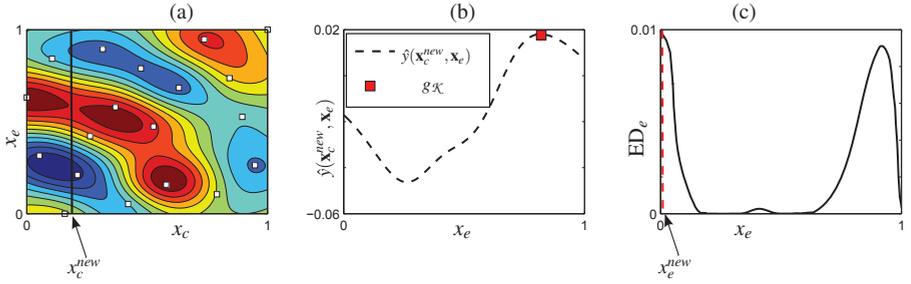


Figure 4.4: Plot (a) show a Kriging metamodel of a two dimensional function, along with the location of \mathbf{x}_c^{new} . The Kriging prediction at \mathbf{x}_c^{new} , corresponding to the response along the black line in plot (a), is plotted with respect to \mathbf{x}_e in (b). ED_e is plotted in (c). The new environment variable location \mathbf{x}_e^{new} is also indicated.

4

to efficiently address the problem affected by both uncertainties in Equation (4.4). Since it is assumed that the function is expensive to evaluate, we devise a scheme that is based on the construction of a surrogate model and the application of robust optimization on the surrogate. Let \mathcal{K}_f be the Kriging model of the objective. Then the problem may be written as

$$r_{\mathcal{K}} = \min_{\mathbf{x}_c \in \mathcal{X}_c} \max_{\mathbf{x}_e \in \mathcal{X}_e, \Delta \in \mathcal{U}} \mathcal{K}_f(\mathbf{x}_c + \Delta, \mathbf{x}_e). \quad (4.12)$$

To accurately identify the global robust optimum it is paramount that Equation (4.12) estimates Equation (4.4) very well, especially in the neighbourhood of the robust optimum. To this end, we propose adapted expected improvement measures that suggest infill sampling points such that the robust optimum, $r_{\mathcal{K}}$, found using Equation (4.12) is close to the true robust optimum after relatively few function evaluations. The alternative to adaptive sampling is to use a fine initial global design of experiments, but as the region of interest is a priori unknown this requires far more function evaluations.

Figure 4.2 shows a flowchart of the algorithm. It can be observed in Figure 4.2 that the algorithm involves an initial surrogate construction phase followed by an iterative process. The robust optimum is estimated on the metamodel at each iteration using Equation (4.12). Thereafter, in Step 5b an adapted expected improvement criterion is employed to find a suitable sampling location, \mathbf{x}_c^{new} , in the control variables space. In Step 5c an expected deterioration measure is used to suggest a new sampling location, \mathbf{x}_e^{new} , in the environment variables space. The process of Kriging model construction and subsequent infill sampling is repeated until the condition statement in Step 7 is violated. At this point, the location of the robust optimum $r_{\mathcal{K}}$ found on the last iteration is returned as the final solution.

The flowchart in Figure 4.2 is globally identical to the flowchart for EGRO [14]. Internally however, Step 5a and Step 5b are not the same for the two algorithms. While in EGRO $r_{\mathcal{K}}$ is found by using Equation (4.8), for the combined problem $r_{\mathcal{K}}$ is estimated by using Equation (4.12). The more important difference between the two algorithms is that the adapted EI criterion used in EGRO to find \mathbf{x}_c^{new} was derived for a problem affected by parametric uncertainties only. On the other hand, the adapted expected improve-

ment criterion EI_c used in this work is derived for a problem affected by both parametric uncertainties and implementation errors. Fundamentally, the rest of the algorithm is the same as EGRO. We refer to the algorithm presented in this work as EGRO-C, where C stands for the *Combination* of uncertainties that the algorithm addresses. In the following discussion, it will be shown how EI_c is derived and how the adaptive sampling enables the problem in Equation (4.4) to be solved efficiently.

4.4.1. IDENTIFYING CONTROL VARIABLE INFILL LOCATION \mathbf{x}_c^{new}

Figure 4.3a shows a Kriging metamodel of a two-dimensional function involving a single control variable and a single environment variable. We scale the range of the design domain to a range of $[0, 1]$ across each dimension when constructing the metamodel and do not consider extrapolation outside this domain. Let the control variable \mathbf{x}_c be affected by an error in implementation $\Delta \in \mathcal{U}$, where $\mathcal{U} \in [-0.05, 0.05]$ for the scaled design domain $x \in [0, 1]$. The worst-case Kriging prediction with respect to the parametric uncertainty is given by

$$\hat{y}_{max, \mathbf{x}_e}(\mathbf{x}_c) = \max_{\mathbf{x}_e \in \mathbb{X}_e} \mathcal{K}_f(\mathbf{x}_c, \mathbf{x}_e), \quad (4.13)$$

and the corresponding maximizer in the environment variable space is denoted as \mathbf{x}_e^{max} . Figure 4.3b shows $\hat{y}_{max, \mathbf{x}_e}$ as a function of the control variable x_c . Figure 4.3b also shows the uncertainty set \mathcal{U} at an arbitrary location on the plot. The maximum value obtained within this uncertainty set gives the worst-case Kriging prediction with respect to both implementation error and parametric uncertainty. The worst-case Kriging prediction with respect to parametric uncertainty and implementation error can therefore be expressed as

$$\hat{y}_{max}(\mathbf{x}_c) = \max_{\Delta \in \mathcal{U}} \hat{y}_{max, \mathbf{x}_e}(\mathbf{x}_c). \quad (4.14)$$

For the uncertainty set \mathcal{U} at the arbitrary location in Figure 4.3b, \hat{y}_{max} , indicated by a red circle, is found on the left bound of the set. The corresponding location \mathbf{x}_c^{max} where \hat{y}_{max} is found is also indicated on the plot. In general, the location \mathbf{x}_c^{max} can be expressed as

$$\mathbf{x}_c^{max} = \mathbf{x}_c + \mathit{arg} \max_{\Delta \in \mathcal{U}} \hat{y}_{max, \mathbf{x}_e}(\mathbf{x}_c). \quad (4.15)$$

Figure 4.3c shows \hat{y}_{max} as a function of x_c . The minimum value obtained on this plot is the robust optimum on the metamodel, $r_{\mathcal{K}}$, since this represents the best worst-case solution.

In order to find an expected improvement measure that suggests a new sampling location in \mathbb{X}_c , we make assumptions similar to those made in formulating the deterministic EI [17]. A normally distributed random variable Y_{max} is used to model the uncertainty in the value of the worst-case Kriging prediction \hat{y}_{max} . The mean value for Y_{max} is given by \hat{y}_{max} while the variance is given by the Kriging mean squared error $s^2(\mathbf{x}_c^{max}, \mathbf{x}_e^{max})$.

To find the new sampling location \mathbf{x}_c^{new} in the control variable space, we seek the location with the highest expectation of improvement over the current robust optimum. The worst-case Kriging prediction \hat{y}_{max} can improve upon the value of $r_{\mathcal{K}}$ if $Y_{max} < r_{\mathcal{K}}$. To find the expectation of this improvement we evaluate the expected value of the

improvement $I_c = \max(r_{\mathcal{X}} - Y_{max}, 0)$,

$$\underbrace{E[I_c(\mathbf{x}_c^{max})]}_{EI_c} = \int_{I_c=0}^{I_c=\infty} I_c \frac{\exp\left(-\frac{t_c^2}{2}\right)}{\sqrt{2\pi}s} dt_c, \quad (4.16)$$

where

$$t_c = \frac{r_{\mathcal{X}} - I_c - \hat{y}_{max}}{s}, \quad s = s(\mathbf{x}_c^{max}, \mathbf{x}_e^{max}). \quad (4.17)$$

The standard normal probability density function is given by

$$\phi(z) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right). \quad (4.18)$$

Using the definition of the standard normal probability density function, EI_c can be expressed as,

$$\begin{aligned} E[I_c(\mathbf{x}_c^{max})] &= (r_{\mathcal{X}} - \hat{y}_{max}) \int_{t_c=-\infty}^{t_c=\frac{r_{\mathcal{X}} - \hat{y}_{max}}{s}} \phi(t_c) dt_c \\ &- s \int_{t_c=-\infty}^{t_c=\frac{r_{\mathcal{X}} - \hat{y}_{max}}{s}} t_c \phi(t_c) dt_c. \end{aligned} \quad (4.19)$$

The first integral in (4.19) is the normal cumulative distribution function $\Phi\left(\frac{r_{\mathcal{X}} - \hat{y}_{max}}{s}\right)$.

The second integral is determined by substituting $z = \frac{-t_c}{2}$. EI_c can then be expressed as,

$$\begin{aligned} E[I_c(\mathbf{x}_c^{max})] &= (r_{\mathcal{X}} - \hat{y}_{max}) \Phi\left(\frac{r_{\mathcal{X}} - \hat{y}_{max}}{s}\right) \\ &+ s \phi\left(\frac{r_{\mathcal{X}} - \hat{y}_{max}}{s}\right). \end{aligned} \quad (4.20)$$

The new control variable location is found by estimating the global maximizer of Equation (4.20). Figure 4.3d shows the expected improvement EI_c as a function of x_c . As in the case of deterministic EI, the function is usually very multimodal. The new infill sampling location x_c^{new} in the control variable space is shown on the plot.

4.4.2. IDENTIFYING ENVIRONMENT VARIABLE INFILL LOCATION \mathbf{x}_e^{new}

In order to have a new infill location \mathbf{x}_{new} at which to sample the expensive function, a new environment variable location \mathbf{x}_e^{new} has to be found to go along with \mathbf{x}_c^{new} . Figure 4.4a shows the same Kriging model as Figure 4.3a. The solid black line on the plot in Figure 4.4a indicates the location of x_c^{new} found via EI_c . Figure 4.4b shows the plot of the Kriging prediction at x_c^{new} as a function of x_e only. The worst-case cost, $g_{\mathcal{X}}$, is indicated on the plot. In general, the worst-case cost with respect to \mathbf{x}_e , for a given \mathbf{x}_c^{new} , may be expressed as

$$g_{\mathcal{X}}(\mathbf{x}_c^{new}, \mathbf{x}_e) = \max_{\mathbf{x}_e \in \mathcal{X}_e} \mathcal{K}_f(\mathbf{x}_c^{new}, \mathbf{x}_e). \quad (4.21)$$

An adapted expected deterioration criterion is needed for the environment variable space, where the goal is to find the location with the highest expectation of deterioration over

$g_{\mathcal{X}}$. Here, a deterioration would be considered the possibility of achieving a higher value for the worst-case cost than $g_{\mathcal{X}}$, i.e. the most unfavourable situation. The criterion EI_e was developed in ur Rehman and Langelaar [14] to describe the expectation of this deterioration and is repeated here for completeness,

$$\underbrace{E[D_e(\mathbf{x}_c^{new}, \mathbf{x}_e)]}_{EI_e} = (\hat{y} - g_{\mathcal{X}}) \Phi\left(\frac{\hat{y} - g_{\mathcal{X}}}{s}\right) + s\phi\left(\frac{\hat{y} - g_{\mathcal{X}}}{s}\right). \quad (4.22)$$

The new environment variable location is found by estimating the global maximizer of Equation (4.22). EI_e is plotted as a function of x_e in Figure 4.4c. The location of the maximizer x_e^{new} is also indicated on the plot.

4.4.3. DISCUSSION

The process of finding \mathbf{x}_c^{new} and \mathbf{x}_e^{new} is represented by Step 5b and Step 5c, respectively, in the flowchart in Figure 4.2. Once the infill location \mathbf{x}_{new} consisting of $(\mathbf{x}_c^{new}, \mathbf{x}_e^{new})$ is found, the expensive function is evaluated at \mathbf{x}_{new} in Step 6. Thereafter, if the total number of samples is exhausted or if the global maximum value found for EI_c is below a predefined threshold, the algorithm terminates and the location of the robust optimum found on the last iteration is returned. The optimizers used to obtain $r_{\mathcal{X}}$, \mathbf{x}_c^{new} and \mathbf{x}_e^{new} are the same as the ones employed in EGRO [14]. We employ a multi-start gradient based optimization strategy for the inner maximization of $r_{\mathcal{X}}$. Jacobian and Hessian information is provided to the optimizer since it is cheaply available for the Kriging surface. Simulated annealing is used for optimization of EI_c and EI_e . For a detailed discussion of the optimizer choices we refer interested readers to ur Rehman and Langelaar [14]. In addition, dedicated optimizers for robust optimization, such as the methods proposed by Bertsimas *et al.* [7], Bertsimas and Nohadani [20], can also be used to estimate the robust optimum $r_{\mathcal{X}}$ at each iteration.

In this work, we determined the Kriging worst-case cost \hat{y}_{\max} as a deterministic quantity on the metamodel and then imposed a normal distribution on this quantity to formulate the expected improvement criterion EI_c . This has been introduced as an approximation of the actual distribution of the maximum [21]. However, computing this actual distribution requires a prohibitively expensive process and moreover leads to an expected improvement expression that cannot be solved analytically. Both aspects would significantly increase the computational cost of the algorithm, without necessarily improving the convergence rate ur Rehman *et al.* [13], ur Rehman and Langelaar [14]. The various test problems in the following section give evidence of the effectiveness of the chosen approach.

We have chosen to add only a single sampling point at each iteration of the algorithm. An alternative could have been to add several sample points at each iteration, particularly when parallel evaluations are possible. For this purpose, locations that gave local expected improvement in the robust optimum could have been chosen. In this scenario all but one of the sampling points would have been added at suboptimal locations. This suboptimal addition of sample points could potentially increase the number of expensive simulations needed to determine the robust optimum. Therefore, we have preferred to add only a single point based on the global maximum of EI_c and EI_e .

Table 4.2: Reference results of all the test problems. The functions are listed in the Appendix.

Test function	\mathbb{X}_c	\mathbb{X}_e	\mathcal{U}	\mathbf{x}_c	\mathbf{x}_e	Reference R.O	n_d
$f_1(\mathbf{x}_c, \mathbf{x}_e)$	$[-5, 5]^2$	$[-5, 5]^2$	$[\pm 12.5\%]^1$	-0.4833	0.0833	13.942	4
				-0.3167	-0.0833		
$f_2(\mathbf{x}_c, \mathbf{x}_e)$	$[-5, 5]^5$	$[-3, 3]^5$	$[\pm 12.5\%]^1$	0.2460	-0.0462	23.225	10
				0.5040	-0.2202		
				0.1410	0.1446		
				-0.5240	0.1128		
				-0.3070	-0.0756		
$f_3(\mathbf{x}_c, \mathbf{x}_e)$	$[-5, 10]$ $[0, 15]$	$[0, 1]$	$[\pm 10\%]^2$	3.4045	1	40.003	3
				1.8855			

4.5. RESULTS

4.5.1. TEST PROBLEMS

To evaluate the ability of the algorithm to add infill sampling points in such a way that the global robust optimum is estimated efficiently, we test it on three numerical and one engineering test problem. The numerical test problems are listed in Appendix B. Problems $f_1(\mathbf{x}_c, \mathbf{x}_e)$ and $f_2(\mathbf{x}_c, \mathbf{x}_e)$ are both assumed to be affected by a one dimensional implementation error, $\mathcal{U} \in [\pm 12.5\%]^1$, that causes a deviation of up to 12.5% in the control variables. Problem $f_3(\mathbf{x}_c, \mathbf{x}_e)$ is a function of two control variables and one environment variable. The problem is assumed to be affected by a two dimensional implementation error, $\mathcal{U} \in [\pm 10\%]^2$, that results in each control variable independently deviating by up to 10%.

Table 4.2 gives the reference robust optimum location and objective value for each numerical test problem. Column 1 lists the function number, while Columns 2 and 3 provide the domain of the control variables and environment variables, respectively. The implementation error uncertainty set \mathcal{U} is given in Column 4. The reference robust optimum location in \mathbf{x}_c , \mathbf{x}_e and the corresponding robust optimum (R.O) value is provided in Column 5, 6 and 7. The last column shows the number of dimensions for each problem. Further discussion of the numerical results is provided in Section 4.5.2 below.

Although the number of dimensions of all the test problems may seem small in the context of optimization problems in general, the problem size is actually quite challenging in a metamodel based robust optimization problem setting. Constructing a high fidelity metamodel of a non-convex problem with more than ten dimensions can require thousands of expensive function evaluations even if the samples are being added using an intelligent adaptive sampling scheme. Therefore, in this study we limit ourselves to showcasing the method on problems of up to ten dimensions.

In addition to applying the algorithm on numerical test problem, it is also tested on a challenging engineering example affected by uncertainties. The problem involves an optical integrated circuit affected by fabrication variations. The device fabricated on the integrated circuit is an optical filter that is affected by both implementation error and parametric uncertainties. Not only is the optical filter expensive to simulate, but it is also very sensitive to the uncertainties. Further details on this engineering problem are

provided in Section 4.5.3.

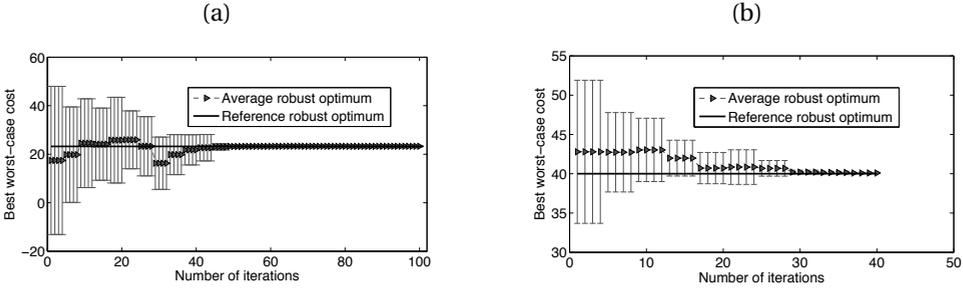


Figure 4.5: The robust optimum found on the metamodel at each iteration of the algorithm for problem $f_2(\mathbf{x}_c, \mathbf{x}_e)$ and $f_3(\mathbf{x}_c, \mathbf{x}_e)$ is plotted. The error bars indicate the standard deviation around the average robust optimum at each iteration. The plot also shows the objective value for the robust optimum on the reference function.

4.5.2. ALGORITHM PERFORMANCE EVALUATION

Since the algorithm is initialized using a non-deterministic space filling technique (LHS), we run each problem 100 times and analyse the average result. All the numerical problems are initialized with $n = 20 \times n_d$ samples. It should be mentioned here that the choice of the number of initial samples employed is problem dependent. If the black-box problem is very non-linear, practitioners would be better served to use a larger number of sample points to construct the initial metamodel. We examine the effect of the initial sample set on EGRO-C by applying it on problem $f_3(\mathbf{x}_c, \mathbf{x}_e)$ with different numbers of initial samples.

Table 4.3 shows the average numerical performance of the algorithm on each problem. Column 1 shows the function number, while Columns 2 and 3 give the average location the robust optimum in the control variables and environment variables domain, respectively. The mean robust optimum (R.O) value is provided in Column 4. The corresponding standard deviation around the robust optimum is shown in Column 5. The last column provides the total number of expensive function calls required by each problem.

Comparing the reference robust optimum objective value, Table 4.2 (Column 7), versus the average robust optimum found by the algorithm, Table 4.3 (Column 4), we observe that the result is almost exactly the same. The standard deviation around the robust optimum, Table 4.3 (Column 5), is also quite low for each test problem.

Figure 4.5(a) shows the average robust optimum estimated at each iteration by EGRO-C for problem $f_2(\mathbf{x}_c, \mathbf{x}_e)$. The standard deviation around the mean value at each iteration is also plotted using error bars. The plot shows that the algorithm converges slowly in the first 30 iterations with the standard deviation going down significantly compared to initial value at the first iteration. After 40 iterations the value for the average robust optimum also coincides with the reference result while the standard deviation goes down even further. After about 50 iterations of the algorithm the standard deviation is small enough such that it cannot be visually observed on the plot.

Table 4.3: The average numerical performance of the algorithm based on 100 runs for each test problem. The table shows the mean and standard deviation of the robust optimum along with the average robust optimum locations in \mathbb{X}_c and \mathbb{X}_e . The total number of expensive function calls is given in the last column.

Test function	Average \mathbf{x}_c	Average \mathbf{x}_e	Average R.O	Standard deviation R.O	Total evaluations n_f
$f_1(\mathbf{x}_c, \mathbf{x}_e)$	-0.4850	0.0850	13.948	0.0042	120
	-0.3150	-0.0850			
$f_2(\mathbf{x}_c, \mathbf{x}_e)$	0.1799	0.2569	23.267	0.0545	300
	0.4700	0.3253			
	0.0898	0.2864			
	-0.5250	0.1230			
$f_3(\mathbf{x}_c, \mathbf{x}_e)$	-0.2747	-0.1279	40.096	0.0811	100
	3.4285	1			
	1.833				

Table 4.4: Comparison of statistics based on 100 runs of EGRO-C and LHS on problem $f_3(\mathbf{x}_c, \mathbf{x}_e)$.

Robust optimum	Mean	Standard deviation	Expensive function evaluations
Reference	40.003	-	-
EGRO-C	40.096	0.0811	100
LHS	46.127	5.1981	100

The same quantities are plotted for problem $f_3(\mathbf{x}_c, \mathbf{x}_e)$ in Figure 4.5(b). The standard deviation starts at a relatively high value and does not diminish in the first 4 iterations. However, it steadily goes down from the fifth iteration onwards. The average robust optimum value stays relatively constant for the first 12 iterations. Thereafter it starts moving towards the reference solution. By the 35th iteration, the average robust optimum has converged to the reference robust optimum, while the standard deviation has fallen to a value low enough such that it is not observable in the plot.

We compare the performance of EGRO-C with the construction of a metamodel using LHS and applying robust optimization on it. The comparison is made on the function $f_3(\mathbf{x}_c, \mathbf{x}_e)$. $f_3(\mathbf{x}_c, \mathbf{x}_e)$ was allowed a budget of 100 expensive function calls, Table 4.3. For a fair comparison we build a metamodel using LHS based on 100 function calls and estimate the robust optimum on the response surface. This process is repeated 100 times since LHS is non-deterministic.

Table 4.4 shows the result of the comparison. Using the same total number of function calls, EGRO-C is able to construct a metamodel that enables the robust optimum to be found with much greater accuracy compared to LHS. The mean for EGRO-C is significantly closer to the reference solution. Furthermore, the standard deviation around the mean is also very low compared to the standard deviation based on 100 runs of LHS.

To investigate the impact of the initial sample size on the performance of EGRO-C, we also apply the algorithm on problem $f_3(\mathbf{x}_c, \mathbf{x}_e)$ with different numbers of initial samples n , chosen via LHS. The initial n samples are varied from $n = 20$ to $n = 80$ in steps of 10 and the algorithm is applied 10 times on the problem for each sample set. EGRO-C is allowed a total computational budget of 100 expensive function calls. Once the 100

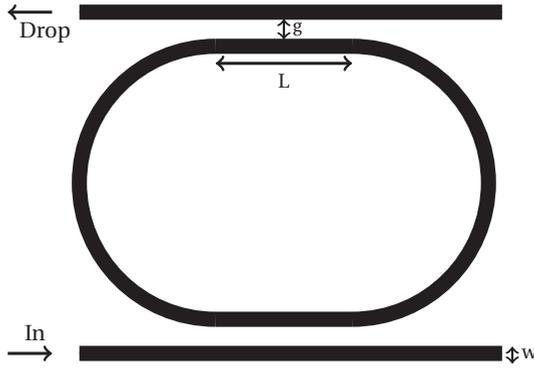


Figure 4.6: Top-view schematic of an optical ring resonator. Light is propagated inside the waveguides (black path lines) on the basis of total internal reflection. A single ring resonator covers an area of approximately 1 mm² on the chip. Reproduced with permission from ur Rehman and Langelaar [14].

expensive function calls (including the respective initial sample sets) are exhausted for each case, the statistics of the robust optimum found are compared.

Table 4.5 shows the comparison of the mean robust optimum found by EGRO-C, after 100 expensive function calls, based on 10 runs on each sample set. The standard deviation around the robust optimum is also given. For $n = 20$, EGRO-C runs for 80 iterations before the termination of the algorithm due to the threshold of 100 expensive function calls. On the other hand, for $n = 80$, EGRO-C only runs for 20 iterations before termination. It is interesting to note that the mean robust optimum closest to the reference solution, in Table 4.2, is found when $n = 20$. However, the lowest standard deviation around the robust optimum is found for $n = 50$. It is clear from Table 4.5 that if we use $n = 60$ samples or more, and thereby allow EGRO-C to run for 40 iterations or less, the mean robust optimum found is not as accurate and has a larger relative standard deviation. The worst solution is in fact found for $n = 80$. In terms of consistency, running EGRO-C with $n = 50$ initial samples seems to give the best result when finding the robust optimum of $f_3(\mathbf{x}_c, \mathbf{x}_e)$ with a total of 100 expensive function calls.

Table 4.5: Comparison of the average robust optimum found on problem $f_3(\mathbf{x}_c, \mathbf{x}_e)$, after 100 expensive simulations, when using different number of initial samples n . The average results are based on 10 runs on each sample set. The acronym R.O. stands for Robust Optimum.

Initial samples n	20	30	40	50	60	70	80
Mean R.O.	40.0292	40.0459	40.0357	40.0348	40.0977	40.1915	41.0421
Standard deviation R.O.	0.0224	0.0389	0.0340	0.0085	0.0564	0.2575	1.1542

4.5.3. ENGINEERING PROBLEM: ROBUST OPTIMIZATION OF AN OPTICAL FILTER

To test the algorithm in a practical setting, we apply it on an optical filter whose performance is very sensitive to uncertainties. The filter is based on an integrated photonic

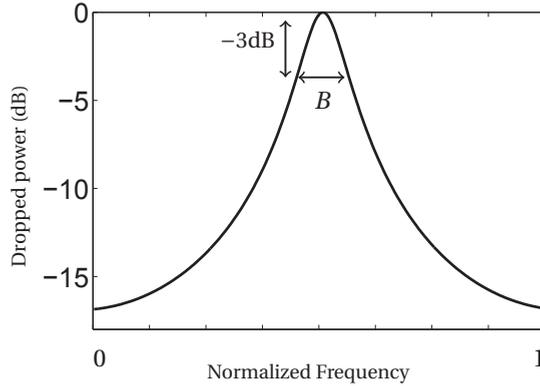


Figure 4.7: Spectral response at the drop port of a ring resonator. B is the filter bandwidth of the resonator. Reproduced with permission from ur Rehman and Langelaar [14].

microring resonator affected by fabrication variations in geometry. Figure 4.6 shows an illustration of the top-view of the resonator which has two straight waveguides that are separated by a ring-shaped waveguide. The waveguide consists of a stripe of Silicon Nitride, typically of width $1 \mu\text{m}$ and thickness 25 to 100 nm, that is buried inside Silicon Dioxide, ([22]). Light propagates inside the waveguide due to the relatively higher refractive index of Silicon Nitride compared to Silicon Dioxide. If we input light at a specific frequency into the in-port it will couple into the ring-shaped resonator, from where it couples into the drop port, Figure 4.6. Resonance only occurs at certain frequencies, therefore the ring resonator behaves like an optical filter.

A typical spectral response at the drop port of a ring resonator is shown in Figure 4.7. An important parameter of interest is the filter bandwidth B , given by the bandwidth of the spectral response at -3dB . For best performance, it is often desired to maximize B . The bandwidth is very sensitive to deviations in the ring resonator geometry which can occur due to fabrication defects. In this work, we aim to estimate the robust optimum of the filter bandwidth using as few expensive simulations as possible. We employ a commercial software package ([23]) in order to simulate the ring resonator. Each simulation takes approximately 10 minutes.

The set of control variables of the problem is $\mathbf{x}_c \in [w \ g \ L]$. w is the width of the waveguides while g is the gap between the straight and the ring section. L is the length of the straight coupling section in the ring. The problem has one implementation error, $\Delta w \in \mathcal{U}$ and one parametric uncertainty Δt . Δw is a geometrical deviation caused by under- or overetching during fabrication and affects both the gap g and the width w . The uncertainty in the out-of-plane thickness of the waveguide is given by Δt . The robust optimization problem is defined as,

$$\min_{w,g,L} \max_{\Delta t, \Delta w \in \mathcal{U}} -B, \quad (4.23)$$

where $w \in [0.9, 1.27] \mu\text{m}$, $g \in [0.9, 1.4] \mu\text{m}$ and $L \in [100, 300] \mu\text{m}$. The range of the implementation error uncertainty set \mathcal{U} is $[-0.1, 0.1] \mu\text{m}$ while the parametric uncertainty set

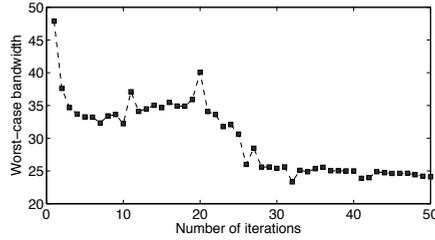


Figure 4.8: The best worst-case bandwidth found on the metamodel at each iteration of the algorithm is plotted with respect to the number of iterations.

Table 4.6: Comparison of the robust optimum found by EGRO, EGRO-C and the deterministic optimum found by EGO for the filter bandwidth. Higher bandwidth values (B) are preferred.

Optimum	w	g	L	Δw	Δt	Nominal B	Worst-case B	Expensive simulations
Nominal (EGO)	1.071	1.045	300	-0.1	-0.003	405.8 GHz	19.5 GHz	150
Robust (EGRO)	1	1.032	233.64	-0.1	0.003	254.1 GHz	23.3 GHz	150
Robust (EGRO-C)	1.043	1.000	244.18	-0.1	0.003	376.4 GHz	30.8 GHz	100

is given by $\Delta t \in [-3, 3]$ nm. The thickness of the waveguides without considering uncertainties is $t = 32$ nm.

ur Rehman and Langelaar [14] applied EGRO on the same engineering problem to estimate the robust optimum. However, the implementation error $\Delta w \in \mathcal{U}$ was treated as a parametric uncertainty. We compare the performance of EGRO-C with EGRO and focus on the number of expensive simulations needed by the two algorithms to estimate the robust optimum. ur Rehman and Langelaar [14] also estimated the nominal optimum of the problem via EGO. In this work, the nominal performance at the robust optimum found by EGRO-C is also compared against the nominal optimum estimated by EGO.

EGRO-C is initialized with 50 samples chosen via LHS. The algorithm is assigned a total budget of 100 expensive simulations, which translates to 50 iterations. These settings are in agreement with the earlier study.

Table 4.6 shows a comparison of the robust optimum estimated by EGRO, EGRO-C and the deterministic optimum found by EGO for the filter bandwidth. Column 2 to 4 provide the optimal location for the design variables for each algorithm. The worst-case locations for the uncertainties Δw and Δt are given in Column 5 and 6, while Column 7 and 8 show the nominal and worst-case filter bandwidth B found at the corresponding optimal location for each algorithm. The last column show the number of expensive simulations used by each algorithm. The quantities Δw and Δt and the nominal, worst-case B are found on the actual simulator ([23]) after each algorithm finishes.

Comparing the worst-case bandwidth B of EGRO and EGRO-C we note that EGRO-C finds a higher value for B even though it used 50 less expensive simulations than the number used by EGRO. As expected, both EGRO and EGRO-C have a higher worst-case bandwidth than that found by EGO. One the other hand, the nominal bandwidth of EGO is higher than the corresponding number for EGRO and EGRO-C.

The result clearly indicates that EGRO-C estimates a relatively better robust optimum

than that found by EGRO using fewer expensive simulations. This seems to suggest that, at least in this case, by rewriting the problem that is affected by both parametric uncertainty and implementation error as one affected only by parametric uncertainties we make the problem more difficult to optimize. As previously mentioned, this can primarily be attributed to the increased difficulty of constructing a high fidelity metamodel of a problem with relatively higher number of dimensions. This explains why EGRO is not able to find a higher value for the worst-case bandwidth despite using more expensive function calls.

Figure 4.8 shows a plot of the best worst-case bandwidth *found on the metamodel* at each iteration of EGRO-C. From the first iteration to the 25th iteration the objective changes quite significantly. Thereafter there is not much change in the best worst-case solution and the trend is sustained until termination at 50 iterations. The fact that the value of the worst-case bandwidth found on the metamodel is slightly lower than the worst-case bandwidth found on the expensive simulator, Table 4.6, suggests that the metamodel ideally needs more than 100 expensive simulations to exactly describe the neighbourhood of the optimum. However, for practical purposes in this application example the robust optimum obtained after 100 simulations is certainly sufficiently accurate.

4.6. CONCLUSION

In this work we addressed robust optimization of expensive computer simulation based problems affected by implementation error *and* parametric uncertainties. A sound infill sampling criterion was developed such that the global robust optimum of the problem could be estimated using only a few expensive function calls. The algorithm was based on the construction and iterative update of a Kriging metamodel of the problem. The infill sampling scheme was developed by modifying the expected improvement criterion so that it was tailored towards estimating the robust optimum instead of the nominal optimum.

The effectiveness of the algorithm was tested on multiple numerical examples as well as on an engineering problem. The algorithm was run 100 times on each of the numerical problems and it managed to consistently converge to the reference global robust optimum on all the problems using relatively few expensive function calls. It was shown that the standard deviation around the average robust optimum estimated by the algorithm was also quite low, demonstrating the reliability of the approach. It was also shown that, in comparison to the proposed approach, treatment of implementation errors as parametric uncertainties significantly increases the complexity of a robust optimization problem and leads to reduced accuracy for fixed sampling budgets.

The engineering problem involved robust optimization of the bandwidth of an optical filter that can be fabricated using an optical integrated circuit. The device was sensitive to manufacturing uncertainties that affected both the design variables and the parameters of the problem. The algorithm's performance on the problem was compared against a redefined version of the robust optimization problem that was affected by only parametric uncertainties (ur Rehman and Langelaar [14]). It was shown that the proposed technique found a superior robust optimum in one-third less number of expensive function calls.

The algorithm, in its current form, can only be applied on problems that do not involve constraints. We aim to extend the infill sampling criterion presented in this work to the general case of robust optimization of constrained problems involving both type of uncertainties i.e. implementation error and parametric uncertainties.

Appendix A: Kriging

In Kriging, the following Gaussian correlation function describes the correlation between any two sample points,

$$\text{Corr}[Y(\mathbf{x}_i), Y(\mathbf{x}_j)] = \exp\left(-\sum_{q=1}^k \theta_q |x_{iq} - x_{jq}|^p\right) \quad (24)$$

where \mathbf{x}_i and \mathbf{x}_j are any two locations in the domain and k represents the total number of dimensions of the problem. We fix p to a constant value of 2. The values for θ_q , μ and σ^2 are chosen such that the likelihood of the observed data is maximized using an optimization-based fitting process. The Maximum Likelihood Estimate (MLE) for the prediction \hat{y} is given by

$$\hat{y}(\mathbf{x}) = \hat{\mu} + \mathbf{r}^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1} \hat{\mu}) \quad (25)$$

where $\hat{\mu}$ is the estimated value for the mean, \mathbf{R} is the $N \times N$ correlation matrix between the N sample points, \mathbf{r} is the vector of correlations between the observed data and the new prediction, while \mathbf{y} is the observed response. The correlation vector \mathbf{r} and the correlation matrix \mathbf{R} are computed using equation (24).

The variance in the Kriging interpolator is found by computing the Mean Squared Error (MSE) in the prediction. This is given by

$$s^2(\mathbf{x}) = \hat{\sigma}^2 \left[1 - \mathbf{r}^T \mathbf{R}^{-1} \mathbf{r} + \frac{1 - \mathbf{1}^T \mathbf{R}^{-1} \mathbf{r}}{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}} \right], \quad (26)$$

The MSE $\hat{s}^2(\mathbf{x})$ is zero at sample point locations since the Kriging prediction interpolates through the true function value at these locations.

Appendix B: Test problems

The test problems on which the algorithm is applied are provided below.

$$f_1(\mathbf{x}_c, \mathbf{x}_e) = 5(x_{c1}^2 + x_{c2}^2) - (x_{e1}^2 + x_{e2}^2) + x_{c1}(-x_{e1} + x_{e2} + 5) + x_{c2}(x_{e1} - x_{e2} + 3). \quad (27)$$

$$\begin{aligned} f_2(\mathbf{x}_c, \mathbf{x}_e) = & 2x_{c1}x_{c5} + 3x_{c4}x_{c2} + x_{c5}x_{c3} + 5x_{c4}^2 + 5x_{c5}^2 - x_{c4}(x_{e4} - x_{e5} - 5) \\ & + x_{c5}(x_{e4} - x_{e5} + 3) + \sum_{i=1}^3 x_{ei}(x_{ci}^2 - 1) - \sum_{i=1}^5 (x_{ei}^2) \end{aligned} \quad (28)$$

$$\begin{aligned} f_3(\mathbf{x}_c, \mathbf{x}_e) = & \left(x_{c2} - \frac{5.1}{4\pi} x_{c2} + \frac{5}{\pi} x_{c1} - 6 \right)^2 + 10 \left(\left(1 - \frac{1}{8\pi} \right) \cos(x_{c1}) + 1 \right) \\ & + (6x_{e1} - 2)^2 \sin(12x_{e1} - 4) + 8x_{e1}, \\ & x_{c1} \in [-5, 10], x_{c2} \in [0, 15], x_{e1} \in [0, 1]. \end{aligned} \quad (29)$$

REFERENCES

- [1] S. P. Gurav, J. F. L. Goosen, and F. VanKeulen, *Bounded-But-Unknown uncertainty optimization using design sensitivities and parallel computing: Application to MEMS*, *Computers & Structures* **83**, 1134 (2005).
- [2] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust optimization* (Princeton University Press, 2009).
- [3] I. Elishakoff, R. T. Haftka, and J. Fang, *Structural design under bounded uncertainty—Optimization with anti-optimization*, *Computers & Structures* **53**, 1401 (1994).
- [4] J. C. Helton, *Uncertainty and sensitivity analysis in the presence of stochastic and subjective uncertainty*, *Journal of Statistical Computation and Simulation* **57**, 3 (1997).
- [5] E. Stinstra and D. den Hertog, *Robust optimization using computer experiments*, *European Journal of Operational Research* **191**, 816 (2008).
- [6] D. Bertsimas, O. Nohadani, and K. M. Teo, *Nonconvex Robust Optimization for Problems with Constraints*, *INFORMS Journal on Computing* **22**, 44 (2010).
- [7] D. Bertsimas, O. Nohadani, and K. M. Teo, *Robust Optimization for Unconstrained Simulation-Based Problems*, *Operations Research* **58**, 161 (2010).
- [8] H. Shah, S. Hosder, S. Koziel, Y. A. Tesfahunegn, and L. Leifsson, *Multi-fidelity robust aerodynamic design optimization under mixed uncertainty*, *Aerospace Science and Technology* **45**, 17 (2015).
- [9] Y. Zhang, S. Hosder, L. Leifsson, and S. Koziel, *Robust airfoil optimization under inherent and model-form uncertainties using stochastic expansions*, in *50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition* (2012).
- [10] D. R. Jones, *A Taxonomy of Global Optimization Methods Based on Response Surfaces*, *Journal of Global Optimization* **21**, 345 (2001).
- [11] A. I. J. Forrester and A. J. Keane, *Recent advances in surrogate-based optimization*, *Progress in Aerospace Sciences* **45**, 50 (2009).
- [12] J. Marzat, E. Walter, and H. Piet-Lahanier, *Worst-case global optimization of black-box functions through kriging and relaxation*, *Journal of Global Optimization* **55**, 707 (2013).
- [13] S. ur Rehman, M. Langelaar, and F. van Keulen, *Efficient Kriging-based robust optimization of unconstrained problems*, *Journal of Computational Science* **5**, 872 (2014).

- [14] S. ur Rehman and M. Langelaar, *Efficient global robust optimization of unconstrained problems affected by parametric uncertainties*, Structural and Multidisciplinary Optimization, 1 (2015).
- [15] A. Forrester, S. Andras, and A. J. Keane, *Engineering design via surrogate modelling : a practical guide* (J. Wiley, 2008).
- [16] H. P. Sacks, J., Welch, W., Mitchell, T. J., and Wynn, *Design and Analysis of Computer Experiments*, Statistical Science 4, 409 (1989).
- [17] D. R. Jones, M. Schonlau, and W. J. Welch, *Efficient Global Optimization of Expensive Black-Box Functions*, Journal of Global Optimization 13, 455 (1998).
- [18] J. Marzat, E. Walter, and H. Piet-Lahanier, *A new strategy for worst-case design from costly numerical simulations*, in *American Control Conference (ACC), 2013* (2013) pp. 3991–3996.
- [19] M. Schonlau and W. Welch, *Global optimization with nonparametric function fitting*, in *Proceedings of the Section on Physical and Engineering Sciences* (American Statistical Association, 1996) pp. 183–186.
- [20] D. Bertsimas and O. Nohadani, *Robust optimization with simulated annealing*, Journal of Global Optimization 48, 323 (2010).
- [21] R. B. Arellano-Valle and M. G. Genton, *On the exact distribution of the maximum of absolutely continuous dependent random variables*, Statistics & Probability Letters 78, 27 (2008).
- [22] R. Heideman, M. Hoekman, and E. Schreuder, *TriPleX-Based Integrated Optical Ring Resonators for Lab-on-a-Chip and Environmental Detection*, Selected Topics in Quantum Electronics, IEEE Journal of 18, 1583 (2012).
- [23] PhoeniX Software, *PhoeniX Software 4.8.4*, (2014).

5

ROBUST OPTIMIZATION OF CONSTRAINED PROBLEMS

5.1. INTRODUCTION

Robustness against uncertainties in a problem is steadily rising as an important aspect in the field of optimization. Uncertainties can be dealt with in different ways depending on the level of information available about the problem. If the probability distributions of all uncertainties are known then the problem is usually addressed with the so called Stochastic Programming approach [1].

However, often full probabilistic data is not available, but bounds on the uncertainties can be given. Such uncertainties are denoted as bounded-but-unknown [2, 3]. Robust optimization can tackle problems of this nature. Robust optimization involves the search for the best worst-case cost, i.e. the minimization of the maximum realizable value of the objective with respect to the uncertainty set, subject to the non-violation of the worst-case constraints. Robust optimization of expensive simulation-based problems is especially challenging since the nested optimization process, when applied to expensive problems, can potentially be very inefficient.

In this work, we present a novel approach for efficient global robust optimization of expensive simulation-based constrained problems affected by bounded-but-unknown uncertainties. The method operates on a surrogate model of the expensive problem which is iteratively improved based on novel infill sampling criteria adapted for constrained robust optimization. The present algorithm enables efficient and accurate determination of the global robust optimum of constrained problems.

Abundant research has been performed in recent decades on robust optimization of problems affected by uncertainties in their problem data. However, much of this work has been focused on solving convex problems. Considerable progress has therefore been made in robust optimization of linear, convex-quadratic, conic-quadratic and semi-definite problems [4]. In contrast, literature related to robust optimization of non-convex problems affected by uncertainties is relatively limited. Bertsimas *et al.* recently

treated robust optimization of non-convex problems with constraints, but the method is aimed at identifying local robust optima only [5]. Additionally, the approach assumes that gradients are available. In general, however, the availability of gradients cannot always be guaranteed.

A vast number of practical problems affected by uncertainties is non-convex. Furthermore, the objective function of such problems is often not explicitly known and therefore has to be treated as a black-box. Such a scenario is typically observed when the objective is a result of a computer simulation. An additional difficulty often encountered is that the inherent simulation is computationally expensive to perform. This further prevents the application of robust optimization in engineering practice.

Applying optimization directly on expensive computer simulations is prohibitively expensive. This is especially true in the case of robust optimization, since robust optimization involves solving a nested min-max optimization problem where the objective to be optimized is itself a result of an optimization. The problem to be tackled therefore renders itself suitable to a surrogate-based optimization strategy where a cheap model is initially constructed via Design of Experiments (DoE) and thereafter, the model is updated using an infill sampling criterion.

Of the various surrogate-based modeling techniques that exist, for the proposed method we employ Kriging [6]. The statistical framework of Kriging provides an estimator of the variance of the Kriging interpolator. Using this potential error, different metrics have been proposed to adaptively sample the design domain such that the deterministic optimum of an unconstrained problem can be found efficiently. The metric of Probability of Improvement (PI) and Expected Improvement (EI) have been shown to be sound statistical infill sampling criteria. By constructing an initial metamodel using a suitable DoE [7] and then employing EI, Jones *et al.* showed that the deterministic global optimum of an unconstrained problem can be found using relatively few expensive simulations [8]. Jones *et al.* used the term Efficient Global Optimization to refer to this method.

In order to apply deterministic optimization on an expensive simulation-based problem with constraints, additional surrogates could be constructed for each constraint. In the Kriging framework, an adaptive sampling scheme for a constrained problem was first explored by Schonlau [9]. Based on the Kriging variance, the method suggests a metric of Probability of Feasibility (PF) for each constraint, analogous to the Probability of Improvement in the objective. The product of probability of feasibility of the constraints and expected improvement of the objective can then be used to suggest new sampling locations. [10, 11] showed that by employing this approach the deterministic global optimum of a constrained problem can be found using relatively few expensive function evaluations.

In addition to providing a basis for an adaptive sampling scheme, Kriging also has the advantage that it generally exhibits superior performance compared to polynomial models when robustness is taken into account [12]. However, a disadvantage of Kriging is that the correlation matrix it generates is prone to ill-conditioning, which may require stabilization [13]. Moreover, Kriging is also known to underestimate the variance in its interpolation [14].

To the best of our knowledge, an infill sampling based approach for surrogate-based global robust optimization of computationally expensive constrained problems affected

by uncertainties has not been previously explored. Recently, Marzat *et al.* demonstrated algorithms for tackling expensive simulation-based problems [15, 16]. Ur Rehman *et al.* also showcased an approach for estimating the global robust optimum of these problems [17, 18]. However this set of methods is limited to unconstrained problems. There is a strong need to address the constrained case since most practical problems affected by uncertainties are, more often than not, subject to constraints. The considered uncertainties are bounded-but-unknown and can affect the parameters as well as the design variables of the problem. Uncertainties affecting the parameters are known as parametric uncertainties. Implementation errors refer to uncertainties that directly affect the design variables.

The primary contribution of this work is to provide a sound criterion for infill sampling of expensive simulation-based constrained problems such that a feasible global robust optimum can be found cheaply. We restrict the focus to problems involving inequality constraints since equality constraints that are affected by uncertainty cannot be satisfied in general. The metamodels of the objective and the constraints are built using Kriging. We derive adapted versions of expected improvement and probability of feasibility to reflect the need for an infill criterion that is suitable for robust optimization instead of deterministic optimization.

The average convergence and mean performance of the method is tested statistically by applying it 100 times on several numerical problems. In addition, the algorithm is applied on an engineering problem that involves an optical filter realized on an integrated circuit which is affected by bounded-but-unknown fabrication uncertainties.

This chapter is organized as follows. Robust optimization of problems with constraints is introduced in Section 5.2. Section 5.3 provides a brief description of Kriging as well as expected improvement and probability of feasibility along with their use in unconstrained and constrained deterministic optimization. We introduce the proposed algorithm for efficient global robust optimization of constrained problems in Section 5.4. Finally, Section 5.5 and 5.6 contain the results and conclusions, respectively.

5.2. ROBUST OPTIMIZATION OF PROBLEMS WITH CONSTRAINTS

A nominal optimization problem subject to constraints may be defined as,

$$\begin{aligned} \min_{\mathbf{x}_c} f(\mathbf{x}) \\ \text{s.t. } h_j(\mathbf{x}_c) \leq 0 \quad \forall j, \end{aligned} \quad (5.1)$$

where $\mathbf{x}_c \in \mathbb{X}_c$ is the set of design variables, $f(\mathbf{x})$ is the objective function and $h_j(\mathbf{x}_c)$ is the set of constraints. If the problem is affected by implementation error $\Delta \in \mathcal{U}$, with \mathcal{U} as the uncertainty set, then this directly impacts the design variables. In this scenario, the robust optimization problem is given by,

$$\begin{aligned} \min_{\mathbf{x}_c} \max_{\Delta} f(\mathbf{x}_c + \Delta) \\ \text{s.t. } \max_{\Delta} h_j(\mathbf{x}_c + \Delta) \leq 0 \quad \forall j. \end{aligned} \quad (5.2)$$

The above formulation shows that robust optimization involves minimizing the worst-case cost instead of the nominal cost. Let us now assume that the problem is affected by

uncertainties in the problem data only. These parametric uncertainties can be modeled using a set of environment variables $\mathbf{x}_e \in \mathbb{X}_e$ where \mathbb{X}_e is the uncertainty set. A robust optimization problem subject to constraints can then be written as,

$$\begin{aligned} \min_{\mathbf{x}_c} \max_{\mathbf{x}_e} f(\mathbf{x}_c, \mathbf{x}_e) \\ \text{s.t. } \max_{\mathbf{x}_e} h_j(\mathbf{x}_c, \mathbf{x}_e) \leq 0 \quad \forall j. \end{aligned} \quad (5.3)$$

Observing the above equation, we note that the *worst-case* constraints with respect to the uncertainty set \mathbb{X}_e should not be violated in order to find a feasible solution. Therefore, the global robust optimum would be the location that provides the best worst-case cost, given that that location does not violate the worst-case constraints.

For some problems, uncertainties could impact both the design variables and the parameters. In this case, the solution has to be robust against parametric uncertainties as well as implementation error. The robust optimization problem, subject to constraints, for this general problem is given by

$$\begin{aligned} \min_{\mathbf{x}_c} \max_{\mathbf{x}_e, \Delta} f(\mathbf{x}_c + \Delta, \mathbf{x}_e) \\ \text{s.t. } \max_{\mathbf{x}_e, \Delta} h_j(\mathbf{x}_c + \Delta, \mathbf{x}_e) \leq 0 \quad \forall j. \end{aligned} \quad (5.4)$$

The objective function and the constraints are considered to be non-convex. Furthermore, we assume that the function and the constraints are based on the response of an expensive computer simulation. Therefore, the ultimate goal of this work is to estimate a feasible global robust optimum of the constrained problem in Equation (5.4) using a relatively small number of expensive simulations. This is performed by operating on cheap Kriging models of the objective and constraints instead of on the expensive computer simulation. The problem is expressed as

$$\begin{aligned} \min_{\mathbf{x}_c} \max_{\mathbf{x}_e, \Delta} \mathcal{K}_f(\mathbf{x}_c + \Delta, \mathbf{x}_e) \\ \text{s.t. } \max_{\mathbf{x}_e, \Delta} \mathcal{H}_j(\mathbf{x}_c + \Delta, \mathbf{x}_e) \leq 0 \quad \forall j, \end{aligned} \quad (5.5)$$

where \mathcal{K}_f is the Kriging model of the objective and $\mathcal{H}_j \forall j$ are the Kriging models of the constraints. In order to estimate the global robust optimum accurately, the surrogate models need to approximate the corresponding reference functions very well, especially in the neighbourhood of the robust optimum. Extra emphasis needs to be paid to the metamodel error in the constraint models $\mathcal{H}_j \forall j$, since a feasible robust optimum on the metamodel should ideally also be feasible on the true function.

In the following section, we will discuss Kriging and its application on deterministic optimization of constrained problems. This will provide a basis for the algorithm proposed in Section 4, which strives to solve Equation (5.4). The scheme uses infill sampling criteria based on Kriging that enable Equation (5.5) to approximate Equation (5.4) increasingly well in potential regions of interest for global robust optimization of a given problem.

5.3. KRIGING-BASED DETERMINISTIC OPTIMIZATION OF CONSTRAINED PROBLEMS

5.3.1. KRIGING

A very brief description of the metamodelling technique known as Kriging is provided herein. For detailed explanation concerning the model construction and interpolation please refer to [6].

Kriging is an interpolation technique that assumes that the function response at any position in the design domain can be described as a normally distributed random variable. It employs a tunable Gaussian basis function, Equation (22), to describe the correlation between any two sample points. Optimal values for the tunable parameters of this basis function are found by maximizing the likelihood of obtaining the observed data. Using this tunable basis function, the Kriging prediction, \hat{y} , is estimated by maximizing the combined likelihood of the observed data and the predicted value, Equation (23). The statistical basis of Kriging provides an estimate of the variance, s^2 , in the Kriging interpolator, Equation (24).

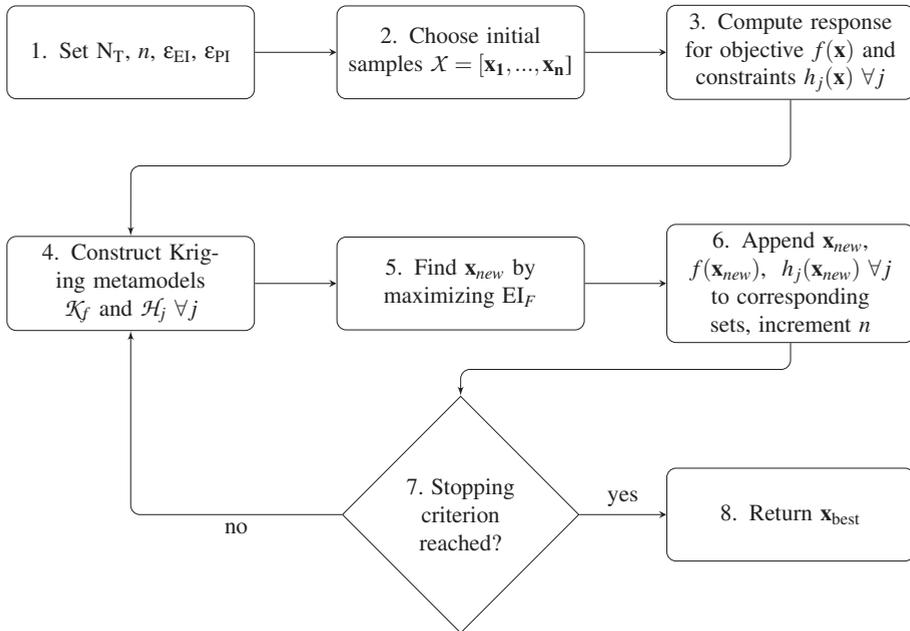


Figure 5.1: Flowchart of deterministic optimization using constrained EI. The algorithm finds the nominal optimum of a constrained problem with relatively few function calls of an expensive to evaluate function.

5.3.2. DETERMINISTIC UNCONSTRAINED OPTIMIZATION

Rather than working with a predetermined and static set of samples, it proves more efficient to adaptively extend the set of samples to refine the approximation. The combi-

nation of the Kriging interpolator and its variance has been successfully used to devise adaptive sampling schemes for efficient optimization of black-box functions. Jones *et al.* proposed the Efficient Global Optimization (EGO) algorithm [8] for deterministic unconstrained optimization based on the Kriging framework. Their method used the adaptive sampling criterion of Expected Improvement (EI).

The EI criterion is developed by assuming that the uncertainty in the predicted value $\hat{y}(\mathbf{x})$ at a position \mathbf{x} can be described in terms of a normally distributed random variable $Y(\mathbf{x})$. The Kriging interpolator $\hat{y}(\mathbf{x})$ is assumed to be the mean of this random variable while the variance is assumed to be given by the Kriging mean squared error $s^2(\mathbf{x})$. There is a possibility for improving on the current observed minimum, y_{\min} , if a part of the distribution $Y(\mathbf{x})$ lies below y_{\min} . Let this improvement be denoted by I . Finding the expectation of the improvement I , i.e. $E[I(\mathbf{x})] = E[\max(y_{\min} - Y, 0)]$, gives us the expected improvement. A cheaply computable analytical expression may be derived for the EI [19],

$$E[I(\mathbf{x})] = (y_{\min} - \hat{y})\Phi\left(\frac{y_{\min} - \hat{y}}{s}\right) + s\phi\left(\frac{y_{\min} - \hat{y}}{s}\right) \quad (5.6)$$

where $\Phi(\cdot)$ is the normal cumulative distribution function and $\phi(\cdot)$ is the normal probability density function. A global optimizer can be used to estimate the global maximizer of EI. The expensive function is evaluated at the maximizer location and the metamodel is rebuilt with the augmented set of samples and responses. By iteratively sampling the metamodel using EI the global deterministic optimum of the problem can be found in relatively few iterations.

5.3.3. DETERMINISTIC CONSTRAINED OPTIMIZATION

The constraints are also considered to be a result of an expensive computer simulation. Therefore, a cheap model has to be built for them as well. An option could be to include each constraint as a penalty term, but for more complex constraints this approach does not work well [11].

In order to deal with constraints a measure known as Probability of Feasibility (PF) was developed by [9]. The criterion is analogous to the probability of improvement for the objective. Let the constraint metamodel be denoted by $\mathcal{H}(\mathbf{x})$ and the Kriging prediction by $\hat{h}(\mathbf{x})$. To derive the expression for probability of feasibility, it is again assumed that the uncertainty in the predicted value $\hat{h}(\mathbf{x})$ at a position \mathbf{x} can be described in terms of a normally distributed random variable $H(\mathbf{x})$ with mean $\hat{h}(\mathbf{x})$ and variance $s^2(\mathbf{x})$. The measure gives the area of the distribution $H(\mathbf{x})$ that is below the constraint limit h_{\min} or $P[F(\mathbf{x}) < h_{\min}]$. For a single constraint the probability of feasibility is given by

$$P[F(\mathbf{x}) < h_{\min}] = \Phi\left(\frac{h_{\min} - \hat{h}}{s}\right). \quad (5.7)$$

Typically, the constraint expression is rearranged so that the constraint limit $h_{\min} = 0$. Just like expected improvement and probability of improvement, the probability of feasibility is an analytical expression that is cheaply computable. The probability of feasibility is basically a metric that gives an indication of possible feasible regions in the domain.

The product of expected improvement in the objective and the probability of feasibility of the constraint can then provide a suitable infill criterion for constrained problems [10, 11],

$$EI_F = E[I(\mathbf{x})]P[F(\mathbf{x}) < h_{min}]. \quad (5.8)$$

By estimating the global maximizer of the constrained expected improvement, EI_F , a suitable location at which to sample both the objective metamodel and constraint metamodels can be found. The method can readily be extended to multiple constraints by using the total probability of feasibility, which is given by the product of the individual probability of feasibility of each constraint.

Figure 5.1 shows the flowchart of the algorithm for deterministic optimization of constrained problems using Kriging and adaptive sampling. The algorithm is initialized in Step 1. A suitable Design of Experiments strategy is used to choose the initial sampling regions in Step 2. Once the responses of the objective and the constraints are found, the Kriging models are constructed in Step 4. Thereafter, the global maximizer of EI_F is estimated and this is assigned as the new location to be sampled, \mathbf{x}_{new} . The response of the objective and the constraints at \mathbf{x}_{new} are computed in Step 6. The process of constructing the objective and constraint metamodels and adaptively sampling the domain is repeated until the stopping criterion in Step 7 is reached. A possible stopping criterion could be the point at which $n = N_T$. Alternatively, depending on the sampling criterion used, the algorithm may be stopped when either the maximum EI_F falls below ϵ_{EI} . At this stage, the feasible sample corresponding to the minimum objective value is returned as the solution \mathbf{x}_{best} . This algorithm has successfully been demonstrated on deterministic constrained problems by [10, 11]. Parr *et al.* used EI_F as the infill sampling criterion in their work.

5.4. EFFICIENT GLOBAL ROBUST OPTIMIZATION OF CONSTRAINED PROBLEMS

A scheme for Kriging-based deterministic optimization of expensive simulation-based constrained problems was introduced in the previous section. We now propose an efficient technique, based on Kriging, for *global robust* optimization of expensive simulation-based constrained problems.

In this section, it is shown how the robust optimum can be found for a problem affected by parametric uncertainties only, Equation (5.3). The basic principle of the algorithm does not change even if the problem to be solved is affected by implementation error only, Equation (5.2) or is affected by both implementation error and parametric uncertainties, Equation (5.4). For clarity, we focus our discussion on an algorithm that solves Equation (5.3). Separate treatment of implementation error can result in efficiency improvement. We refer to [17] for a detailed discussion of this aspect.

Figure 5.2 illustrates the steps that are involved in estimating the robust optimum. The foundation of the method is the same as the one for deterministic Kriging-based optimization. Both approaches depend on the same initialization phase, i.e. Step 1 to Step 4 are identical. This is followed by an iterative surrogate update process where a single new adaptive sample is added in each iteration.

The significant difference between the two methods is the actual process by which

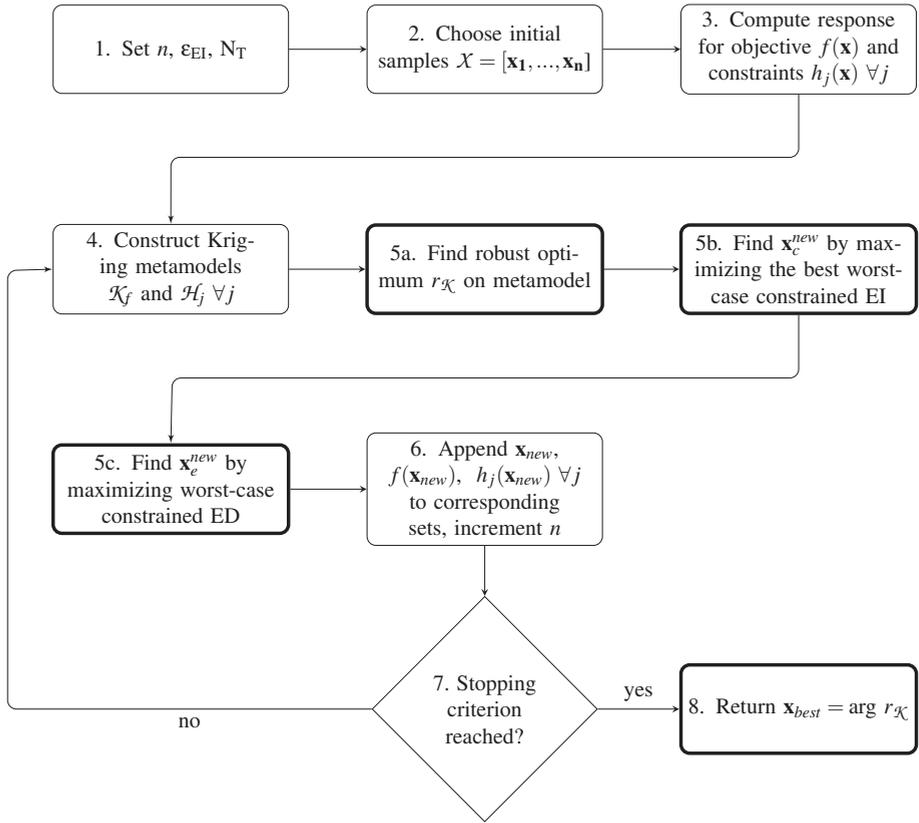


Figure 5.2: Flowchart shows the algorithm for efficient global robust optimization of constrained problems. The steps with the bold borders represent the changes that have been made to the algorithm in Figure 5.1 in order to reflect the fact that we are searching for a robust optimum.

this new sample \mathbf{x}_{new} is found at each iteration. In deterministic optimization, the search for \mathbf{x}_{new} simply involved maximizing Equation (5.8). However for robust optimization this process has to be broken down into several steps. A reference metric for the robust optimum is first required. This is given by the constrained robust optimum, $r_{\mathcal{X}}$, on the metamodel.

When estimating the robust optimum $r_{\mathcal{X}}$ on the metamodel, the effect of the metamodel error also has to be included. In particular, errors in the constraint surrogate can result in an infeasible solution being chosen as the robust optimum. To mitigate the effect of the metamodel error, [12] suggested a method that makes use of the variance in the Kriging interpolator of the constraints. The strategy basically involved adding the standard deviation of the constraint metamodel to the Kriging prediction of the constraint. This would result in a more conservative constraint metamodel, especially in regions with high uncertainty and thereby reduce the chance of obtaining an infeasible

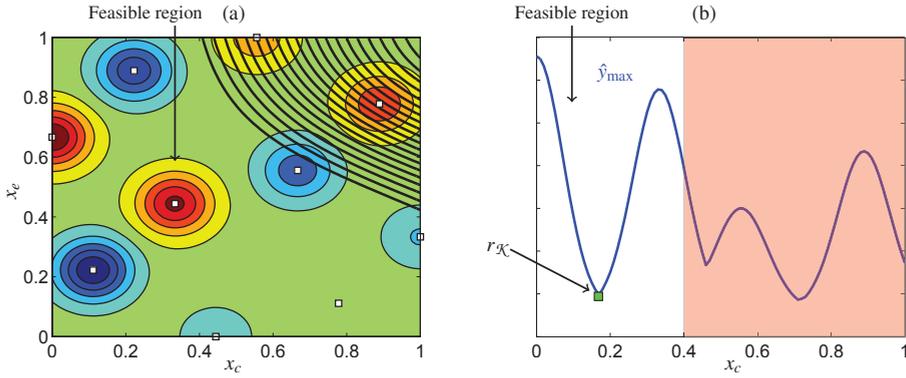


Figure 5.3: Plot (a) shows a Kriging metamodel of a two-dimensional function. The constraint boundary of the constraint metamodel is also plotted. The infeasible region is given by the area covered by the black lines. Plot (b) shows the worst-case Kriging metamodel as well as the worst-case constraint boundary. The infeasible region in (b) is shaded in pink. The robust optimum location is also indicated on the plot.

solution. The robust optimum $r_{\mathcal{K}}$ is therefore found via,

$$\begin{aligned} \min_{\mathbf{x}_c} \max_{\mathbf{x}_e} \mathcal{H}_f(\mathbf{x}_c, \mathbf{x}_e) \\ \text{s.t. } \max_{\mathbf{x}_e} \mathcal{H}_j(\mathbf{x}_c, \mathbf{x}_e) + \kappa s_j(\mathbf{x}_c, \mathbf{x}_e) \leq 0 \quad \forall j, \end{aligned} \tag{5.9}$$

where $s_j(\mathbf{x}_c, \mathbf{x}_e)$ is the metamodel standard deviation for the j^{th} constraint metamodel at location $(\mathbf{x}_c, \mathbf{x}_e)$. The parameter κ , chosen between $[0, 1]$ is a measure of how conservative we want to be with respect to the metamodel error. A value of zero for κ means that the metamodel error is not included, while higher values indicate a more conservative approach.

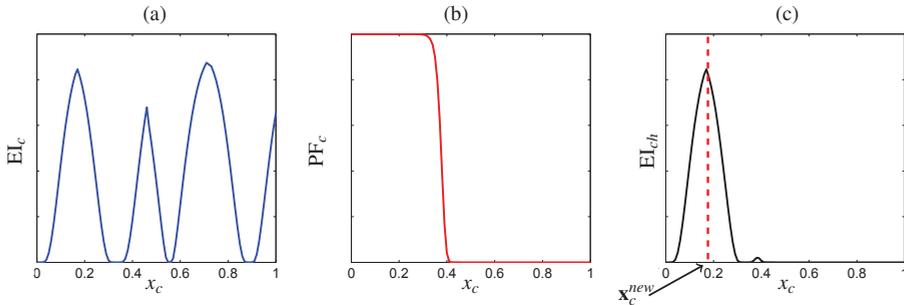


Figure 5.4: Plot (a) shows the expected improvement, EI_c , in the objective. The Probability of Feasibility of the constraint is plotted in (b). Plot (c) shows the product of EI_c and PF_c . The new control variable location $\mathbf{x}_c^{\text{new}}$ is also indicated.

After locating $r_{\mathcal{K}}$, the search is divided into two parts. First, the optimal sampling location in the control variable space, \mathbb{X}_c , is found and then we search for the optimal sampling location in the environment variable space, \mathbb{X}_e . The adaptive sampling measures needed to perform this search are also suitably adapted for estimating regions of interests for locating the robust optimum rather than the deterministic optimum.

5.4.1. OPTIMAL SAMPLING LOCATION IN \mathbb{X}_c

The optimal sampling location $\mathbf{x}_c^{new} \in \mathbb{X}_c$ should be the infill location corresponding to the highest expectation of improvement over the current constrained robust optimum $r_{\mathcal{K}}$, Equation (5.9). The search for \mathbf{x}_c^{new} is performed in Step 5b of the flowchart in Figure 5.2.

To illustrate how this sampling location is found, we make use of an example function of a single control variable x_c and a single environment variable x_e . The problem has one constraint, which is also a function of both x_c and x_e . Figure 5.3(a) shows the Kriging metamodel \mathcal{K}_f of the two dimensional function, based on a set of initial samples and responses. The plot also contains the constraint boundary of the constraint metamodel \mathcal{H} . The feasible region is also indicated on the plot. The construction of the Kriging metamodel of the objective and the constraint, Equation (5.9), involves going through Step 1 to Step 4 of the flowchart in Figure 5.2.

Figure 5.3(b), on the other hand, shows the worst-case Kriging metamodel \hat{y}_{max} ,

$$\hat{y}_{max}(\mathbf{x}_c) = \max_{\mathbf{x}_e \in \mathbb{X}_e} \mathcal{K}_f(\mathbf{x}_c, \mathbf{x}_e). \quad (5.10)$$

The maximizer of Equation (5.10) is denoted by \mathbf{x}_e^{max} . The region where the worst-case constraint, \hat{h}_{max} , has a predicted response greater than the constraint limit is indicated in pink in Figure 5.3(b),

$$\hat{h}_{max}(\mathbf{x}_c) = \max_{\mathbf{x}_e \in \mathbb{X}_e} \mathcal{H}(\mathbf{x}_c, \mathbf{x}_e). \quad (5.11)$$

The minimum value for \hat{y}_{max} within the feasible region gives the constrained robust optimum $r_{\mathcal{K}}$. The process of estimating the robust optimum corresponds to Step 5a in the flowchart in Figure 5.2.

A constrained expected improvement criterion is required for identifying a promising location at which to sample in \mathbb{X}_c . Following the method described for deterministic optimization, this would involve obtaining an expected improvement expression for the objective and a probability of feasibility for the constraint.

To formulate the EI in the objective, it is assumed that the uncertainty in the worst-case Kriging prediction \hat{y}_{max} , at any location $(\mathbf{x}_c, \mathbf{x}_e^{max})$, can be described in terms of a normally distributed random variable Y_{max} with mean \hat{y}_{max} and variance $s^2(\mathbf{x}_c, \mathbf{x}_e^{max})$. We can improve over the current robust optimum $r_{\mathcal{K}}$ when $Y_{max} < r_{\mathcal{K}}$. It was shown by [18] that the expectation of this improvement, I_c , is given by

$$\underbrace{E[I_c(\mathbf{x}_c)]}_{EI_c} = (r_{\mathcal{K}} - \hat{y}_{max}) \Phi \left(\frac{r_{\mathcal{K}} - \hat{y}_{max}}{s} \right) + s \phi \left(\frac{r_{\mathcal{K}} - \hat{y}_{max}}{s} \right). \quad (5.12)$$

The plot in Figure 5.4(a) shows the expected improvement EI_c as a function of the control variable x_c .

To come up with the probability of feasibility expression, we again make use of a normal distribution to model the uncertainty in the worst-case constraint, \hat{h}_{max} . Therefore, the uncertainty in the worst-case constraint, \hat{h}_{max} , at any location $(\mathbf{x}_c, \mathbf{x}_e^{max})$ is treated in terms of a normally distributed random variable H_{max} with mean \hat{h}_{max} and variance $s^2(\mathbf{x}_c, \mathbf{x}_e^{max})$. The probability of feasibility is then given by the area of the distribution H_{max} that is below the constraint limit h_{min} . The expression for a single constraint can be written as,

$$\underbrace{P[F_c(\mathbf{x}_c)]}_{PF_c} = \Phi\left(\frac{h_{min} - \hat{h}_{max}}{s}\right). \quad (5.13)$$

The plot in Figure 5.4(b) shows the probability of feasibility PF_c as a function of the control variable x_c when h_{min} is considered to be at the constraint limit.

As in deterministic constrained optimization, a suitable infill criterion in \mathbb{X}_c can be found by maximizing the product of the expected improvement, EI_c , in the objective and the probability of feasibility, PF_c , in the constraint,

$$EI_{ch} = E[I_c(\mathbf{x}_c)] P[F_c(\mathbf{x}_c)]. \quad (5.14)$$

The new sampling location, $\mathbf{x}_c^{new} \in \mathbb{X}_c$ is obtained by determining the global maximizer of Equation (5.14). Multiple constraints are handled by using the total probability of feasibility which is given by the probability of feasibility of each constraint. Figure 5.4(c) shows a plot of EI_{ch} . The new control variable location \mathbf{x}_c^{new} , given by the location of the global maximum, is also indicated. \mathbf{x}_c^{new} is determined in Step 5b in the flowchart in Figure 5.2.

5.4.2. OPTIMAL SAMPLING LOCATION IN \mathbb{X}_e

After choosing \mathbf{x}_c^{new} , the algorithm searches for the optimal infill sampling location, \mathbf{x}_e^{new} , in the environment variable space \mathbb{X}_e . Figure 5.5(a) shows the same Kriging metamodel of the objective and the constraint boundary of the constraint metamodel along with the feasible region. The location of \mathbf{x}_c^{new} is also shown on the plot. Figure 5.5(b) shows the Kriging prediction of the objective at \mathbf{x}_c^{new} , corresponding to the line of plot (a), plotted with respect to \mathbf{x}_e . The worst-case cost $g_{\mathcal{N}}$ is also shown on the plot. The worst-case is given by,

$$g_{\mathcal{N}}(\mathbf{x}_c^{new}, \mathbf{x}_e) = \max_{\mathbf{x}_e \in \mathbb{X}_e} \mathcal{K}_f(\mathbf{x}_c^{new}, \mathbf{x}_e). \quad (5.15)$$

Figure 5.5(c) shows the Kriging prediction of the constraint at \mathbf{x}_c^{new} . Again the worst-case constraint value $g_{\mathcal{H}}$ is also shown on the plot. In general, the worst-case constraint value is given by,

$$g_{\mathcal{H}}(\mathbf{x}_c^{new}, \mathbf{x}_e) = \max_{\mathbf{x}_e \in \mathbb{X}_e} \mathcal{H}(\mathbf{x}_c^{new}, \mathbf{x}_e). \quad (5.16)$$

An adaptive sampling criterion is needed in the environment variable space to suggest \mathbf{x}_e^{new} . Choosing \mathbf{x}_e^{new} involves finding a location that could potentially give a higher, i.e. more pessimistic, value than $g_{\mathcal{N}}$ and $g_{\mathcal{H}}$. This is the goal since the aim is to find the

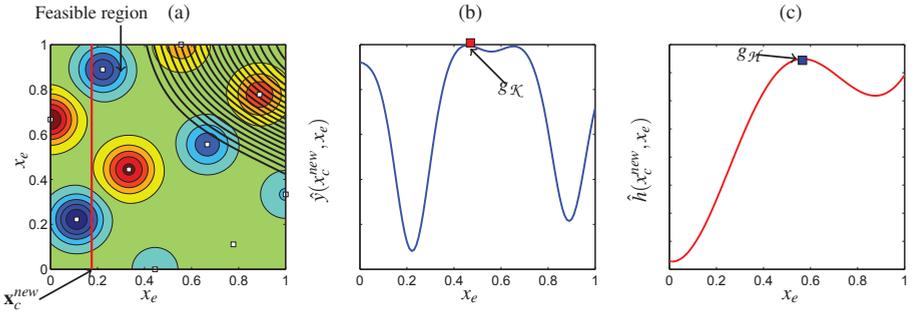


Figure 5.5: Plot (a) shows the same Kriging metamodel along with the constraint boundary of the constraint metamodel and the infeasible region, covered by the black lines. The location of x_c^{new} is also shown. The Kriging prediction at x_c^{new} , corresponding to the response along the red line in plot (a), is plotted with respect to x_e in (b). Plot (c) shows the Kriging prediction of the constraint at x_c^{new} .

5

most adverse situation in the environment variable space \mathbb{X}_e . An expected *deterioration* (ED) criterion for the objective should therefore help identify a location with the highest expected value relative to $g_{\mathcal{X}}$. Similarly, an ED measure for the constraint should aid in estimating a location with the highest expected constraint value relative to $g_{\mathcal{H}}$.

In an earlier work [18] the authors have derived the expression for the expected deterioration in the objective with respect to the environment variable space. This is given

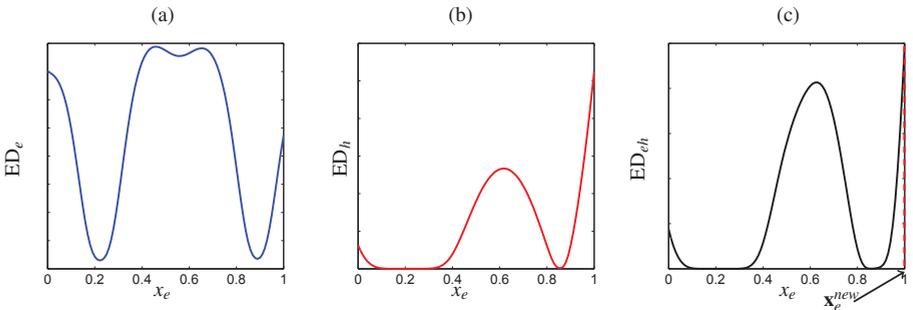


Figure 5.6: Plot (a) shows the expected deterioration, ED_e , in the objective with respect to the environment variable. ED_h is plotted in (b). Plot (c) shows the constrained expected deterioration ED_{eh} . The new environment variable location x_e^{new} is also indicated.

by,

$$\underbrace{E[D_e(\mathbf{x}_c^{new}, \mathbf{x}_e)]}_{EI_e} = (\hat{y} - g_{\mathcal{X}}) \Phi \left(\frac{\hat{y} - g_{\mathcal{X}}}{s} \right) + s \phi \left(\frac{\hat{y} - g_{\mathcal{X}}}{s} \right). \quad (5.17)$$

The plot in Figure 5.6(a) shows the expected deterioration, EI_e , as a function of the environment variable, x_e . The expected deterioration in the constraint is completely analogous to EI_e in the objective. The expression is given by,

$$\underbrace{E[D_h(\mathbf{x}_c^{new}, \mathbf{x}_e)]}_{ED_h} = (\hat{h} - g_{\mathcal{H}}) \Phi \left(\frac{\hat{h} - g_{\mathcal{H}}}{s} \right) + s \phi \left(\frac{\hat{h} - g_{\mathcal{H}}}{s} \right). \quad (5.18)$$

In the case of multiple constraints, the total expected deterioration in the constraint, ED_h , can be found by taking the product of the individual ED for each constraint. The plot in Figure 5.6(b) shows the expected deterioration ED_h in the constraint as a function of the environment variable x_e .

The new sampling location \mathbf{x}_e^{new} can be found by determining the maximizer of the product of the expected deterioration in the objective, EI_e , and the expected deterioration in the constraint, ED_h ,

$$ED_{eh} = E[D_e(\mathbf{x}_c^{new}, \mathbf{x}_e)] E[D_h(\mathbf{x}_c^{new}, \mathbf{x}_e)]. \quad (5.19)$$

Figure 5.6(c) shows a plot of the product ED_{eh} along with the location of the new environment variable location, \mathbf{x}_e^{new} . Step 5c in the flowchart in Figure 5.2 involves the search for the new environment variable, \mathbf{x}_e^{new} .

5.4.3. IMPLEMENTATION ASPECTS

Once \mathbf{x}_{new} is identified, the objective and the constraints are evaluated using the expensive simulation at the new location. Thereafter, if the stopping criterion has not been reached yet, the metamodel is rebuilt and the process of searching for \mathbf{x}_c^{new} and \mathbf{x}_e^{new} is repeated. The algorithm is stopped when the total number of function evaluations available N_T is exhausted. Additionally, the algorithm can also be terminated if the robust optimum, $r_{\mathcal{X}}$, found over the last few iterations, does not change significantly. For this purpose, we maintain a history set, \mathcal{S}_h , that consists of the robust optimum found at each iteration.

Apart from its use as a termination criterion, the history set can aid in the search for the robust optimum. Whenever a new search for the robust optimum $r_{\mathcal{X}}$ is initiated at a particular iteration, the starting points for the search can include the history set, \mathcal{S}_h , of the robust optima locations found in the previous iterations. In this manner, it is ensured that a possible robust optimum location found in previous iterations is not

missed by the global search in the current iteration. By doing so, we also systematically reduce the estimation error of the internal global optimizers.

Once the algorithm terminates, the location of the robust optimum, $r_{\mathcal{X}}$, found at the last iteration is returned as the final solution. The result from the final iteration is chosen instead of any other iteration since the last iteration includes the most information about the problem.

5.5. RESULTS

5.5.1. TESTING METHODOLOGY

In order to test the ability of the algorithm to reliably and efficiently converge to the global robust optimum, its numerical performance is evaluated on five analytical test problems and one engineering case study. The test problems are provided in the Appendix. The objective functions for the first four problems are well known benchmark problems which were originally employed as test problems for unconstrained min-max optimization by [20]. The corresponding expressions for the constraints have been chosen such that a feasible solution exists for all problems while, at the same time, ensuring that the global robust optimum is not given by a trivial solution.

We assess the ability of the method to find the robust optimum of a constrained problem by testing it on Problem P1, P2 and P4. All three problems have a single constraint. To get insight into the capacity of the technique to handle more constraints, it is tested on Problem P3 and P5, which have two inequality constraints. Similarly, the ability of the algorithm to deal with different kinds of constraints is analyzed by choosing some constraints to be nonlinear, e.g. P1 and P2, while keeping others linear, e.g. P4. Additionally, P3 has both linear and nonlinear constraints. Another important aspect is the evaluation of the scalability of the algorithm. To this end, Problem P4, which is a function of 10 dimensions in both the objective and the constraint, is used as a test case. P1, P2 and P3 are a function of 2 control variables and 2 environment variables, while P10 is a function of 5 control variables and 5 environment variables. Note that while a problem with 10 variables may not qualify as a large problem in deterministic optimization, it is substantially more challenging in the robust case. The nested nature of robust optimization makes that computational costs increase significantly faster with problem size than in the deterministic case.

Since the initial sampling, performed via space-filling, is random, the results of each run may be different. However, the method should be able to converge regardless of the initial samples. In order to test the repeatability and reproducibility of the algorithm, it is run 100 times on each test problem and the statistical results are analyzed. The number of initial samples n are chosen as $n = 10 \times n_d$, where n_d is the number of dimensions of the problem. The maximum function evaluations available, N_T , is set to 150 for P1, P2 and P3. For the larger problem P4, $N_T = 450$.

The algorithm's performance is also tested on a polynomial problem proposed by Bertsimas *et al.* as a test case for robust optimization of constrained problems [5]. The test case, listed as problem P5 in the Appendix, is a 2-dimensional non-convex problem with two non-linear constraints. The problem is assumed to be affected by implementation error $\Delta = [\Delta x_{c1} \ \Delta x_{c2}]$ such that $\|\Delta\|_2 \leq 0.5$. The uncertainty set, which takes the

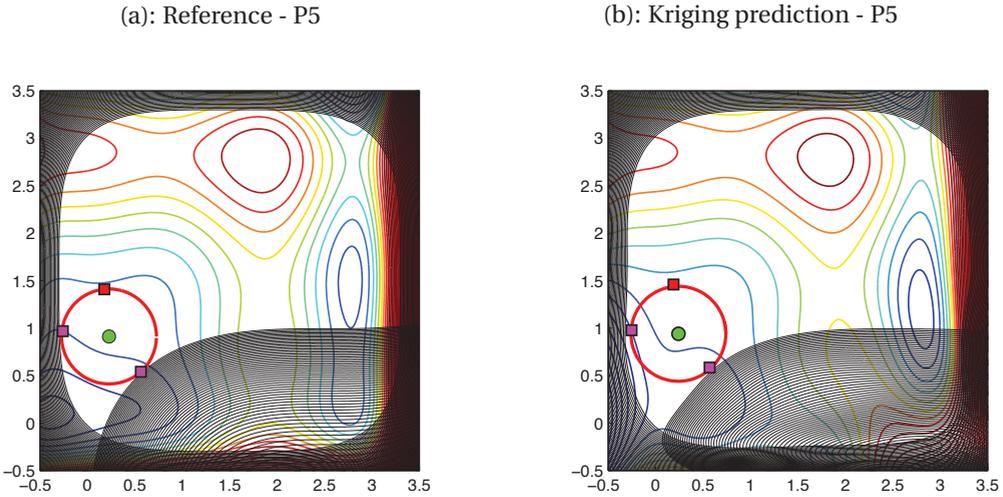


Figure 5.7: Plot (a) shows contour lines of the reference function and constraints for problem P5 along with the location of the robust optimum, indicated by the small green circle. Plot (b) shows contour lines of the Kriging prediction of the function and constraints along with the location of the robust optimum on the Kriging surface, (small green circle). The big red circle, in both plots, indicates the extent of the 2-norm uncertainty set. The red square, in the two plots, shows the location of the best worst-case cost for the objective. Finally, the magenta squares, in both plots, indicate the worst-case cost with respect to the constraints.

form of a circle, is convex. The maximum function evaluations for P5 is set to $N_T = 65$.

In addition to testing it on analytical benchmark problems, the algorithm is also applied on an engineering case study. The problem consists of an optical filter based on a ring resonator that is fabricated as an optical integrated circuit. The fabrication is affected by variations. The behavior of the filter is highly sensitive to these manufacturing uncertainties. Therefore the problem lends itself to evaluation of the effectiveness of the algorithm in a practical situation.

5.5.2. NUMERICAL PERFORMANCE EVALUATION

A TYPICAL EXAMPLE

Before discussing the statistical performance of the algorithm, we visually compare the metamodel and the robust optimum at the final iteration of the algorithm against the reference function for problem P5. Test problem P5 was used by Bertsimas *et al.* to demonstrate their method on robust optimization of constrained problems [5]. In this work, we use the problem simply as a benchmark example. The purpose therefore is not to compare the proposed method against the approach of Bertsimas *et al.*, since their approach is complimentary to this work and can be integrated with the presented algorithm.

The test problem is non-linear and non-convex. Therefore it serves as a challenging test case to analyze the ability of the proposed algorithm to estimate the global robust optimum efficiently. Problem P5 is affected by implementation error $\Delta = [\Delta x_{c1} \Delta x_{c2}] \in$

Table 5.1: Reference results of all the test problems. The functions are listed in the Appendix. n_h represents the number of constraints in the problem. $f(\mathbf{x}_c, \mathbf{x}_e)$ is the optimum objective value. n_d gives the total number of dimensions.

Problem no.	n_h	\mathbb{X}_c	\mathbb{X}_e	\mathbf{x}_c	\mathbf{x}_e	f	n_d
P1	1	$[-5, 5]^2$	$[-5, 5]^2$	-3.9462 -2.6972	0.6251 -0.6247	87.19	4
P2	1	$[-5, 5]^2$	$[-5, 5]^2$	-1.1005 -1.5803	0.3051 2.5455	44.87	4
P3	2	$[-5, 5]^2$	$[-5, 5]^2$	-0.3502 2.5	-0.0509 5	59.59	4
P4	1	$[-5, 5]^5$	$[-3, 3]^5$	-2.0935 1.8516 -1.4695 -1.0453 0.2691	1.6765 1.2508 0.5925 0.6599 -0.6599	-0.3866	10
P5	2	$[-0.5, 3.5]^2$	-	0.228 0.912	-	7.09	2

Table 5.2: The average numerical performance of the algorithm based on 100 runs evaluated on the test problems provided in the Appendix. The table shows the mean and standard deviation of the robust optimum along with the average robust optimum locations in \mathbb{X}_c and \mathbb{X}_e . The total number of evaluations on the expensive function is shown and the number of dimensions n_d are also shown.

Problem no.	Mean \mathbf{x}_c	Mean \mathbf{x}_e	Mean f	SD f	Total evaluations	n_d	Infeasible solutions
P1	-3.8042 -2.6736	0.6521 -0.6306	87.11	0.271	77	4	2%
P2	-1.0256 0.1478	0.2655 2.7550	44.99	1.513	83	4	1%
P3	-0.3497 2.5017	0.0474 5.0000	59.65	0.2699	67	4	0%
P4	-2.1160 1.8492 -1.4000 -1.0605 0.2625	1.7102 1.1378 0.5209 0.7449 -0.6598	-0.3945	0.0227	392	10	0%
P5	0.2321 0.9243	-	7.162	0.058	60	2	3%

\mathcal{U} such that $\|\Delta\|_2 \leq 0.5$. Since the problem has only 2 control variables that are both affected by implementation error, it is easy to visualize the function and constraints surface. Figure 5.7(a) shows a contour plot of the reference function and constraints for problem P5. Figure 5.7(b) shows the contour plot of the Kriging prediction of P5 after 45 iterations of the algorithm and 65 expensive simulations have been performed. On both plots, the location of the global robust optimum is given by a green circle. The robust optimum is circumscribed by a red circle in both Figure 5.7(a) and Figure 5.7(b). The region bounded by this red circle is the 2-norm uncertainty set, $\|\Delta\|_2 \leq 0.5$. The red square, in the two plots, shows the location of the best worst-case cost for the objective. On the other hand, the magenta squares, in both plots, indicate regions with the highest risk of potential constraint violation for each constraint.

Visually, Figure 5.7(a) and 5.7(b) seem quite similar. The location of the reference robust optimum as well as the best worst-case cost with respect to the objective also visually matches on both plots. The sample points are added to the problem in such a way that the Kriging prediction for the objective and the constraints is trustworthy in the local region in the neighborhood of the robust optimum. The figure shows that the algorithm samples the expensive function in such a way that after 65 simulations it is able to accurately estimate the location of the global robust optimum. Additionally, the constrained expected improvement criterion ensures that the whole design domain is explored and a potential solution is not missed due to any inaccuracy in local regions. In the next subsection it will be shown, based on 100 runs of the algorithm on problem P5, that the proposed approach shows consistent convergence to the robust optimum.

BENCHMARK STATISTICS

The reference robust optima and their corresponding locations for the five numerical test problems are shown in Table 5.1. These optima were obtained by direct robust optimization using the analytical expressions, i.e. without metamodel error. The number of constraints, n_h , is shown in column 2 while the number of total dimensions of the problem n_d is given in the last column. The domain size in \mathbb{X}_c and \mathbb{X}_e is provided in column 3 and 4, respectively. Column 5 shows the robust optimum location for \mathbf{x}_c while column 6 gives the robust optimum location for \mathbf{x}_e . The robust optimum objective value is given by $f(\mathbf{x}_c, \mathbf{x}_e)$ in the second last column.

It is important to realize that the worst-case location for \mathbf{x}_e is different for the objective as opposed to the worst-case value for \mathbf{x}_e in the case of a constraint. The locations for \mathbf{x}_e listed in the table represent only the robust optimum location in \mathbb{X}_e . On the other hand, the maximizer in \mathbb{X}_e for each constraint has not been listed.

Table 5.2 shows the average numerical performance of the proposed approach based on 100 runs of each test problem. The problem number is given by the first column. The second and third column provide the mean robust optimum location in \mathbb{X}_c and \mathbb{X}_e based on the 100 runs, respectively. The mean and standard deviation of the objective value at the robust optimum for each function are given in column four and five, respectively. The average total number of expensive function evaluations required to achieve this average performance is given in the sixth column. The second last column gives the number of dimensions of each problem.

The ratio of the mean robust optimum objective value (column 4 in Table 5.2) to the reference robust optimum (column 7 in Table 5.1) is plotted in Fig. 5.8 for the five test

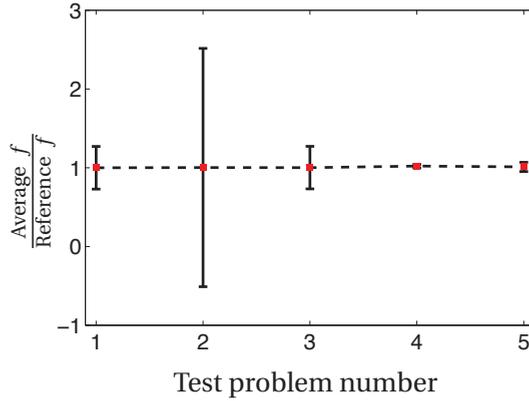


Figure 5.8: Ratio of the mean robust optimum, found by the algorithm based on 100 runs, to the reference robust optimum is plotted. The error bars show the standard deviation around the mean value for each problem.

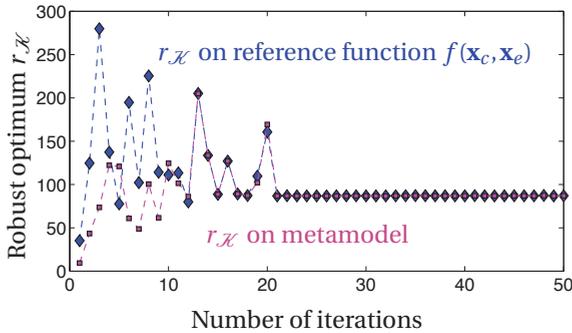


Figure 5.9: The robust optimum found on the metamodel at each iteration of the algorithm for problem P1 is plotted. The metamodel is initially constructed using only 10 initial samples. The plot also shows the objective value for the robust optimum on the reference function.

problems. The error bars indicate the standard deviation around the optimum for each test problem. The standard deviation varies dramatically from one problem to another. This difference is a function of the local gradient in the neighborhood of the robust optimum for the individual problems. Obviously, higher gradients lead to greater relative deviation even when there is a small change in the design variables. In this context, the location of the robust optimum and their relative accuracy is also highlighted. In almost all cases, the numbers compare quite well with the reference optima locations. Where there are larger local deviations in a particular variable, this can be attributed to the fact that the objective could be locally very flat with respect to that variable in the neighborhood of the robust optimum. Additionally, in some cases two different values of a particular variable can lead to the same robust optimum. This is the case for x_{c2} for

problem P2. Therefore, the average value for x_{c2} for problem P2 is completely different from the reference location.

It is also pertinent to point out that the average locations given in Table 5.2 are only meant to show the mean closeness of the result found to the reference location. Since the locations are averages they cannot be used to evaluate the feasibility of the final solution. The last column in Table 5.2 shows the percentage of solutions that were found to be infeasible when evaluated on the respective functions as a post-processing step. The results indicate that the number of optimization results that are feasible on the meta-model but infeasible on the reference function is, in general, very low.

The most crucial numbers in Table 5.2 are given in the sixth column. This column states the total number of evaluations required, on average, to estimate the robust optimum. We note that the total number of function evaluations is quite small for all five problems. Apart from problem P5, all of the problems require less than 4 samples per dimension. The largest problem, P10, in fact requires much less than 2 samples per dimension, ($2^{10} = 1024$ samples), to achieve the reported average performance.

INDIVIDUAL RUN ANALYSIS

Apart from studying the average performance, it is instructive to analyze individual runs for the different test cases. For problem P1, we compare the effect of choosing different number of starting points on the intermediate accuracy of the metamodel at each iteration as well as on the convergence of the algorithm. To this end, Figure 5.9 shows the robust optimum found on the objective metamodel for the problem P1 when the meta-model is initialized with only 10 initial samples. The corresponding robust optimum location's objective value on the reference function $f(\mathbf{x}_c, \mathbf{x}_e)$ at each iteration is also plotted. As expected, in the beginning the robust optimum on the metamodel and the corresponding objective value on $f(\mathbf{x}_c, \mathbf{x}_e)$ do not match. But steadily, the values become closer to each other until by about the 15th iteration they are almost the same. Figure 5.10 shows the same plot for problem P1, but now the number of initial samples is 40. It is interesting to observe that the robust optima found on the metamodel and the ref-

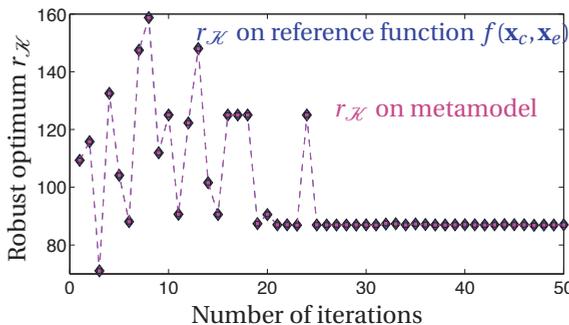


Figure 5.10: The robust optimum found on the metamodel at each iteration of the algorithm for problem P1 is plotted. The metamodel is initially constructed using 40 initial samples. The plot also shows the objective value for the robust optimum on the reference function.

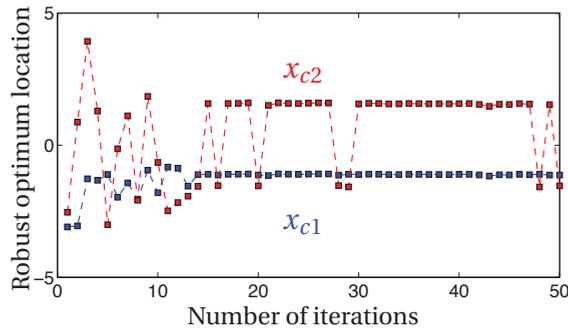


Figure 5.11: The robust optimum location for x_{c1} and x_{c2} found on the metamodel at each iteration of the algorithm for problem P2 is shown. The robust optimum has been reached by the 15th iteration even though the location of x_{c2} changes in the following iterations as well.

5

erence function are already indistinguishable from the first iteration. But this does not automatically guarantee that the algorithm will converge to the robust optimum faster than for the 10 initial samples, Figure 5.9. In fact, in this particular case, the algorithm converges at the same speed for both runs. This suggests that having a larger number of samples for the initial space filling step does not always lead to faster convergence.

Comparison of problem P2 and problem P3 is also enlightening since the two problems have the same objective function and the same first constraint. Problem P3 has a second linear constraint that is not present in P2. It was mentioned in the discussion of the average results in Table 5.2 that x_{c2} in P2 can attain two values. This is exhibited by Figure 5.11 which shows that while x_{c1} has attained a constant value by 15th iteration, x_{c2} sometimes jumps up and down. The algorithm has also converged to the robust optimum by the 15th iteration. It is easy to observe why this happens by turning our attention to the function $f(\mathbf{x}_c, \mathbf{x}_e)$ in problem P2 in the Appendix. It can be seen that x_{c2}

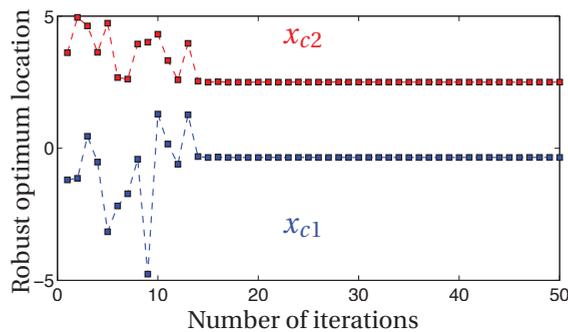


Figure 5.12: The robust optimum location for x_{c1} and x_{c2} found on the metamodel at each iteration of the algorithm for problem P3 is plotted. The robust optimum has been reached by the 15th iteration.

appears only once in the objective function and it makes an appearance as a quadratic term. The domain of x_{c2} in \mathbb{X}_c is $[-5, 5]$ as shown by Table 5.1. The quadratic term and the symmetric domain suggests that as long as the constraint function does not hinder x_{c2} from taking both positive and negative versions of its optimum location, both locations will be equally optimal. Therefore, x_{c2} is able to attain a value of 1.58 or -1.58 without affecting the robust optimum objective value.

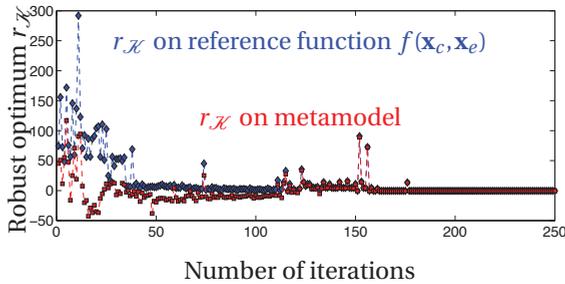


Figure 5.13: The robust optimum found on the metamodel at each iteration of the algorithm is plotted for problem P4. The metamodel is initially constructed using 100 initial samples. The plot also shows the objective value for the robust optimum on the reference function.

The situation changes, however, when the second constraint is taken into account in problem P3. Figure 5.12 shows the optimum robust optimum location for x_{c1} and x_{c2} . The algorithm has converged to the robust optimum by the 15th iteration for this run. With the presence of the second constraint, x_{c2} is not allowed to have a value below 2.5 which means that we no longer observe the phenomenon exhibited in Figure 5.11.

To investigate how the metamodel accuracy is affected by the number of dimensions in a problem, we check how the algorithm fares on the 10 dimensional problem. Figure 5.13 shows the robust optimum found on the objective metamodel for the problem P4 when the metamodel is initialized with 100 initial samples. Again, the corresponding robust optimum location's objective value on the reference function is also plotted. From iteration 1 to iteration 50, the objective values on the metamodel and the reference function are often completely different. Steadily, from iteration 51 to iteration 100, the metamodel starts giving a more accurate picture of the reference function. By the 150th iteration, the two plots are practically indistinguishable. It is clear that by the 200th iteration, the algorithm has converged to the robust optimum, and the same result is given by the reference function, as exhibited by the overlap of the plots.

5.5.3. ENGINEERING CASE STUDY: ROBUST OPTIMIZATION OF AN OPTICAL FILTER

The algorithm is applied on an engineering problem that is very sensitive to uncertainties. The problem involves an optical filter that is fabricated as part of an optical integrated circuit. A schematic of the integrated device is shown in Fig.5.14. Light is input at the in-port, Fig. 5.14. Light is guided, via the principle of total internal reflection, in a relatively higher refractive index layer of SiN, black path in Fig. 5.14, that is embedded

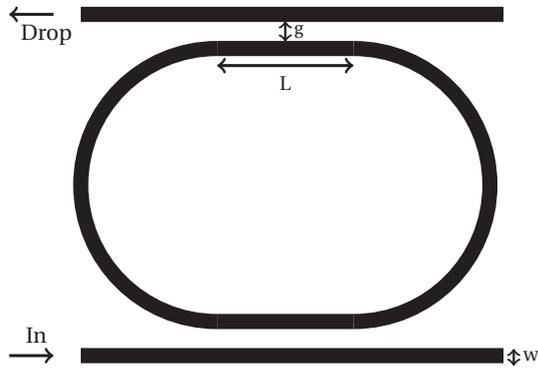


Figure 5.14: Top-view schematic of an optical ring resonator. The area occupied by this integrated photonic device is less than 1 mm^2 . Reproduced with permission from [18].

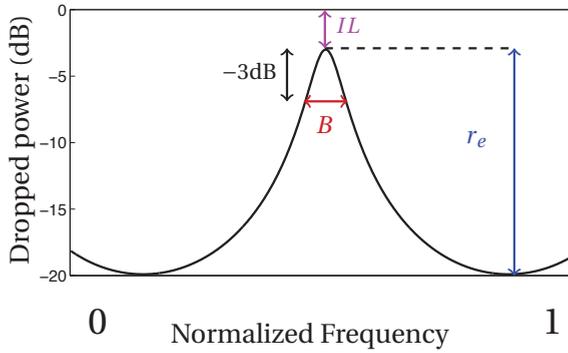


Figure 5.15: Spectral response at the drop port of a ring resonator. The Bandwidth (B), Insertion Loss (IL) and Extinction Ratio (r_e) of the ring resonator are also indicated on the plot.

Table 5.3: Comparison of the robust optimum found by the proposed approach with the deterministic optimum for the filter bandwidth. Higher values for B indicated better performance.

Optimum	w	g	L	Δw	Δt	Nominal B	Worst-case B	Expensive simulations
Nominal	1.11	1.259	220.48	-0.1	-0.003	6.16 GHz	Infeasible	225
Robust	1.17	1.088	108.70	0	-0.003	3.42 GHz	2.19 GHz	225

in a lower refractive index medium SiO_2 . The path through which the light is guided is known as a waveguide. The device is affected by fabrication defects that directly impact the geometry of the cross section of the waveguide. These fabrication defects, in turn, affect the optical performance of the device.

The filter is realized by placing a ring shaped waveguide in between two straight waveguides. When the waveguides are within a certain proximity, light at particular wavelengths is coupled from the straight waveguide into the ring-shaped waveguide and from thereon it couples again into the other straight waveguide. Details related to the physics and operation of ring resonators may be found in the work by [21]. The response, Fig. 5.15 at the drop port shows that the filter resonates at a certain frequency and drops the power at other frequencies. In this work, we are interested in optimizing the bandwidth, B , of the filter at -3dB, Fig. 5.15. The problem involves two constraints related to the Insertion loss IL and the extinction ratio r_e , Fig. 5.15. A commercial software package, [22], is used to simulate the optical filter. The cost of a single simulation is approximately ten minutes.

The quantities of interest are very sensitive to deviations caused by fabrication defects. Therefore, performing robust optimization on the device can lead to significant improvement in the overall yield. The problem is affected by both implementation error type uncertainties and parametric uncertainties. The design variables of the problem are $\mathbf{x}_c \in [w \ g \ L]$ where w is the width of the waveguides while g is the gap between the straight and the ring section, Fig. 5.14. L is the length of the straight coupling section in the ring. An implementation error, $\Delta w \in \mathcal{U}$ affects both the gap g and the width w . The parametric uncertainty Δt is the uncertainty in the out-of-plane thickness of the waveguide. The robust optimization problem is defined as,

$$\begin{aligned}
 & \min_{w,g,L} \max_{\Delta t, \Delta w \in \mathcal{U}} -B, \\
 \text{s.t. } & \max_{\Delta t, \Delta w \in \mathcal{U}} -r_e + 10\text{dB} \leq 0 \\
 & \max_{\Delta t, \Delta w \in \mathcal{U}} IL + 20\text{dB} \leq 0
 \end{aligned} \tag{5.20}$$

where $w \in [0.9, 1.27]\mu\text{m}$, $g \in [0.9, 1.4]\mu\text{m}$ and $L \in [100, 300]\mu\text{m}$. The range of the implementation error uncertainty set \mathcal{U} is $[-0.1, 0.1]\mu\text{m}$ while the parametric uncertainty set is given by $\Delta t \in [-3, 3]\text{nm}$. The nominal waveguide thickness is $t = 32\text{nm}$ and the radius of the ring section is $600\mu\text{m}$.

The proposed method is applied to identify the robust optimum of the filter. The result is compared to the nominal optimum of the problem that is determined by applying the nominal optimization using constrained expected improvement [10]. The optimiza-

tion problem without uncertainties is defined as,

$$\begin{aligned} & \min_{w,g,L} -B, \\ \text{s.t. } & -r_e + 10\text{dB} \leq 0 \\ & IL + 20\text{dB} \leq 0. \end{aligned} \tag{5.21}$$

Both optimization runs are allowed a total of 225 expensive simulations, equivalent to 37.5 hours of simulation time. A large number of total simulations is used since the problem is quite non-linear. Table 5.3 shows a comparison of the robust optimum estimated by the proposed approach with the deterministic optimum found via constrained expected improvement [10]. Column 2 to 4 provide the optimal locations for W , g and L for both algorithms. Column 5 and 6 give the worst-case location with respect to the objective in ΔW and Δt . The next two columns show the nominal and worst-case bandwidth for the two methods. Apart from the worst-case location for the nominal optimum all other solutions were found to be feasible since neither of the two constraints was violated. The feasibility as well as the actual value of the bandwidth B for the nominal worst-case location was found on the simulator as a post processing step. The nominal optimum is found at the boundary of the first constraint, therefore the worst-case bandwidth at nominal location turns out to be infeasible.

It is interesting to note that the worst-case location with respect to the objective for the robust optimum does not occur at the boundary of the set Δw . This shows that the behavior of the function inside the uncertainty set is non-convex. While the nominal performance of the robust optimum is suboptimal compared to the nominal optimum, the robust solution has the advantage that it remains feasible even in the worst case.

5.6. CONCLUSION

In this work, we have presented a novel technique for efficient global robust optimization of expensive simulation based problems involving constraints. The efficiency of the approach derives from the surrogate-based optimization strategy employed. Kriging was chosen as the surrogate due to the availability of an estimate of the error in the interpolator.

We extended the applicability of the Kriging-based constrained optimization framework to the non-deterministic case, where the problem was affected by uncertainties in its parameters. Adapted infill sampling criteria for expected improvement and probability of feasibility were developed that enabled fast convergence to the global robust optimum of constrained problems. The proposed technique provides a viable alternative to fixed Design of Experiments based approaches. The fact that the metamodels are iteratively improved in regions of interest for robust optimization gives the proposed technique a significant advantage over traditional DoE based strategies. The efficiency of the adaptive sampling scheme is particularly important for higher dimensional problems, for which typical space-filling techniques fail to obtain a reasonable solution using a limited number of simulations. In addition to its robustness against the error arising from parametric uncertainties, the method was also made robust against metamodel error using the strategy suggested by [12]. It is pertinent to point out here that the applicability of the method described in this work is not limited to Kriging. Instead, any

interpolation strategy that also provides an error estimate in the interpolation can replace the Kriging framework employed in the proposed technique.

Several benchmark problems were used to analyze the numerical performance of the algorithm. Due to the random nature of the initial sampling, the algorithm was run 100 times on each test problem and the statistical results were investigated. It was shown that the algorithm exhibited efficient reliable convergence to the global robust optimum for all test problems. In addition, the algorithm was also applied on an engineering problem where the bandwidth of an optical filter was optimized. It was shown that while the nominal optimum of the problem gives an infeasible solution in the worst-case, the robust optimum always remains feasible even when considering the uncertainties.

While the method was applied on cheap models developed via Kriging, the approach can equally easily be applied using any other interpolation method that provides an error estimate in the interpolation. The proposed technique is therefore widely applicable and presents a novel opportunity to efficiently investigate robust optimization of different expensive computer simulation based problems affected by uncertainties.

Appendix A: Kriging

The basis function in Kriging is given by the following Gaussian correlation function,

$$\text{Corr} [Y(\mathbf{x}_i), Y(\mathbf{x}_j)] = \exp \left(- \sum_{q=1}^k \theta_q |x_{iq} - x_{jq}|^p \right) \quad (22)$$

where \mathbf{x}_i and \mathbf{x}_j are any two locations in the domain and k represents the total number of dimensions of the problem. The parameter p is assigned a constant value of 2. θ_q , μ and σ^2 are varied such that the likelihood of the observed data is maximized. The Maximum Likelihood Estimate (MLE) for the prediction \hat{y} is given by

$$\hat{y}(\mathbf{x}) = \hat{\mu} + \mathbf{r}^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1} \hat{\mu}) \quad (23)$$

where $\hat{\mu}$ is the estimated value for the mean, \mathbf{R} is the $N \times N$ correlation matrix between the N sample points, \mathbf{r} is the vector of correlations between the observed data and the new prediction, while \mathbf{y} is the observed response. The correlation vector \mathbf{r} and the correlation matrix \mathbf{R} may be found via equation (22).

The Mean Squared Error (MSE) in the Kriging prediction is given by

$$s^2(\mathbf{x}) = \hat{\sigma}^2 \left[1 - \mathbf{r}^T \mathbf{R}^{-1} \mathbf{r} + \frac{1 - \mathbf{1}^T \mathbf{R}^{-1} \mathbf{r}}{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}} \right], \quad (24)$$

The MSE $s^2(\mathbf{x})$ is zero at sample point locations because the true function value is known at these locations.

Appendix B: Test problems

The following test problems have been used to test the ability of the algorithm to converge to the global robust optimum. The objective functions, **P1** to **P4**, were originally used as example problems for unconstrained min-max optimization in [20].

P1

$$\begin{aligned}
f(\mathbf{x}_c, \mathbf{x}_e) &= 5(x_{c1}^2 + x_{c2}^2) - (x_{e1}^2 + x_{e2}^2) \\
&\quad + x_{c1}(-x_{e1} + x_{e2} + 5) + x_{c2}(x_{e1} - x_{e2} + 3). \\
h(\mathbf{x}_c, \mathbf{x}_e) &= -x_{c1}^2 + 5x_{c2} - x_{e1} + x_{e2}^2 - 1.
\end{aligned} \tag{25}$$

P2

$$\begin{aligned}
f(\mathbf{x}_c, \mathbf{x}_e) &= 4(x_{c1} - 2)^2 - 2x_{e1}^2 + x_{c1}^2 x_{e1} - x_{e2}^2 + 2x_{c2}^2 x_{e2}. \\
h(\mathbf{x}_c, \mathbf{x}_e) &= 5x_{c1} - x_{c2}^2 + x_{e1} + x_{e2} - 2.
\end{aligned} \tag{26}$$

P3

$$\begin{aligned}
f(\mathbf{x}_c, \mathbf{x}_e) &= 4(x_{c1} - 2)^2 - 2x_{e1}^2 + x_{c1}^2 x_{e1} - x_{e2}^2 + 2x_{c2}^2 x_{e2}. \\
h_1(\mathbf{x}_c, \mathbf{x}_e) &= 5x_{c1} - x_{c2}^2 + x_{e1} + x_{e2} - 2. \\
h_2(\mathbf{x}_c, \mathbf{x}_e) &= -2x_{c2} + x_{e1}.
\end{aligned} \tag{27}$$

P4

$$\begin{aligned}
f(\mathbf{x}_c, \mathbf{x}_e) &= 2x_{c1}x_{c5} + 3x_{c4}x_{c2} + x_{c5}x_{c3} + 5x_{c4}^2 \\
&\quad + 5x_{c5}^2 - x_{c4}(x_{e4} - x_{e5} - 5) \\
&\quad + x_{c5}(x_{e4} - x_{e5} + 3) + \sum_{i=1}^3 x_{ei}(x_{ci}^2 - 1) - \sum_{i=1}^5 (x_{ei}^2). \\
h(\mathbf{x}_c, \mathbf{x}_e) &= 5x_{c1} - x_{c2} + x_{c3} + x_{c4} \\
&\quad - x_{c5} + x_{e1} - x_{e2} + x_{e3} + x_{e4} - x_{e5}.
\end{aligned} \tag{28}$$

P5

$$\begin{aligned}
f(\mathbf{x}_c) &= 2x_{c1}^6 - 12.2x_{c1}^5 + 21.2x_{c1}^4 + 6.2x_{c1} - 6.4x_{c1}^3 \\
&\quad - 4.7x_{c1}^2 + x_{c2}^6 - 11x_{c2}^5 + 43.3x_{c2}^4 - 10x_{c2} - 74.8x_{c2}^3 + 56.9x_{c2}^2 \\
&\quad - 4.1x_{c1}x_{c2} - 0.1x_{c2}^2x_{c1}^2 + 0.4x_{c2}^2x_{c1} + 0.4x_{c1}^2x_{c2}. \\
h_1(\mathbf{x}_c) &= (x_{c1} - 1.5)^4 + (x_{c2} - 1.5)^4 - 10.125. \\
h_2(\mathbf{x}_c) &= -(2.5 - x_{c1})^3 - (x_{c2} + 1.5)^3 + 15.75.
\end{aligned} \tag{29}$$

Problem P5 is affected by implementation error $\Delta = [\Delta x_{c1} \ \Delta x_{c2}] \in \mathcal{U}$ such that $\|\Delta\|_2 \leq 0.5$.

REFERENCES

- [1] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming* (Springer Verlag, New York, NY, USA, 1997).

- [2] S. P. Gurav, J. F. L. Goosen, and F. van Keulen, *Bounded-But-Unknown uncertainty optimization using design sensitivities and parallel computing: Application to MEMS*, *Computers & Structures* **83**, 1134 (2005).
- [3] I. Elishakoff and M. Ohsaki, *Optimization and anti-optimization of structures under uncertainty* (Imperial College Press, London, 2010).
- [4] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust optimization* (Princeton University Press, 2009).
- [5] D. Bertsimas, O. Nohadani, and K. M. Teo, *Nonconvex Robust Optimization for Problems with Constraints*, *INFORMS Journal on Computing* **22**, 44 (2010).
- [6] J. Sacks, W. Welch, T. J. Mitchell, and H. P. Wynn, *Design and Analysis of Computer Experiments*, *Statistical Science* **4**, 409 (1989).
- [7] M. Morris and T. J. Mitchell, *Exploratory designs for computational experiments*, *Journal of Statistical Planning and Inference* **43**, 381 (1995).
- [8] D. R. Jones, M. Schonlau, and W. J. Welch, *Efficient Global Optimization of Expensive Black-Box Functions*, *Journal of Global Optimization* **13**, 455 (1998).
- [9] M. Schonlau, *Computer experiments and global optimization*, Ph.D. thesis, University of Waterloo (1997).
- [10] J. Parr, A. Forrester, A. J. Keane, and C. M. Holden, *Enhancing infill sampling criteria for surrogate-based constrained optimization*, *Journal of Computational Methods in Sciences and Engineering* **12**, 25 (2012).
- [11] J. M. Parr, A. J. Keane, A. I. Forrester, and C. M. Holden, *Infill sampling criteria for surrogate-based optimization with constraint handling*, *Engineering Optimization* **44**, 1147 (2012).
- [12] E. Stinstra and D. den Hertog, *Robust optimization using computer experiments*, *European Journal of Operational Research* **191**, 816 (2008).
- [13] D. R. Jones, *A Taxonomy of Global Optimization Methods Based on Response Surfaces*, *Journal of Global Optimization* **21**, 345 (2001).
- [14] D. den Hertog, J. P. C. Kleijnen, and A. Y. D. Siem, *The Correct Kriging Variance Estimated by Bootstrapping*, *The Journal of the Operational Research Society* **57**, pp. 400 (2006).
- [15] J. Marzat, E. Walter, and H. Piet-Lahanier, *Worst-case global optimization of black-box functions through Kriging and relaxation*, *Journal of Global Optimization* , 1 (2012).
- [16] J. Marzat, E. Walter, and H. Piet-Lahanier, *A new strategy for worst-case design from costly numerical simulations*, in *American Control Conference (ACC), 2013* (2013) pp. 3991–3996.

- [17] S. ur Rehman, M. Langelaar, and F. van Keulen, *Efficient Kriging-based robust optimization of unconstrained problems*, *Journal of Computational Science* **5**, 872 (2014).
- [18] S. ur Rehman and M. Langelaar, *Efficient global robust optimization of unconstrained problems affected by parametric uncertainties*, *Structural and Multidisciplinary Optimization*, **1** (2015).
- [19] M. Schonlau and W. Welch, *Global optimization with nonparametric function fitting*, in *Proceedings of the Section on Physical and Engineering Sciences* (American Statistical Association, 1996) pp. 183–186.
- [20] B. Rustem and M. Howe, *Algorithms for Worst-Case Design and Applications to Risk Management* (Princeton University Press, 2002) p. 408.
- [21] W. Bogaerts, P. De Heyn, T. Van Vaerenbergh, K. De Vos, S. Kumar Selvaraja, T. Claes, P. Dumon, P. Bienstman, D. Van Thourhout, and R. Baets, *Silicon microring resonators*, *Laser & Photonics Reviews* **6**, 47 (2012).
- [22] PhoeniX Software, *PhoeniX Software 4.8.4*, (2014).

II

COMPUTATIONAL METHODS - SYSTEM LEVEL DETERMINISTIC AND ROBUST OPTIMIZATION

6

DETERMINISTIC OPTIMIZATION OF SYSTEMS WITH INDEPENDENT COMPONENTS

6.1. INTRODUCTION

Optimization of systems has been dealt with in different ways depending on the problem type. The field of multidisciplinary design optimization has evolved at a fast rate in the past decades with the development of various methods and techniques to address optimization of systems with many components or disciplines. In multidisciplinary design optimization the primary focus has been on optimization of systems involving multiple interactions between disciplines which are coupled with one another via coupling variables. Various techniques have been developed that aid in maintaining consistency between these disciplines during the optimization process [1].

A majority of systems falls under the category of a problem where multiple components have complex interactions with one another. However, a subset of engineering and non-engineering systems consists of components that are not interdependent. Such systems may contain several components but the response of these components can be evaluated independently of one another. In other words, the bi-level problem has a hierarchical structure where, at the lower level, each component in the system is only a function of design variables and there is no exchange of coupling variables between components. In this scenario there is no requirement for maintaining consistency between the different disciplines or components and the system optimization process is therefore much simpler.

In addition, components could be expensive to evaluate but the system transformation may be cheap. Systems of this type are common in the field of integrated optics [2]. The response of integrated optical components such as power splitters, directional couplers and phase shifters can be obtained independently of other components. Finding the component response usually requires computationally expensive electromagnetic

simulations. Multiple such components can be used to develop complex integrated optical systems. An example of such a system is an integrated optical serial ring resonator [3].

This work addresses optimization of hierarchical systems based on independent components that are expensive to evaluate. It is assumed that, once the component responses are available, the system response is cheap to compute. In order to optimize the system we construct response surfaces of the underlying components. The accuracy of the response surfaces depends on the number of samples and their locations. The challenge is to perform the optimization with a desired accuracy at very low computational cost. The response surfaces are built using Kriging [4]. Kriging provides an estimator of the mean squared error in its interpolation. This mean squared error was used by Jones *et al.* [5] to develop the Expected Improvement (EI) criterion. The EI criterion enables adaptive sampling of unconstrained problems in order to estimate the optimum using a few expensive simulations.

We develop a system level expected improvement criterion that is derived from a *system level* Mean Squared Error (MSE). The system MSE is found by performing a linear transformation of the component level mean squared error generated by each component metamodel. The system level EI suggests a potential location to be adaptively sampled at each iteration of the algorithm and the component metamodels are updated at the corresponding location. The iterative update of the component metamodels results in a higher fidelity system response with each passing iteration. The process enables the optimum of the system to be estimated using only a few component simulation calls.

Metamodels have often been used in multidisciplinary optimization [6]. Improving systems by adaptively sampling component metamodels is also a fairly well known concept [7, 8]. Optimization of multilevel decomposed systems that have a hierarchical structure has also received attention [9]. Similarly, [10] employed different strategies for component metamodel construction and update for system level optimization. Research has also been previously performed on partitioning expensive problems with a large number of variables so that they can be treated using component metamodels in a multilevel system framework [11]. On the other hand, [12] specifically studied the quantification of error at system level due to component level metamodels in a hierarchical system.

This work also aims to optimize a multi-level problem with a hierarchical structure. The bi-level problem consists of a cheap system transformation at the upper level and expensive component models at the lower level. To our knowledge, there has been no previous work on a system level *expected improvement* based sampling strategy for optimization of systems involving expensive to simulate components that are independent of each other. The proposed approach can efficiently optimize such systems, with the caveat that the number of component variables is not so large such that high fidelity component metamodels cannot be built using a reasonable number of simulations.

The method is especially relevant for systems in which the component behavior is easier to approximate than system behavior. Also, it may be advantageous to employ the technique in situations where many similar components are present in the system since a single metamodel can then often replace multiple components. Furthermore, the approach is attractive for fields in which the systems are typically composed of a

selection of a small set of components.

The chapter is organized as follows. The system level optimization problem of systems with independent components is described in Section 6.2. In Section 6.3, we introduce Kriging and Efficient Global Optimization and then present the system level expected improvement approach for adaptive sampling of the system. The algorithm is tested on several well known numerical test problems and the result is analyzed in Section 6.4. Finally, Section 6.5 contains the conclusions and suggestions for future work.

6.2. OPTIMIZATION OF SYSTEMS WITH INDEPENDENT COMPONENTS

We introduce unconstrained optimization of the system in this section, but the method also applies to constrained problems. Let S represent the response of a system of a set of \mathcal{N} uncoupled components \mathbf{c} . Each component c_i , where $\{i | i \in \mathbb{N}, i \leq \mathcal{N}\}$, is a function of design variables $\mathbf{x}_i \in \mathbb{X}_i \subseteq \mathbb{X}$. Since design variables could potentially be shared across components, the sets \mathbb{X}_1 to $\mathbb{X}_{\mathcal{N}}$ may or may not be disjoint. In addition, some design variables $\mathbf{x}_s \in \mathbb{X}_s$ may only be present at system level, or may be present at both component and system level. The union of sets $\mathbb{X}_i \forall i \in \mathcal{N}$ and \mathbb{X}_s is the set \mathbb{X} . We define design variables that are only present in a single component as local variables. Design variables that are present in multiple component or at both component and system level are referred to as global variables. The optimization of the system response can be written as,

$$\min_{\mathbf{x} \in \mathbb{X}} S(c_1(\mathbf{x}_1), c_2(\mathbf{x}_2), c_3(\mathbf{x}_3), \dots, c_{\mathcal{N}}(\mathbf{x}_{\mathcal{N}}), \mathbf{x}_s). \quad (6.1)$$

If the system response is based on expensive to simulate components, then applying optimization directly on the system response is prohibitively costly. A more viable option instead is to construct lower dimensional (i.e. cheaper) metamodels \mathcal{K}_{c_i} of the components and to apply optimization on the resulting system,

$$\min_{\mathbf{x} \in \mathbb{X}} S_{\mathcal{K}}(\mathcal{K}_{c_1}(\mathbf{x}_1), \mathcal{K}_{c_2}(\mathbf{x}_2), \dots, \mathcal{K}_{c_{\mathcal{N}}}(\mathbf{x}_{\mathcal{N}}), \mathbf{x}_s). \quad (6.2)$$

The process is visualized in Figure 6.1. In this work, we construct the metamodels using Kriging which provides an estimator of the error in its interpolation. We then adaptively sample the component metamodels in such a way that the system optimum is found using only a limited number of expensive simulations of the components.

The fidelity of the system response that is based on component metamodels will heavily depend on the error of the underlying component response surfaces. Furthermore, the relative amount of system error is also governed by the operation that is performed on the component responses in order to arrive at the system response. This operation could turn a small component error into a large error contribution on the system level and vice versa. Therefore, it is important that any adaptive sampling of the components takes into account the error generated at system level. In the following section we show how such an adaptive sampling scheme can be developed.

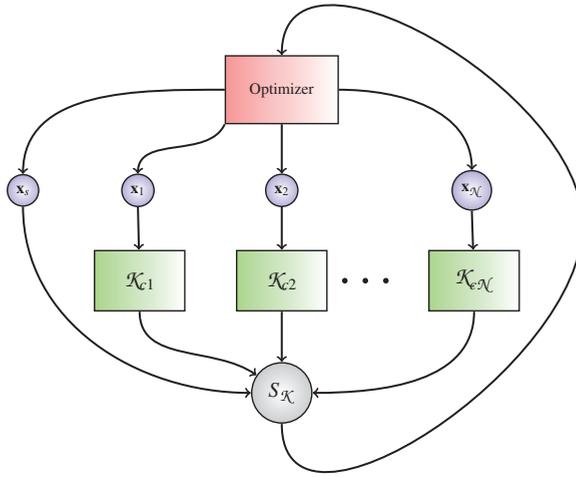


Figure 6.1: Process of optimization of the system based on metamodels of the independent components is shown.

6.3. INFILL SAMPLING CRITERION FOR SYSTEM LEVEL OPTIMIZATION

6.3.1. KRIGING AND EFFICIENT GLOBAL OPTIMIZATION

KRIGING

Kriging is a metamodeling method that constructs the most likely interpolation through a set of responses based on certain statistical assumptions. The most important of these assumptions is that the function response is a normally distributed random variable. However, Kriging is also a popular method for constructing response surfaces of fully deterministic simulation results [13]. Before introducing our new method in Section 6.3.2, we provide a condensed explanation of the main steps in Kriging metamodel construction. Some equations are included in the Appendix, for reference. Sacks *et al.* [4] provide a full description of Kriging.

Kriging uses a tunable basis function which usually takes the form of a parameterized Gaussian correlation function, Equation (30), to measure the correlation between sample points. Maximum likelihood estimation (MLE) is used to estimate the parameters that best explain the observed responses. Kriging prediction \hat{y} , Equation (31), at a previously unsampled location is then also performed using maximum likelihood estimation.

An important aspect of Kriging is that a potential error in the interpolation can be estimated. This Mean Squared Error (MSE) in the prediction is given by Equation (32). The error is basically inversely proportional to the curvature of the likelihood function

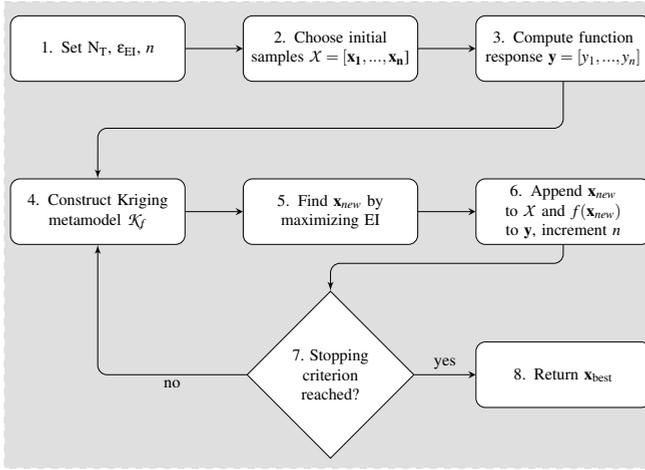


Figure 6.2: Flowchart shows the steps involved in Efficient Global Optimization [5].

used for the Kriging prediction.

EFFICIENT GLOBAL OPTIMIZATION

The combination of the Kriging prediction $\hat{y}(\mathbf{x})$ and MSE $s^2(\mathbf{x})$ can be used to adaptively sample the design domain in order to quickly estimate the global minimum [5]. One of the most effective Kriging based adaptive sampling criteria is that of Expected Improvement (EI). EI is formulated by assuming that the metamodel uncertainty in response $\hat{y}(\mathbf{x})$ at \mathbf{x} may be expressed as a normally distributed random variable $Y(\mathbf{x})$ with mean $\hat{y}(\mathbf{x})$ and variance $s^2(\mathbf{x})$. An improvement can be made over the current observed minimum y_{\min} at \mathbf{x} if some part of $Y(\mathbf{x})$ is below y_{\min} . The Probability of Improvement (PI) over y_{\min} may be expressed as,

$$P[I(\mathbf{x})] = \Phi\left(\frac{y_{\min} - \hat{y}}{s}\right). \quad (6.3)$$

A new sampling location that gives the highest probability of improvement over y_{\min} can be found by maximizing Equation (6.3).

A more refined method of sampling involves finding the expectation of the amount of improvement over y_{\min} . The expected improvement can be calculated by taking the expectation $E[I(\mathbf{x})] = E[\max(y_{\min} - Y, 0)]$. A cheap analytical expression for EI can be found in terms of the normal cumulative distribution function $\Phi(\cdot)$ and the normal probability density function $\phi(\cdot)$ [14]. The expected improvement over the minimum observed response at any location \mathbf{x} is

$$E[I(\mathbf{x})] = (y_{\min} - \hat{y})\Phi\left(\frac{y_{\min} - \hat{y}}{s}\right) + s\phi\left(\frac{y_{\min} - \hat{y}}{s}\right). \quad (6.4)$$

A new sampling location that gives the maximum *expected* improvement can be found by estimating the maximizer of Equation (6.4).

The adaptive sampling strategy can be explained using the flowchart in Figure 6.2. An initial metamodel \mathcal{K}_f is constructed using n samples chosen via Design of Experiments, e.g. Latin Hypercube sampling (LHS). Equation (6.4) is then maximized to estimate the new adaptive sampling location \mathbf{x}_{new} . The sampling location and response is added to the set and the metamodel is constructed again with the augmented set of samples and responses until either EI_{max} , the maximum EI at the last iteration, becomes less than ϵ_{EI} or the computational budget N_T is exhausted. The algorithm is referred to as Efficient Global Optimization (EGO) [5]. This EGO algorithm forms the main inspiration for the present work.

OPTIMIZATION OF CONSTRAINED PROBLEMS

An infill sampling strategy based on Kriging for computationally expensive constrained problems was first suggested by [15]. The method involved building metamodels of the objective and constraints and iteratively improving both sets of metamodels via a single adaptive sampling scheme. This infill sampling criterion was simply a product of probability of improvement in the objective and a probability of *feasibility* for the constraint.

The probability of feasibility measure operates in much the same way as the probability of improvement for the objective. Metamodels are built for the expensive to evaluate constraint(s). Let the Kriging prediction and MSE for such a constraint metamodel be given by $\hat{h}(\mathbf{x})$ and $s^2(\mathbf{x})$. The metamodel uncertainty for each constraint metamodel is again described in terms of a normally distributed random variable $H(\mathbf{x})$ with mean $\hat{h}(\mathbf{x})$ and variance $s^2(\mathbf{x})$.

Let h_{min} be the constraint limit. The standard notation for constrained problems usually assigns h_{min} to be equal to zero. The constrained problem at \mathbf{x} remains feasible as long as the constraint is less than or equal to zero.

The probability of feasibility is given by the area of the distribution $H(\mathbf{x})$ that is below the constraint limit h_{min} ,

$$P[F(\mathbf{x}) < h_{min}] = \Phi\left(\frac{h_{min} - \hat{h}}{s}\right). \quad (6.5)$$

A possible choice for a new *feasible* sampling criterion for a constrained problem can then be the product of the probability of improvement of the objective and the probability of feasibility of constraints. For a problem with a single constraint this may be written as,

$$PI_F = P[I(\mathbf{x})]P[F(\mathbf{x}) < h_{min}] \quad (6.6)$$

Alternatively, the expected improvement in the objective can be multiplied by the product of the probability of feasibility of the constraints,

$$EI_F = E[I(\mathbf{x})]P[F(\mathbf{x}) < h_{min}]. \quad (6.7)$$

6.3.2. SYSTEM LEVEL EI CRITERION

The aim of this work is to optimize Equation (6.2) by means of building and iteratively improving a set of component metamodels such that the optimum of the original system

problem, Equation (6.1), is estimated. An additional goal is to solve system problems involving constraints. Changes required in the iterative sampling strategy to accommodate this goal are discussed in Section 6.3.3.

The algorithm is initialized by building Kriging metamodels of all components based on a set of initial samples. In a software implementation, these could also be pre-built library metamodels. The mean squared error of the individual component metamodels, denoted as $s_i^2(\mathbf{x})$, is given by Equation (32). Let the system level response be given by

$$\hat{y}_{\text{sys}} = S_{\mathcal{K}}(\hat{y}_1(\mathbf{x}_1), \hat{y}_2(\mathbf{x}_2), \dots, \hat{y}_{\mathcal{N}}(\mathbf{x}_{\mathcal{N}}), \mathbf{x}_s), \quad (6.8)$$

where \hat{y}_i is the Kriging prediction of each component \mathcal{K}_{ci} .

In order to update the metamodels in regions of interest for system optimization, an adaptive sampling strategy at system level is required. To derive such a system infill sampling criterion, an error estimate is needed of the system level response, Equation (6.8).

The EI criterion in Section (6.3.1) was found by assuming that the uncertainty in the predicted value $\hat{y}(\mathbf{x})$ at a position \mathbf{x} can be described in terms of a normally distributed random variable $Y(\mathbf{x})$ with mean given by the Kriging prediction $\hat{y}(\mathbf{x})$ and variance given by the Kriging MSE $s^2(\mathbf{x})$. Following the same process, we assume that the uncertainty in the system level response \hat{y}_{sys} at a position \mathbf{x} can be expressed in terms of a random variable $Y_{\text{sys}}(\mathbf{x})$ with mean \hat{y}_{sys} and variance s_{sys}^2 . Furthermore, we retain the assumption that the uncertainty in the predicted value of each component \mathcal{K}_{ci} can be described as a normally distributed random variable $Y_i(\mathbf{x})$ with mean given by the Kriging prediction \hat{y}_i and variance given by the Kriging MSE s_i^2 .

The random variable describing system level uncertainty, $Y_{\text{sys}}(\mathbf{x})$, can basically be described in terms of the system operation on the component level random variables $\mathbf{Y} = [Y_1, Y_2, \dots, Y_{\mathcal{N}}]$ and the deterministic system variable \mathbf{x}_s ,

$$Y_{\text{sys}}(\mathbf{x}) = S_{\mathcal{K}}(Y_1(\mathbf{x}_1), Y_2(\mathbf{x}_2), \dots, Y_{\mathcal{N}}(\mathbf{x}_{\mathcal{N}}), \mathbf{x}_s), \quad (6.9)$$

where the normal random variables $Y_1(\mathbf{x}_1), Y_2(\mathbf{x}_2), \dots, Y_{\mathcal{N}}(\mathbf{x}_{\mathcal{N}})$ are independent. A linear approximation of the right hand side of Equation (6.9) can be derived by performing the Taylor series expansion of $S_{\mathcal{K}}(\mathbf{Y}, \mathbf{x}_s)$ about the mean values \hat{y}_i of \mathbf{Y} and truncating the series to include only the linear terms, i.e. the first two terms,

$$S_{\mathcal{K}}(\mathbf{Y}, \mathbf{x}_s) = S_{\mathcal{K}}(\hat{y}_1, \hat{y}_2, \dots, \hat{y}_{\mathcal{N}}, \mathbf{x}_s) + \sum_{i=1}^{\mathcal{N}} (Y_i - \hat{y}_i) \left. \frac{\partial S_{\mathcal{K}}}{\partial Y_i} \right|_{\hat{y}_i}. \quad (6.10)$$

Since the above expansion is linear, Y_{sys} is a normal random variable [16]. It can be shown [17] that the first order mean of Y_{sys} is given by

$$E(Y_{\text{sys}}) = \hat{y}_{\text{sys}} = S_{\mathcal{K}}[\hat{y}_1, \hat{y}_2, \dots, \hat{y}_{\mathcal{N}}, \mathbf{x}_s] \quad (6.11)$$

and the first order variance of Y_{sys} is given by

$$\text{Var}(Y_{\text{sys}}) = s_{\text{sys}}^2 = \sum_{i=1}^{\mathcal{N}} b_i^2 s_i^2, \quad (6.12)$$

where $b_i = \left. \frac{\partial S_{\mathcal{K}}}{\partial Y_i} \right|_{\hat{y}_i}$.

The derivatives b_i can be computed cheaply using finite difference in case of black-box system functions, or analytically otherwise. This is based on the assumption that the components are expensive to evaluate, but the system is cheap. Since Y_{sys} is normally distributed, an analytical expression similar to EI, Equation (6.4), can be derived for the system level expected improvement as well.

Let the optimum on the system response, Equation (6.2), be denoted by $d_{\mathcal{K}}$. We can improve on the value $d_{\mathcal{K}}$ if $Y_{sys} < d_{\mathcal{K}}$. The expectation of this improvement, $I_{sys} = \max(d_{\mathcal{K}} - Y_{sys}, 0)$, can be found by

$$\underbrace{E[I_{sys}(\mathbf{x})]}_{EI_{sys}} = \int_{I_{sys}=0}^{I_{sys}=\infty} I_{sys} \frac{\exp\left(-\frac{t_{sys}^2}{2}\right)}{\sqrt{2\pi}s_{sys}} dI_{sys}, \quad (6.13)$$

where

$$t_{sys} = \frac{d_{\mathcal{K}} - I_{sys} - \hat{y}_{sys}}{s_{sys}}. \quad (6.14)$$

The standard normal probability density function is given by

$$\phi(z) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right). \quad (6.15)$$

Plugging in Equation (6.15) into Equation (6.13), EI_{sys} can be written as,

$$E[I_{sys}(\mathbf{x})] = (d_{\mathcal{K}} - \hat{y}_{sys}) \int_{t_{sys}=-\infty}^{t_{sys}=\frac{d_{\mathcal{K}}-\hat{y}_{sys}}{s_{sys}}} \phi(t_{sys}) dt_{sys} - s_{sys} \int_{t_{sys}=-\infty}^{t_{sys}=\frac{d_{\mathcal{K}}-\hat{y}_{sys}}{s_{sys}}} t_{sys} \phi(t_{sys}) dt_{sys}. \quad (6.16)$$

The first integral in Equation (6.16) can be recognized as the normal cumulative distribution function $\Phi\left(\frac{d_{\mathcal{K}}-\hat{y}_{sys}}{s_{sys}}\right)$. The second integral can be solved by using the substitution $z = \frac{-t_{sys}^2}{2}$. The final analytical expression for the infill sampling criterion at system level, EI_{sys} , is

$$E[I_{sys}(\mathbf{x})] = (d_{\mathcal{K}} - \hat{y}_{sys}) \Phi\left(\frac{d_{\mathcal{K}} - \hat{y}_{sys}}{s_{sys}}\right) + s_{sys} \phi\left(\frac{d_{\mathcal{K}} - \hat{y}_{sys}}{s_{sys}}\right). \quad (6.17)$$

The location of the global maximum of Equation (6.17) gives the next infill sampling location \mathbf{x}_{new} .

The algorithm is referred to as Bilevel Efficient Global Optimization (BEGO). We illustrate the main idea of the algorithm via the following illustrative example,

$$\begin{aligned} S(\mathbf{c}(x)) &= \sin(c_1(x)) + \cos(c_2(x)), \\ \text{where } c_1(x) &= 10x^4, \quad c_2(x) = 10(1-x)^3, \quad x \in [0, 1]. \end{aligned} \quad (6.18)$$

The system response is a function of two components that in turn are a function of the same variable x . Figure 6.3 shows the plot of $S(\mathbf{c}(x))$, $c_1(x)$ and $c_2(x)$. The system

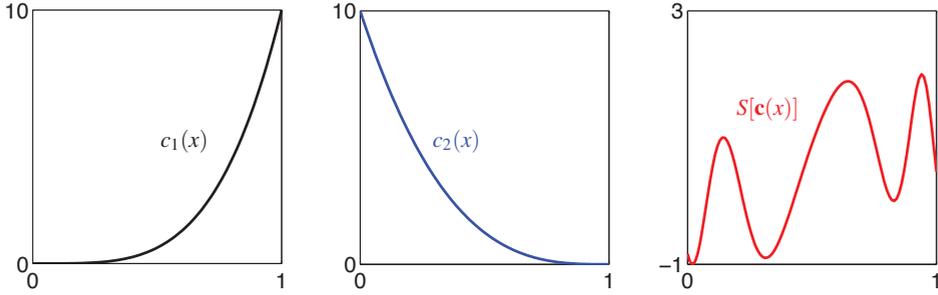


Figure 6.3: The components $c_1(x)$ and $c_2(x)$ and the system response $S(\mathbf{c}(x))$ are plotted in the domain $x \in [0, 1]$.

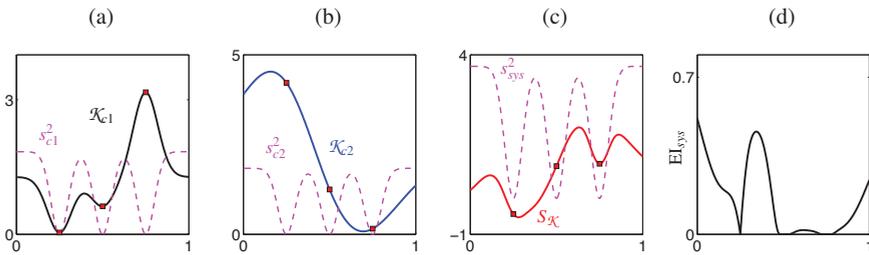


Figure 6.4: The Kriging metamodel of components $c_1(x)$ and $c_2(x)$ and the system response $S_{\mathcal{K}}$ are plotted along with their respective error estimate in (a), (b) and (c). Plot (d) shows the system expected improvement EI_{sys} .

response has three local minima, two of which are very close in terms of their objective value.

We construct Kriging metamodels of the components $c_1(x)$ and $c_2(x)$ based on a set of initial samples. The true component and system responses are treated as black-box. The system response based on metamodels may be expressed as,

$$S_{\mathcal{K}}(\mathcal{K}_{\mathbf{c}}(x)) = \sin(\mathcal{K}_{c_1}(x)) + \cos(\mathcal{K}_{c_2}(x)), \quad x \in [0, 1]. \quad (6.19)$$

Figure 6.4 shows the Kriging metamodels, \mathcal{K}_{c_1} and \mathcal{K}_{c_2} , of the components c_1 and c_2 along with the system response $S_{\mathcal{K}}$ and the system expected improvement EI_{sys} . The metamodels have been initialized with three samples. The mean squared error for \mathcal{K}_{c_1} and \mathcal{K}_{c_2} and the error estimate for $S_{\mathcal{K}}$ are also shown in plots (a), (b) and (c), respectively. The error estimate for $S_{\mathcal{K}}$ is constructed from the individual errors of \mathcal{K}_{c_1} and \mathcal{K}_{c_2} , by employing the linear transformation, Equation 6.12, at each value of x .

As expected, the error estimates are zero at the sample locations in plots (a), (b) and (c). Similarly, the plot of EI_{sys} shows that no improvement is expected at locations that have already been sampled. EI_{sys} is maximum at $x = 0$ and this would be chosen as the new infill sampling location for this iteration of the algorithm.

Figure 6.5 shows a flowchart of the main steps involved in system optimization using Kriging based system expected improvement. It closely resembles the EGO flowchart, in Figure 6.2, but important differences should be noted. After constructing the component metamodels in Step 4, the new infill sampling location \mathbf{x}_{new} is estimated in Step 5. This involves a few substeps. Firstly, the deterministic optimum, $d_{\mathcal{K}}$ on $S_{\mathcal{K}}$ has to be estimated. Thereafter, the linear transformation of the component level mean squared error(s) is performed to estimate the system level error, s_{sys}^2 . The system error estimate s_{sys}^2 is used along with the deterministic optimum $d_{\mathcal{K}}$ to evaluate EI_{sys} in Equation (6.17). The global maximum of Equation (6.17) gives the new sampling location \mathbf{x}_{new} . In Step 6 the component response is evaluated at \mathbf{x}_{new} . If the stopping criterion has been reached already then the argument of $d_{\mathcal{K}}$, the minimizer of $S_{\mathcal{K}}$, is returned as the final system optimum, otherwise the algorithm returns to Step 4 and the loop is repeated until termination.

6.3.3. CONSTRAINED OPTIMIZATION OF SYSTEMS

If the system involves constraints then the sampling criterion must also be adapted to deal with these constraints. Similar to the formulation of system-level EI above, a probability of feasibility measure at system level can be derived to ensure that samples are added in areas of interest for constrained optimization instead of unconstrained optimization.

Let S_h represent the transformation of the component metamodels in order to get the constraint response \hat{h}_{sys} .

$$\hat{h}_{sys} = S_h(\hat{y}_1(\mathbf{x}_1), \hat{y}_2(\mathbf{x}_2), \dots, \hat{y}_{\mathcal{N}}(\mathbf{x}_{\mathcal{N}}), \mathbf{x}_s). \quad (6.20)$$

The error in the constraint response \hat{h}_{sys} can be derived in a manner that is completely analogous to the system error, s_{sys} , found for the objective. The system level variance for the constraint is denoted by s_h^2 . The probability of feasibility, PF_{sys} , for the constraint S_h can be expressed as,

$$P[F(\mathbf{x}) < h_{min}] = \Phi\left(\frac{h_{min} - \hat{h}_{sys}}{s_h}\right). \quad (6.21)$$

where h_{min} is the constraint limit that dictates whether the constraint is feasible or infeasible. The product of the system level expected improvement EI_{sys} and the system level probability of feasibility, PF_{sys} , gives a criterion for sampling a constrained problem. For a single constraint this criterion can simply be expressed as,

$$EI_{sf} = EI_{sys} \cdot PF_{sys}. \quad (6.22)$$

The maximizer of Equation (6.22) is then sampled before constructing the component metamodels again. If there are multiple constraints, the product of individual probability of feasibility of all constraints will replace PF_{sys} in Equation (6.22).

6.4. RESULTS

6.4.1. NUMERICAL PERFORMANCE EVALUATION

The algorithm is tested on several test problems to investigate its ability to estimate the deterministic optimum of the system accurately and consistently. The performance of

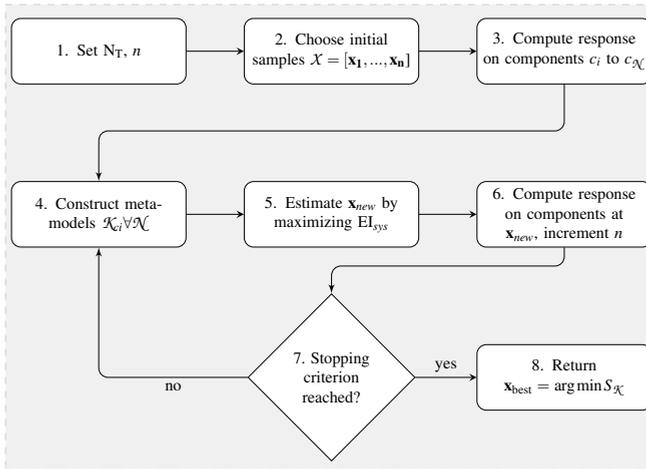


Figure 6.5: Flowchart shows the steps involved in system optimization.

the algorithm is compared against treating the entire system as a component and applying optimization using the EGO method [5]. In the following discussion we refer to the system decomposed into several components with the term DEC, (DEComposed), and the system treated as a single component with the term SAC, where SAC stands for System-As-Component. Similarly, we compare the performance of the approach with a Latin Hypercube based sampling scheme for approximate response construction and optimization of the SAC response. The advantages of using an adaptive sampling scheme for the decomposed system problem are exhibited by comparing the effectiveness of the method with optimizing a system response for a constrained problem based on component metamodels built via LHS.

ALGORITHM EVOLUTION EXAMPLE

Before performing an in-depth analysis, we illustrate the evolution of the algorithm on the one-variable system that consisted of two non-linear components. The problem was introduced in Section 6.3, Equation (6.18). The plot of the reference components and system response was given in Figure 6.3.

Figure 6.6 shows five iterations of the algorithm, after it has been initialized with three samples at 0.25, 0.5 and 0.75 within the domain $x \in [0, 1]$. The component metamodels and the system response along with their respective error estimates are given in plot (a), (b) and (c) for each iteration. Plot (d) shows the system expected improvement found at each iteration. Each subsequent iteration updates the component metamodel with \mathbf{x}_{new} , the location of the maximum value for EI_{sys} attained in the previous iteration. By the 5th iteration the maximum EI_{sys} falls to an insignificantly small value and the global optimum of the problem, $S = -1$, is found at $x = 0.0196$. The error estimates for the components and system response also drop fast. At the 5th iteration the component metamodels and the system response $S_{\mathcal{X}}$ match the reference in Figure 6.3 and the

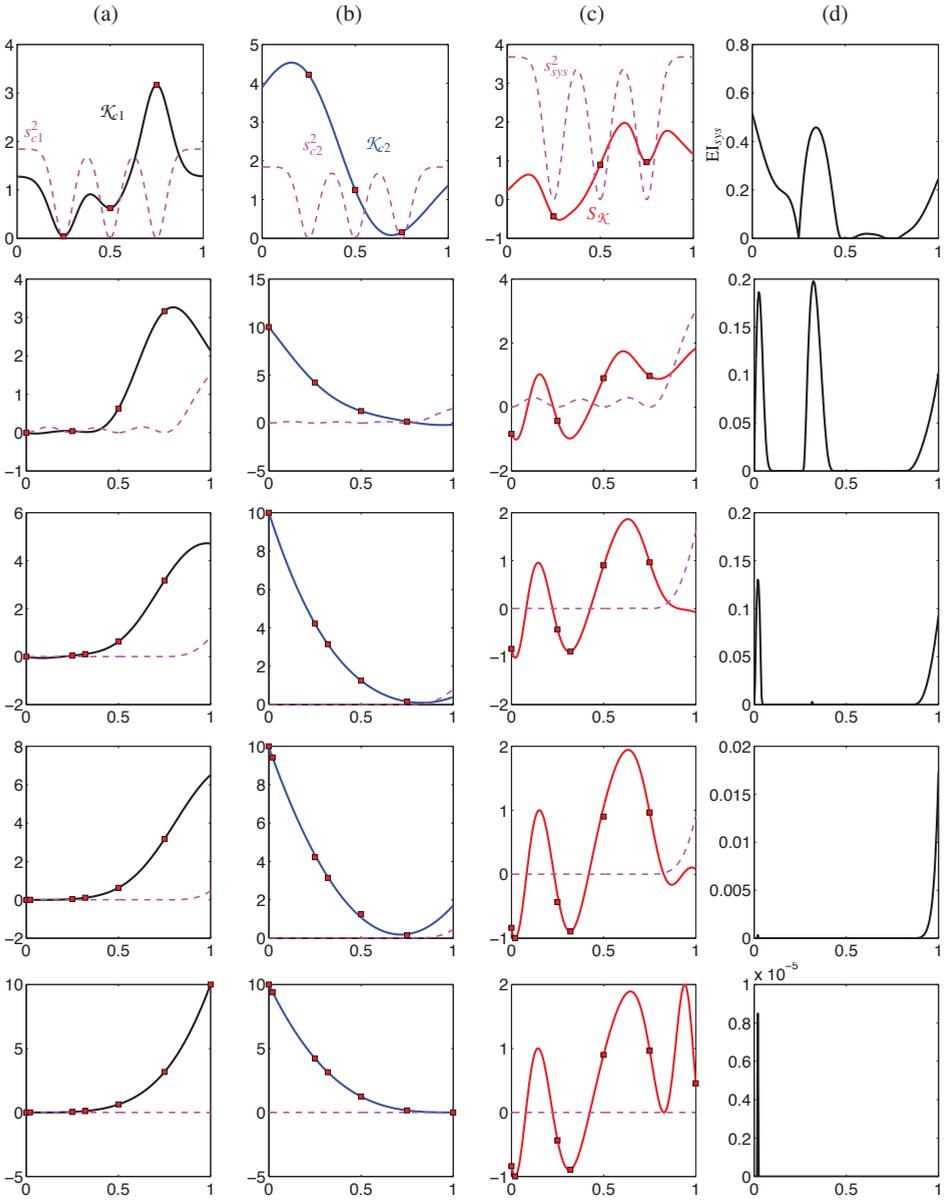


Figure 6.6: The evolution of the method is shown as new sampling points are added at each iteration. The Kriging metamodel of components $c_1(x)$ and $c_2(x)$ and the system response $S_{\mathcal{X}}$ are plotted along with their respective error estimate in (a), (b) and (c). Plot (d) shows the system expected improvement El_{sys} .

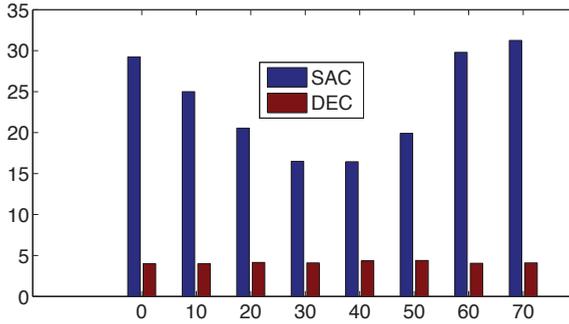


Figure 6.7: Comparison of the average number of simulations, based on 20 runs, needed by DEC and SAC to converge to the optimum of the problem.

respective error estimates are also very low.

The DEC and SAC algorithms were applied 20 times on the illustrative system problem S_1 . Both methods were initialized with two random sampling locations for each run. The average number of iterations needed by DEC to converge to the optimum was 6.6 while the corresponding number of iterations for SAC was 14.5. In part, this disparity is caused by the difference in nonlinearity of the system and component responses, which require varying amounts of samples to approximate appropriately. To further illustrate this aspect, additional test problems are discussed below.

UNCONSTRAINED TEST PROBLEMS

The algorithm is tested on a set of one-dimensional systems based on the following numerical problem,

$$\begin{aligned}
 S_p[\mathbf{c}(x)] &= \sin(c_1(x)) + \cos(c_2(x)), \\
 c_1(x) &= px, \quad c_2(x) = (75 - p)x, \quad x \in [0, 1],
 \end{aligned}
 \tag{6.23}$$

where p changes from $p = 0$ to $p = 70$ in steps of 10. This results in 8 different system problems with different values of the frequency p . Figure 6.7 shows the number of iterations needed to find the optimum for each of these 8 problems by SAC and DEC. Each problem was initialized with two random sampling locations for both SAC and DEC. In order to ensure that the result is not skewed by fortuitously hitting the optimum, both algorithms are only terminated when the optimal solution is repeated over 2 consecutive iterations. The bar chart shows the average performance over 20 runs for both methods. DEC found the optimum for each problem using less than 5 iterations. On the other hand SAC needed more than 15 iterations in all cases. For all values of p , the functions had multiple local optima that were relatively close to each other, so the problem was relatively difficult to optimize even in one variable.

A question that arises from the clear difference between the performance of DEC and SAC in Figure 6.7 is why DEC does so much better than SAC on all problems. The fact is that the metamodels that DEC has to build are very simple linear functions that require only relatively few function calls to fit. On the other hand SAC has to construct

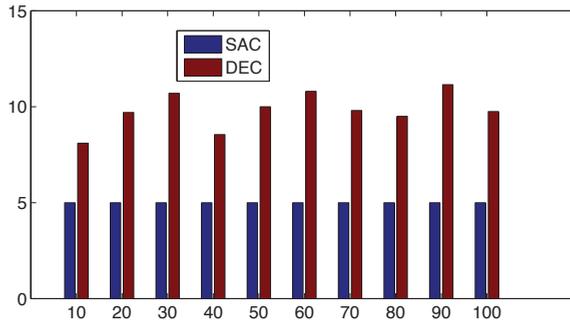


Figure 6.8: The average number of simulations, based on 20 runs, needed by DEC and SAC to converge to the optimum of the problem. The horizontal axis is the value of problem parameter q .

the metamodel of the system response which is highly nonlinear. Therefore SAC requires many more iterations to fit the response well enough to estimate the optimum correctly.

To test this argument, we apply the following problem on SAC and DEC,

$$S_q[\mathbf{c}(x)] = c_1(x) + c_2(x),$$

$$c_1(x) = (5(x - 0.5))^2 + 20 \sin(qx), \quad c_2(x) = (5(x - 0.5))^2 - 20 \sin(qx), \quad x \in [0, 1], \quad (6.24)$$

where the frequency q changes from $q = 10$ to $q = 100$ in steps of 10. This results in ten problems with differing frequency of the components. The system transformation is linear while the component responses are highly nonlinear. It should be noted however that if SAC is used to optimize Equation (6.24) then the response is always linear and always the same, no matter what value is chosen for q . This is due to the fact that the sinusoidal parts of the components cancel each other at system level.

Both SAC and DEC are initialized with two random samples. Again, in order to ensure that the algorithm is not terminated when the algorithm accidentally finds the optimum, both algorithms are stopped when the optimum is repeated over two consecutive iterations. Based on an average of 20 runs SAC requires only 5 total function calls to locate the optimum. Figure 6.8 shows the performance of DEC for different values of q . For all values of q , DEC requires more simulations than SAC to converge to the optimum. But interestingly the difference between the performance of SAC and DEC is not as dramatic in Figure 6.8 as was the case for the problems shown in Figure 6.7.

The algorithm is next tested on the three-hump camel function, which is an established test problem [18] for unconstrained optimization in two dimensions,

$$f_{camel} = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2, \quad x_i \in [-5, 5] \forall i \in \mathcal{N}. \quad (6.25)$$

In the decomposed system level problem, we assume that there are three expensive components and the rest of the function is cheap to compute. The problem is decomposed

Table 6.1: Comparison of adaptive system optimization of decomposed Camel problem, Equation (6.26), versus application of LHS on the original Camel function, Equation (6.25). The mean and standard deviation of the global optimum found by each method is shown based on 100 runs.

Optimum	Mean	Standard deviation	Expensive function evaluations
Reference	0	-	-
Adaptive system sampling	0.00	0.001	12
Latin Hypercube sampling	1.02	0.015	12

and written in terms of the system and components as,

$$\begin{aligned}
 S_{camel}[\mathbf{c}(\mathbf{x})] &= 2c_1(x_1) - 1.05c_1^2(x_1) + \frac{c_1^3(x_1)}{6} + c_2(x_1, x_2) + c_3(x_2), \\
 c_1(x_1) &= x_1^2, \quad c_2(x_1, x_2) = x_1 x_2, \quad c_3(x_2) = x_2^2, \\
 x_i &\in [-5, 5] \forall i \in \mathcal{N}.
 \end{aligned} \tag{6.26}$$

The proposed algorithm is applied 100 times on the system in Equation (6.26). The component models are constructed using 4 initial Latin Hypercube samples (LHS). The number of expensive simulations is limited to 12. Therefore the algorithm can run for 8 iterations.

The mean performance of the 100 runs of the system level approach is compared against applying optimization on a Kriging metamodel of the original function in Equation (6.25) which is constructed using 12 expensive simulations chosen via LHS. The Kriging metamodel is also constructed 100 times based on 100 different combinations of LHS samples and the mean performance is analyzed.

Table 6.1 shows the comparison of the two approaches along with the reference global optimum of the function. The mean and standard deviation of the objective value at the global optimum location found by both methods are shown. The objective values shown have been generated on the original function, Equation (6.25), as a post-processing step. The mean optimum found by adaptive system sampling approach for the decomposed problem is closer to the reference solution than the one found by LHS. In addition, the standard deviation for the system sampling scheme is also much lower compared to the one found by using LHS. This indicates that performing adaptive system sampling of the decomposed problem in Equation (6.26) is more efficient than applying LHS on the original problem in Equation (6.25). However, a comparison of the performance of LHS and the adaptive sampling scheme on a decomposed problem would also be interesting. In the following subsection we show this comparison for the more widely applicable case of a constrained problem.

CONSTRAINED TEST PROBLEM

The algorithm is now applied on a constrained benchmark problem. The objective is a modified version of the Rastrigin function. The constrained problem is given below,

$$f(\mathbf{x}) = 0.01 \left(100 + \sum_{i=1}^{10} [x_i^2 - 10 \cos(0.2\pi x_i)] \right)^2 \quad (6.27)$$

$$\text{s.t. } 35 - \sum_{i=1}^4 x_i \leq 0,$$

$$x_i \in [-5.12, 5.12] \forall i \in \mathcal{N}.$$

The problem consists of 10 variables. We choose to decompose the problem into a system with four expensive components. Based on this decomposition, the system can be written as,

$$S[\mathbf{c}(\mathbf{x})] = 0.01 (100 + c_1 - 10c_2 + c_3 - 10c_4)^2 \quad (6.28)$$

$$\text{s.t. } 35 - c_1 \leq 0,$$

$$c_1 = \sum_{i=1}^4 x_i^2, \quad c_2 = \sum_{i=1}^4 \cos(0.2\pi x_i), \quad c_3 = \sum_{i=5}^{10} x_i^2, \quad c_4 = \sum_{i=5}^{10} \cos(0.2\pi x_i),$$

$$x_i \in [-5.12, 5.12] \forall i \in \mathcal{N}.$$

6

The system non-linearly transforms the response of the four non-linear components. The components c_1 and c_2 are a function of four variables which are shared across both the components. On the other hand, c_3 and c_4 are a function of six different variables, which are also shared across these two components. The component c_1 makes an appearance in both the objective and the constraint. The metamodel of c_1 can therefore be used for both the system transformation, $S_{\mathcal{N}}$, of the objective as well as in the system transformation, S_h , of the constraint.

The decomposed problem in Equation (6.28) is used to compare the performance of BEGO with an LHS based strategy. For this problem, metamodels are made for the component response instead of the system response for the LHS based approach as well. Both methods are run 20 times on the proposed problem, Equation (6.28), and the mean performance is compared.

In the case of BEGO, the component metamodels are initialized with a different number of samples based on the number of dimensions of the component. The four dimensional components c_1 and c_2 are initialized with 20 samples, while the six dimensional components c_3 and c_4 are initialized with 30 samples. The algorithm is allowed to run for 80 iterations. This means that at termination, 100 samples would have been added to \mathcal{K}_{c_1} and \mathcal{K}_{c_2} , while \mathcal{K}_{c_3} and \mathcal{K}_{c_4} would be based on 110 samples.

The LHS based approach is therefore given a total budget of 100 samples for c_1 and c_2 and 110 samples for c_3 and c_4 . The system response $S_{\mathcal{N}}$ based on the LHS based metamodels is optimized and the average result of the 20 runs is analyzed.

We also compare the performance for the case in which both LHS and BEGO are given 25 less expensive simulations. This means that BEGO is terminated after 55 iterations. On the other hand, for LHS \mathcal{K}_{c_1} and \mathcal{K}_{c_2} are constructed based on 75 samples and \mathcal{K}_{c_3} and \mathcal{K}_{c_4} are built using 85 samples.

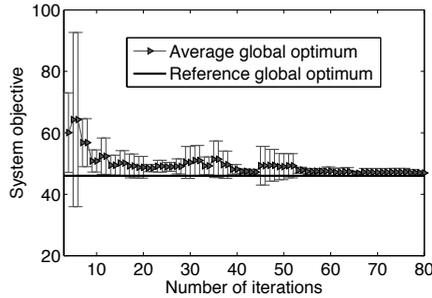


Figure 6.9: The mean and standard deviation of the global optimum found on each iteration of BEGO based on the 20 runs for the test problem defined in Equation (6.28).

Figure 6.9 shows the average global optimum found at each iteration of BEGO, based on 20 runs. The objective is evaluated, as a post-processing step, on the reference system response based on the optimal location found at each iteration of BEGO. The reference global optimum is also found on the reference system. The errorbars indicate the standard deviation around the optimum. After about 60 iterations of BEGO, the algorithm has converged to a value close to the reference global optimum and the standard deviation is also relatively low.

Table 6.2 shows the comparison of the statistics of the global optimum found by LHS and BEGO. Once again the objective values given here have been found by evaluating the optima found by each method on the reference system response as a post-processing step. The results indicate that BEGO performs better than LHS in terms of mean closeness of the optimum to the reference solution, given the same number of component evaluations. The standard deviation of BEGO is also lower than for LHS. Interestingly, for LHS with a higher number of samples, 2 out of the 20 optimal locations found result in a constraint violation. On the other hand, the optimal locations of BEGO do not violate the constraint for any of the 20 runs. Comparing the result of the higher number of evaluations (second set of rows in Table 6.2) with the result for the lower number of evaluations, we note that convergence is steadily taking place for both approaches. The mean and standard deviation for BEGO based on the lower number of evaluations are both better than the corresponding numbers for LHS with a higher number of evaluations.

Using the constrained and unconstrained problems, we have evaluated the performance of BEGO in different settings. The method converges well on the problems and also performs better than a space-filling based strategy. The primary advantage of the adaptive technique lies in reducing the computational budget needed to find the global optimum. Since each extra component simulation significantly increases the time and computational costs of optimization, any amount of reduction in these costs brings significant efficiency improvement. We now test the algorithm in a more practical setting by applying it on an engineering problem.

Table 6.2: Comparison of the mean and standard deviation of the global optimum found by BEGO and LHS based on 20 runs.

Optimum	Reference	BEGO	LHS	Evaluations c_1, c_2	Evaluations c_3, c_4
Mean	46.03	47.43	48.89	75	85
Standard deviation	-	0.97	2.12	75	85
Constraint violations	-	0	0	75	85
Mean	46.03	46.93	47.70	100	110
Standard deviation	-	0.66	1.34	100	110
Constraint violations	-	0	2	100	110

6.4.2. ENGINEERING PROBLEM: DETERMINISTIC OPTIMIZATION OF AN OPTICAL FILTER

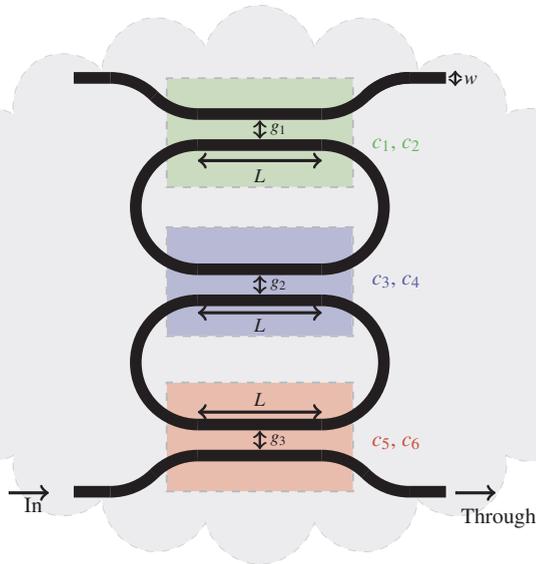


Figure 6.10: Top-view schematic of a second order serial optical ring resonator. The cloud encompasses the complete system. The colored boxes, on the other hand, represent the six expensive to evaluate components.

BACKGROUND OF THE PROBLEM

We perform system optimization of an integrated photonic serial ring resonator that behaves as an optical filter. Light is confined in a high refractive index SiN core layer which is buried inside a relatively lower refractive index SiO₂ cladding layer. The light is guided inside these optical integrated circuits via the principle of total internal reflection. Figure 6.10 shows the top-view schematic of the system. When light at a certain wavelength is launched into the waveguide (black paths lines) at the 'In' port, it partially couples

into the adjacent waveguide. This coupled light travels through the first ring section and partially couples again into the second ring section. A portion of this coupled light is dropped at the ‘Drop’ port. The rest of the light exits via the ‘Through’ port.

The amount of coupling can be varied by changing the gap and the length of the coupling sections (area enclosed by the colored boxes in Figure 6.10). This change in coupling gives rise to a change in the optical response at the ‘Drop’ and ‘Through’ port. Different optical filter responses can therefore be generated by varying the coupling ratio of each coupler. The objective in this study is to achieve the band-pass filter response given in Figure 6.11 at the ‘Through’ port.

In order to compute the power coupling ratio of the coupler, two expensive electro-magnetic simulations have to be performed. The first simulation computes the power coupling ratio, P_{L0} , when the coupling length is zero. The second simulation computes the beat length, L_{π} , which is defined as the coupling length needed to completely couple light from the first waveguide into the second waveguide and back into the first waveguide. Since the exchange of light between adjacent waveguides follows a sinusoidal

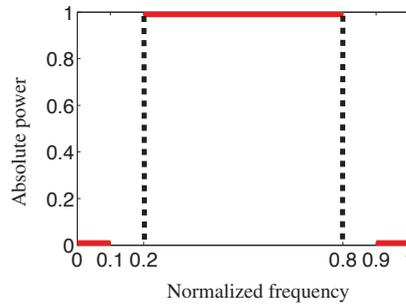


Figure 6.11: Desired band-pass filter spectral response (red line) of the serial ring resonator. No preference is specified in the intervals $[0.1, 0.2]$ and $[0.8, 0.9]$.

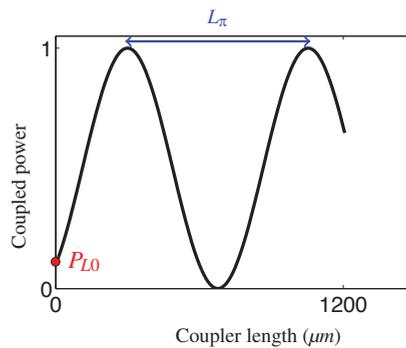


Figure 6.12: Power coupling ratio as a function of the coupler length for a certain coupler gap.

curve, the two parameters, P_{L0} and L_π , are enough to compute the coupled power at any coupling length given a certain gap [19]. Figure 6.12 shows the power coupling ratio as a function of the coupling length based on a certain simulated value of P_{L0} and L_π for a given gap.

Once the power coupling ratio of each coupler is known, the transfer function at the ‘Through’ port can be cheaply computed as a function of the normalized frequency using scattering matrix analysis [3]. Let $H(n_f)$ represent the transfer function for the normalized frequency $n_f \in [0, 1]$. We can then define a system objective for the desired bandpass spectral response as,

$$\min_{\mathbf{x} \in \mathbb{X}} \frac{1-b}{2} \|\bar{H}_{stop1}\|_p + b \left[1 - \|1 - \bar{H}_{pass}\|_p \right] + \frac{1-b}{2} \|\bar{H}_{stop2}\|_p, \quad (6.29)$$

where \bar{H}_{stop1} , \bar{H}_{pass} and \bar{H}_{stop2} represent the vector of responses for $n_f \in [0, 0.1]$, $n_f \in [0.2, 0.8]$ and $n_f \in [0.9, 1]$, respectively. The five design variables of the problem, depicted in Figure 6.10, are $\mathbf{x} \in [w, g_1, g_2, g_3, L]$, where $w \in [1, 1.15]\mu m$, $L \in [0, 2400]\mu m$ and all the gaps are in the range $[1, 1.3]\mu m$. If p is large then the objective basically involves minimization of the weighed sum of the maximum value in the two stop bands and the minimum value in the pass-band. The p -norm is used instead of the maximum and minimum value in order to ensure that the objective remains continuously differentiable. The weights are based on the value of b . In this work, we choose $p = 20$ and $b = 0.6$.

6

SYSTEM OPTIMIZATION

The system response, S , can be modeled in terms of two expensive components that are repeated three times in the second order resonator since each coupler requires two expensive simulations and there are three couplers in the system. Figure 6.10 shows the six components c_1 to c_6 as well as the five design variables of the problem $[w, g_1, g_2, g_3, L]$. We treat the responses P_{L0} and L_π , generated from each coupling section, as the components of the problem since this suits the mathematical structure of this problem. The expensive components are only a function of the width and the respective gaps. The components c_1 , c_3 and c_5 give the power coupling ratio P_{L0} for each coupler while the beat length L_π for each coupler is found by components c_2 , c_4 and c_6 .

A commercial electromagnetic solver, Phoenix Software, is used to compute P_{L0} and L_π . Both simulations require approximately 10 minutes. Initial Kriging metamodels for c_1 and c_2 are built based on $n_d = 9$ samples for w and g_1 chosen via Latin Hypercube sampling. We assign g_2 and g_3 the same initial sample values as g_1 . Since the design domain for all the components is the same we can essentially use \mathcal{K}_{c_1} as the approximate response for P_{L0} for all three components c_1 , c_3 and c_5 . Similarly \mathcal{K}_{c_2} can be used to give the beat length L_π for components c_2 , c_4 and c_6 .

Once the component metamodels that predict the value for P_{L0} and L_π for each coupler have been constructed, the following operations to evaluate the system objective are performed at system level. In order to evaluate the objective to be optimized, Equation (6.29), the power coupling ratio of each coupler is needed. This operation is performed on system level since the value of P_{L0} and L_π can cheaply be predicted by the Kriging metamodels for each coupler. The computation of the transfer function via scattering matrix analysis involves simple operations on small matrices and is therefore also per-

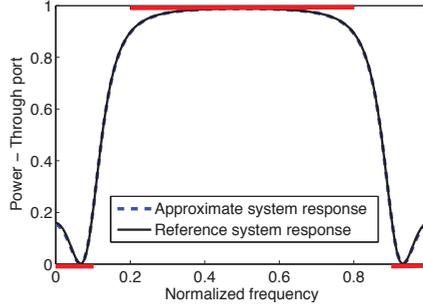


Figure 6.13: Plot shows the approximate system response and the reference system response at the global optimum found by BEGO.

formed on system level. Once the transfer function is known, it has to be plugged into Equation (6.29) to find the system objective.

A total computational budget of 21 expensive simulations for evaluating P_{L0} and L_π is reserved in addition to the 9 initial simulations. At each iteration of the system algorithm, P_{L0} and L_π is simulated three times for the three different combinations $[w \ g_1]$, $[w \ g_2]$ and $[w \ g_3]$ of the location for infill sampling suggested by the algorithm. The total number of simulations available translates to 7 iterations of the system optimization algorithm since $7 \times 3 = 21$ simulations. The three new P_{L0} responses at each iteration are all added to the Kriging metamodel \mathcal{K}_{c1} . On the other hand, \mathcal{K}_{c2} is augmented with the set of three new beat length responses L_π at each iteration.

Table 6.3: The location of the global optimum is given along with the objective on the approximate system and the reference system

	w	g1	g2	g3	L	$S_{\mathcal{K}}$	S
Optimum	1.0699	1.085	1.100	1.084	1940.2	0.0159	0.0160

Figure 6.13 shows the optimal system response found after 7 iterations of BEGO. The approximate system response $S_{\mathcal{K}}$ is plotted along with the system response obtained when the optimal result is fed into the expensive component simulators. It can be seen from the figure that the system response based on the component metamodels closely

Table 6.4: The response of the individual components at the global optimum are given for the metamodels and the reference simulator.

Components	$c_1(P_{L0})$	$c_3(P_{L0})$	$c_5(P_{L0})$	$c_2(L_\pi)$	$c_4(L_\pi)$	$c_6(L_\pi)$
Reference	0.1203	0.1126	0.1203	903.45	937.83	903.45
Kriging	0.1203	0.1130	0.1203	903.62	938.00	903.62

resembles the actual system response at the global optimum. The optimum found by BEGO seems to perform quite well in the pass band region, i.e. almost all the light in the normalized frequency region $[0.2 \ 0.8]$ is being passed through. The amount of power in the stop bands $[0 \ 0.1]$, $[0.9 \ 1]$ is still not very low. This is to be expected since we are optimizing a second order filter, i.e. there are only two rings in the serial resonator structure. Increasing the filter order will further improve the performance.

Table 6.3 shows the location at which the global optimum of the serial ring resonator is found. Although we do not impose the restriction that $g_3 = g_1$, the optimal result produces a resonator with symmetric gaps. The numerical objective value for S and $S_{\mathcal{K}}$ confirm the result shown in Figure 6.13. Table 6.4 gives the response of the individual components at the global optimum. As expected, the Kriging metamodel responses for P_{L0} and L_{π} for each directional coupler section is close to the reference component response.

6.5. CONCLUSION

In this work, we have developed an efficient strategy for global optimization of systems involving independent components. The approach, referred to as BEGO, targets systems involving expensive to evaluate components that do not interact with each other. Kriging metamodels were employed to construct the response of the expensive components. A novel system level infill sampling strategy was derived in order to sample the components in regions of interest for system optimization. A linear Taylor series approximation of the system transformation of the Kriging components was performed in order to obtain an analytical expression for system level expected improvement. The infill sampling criterion was modified for constrained system optimization by deriving a system level probability of feasibility for each constraint.

The system level optimization approach was first compared with treating the problem as a component and applying Efficient Global Optimization [5] for obtaining the optimum. Similarly, the effectiveness of the technique was compared with building a response surface of the system via LHS and globally optimizing the cheap response. Both comparisons exhibited that, in general, there is clear efficiency improvement when optimization is performed on the decomposed problem. However, if the component responses are highly nonlinear and the system response is linear then decomposition may not always result in efficiency improvement.

We demonstrated the advantages of using an adaptive sampling scheme for the decomposed system problem by comparing the effectiveness of the strategy with performing optimization of a system response based on component metamodels constructed using Latin Hypercube sampling. The two approaches were applied on a modified and constrained version of the Rastrigin function. Based on an average of 20 runs, BEGO converged comparatively faster to the optimum of the problem as opposed to the Latin Hypercube sampling based optimization strategy.

In addition, BEGO was applied on an engineering system involving an optical filter based on an integrated photonic serial ring resonator. The engineering problem consisted of a system with six uncoupled components. It was shown that BEGO was able to determine the global optimum of the system problem using only a limited number of expensive component simulations.

The algorithm is especially relevant for problems in which many multiple components are part of the system. In this scenario the same metamodel can be used to construct the response for the multiple components. Another advantage of employing the bilevel framework is that the number of dimensions for each component metamodel is often much lower than the total dimension size at system level. This enables the approximate system response to converge fast to the actual response with each passing component level simulation. Since the component response is treated as a black-box, the method is applicable to any hierarchical system with low dimensional components.

Appendix A: Kriging

The correlation between sample points is given by the following expression

$$\text{Corr} [Y(\mathbf{x}_i), Y(\mathbf{x}_j)] = \exp \left(- \sum_{q=1}^k \theta_q |x_{iq} - x_{jq}|^2 \right) \quad (30)$$

where \mathbf{x}_i and \mathbf{x}_j are any two locations in the domain and k is the number of dimensions and θ_q , μ , σ^2 are the parameters estimated via Maximum Likelihood Estimation. The Kriging prediction \hat{y} is expressed as

$$\hat{y}(\mathbf{x}) = \hat{\mu} + \mathbf{r}^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1} \hat{\mu}) \quad (31)$$

where $\hat{\mu}$ is the estimated mean, \mathbf{R} is the $N \times N$ correlation matrix between the N samples, \mathbf{r} is the vector of correlations between the observed data and the new prediction and \mathbf{y} is the observed response. Equation (30) is used to find \mathbf{R} and \mathbf{r} .

The Mean Squared Error (MSE) estimate in the Kriging prediction is expressed as,

$$s^2(\mathbf{x}) = \hat{\sigma}^2 \left[\mathbf{1} - \mathbf{r}^T \mathbf{R}^{-1} \mathbf{r} + \frac{\mathbf{1} - \mathbf{1}^T \mathbf{R}^{-1} \mathbf{r}}{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}} \right], \quad (32)$$

where $\hat{\sigma}^2$ is the Maximum Likelihood Estimate (MLE) of σ^2 .

REFERENCES

- [1] J. R. R. A. Martins and A. B. Lambe, *Multidisciplinary Design Optimization: A Survey of Architectures*, AIAA Journal **51**, 2049 (2013).
- [2] G. Lifante, *Integrated photonics: fundamentals* (J. Wiley, 2003).
- [3] O. S. Ahmed, M. A. Swillam, M. H. Bakr, and X. Li, *Efficient Design Optimization of Ring Resonator-Based Optical Filters*, Lightwave Technology, Journal of **29**, 2812 (2011).
- [4] J. Sacks, W. Welch, T. J. Mitchell, and H. P. Wynn, *Design and Analysis of Computer Experiments*, Stat Sci **4**, 409 (1989).
- [5] D. R. Jones, M. Schonlau, and W. J. Welch, *Efficient Global Optimization of Expensive Black-Box Functions*, Journal of Global Optimization **13**, 455 (1998).

- [6] F. A. C. Viana, T. W. Simpson, V. Balabanov, and V. Toropov, *Metamodeling in Multidisciplinary Design Optimization: How Far Have We Really Come?* AIAA Journal **52**, 670 (2014).
- [7] G. Li, S. Azarm, A. Farhang-Mehr, and A. R. Diaz, *Approximation of multiresponse deterministic engineering simulations: a dependent metamodeling approach*, Structural and Multidisciplinary Optimization **31**, 260 (2006).
- [8] W. Hu, M. Li, S. Azarm, and A. Almansoori, *Multi-Objective Robust Optimization Under Interval Uncertainty Using Online Approximation and Constraint Cuts*, Journal of Mechanical Design **133**, 061002 (2011).
- [9] M. Kokkolaras, Z. P. Mourelatos, and P. Y. Papalambros, *Design optimization of hierarchically decomposed multilevel systems under uncertainty*, in *ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (American Society of Mechanical Engineers, 2004) pp. 613–624.
- [10] R. R. Barton, *Design of experiments for fitting subsystem metamodels*, in *Proceedings of the 29th conference on Winter simulation* (IEEE Computer Society, 1997) pp. 303–310.
- [11] P. N. Koch, D. Mavris, and F. Mistree, *Partitioned, Multilevel Response Surfaces for Modeling Complex Systems*, AIAA Journal **38**, 875 (2000).
- [12] R. Xiu, X. Zhang, Y. Liu, and H.-Z. Huang, *Metamodeling uncertainty quantification in multi-level engineering system design*, in *Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE), 2013 International Conference on* (2013) pp. 449–454.
- [13] A. Forrester, S. Andras, and A. J. Keane, *Engineering design via surrogate modelling : a practical guide* (J. Wiley, 2008).
- [14] M. Schonlau and W. Welch, *Global optimization with nonparametric function fitting*, in *Proceedings of the Section on Physical and Engineering Sciences* (American Statistical Association, 1996) pp. 183–186.
- [15] M. Schonlau, *Computer experiments and global optimization*, Ph.D. thesis, University of Waterloo (1997).
- [16] M. Taboga, *Lectures on probability theory and mathematical statistics.*, 2nd ed. (CreateSpace Independent Publishing Platform, 2012) p. 656.
- [17] B. M. Ayyub, *Risk Analysis in Engineering and Economics* (Chapman & Hall/CRC, Boca Raton, FL, 2003).
- [18] D. Surjanovic, S. & Bingham, *Virtual Library of Simulation Experiments: Test Functions and Datasets*, (2013).
- [19] Syms, R and J. Cozens, *Optical guided waves and devices* (McGraw-Hill, 1992).

7

ROBUST OPTIMIZATION OF SYSTEMS WITH INDEPENDENT COMPONENTS

7.1. INTRODUCTION

The optimization of systems consisting of multiple subsystems or disciplines with complex interaction between them has historically been addressed within the framework of Multidisciplinary Design Optimization (MDO) [1]. MDO basically involves decomposition of the system into components or disciplines, and the development of algorithms to improve both the inter-disciplinary consistency and system optimality. While the system optimization work has typically focused on deterministic problems, many practical engineering problems also involve uncertainties. Specific approaches have been developed to deal with the resulting non-deterministic system optimization problem [2], recent examples include for example the work by [3–5].

However, problems affected by uncertainties could also consist of systems with independent components. In this subclass, each component affects the response of the system but there is no coupling between components. In this chapter, we consider the scenario where the components are computationally expensive to simulate but the computation of the system response from component output is inexpensive. System optimization under uncertainty of these problems therefore also becomes a computationally intensive task, despite the cheap system evaluation step.

An efficient approach for robust optimization of systems with expensive to simulate independent components could directly benefit many problems within the discipline of integrated photonics. On-chip systems are designed to process photonic signals for example data communication, signal conditioning or measurement purposes. These systems consist of interconnected basic components such as directional couplers [6] and MMI couplers [7]. The component responses as a function of design parameters are expensive to evaluate. However, once the component responses are available, a computa-

tionally cheap system transformation of the component response can generate complex systems such as interferometers and optical filters [8]. Equivalent engineering problems that could be decomposed in this way may also be present in other disciplines.

A complicating factor is the presence of uncertainties in the problem. The system optimum should be robust with respect to these uncertainties. The probability distributions of the uncertainties in the problem may either be known completely or only the intervals of the uncertainties may be available. The latter situation involving bounded-but-unknown uncertainties typically occurs in the field of integrated photonics where the probability distribution of the fabrication uncertainties is protected information of the foundries. In this scenario, robust optimization [9] has to be applied in order to estimate a relatively insensitive optimal solution.

Since robust optimization involves solving a nested optimization problem, applying it directly on a system based on expensive subsystems is prohibitively costly. A viable strategy for optimizing this class of problems is to replace the expensive subsystems with cheaper models. Optimization under uncertainty can then be applied on the system response derived from the cheaper surrogate subsystems. Such a strategy would be even more beneficial if the component responses are relatively simpler to approximate than the system response.

System optimization using metamodels at subsystem level is an attractive approach for efficiently estimating an optimum. As an additional benefit, if the system has multiple copies of the same component, a single metamodel could replace all the components. Furthermore, since the metamodels are built at component level they are much lower dimensional than the number of variables of the system. The curse of dimensionality, therefore, does not impact this approach as acutely as would be the case if a single metamodel of the system response is constructed. Additionally, much fewer samples are usually needed to build a high fidelity metamodel for the lower dimensional components.

However, the introduction of metamodel error at subsystem level could lead to an inaccurate estimation of the optimum. In order to mitigate this problem, the subsystem level metamodels could adaptively be improved in regions of interest and the optimum could be estimated via an iterative procedure. But identifying sampling locations at subsystem level based on the response at system level is a non-trivial exercise. The most optimal sample placement not only depends on the subsystem response but also on how it transforms into the system level response. Critically, any subsystem level metamodel error would also undergo a transformation. Therefore, simply adding sample points at locations where the subsystem level metamodel error is high is not a sound approach.

Metamodels have commonly been used in system optimization due to the computational complexity of evaluating certain component level responses [10, 11]. System robust optimization based on component metamodels has been explored previously by [12]. There has also been work on methods for improving the approximate system response by adaptively sampling the components so that the system robust optimum is found [3, 13]. This set of work was focused on the typical scenario in which the system has components that exchange coupling variables with one another. We instead focus exclusively on systems with independent components, which allows us to formulate an efficient approach by exploiting the specific structure of this class of problems.

This chapter presents an infill sampling strategy in order to sample component meta-models such that the global robust optimum of systems with expensive to evaluate *independent* components is found efficiently. We address the problem by constructing meta-models of each component and adaptively sampling the metamodels at each iteration based on a system level infill sampling criterion. The metamodels are built via Kriging [14], which also provides an estimator for the error in the component metamodels. A first order transformation is performed on the component mean squared error to obtain a system level error estimator. This error estimator is used together with the current system robust optimum to derive system level Expected Improvement (EI) criteria for the design variables and uncertainties in the problem. It is important to point out here that no metamodel is constructed at system level. The optimization and infill sampling processes are driven by the actual computed cheap system responses and EI values, without interpolation.

The proposed approach builds on the component level EI criterion for deterministic optimization suggested by [15] and its extension to robust optimization by [16], as well as on the system level EI criterion for deterministic optimization recently introduced by [17]. The primary novelty of this work is the extension of the work by [17] to the scenario in which the system problem is affected by uncertainties. The technique is applied on a constrained benchmark robust optimization problem as well as on an engineering example of an integrated photonic serial ring resonator affected by fabrication uncertainties. The method efficiently converges to the robust optimum of the problem using a limited computational budget.

This chapter is organized as follows. The system level robust optimization problem is introduced in Section 7.2. Section 7.3 provides a background for Kriging and adaptive sampling strategies previously developed for deterministic and robust optimization at device and system level. We introduce the proposed infill sampling criteria for robust optimization of systems with independent components in Section 7.4. Finally, Sections 7.5 and 7.6 contain the results and conclusion, respectively.

7.2. ROBUST OPTIMIZATION OF SYSTEMS WITH INDEPENDENT COMPONENTS

A system response S based on \mathcal{N} uncoupled components \mathbf{c} can be described as

$$S(c_1(\mathbf{x}_{d1}), c_2(\mathbf{x}_{d2}), c_3(\mathbf{x}_{d3}), \dots, c_{\mathcal{N}}(\mathbf{x}_{d\mathcal{N}}), \mathbf{x}_{ds}), \quad (7.1)$$

where each component c_i , $\{i | i \in \mathbb{N}, i \leq \mathcal{N}\}$, is a function of $\mathbf{x}_{di} \in \mathbb{X}_{di} \subseteq \mathbb{X}_d$. Some design variables, $\mathbf{x}_{ds} \in \mathbb{X}_{ds}$ of the problem appear directly at the system level. The union of $\mathbb{X}_{di} \forall i \in \mathcal{N}$ and \mathbb{X}_{ds} gives the total design space \mathbb{X}_d .

The components in the system are independent since they do not exchange any coupling variables with one another and only affect the system response. However, it is pertinent to note here that the component level design variables, $\mathbf{x}_{di} \in \mathbb{X}_{di}$, may affect multiple components. Similarly, the system level variables \mathbf{x}_{ds} may be present in some components as well. Consequently, the sets \mathbb{X}_{d1} to $\mathbb{X}_{d\mathcal{N}}$ will not necessarily be disjoint. Design variables that appear only in one component may be considered as local design variables while shared variables are global design variables.

If the system problem is affected by uncertainties then these may impact the component level response as well as the system response. Let $\mathbf{x}_e \in \mathbb{X}_e$ describe the set of parametric uncertainties or environment variables.

The system response for an uncertain system based on independent components may be written as

$$S(c_1(\mathbf{x}_{d1}, \mathbf{x}_{e1}), c_2(\mathbf{x}_{d2}, \mathbf{x}_{e2}), c_3(\mathbf{x}_{d3}, \mathbf{x}_{e3}), \dots, c_{\mathcal{N}}(\mathbf{x}_{d\mathcal{N}}, \mathbf{x}_{e\mathcal{N}}), \mathbf{x}_{dS}, \mathbf{x}_{eS}), \tag{7.2}$$

where $\mathbf{x}_{ei} \in \mathbb{X}_{ei} \subseteq \mathbb{X}_e$. A subset, $\mathbf{x}_{eS} \in \mathbb{X}_{eS}$, of uncertainties appear at system level. Uncertainties can be component-specific or may affect multiple components.

In this work, we address the case in which only intervals of the uncertainties are known while the probability distribution is not available. Robust optimization at system level then requires the estimation of the best worst-case cost for S [9]. In this section, for clarity we discuss unconstrained robust optimization only. However, the proposed approach is applicable to constrained problems as well. We expand upon the constrained optimization case in Section 7.4.3. Finding the best worst-case cost for the unconstrained problem involves the following nested optimization procedure,

$$\min_{\mathbf{x}_d \in \mathbb{X}_d} \max_{\mathbf{x}_e \in \mathbb{X}_e} S(\mathbf{c}(\mathbf{x}_d, \mathbf{x}_e), \mathbf{x}_{dS}, \mathbf{x}_{eS}). \tag{7.3}$$

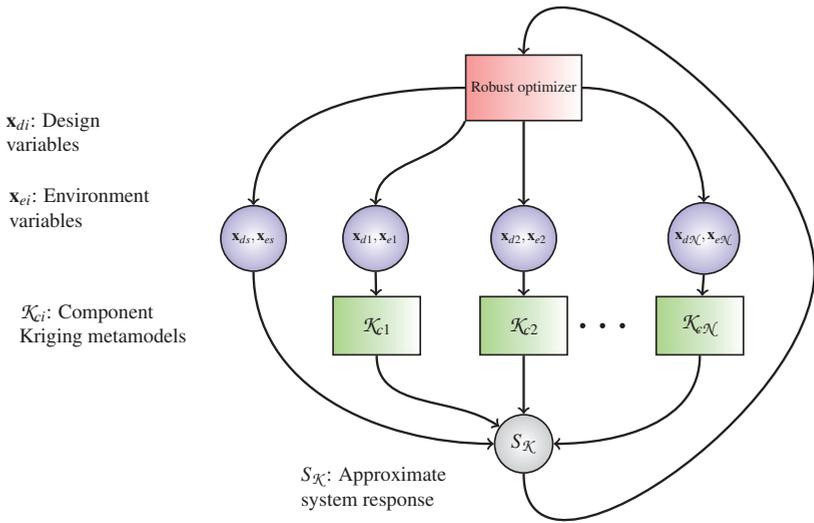


Figure 7.1: Robust optimization of the approximate system response based on component metamodels.

Minimizing the worst-case cost is computationally intensive since estimating the worst-case cost also requires a global optimization procedure. In this scenario it becomes imperative that the system response is cheap to compute. If the underlying components are expensive to evaluate then they can be replaced by metamodels in order to ensure a cheap system level objective evaluation. Robust optimization of a system with cheap independent component metamodels can be described as,

$$\min_{\mathbf{x}_d \in \mathcal{X}_d} \max_{\mathbf{x}_e \in \mathcal{X}_e} S_{\mathcal{K}}(\mathcal{K}_{c1}(\mathbf{x}_{d1}, \mathbf{x}_{e1}), \mathcal{K}_{c2}(\mathbf{x}_{d2}, \mathbf{x}_{e2}), \dots, \mathcal{K}_{cN}(\mathbf{x}_{dN}, \mathbf{x}_{eN}), \mathbf{x}_{ds}, \mathbf{x}_{es}). \quad (7.4)$$

where \mathcal{K}_{ci} represents the metamodel of component c_i . Figure 7.1 shows the process of robust optimization of the system problem and depicts the relationship between the approximate system response $S_{\mathcal{K}}$ and the underlying components. Note that no metamodel is constructed at system level.

The aim of this work is to estimate the global robust optimum of S by applying robust optimization on $S_{\mathcal{K}}$, Equation (7.4). As soon as the expensive components are replaced by metamodels, the error inherent in them is also introduced in the system response, $S_{\mathcal{K}}$. A strategy to iteratively improve the component metamodels in regions of interest for system level global robust optimization is therefore required.

Finding such a system level infill sampling criterion is non-trivial since a small metamodel error at component level could result in a large error at system level due to the transformation from component to system level. In addition, to help locate the global robust optimum, the samples should be added in such a way that the impact of the parametric uncertainties, \mathbf{x}_e , on the system response is also included. Any formulation of a system level infill sampling strategy must therefore take these effects into consideration. In this scenario, an interpolation method that provides a sound error estimator in the interpolation is needed to help formulate a suitable infill sampling criteria. Kriging is one of the interpolation techniques that provide an interpolation error estimate. While it is based on statistical principles of maximum likelihood, Kriging has proven to also be a valuable and successful procedure for interpolation of deterministic data, e.g. response surfaces [18, 19].

7.3. BACKGROUND

In this section, we provide a brief background of Kriging [14], Component level Efficient Global Optimization [15], Component level Efficient Global Robust Optimization [16] and System level Efficient Global Optimization [17]. Experts in these areas may skip to the next section to directly peruse the novel method proposed in this work.

7.3.1. KRIGING

Kriging is a mathematical modeling technique that interpolates the response between samples via a linear combination of tunable basis functions. The statistical approach employed by Kriging enables the estimation of the potential error in the interpolated model. This mean squared error is an essential ingredient in the derivation of adaptive sampling strategies for iteratively improving the response surface. A very concise description of fundamental aspects in Kriging based optimization is provided here. For detailed derivation readers are referred to the work by [14].

The statistical framework of Kriging utilizes the starting assumption that the function response is a normally distributed random variable. The tunable basis function used in Kriging usually takes the form of a parameterized Gaussian correlation function. The Gaussian correlation function between any two points \mathbf{x}_i and \mathbf{x}_j in the domain is given by

$$\text{Corr}[Y(\mathbf{x}_i), Y(\mathbf{x}_j)] = \exp\left(-\sum_{q=1}^k \theta_q |x_{iq} - x_{jq}|^2\right). \quad (7.5)$$

θ_q , μ and σ^2 are the parameters found via Maximum Likelihood Estimation while k is the total number of dimensions. The estimates for θ_q , μ and σ^2 are denoted by $\hat{\theta}_q$, $\hat{\mu}$ and $\hat{\sigma}^2$ respectively.

The Kriging prediction is also derived using maximum likelihood estimation [14]. The prediction is expressed in terms of the vector of observed responses \mathbf{y} , the correlation matrix \mathbf{R} between the N sample points and the correlation vector, \mathbf{r} , between the observed data and the new unsampled location,

$$\hat{y}(\mathbf{x}) = \hat{\mu} + \mathbf{r}^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1}\hat{\mu}). \quad (7.6)$$

Equation (7.5) is used to find the matrix \mathbf{R} and the vector \mathbf{r} .

The potential Mean Squared Error (MSE) in the Kriging interpolation, Equation (7.7), is also derived based on the underlying statistical assumptions. The error in the Kriging interpolation is given by,

$$s^2(\mathbf{x}) = \hat{\sigma}^2 \left[1 - \mathbf{r}^T \mathbf{R}^{-1} \mathbf{r} + \frac{1 - \mathbf{1}^T \mathbf{R}^{-1} \mathbf{r}}{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}} \right]. \quad (7.7)$$

7.3.2. COMPONENT LEVEL EFFICIENT GLOBAL OPTIMIZATION

The combination of the Kriging predictor and Mean Squared Error can be used for balancing exploration and exploitation of the metamodel [20]. The adaptive sampling strategy of Expected Improvement (EI), proposed by [15], effectively balances the need for exploring regions with high potential error with the requirement of sampling areas that could help in estimating the optimal location. The sampling strategy enables the global minimum of an unconstrained problem to be found in relatively few expensive function calls.

The EI criterion was developed by assuming that the uncertainty in the Kriging prediction, $\hat{y}(\mathbf{x})$, takes the form of a normal random variable whose mean and variance is given by the Kriging prediction $\hat{y}(\mathbf{x})$ and MSE $s^2(\mathbf{x})$, respectively. Using this information along with the knowledge of the current minimum amongst the observed data, y_{\min} , enables the formulation of an expectation of improvement over y_{\min} . An improvement I over y_{\min} could be possible at a position \mathbf{x} if a portion of the distribution $Y(\mathbf{x})$ lies below y_{\min} . The probability of this improvement is given by the area of the normal distribution below the current minimum y_{\min} . The expectation, $E[I(\mathbf{x})] = E[\max(y_{\min} - Y, 0)]$, of improvement [21] is given by the first moment of this area. Mathematically, the EI may be written as,

$$E[I(\mathbf{x})] = (y_{\min} - \hat{y}) \Phi\left(\frac{y_{\min} - \hat{y}}{s}\right) + s \phi\left(\frac{y_{\min} - \hat{y}}{s}\right) \quad (7.8)$$

where $\Phi(\cdot)$ is the normal cumulative distribution function and $\phi(\cdot)$ is the normal probability density function. If the term before the addition sign on the right side is high then emphasis will be placed on local search. On the other hand, if the term after the addition sign is relatively high then exploration in areas with large potential error will be prioritized. By performing global optimization on the cheap to evaluate expression in (7.8) a location that maximizes the expected improvement can be estimated.

Efficient Global Optimization (EGO) [15], Algorithm 1, uses expected improvement to adaptively sample the expensive function so that the global optimum is found using relatively few iterations. The initial sampling locations are found via a space filling technique such as Latin Hypercube sampling. The algorithm runs until either the total number of simulations N_T is exhausted or the maximum EI falls below the threshold ϵ_{EI} .

7.3.3. COMPONENT LEVEL EFFICIENT GLOBAL ROBUST OPTIMIZATION

[16] extended the EGO algorithm to the non-deterministic case. They proposed a method known as Efficient Global Robust Optimization (EGRO) which specifically targeted problems affected by bounded-but-unknown parametric uncertainties. The response was assumed to originate from an expensive to simulate black-box function. An iterative sampling strategy similar to the deterministic EI was employed in order to efficiently estimate the global robust optimum of the following problem,

$$\min_{\mathbf{x}_d \in \mathbb{X}_d} \max_{\mathbf{x}_e \in \mathbb{X}_e} f(\mathbf{x}_d, \mathbf{x}_e), \quad (7.9)$$

where $\mathbf{x}_d \in \mathbb{X}_d$ are the design variables and $\mathbf{x}_e \in \mathbb{X}_e$ are the parametric uncertainties. Algorithm 2 shows the essential steps in EGRO. The proposed adaptive sampling strategy involved a separate infill sampling criterion, EI_d , for the control variable and a different infill sampling criterion, EI_e , for the parametric uncertainties. EI_d was geared towards computing the expectation of improvement over the current robust optimum $r_{\mathcal{X}}$. On the other hand, the expectation of deterioration EI_e involved evaluating the expectation

Algorithm 1 Component level Efficient Global Optimization

- 1: $n \leftarrow$ Number of initial samples
 - 2: $N_T \leftarrow$ Total number of simulations
 - 3: $\epsilon_{EI} \leftarrow$ Expected Improvement threshold
 - 4: Choose initial samples $\mathcal{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]$
 - 5: Compute function response $\mathbf{y} = [y_1, \dots, y_n]$
 - 6: **while** $n < N_T$ and $EI_{\max} > \epsilon_{EI}$ **do**
 - 7: Construct Kriging metamodel \mathcal{K}_f
 - 8: $y_{\min} \leftarrow$ Minimum value in set of observed responses
 - 9: Estimate \mathbf{x}_{new} by maximizing EI on metamodel.
 - 10: $EI_{\max} \leftarrow \max(E[I(x)])$
 - 11: Append \mathbf{x}_{new} to \mathcal{X} and $f(\mathbf{x}_{new})$ to \mathbf{y}
 - 12: $n \leftarrow n + 1$.
 - 13: Return $\mathbf{x}_{best} = \arg y_{min}$.
-

Algorithm 2 Component level Efficient Global Robust Optimization

```

1:  $n \leftarrow$  Number of initial samples
2:  $N_T \leftarrow$  Total number of simulations
3:  $\epsilon_{EI} \leftarrow$  Expected Improvement threshold
4: Choose initial samples  $\mathcal{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]$ 
5: Compute function response  $\mathbf{y} = [y_1, \dots, y_n]$ 
6: while  $n < N_T$  and  $EI_d^{\max} > \epsilon_{EI}$  do
7:   Construct Kriging metamodel  $\mathcal{K}_f$ 
8:   Estimate global robust optimum  $r_{\mathcal{X}}$  on metamodel
9:   Estimate  $\mathbf{x}_d^{new}$  by maximizing  $EI_d$ 
10:   $EI_d^{\max} \leftarrow \max(EI_d)$ 
11:  Estimate  $\mathbf{x}_e^{new}$  by maximizing  $EI_e$ 
12:  Append  $\mathbf{x}_{new} = (\mathbf{x}_d^{new}, \mathbf{x}_e^{new})$  to  $\mathcal{X}$  and  $f(\mathbf{x}_{new})$  to  $\mathbf{y}$ 
13:   $n \leftarrow n + 1$ .
14: Return  $\mathbf{x}_{best} = \arg r_{\mathcal{X}}$ .

```

of finding a worse value than the predicted worst-case cost at the control variable sampling location, \mathbf{x}_d^{new} . On termination of the algorithm, the argument of the best worst-case cost found, $r_{\mathcal{X}}$, was returned as the final solution.

7.3.4. BILEVEL EFFICIENT GLOBAL OPTIMIZATION (BEGO)

Recently, [17] proposed an adaptive sampling strategy for global optimization of systems with independent components, i.e. minimization of Equation (7.1). A system level error was needed in order to derive a system level expected improvement criterion. The approach is based on a linear transformation of the Kriging error predicted at the component level to arrive at a system level error estimator.

Algorithm 3 System level Efficient Global Optimization for unconstrained problems

```

1:  $n_i \leftarrow$  Number of initial samples for  $i$ th component
2:  $N \leftarrow$  Total number of iterations
3: Choose initial samples  $\mathcal{X}_i \forall \mathcal{N}$ 
4: Compute response on components  $c_i$  to  $c_{\mathcal{N}}$ 
5:  $j \leftarrow 0$ 
6: while  $j < N$  do
7:   Construct Kriging metamodels  $\mathcal{K}_{c_i} \forall \mathcal{N}$ 
8:   Estimate global minimum  $d_{\mathcal{X}}$  on system response  $S_{\mathcal{X}}$ 
9:   Estimate  $\mathbf{x}_{new}$  by maximizing  $EI_{sys}$ 
10:  Compute response on components at  $\mathbf{x}_{new}$ 
11:   $j \leftarrow j + 1$ .
12: Return  $\mathbf{x}_{best} = \arg d_{\mathcal{X}}$ .

```

The system level error estimator was used to predict the expectation of improvement over the current global optimum, $d_{\mathcal{X}}$, on the system response. Algorithm 3 shows the

steps in BEGO. Details related to the derivation of EI_{sys} and the implementation of the algorithm may be found in [17].

Algorithm 4 System level Efficient Global Robust Optimization for unconstrained problems

- 1: $n_i \leftarrow$ Number of initial samples for i th component
 - 2: $N \leftarrow$ Total number of iterations
 - 3: Choose initial samples $\mathcal{X}_i \forall \mathcal{N}$
 - 4: Compute response on components c_i to $c_{\mathcal{N}}$
 - 5: $j \leftarrow 0$
 - 6: **while** $j < N$ **do**
 - 7: Construct Kriging metamodels $\mathcal{K}_{c_i} \forall \mathcal{N}$
 - 8: Estimate global robust optimum $r_{\mathcal{K}}^{sys}$ on system response $S_{\mathcal{K}}$
 - 9: Estimate \mathbf{x}_d^{new} by maximizing $EI_{sys,d}$
 - 10: Estimate \mathbf{x}_e^{new} by maximizing $ED_{sys,e}$
 - 11: Compute response on components at $\mathbf{x}_{new} = (\mathbf{x}_d^{new}, \mathbf{x}_e^{new})$
 - 12: $j \leftarrow j + 1$.
 - 13: Return $\mathbf{x}_{best} = \arg r_{\mathcal{K}}^{sys}$.
-

7.4. BILEVEL EFFICIENT GLOBAL ROBUST OPTIMIZATION (BEGRO)

In this work we aim to combine component level EGRO, (Algorithm 2, [16]), with System level EGO, (Algorithm 3, [17]), to arrive at a system level efficient global robust optimization strategy. Algorithm 4 shows the main steps involved in this process. The responses of the components are computed (Step 4) based on initial samples chosen via Latin Hypercube sampling (Step 3). Each component metamodel can have a different number of initial samples. Thereafter, the main loop of the algorithm is initiated in Step 6. The algorithm is referred to as Bilevel Efficient Global Robust Optimization (BEGRO).

Kriging metamodels are constructed for all components based on the set of samples and responses in Step 7. At the next step, the global robust optimum, $r_{\mathcal{K}}^{sys}$, is estimated on the system response, Equation (7.4), based on the component metamodels.

The two main steps of the proposed algorithm are Step 9 and Step 10. The new sample points in the control variable space \mathbb{X}_d and parametric uncertainties space \mathbb{X}_e are estimated in these two steps. In order to perform this, a separate system level expected improvement criterion is derived to find the new set of control variable samples, \mathbf{x}_d^{new} , and a separate system level expected deterioration criterion is derived to find the new set of parametric uncertainty samples, \mathbf{x}_e^{new} . The approach is similar to the one described for Component level EGRO, Algorithm 2, in that separate criteria are derived for the control variable space and the parametric uncertainties space.

7.4.1. ESTIMATION OF \mathbf{x}_d^{new}

A system level adaptive sampling scheme in the control variable space, \mathbb{X}_d , should ideally provide the most likely location that will improve over the current global system level

robust optimum, $r_{\mathcal{K}}^{sys}$. Such an infill sampling criterion would need some measure of the system level error estimate for the response $S_{\mathcal{K}}$.

A system level error estimate depends not only on the error in the individual component metamodels but also on the transformation that takes place from component to system level. An error estimator is available for each component metamodel, \mathcal{K}_{ci} , via the Kriging mean squared error, $s_i^2(\mathbf{x}_d, \mathbf{x}_e)$, found using Equation (7.7). Following the approach of [17] for System level EGO, we perform a linear transformation of the component error to find a system level error estimate.

Let \hat{y}_i represent the Kriging prediction for the i^{th} component metamodel \mathcal{K}_{ci} . Then the system level response, \hat{y}_{sys} , can be expressed as,

$$\hat{y}_{sys} = S_{\mathcal{K}}(\hat{y}_1(\mathbf{x}_{d1}, \mathbf{x}_{e1}), \hat{y}_2(\mathbf{x}_{d2}, \mathbf{x}_{e2}), \dots, \hat{y}_{\mathcal{N}}(\mathbf{x}_{d\mathcal{N}}, \mathbf{x}_{e\mathcal{N}}), \mathbf{x}_{cs}, \mathbf{x}_{es}). \quad (7.10)$$

In the derivation of an EI criterion for the component level EGO algorithm described in Section 7.3.2 it was assumed that the *metamodel uncertainty* in the component response can be described in terms of a normally distributed random variable. We hold on to this assumption in order to derive a system level robust EI criterion. In addition we assume that the error in the system level response, \hat{y}_{sys} , can be modeled as a normal random variable with mean \hat{y}_{sys} and variance s_{sys}^2 , where s_{sys}^2 is a system level mean squared error.

For multiple components, this means that the *metamodel uncertainty* in the predicted response of each component \mathcal{K}_{ci} can be described as a normally distributed random variable $Y_i(\mathbf{x}_{di})$ with mean given by the Kriging prediction \hat{y}_i and variance given by the Kriging MSE s_i^2 . It is emphasized here that the uncertainty being referred to is the metamodel uncertainty and not the parametric uncertainty \mathbb{X}_e , whose distribution is bounded-but-unknown.

In this setting, the system response can also be defined as a random variable $Y_{sys}(\mathbf{x}_d, \mathbf{x}_e)$, which may be expressed as a system transformation of the independent component level random variables $\mathbf{Y} = [Y_1, Y_2, \dots, Y_{\mathcal{N}}]$ and the deterministic system variables $(\mathbf{x}_{cs}, \mathbf{x}_{es})$,

$$Y_{sys}(\mathbf{x}_d, \mathbf{x}_e) = S_{\mathcal{K}}(\mathbf{Y}, \mathbf{x}_{cs}, \mathbf{x}_{es}). \quad (7.11)$$

The system operation may or may not be linear. A non-linear transformation will cause the component level normal distributions to transform non-linearly as well. In such a scenario Y_{sys} would no longer be normally distributed. This would make valuation of quantities such as expected improvement computationally cumbersome.

In order to maintain a normal distribution for Y_{sys} , a linear approximation of Equation (7.11) can be applied. We linearize the expression by performing a Taylor series expansion of $S_{\mathcal{K}}(\mathbf{Y}, \mathbf{x}_{cs}, \mathbf{x}_{es})$ about the mean values \hat{y}_i of \mathbf{Y} and retaining the first two terms only,

$$S_{\mathcal{K}}(\mathbf{Y}, \mathbf{x}_{cs}, \mathbf{x}_{es}) = S_{\mathcal{K}}(\hat{y}_1, \hat{y}_2, \dots, \hat{y}_{\mathcal{N}}, \mathbf{x}_{cs}, \mathbf{x}_{es}) + \sum_{i=1}^{\mathcal{N}} (Y_i - \hat{y}_i) \frac{\partial S_{\mathcal{K}}}{\partial Y_i} \Big|_{\hat{y}_i}. \quad (7.12)$$

The mean of the linear expression for Y_{sys} is the system level response [22],

$$E(Y_{sys}) = \hat{y}_{sys} = S_{\mathcal{K}}(\hat{y}_1, \hat{y}_2, \dots, \hat{y}_{\mathcal{N}}, \mathbf{x}_{cs}, \mathbf{x}_{es}) \quad (7.13)$$

while the variance is given by

$$\text{Var}(Y_{sys}) = s_{sys}^2 = \sum_{i=1}^{\mathcal{N}} b_i^2 s_i^2, \quad (7.14)$$

where $b_i = \left. \frac{\partial S_{\mathcal{K}}}{\partial Y_i} \right|_{\hat{y}_i}$.

Finite difference can be used on the inexpensive system-level functions to numerically approximate the derivatives b_i .

Using the system level error estimate s_{sys}^2 we can derive an EI criterion in \mathbb{X}_d . $r_{\mathcal{K}}^{sys}$ represents the best worst-case cost on the system response. To improve over $r_{\mathcal{K}}^{sys}$ a location is sought that could potentially have a lower *worst-case* cost. Let $\hat{y}_{sys}^{\max}(\mathbf{x}_d)$ represent the worst-case cost for a given value of \mathbf{x}_d ,

$$\hat{y}_{sys}^{\max}(\mathbf{x}_d) = \max_{\mathbf{x}_e \in \mathbb{X}_e} S_{\mathcal{K}}. \quad (7.15)$$

The corresponding location in \mathbb{X}_e where the worst-case cost is obtained is given by \mathbf{x}_e^{\max} . Let Y_{sys}^{\max} represent the normal distribution at $(\mathbf{x}_d, \mathbf{x}_e^{\max})$. The mean and standard deviation at Y_{sys}^{\max} is denoted by \hat{y}_{sys}^{\max} and s_{sys}^{\max} , respectively.

Having defined the nomenclature, an improvement over $r_{\mathcal{K}}^{sys}$ can now formally be written as $I_{sys} = \max(r_{\mathcal{K}}^{sys} - Y_{sys}^{\max}, 0)$. The expected value of this improvement is [16],

$$\underbrace{E[I_{sys}(\mathbf{x}_d)]}_{EI_{sys,d}} = \int_{I_{sys}=0}^{I_{sys}=\infty} I_{sys} \frac{\exp\left(-\frac{t_{sys}^2}{2}\right)}{\sqrt{2\pi}s_{sys}^{\max}} dt_{sys}, \quad (7.16)$$

where

$$t_{sys} = \frac{r_{\mathcal{K}}^{sys} - I_{sys} - \hat{y}_{sys}^{\max}}{s_{sys}^{\max}}. \quad (7.17)$$

The integration in Equation (7.16) can be solved by substitution [16]. The final analytical expression for $EI_{sys,d}$ can be written in terms of the normal cumulative distribution function Φ and the normal probability density function ϕ as

$$E[I_{sys}(\mathbf{x}_d)] = (r_{\mathcal{K}}^{sys} - \hat{y}_{sys}^{\max}) \Phi\left(\frac{r_{\mathcal{K}}^{sys} - \hat{y}_{sys}^{\max}}{s_{sys}^{\max}}\right) + s_{sys}^{\max} \phi\left(\frac{r_{\mathcal{K}}^{sys} - \hat{y}_{sys}^{\max}}{s_{sys}^{\max}}\right). \quad (7.18)$$

The global maximizer, \mathbf{x}_d^{new} , of Equation (7.18) is estimated in Step 9 of Algorithm 4 to give the new infill sample in \mathbb{X}_d .

7.4.2. ESTIMATION OF \mathbf{x}_e^{new}

The infill sample \mathbf{x}_d^{new} in \mathbb{X}_d provides the location expected to give the highest improvement in the best worst-case system level response. Given the infill sampling location \mathbf{x}_d^{new} in \mathbb{X}_d , we now find an infill sampling location in \mathbb{X}_e . Let $g_{\mathcal{K}}^{sys}$ represent the worst-case cost at \mathbf{x}_d^{new} evaluated on the system response $S_{\mathcal{K}}$,

$$g_{\mathcal{K}}^{sys} = \max_{\mathbf{x}_e \in \mathbb{X}_e} S_{\mathcal{K}}(\mathbf{x}_d^{new}, \mathbf{x}_e). \quad (7.19)$$

An expected deterioration criterion in \mathbb{X}_e is proposed which estimates the location with the highest expectation of deterioration over $\mathbf{g}_{\mathcal{K}}^{sys}$. In this scenario, a deterioration means finding a higher value than $\mathbf{g}_{\mathcal{K}}^{sys}$. This deterioration may mathematically be expressed as $D_e = \max(Y_{sys} - \mathbf{g}_{\mathcal{K}}^{sys}, 0)$. The expectation of this deterioration can be derived in exactly the same way as $EI_{sys,d}$. The expectation of deterioration is given by,

$$\underbrace{E[D_e(\mathbf{x}_d^{new}, \mathbf{x}_e)]}_{ED_{sys,e}} = \int_{I_e=0}^{I_e=\infty} D_e \frac{\exp\left(-\frac{I_e^2}{2}\right)}{\sqrt{2\pi}s_{sys}} dD_e, \quad (7.20)$$

where

$$t_e = \frac{\hat{y}_{sys} - I_e - \mathbf{g}_{\mathcal{K}}^{sys}}{s_{sys}}, \quad s_{sys} = s_{sys}(\mathbf{x}_d^{new}, \mathbf{x}_e). \quad (7.21)$$

Following exactly the same steps as described for deriving $EI_{sys,d}$, the final expression becomes,

$$\underbrace{E[D_e(\mathbf{x}_d^{new}, \mathbf{x}_e)]}_{ED_{sys,e}} = (\hat{y}_{sys} - \mathbf{g}_{\mathcal{K}}^{sys}) \Phi\left(\frac{\hat{y}_{sys} - \mathbf{g}_{\mathcal{K}}^{sys}}{s_{sys}}\right) + s_{sys} \phi\left(\frac{\hat{y}_{sys} - \mathbf{g}_{\mathcal{K}}^{sys}}{s_{sys}}\right). \quad (7.22)$$

The new sampling location in \mathbb{X}_e is found in Step 10 of Algorithm 4. The response of the components at the new location $\mathbf{x}_{new} = (\mathbf{x}_d^{new}, \mathbf{x}_e^{new})$ is computed. The loop continues until the total number of iterations N are completed. At this stage, the argument of the robust optimum $r_{\mathcal{K}}^{sys}$ is returned as the final solution.

7

Algorithm 5 System level Efficient Global Robust Optimization for constrained problems

- 1: $n_i \leftarrow$ Number of initial samples for i th component
 - 2: $N \leftarrow$ Total number of iterations
 - 3: Choose initial samples $\mathcal{X}_i \forall \mathcal{N}$
 - 4: Compute response on components c_i to $c_{\mathcal{N}}$
 - 5: $j \leftarrow 0$
 - 6: **while** $j < N$ **do**
 - 7: Construct Kriging metamodels $\mathcal{K}_{ci} \forall \mathcal{N}$
 - 8: Estimate constrained global robust optimum $r_{\mathcal{K}}^{sys}$
 - 9: Estimate \mathbf{x}_d^{new} by maximizing EI_{sc}
 - 10: Estimate \mathbf{x}_e^{new} by maximizing ED_{se}
 - 11: Compute response on components at $\mathbf{x}_{new} = (\mathbf{x}_d^{new}, \mathbf{x}_e^{new})$
 - 12: $j \leftarrow j + 1$.
 - 13: **Return** $\mathbf{x}_{best} = \arg r_{\mathcal{K}}^{sys}$.
-

7.4.3. INFILL SAMPLING CRITERIA FOR CONSTRAINED SYSTEMS

The infill sampling criteria $EI_{sys,d}$ and $ED_{sys,e}$ described in Section 7.4.1 and Section 7.4.2 would have to be modified if the system optimization problem involves constraints.

The modifications are needed to incorporate the presence of constraints in the design domain so that a *feasible* global robust optimum is reached efficiently.

A Probability of Feasibility (PF) measure was developed by [23] for constrained optimization problems in order to perform infill sampling on objective and constraint metamodels. The infill sampling strategy was basically a product of the EI in the objective, Equation (7.8), and the probability of feasibility of the constraint [24]. [25] extended this constrained EI strategy to the case where the robust optimum was sought for the constrained problem. Recently, [17] derived a system level constrained EI criterion for global optimization of systems with constraints.

In this work, separate constrained EI expressions are found for \mathbb{X}_d and \mathbb{X}_e . Given a problem with a single inequality constraint, let \hat{h}_{sys} represent the system level constraint response based on component metamodels,

$$\hat{h}_{sys} = S_h(\hat{y}_1(\mathbf{x}_1), \hat{y}_2(\mathbf{x}_2), \dots, \hat{y}_N(\mathbf{x}_N), \mathbf{x}_{cs}, \mathbf{x}_{es}). \quad (7.23)$$

where S_h is the system operation performed to find the constraint response. The system constraint error, s_h , can be found for the constraint response \hat{h}_{sys} using the same derivation as was used to find s_{sys} , Section 7.4.1. Once s_h is known, the derivation of the infill sampling criteria for both \mathbb{X}_d and \mathbb{X}_e is completely equivalent to the procedure described by [25]. The final expression for constrained system expected improvement, EI_{sc} , in \mathbb{X}_d is,

$$EI_{sc} = EI_{sys,d} \cdot PF_{sys,d}, \quad (7.24)$$

where

$$\underbrace{P[F(\mathbf{x}_d) < h_{min}]}_{PF_{sys,d}} = \Phi\left(\frac{h_{min} - \hat{h}_{sys}^{max}}{s_h}\right). \quad (7.25)$$

h_{min} is the constraint limit, \hat{h}_{sys}^{max} is the worst-case prediction for the constraint at \mathbf{x}_d and s_h is the system error at the corresponding location where the worst-case was found.

The expression for constrained system expected deterioration, ED_{se} , in \mathbb{X}_e is,

$$ED_{se} = ED_{sys,e} \cdot ED_{sys,h}, \quad (7.26)$$

where

$$\underbrace{E[D_h(\mathbf{x}_d^{new}, \mathbf{x})]}_{ED_{sys,h}} = (\hat{h}_{sys} - g_h^{sys}) \Phi\left(\frac{\hat{h}_{sys} - g_h^{sys}}{s_{sys}}\right) + s_{sys} \phi\left(\frac{\hat{h}_{sys} - g_h^{sys}}{s_{sys}}\right). \quad (7.27)$$

g_h^{sys} represents the worst-case constraint value found at \mathbf{x}_d^{new} .

The infill sampling criteria EI_{sc} and ED_{se} enable the constrained system level problem to be sampled in such a way that a feasible global system robust optimum is estimated efficiently. The derivation of EI_{sc} and ED_{se} is analogous to the infill criteria at device level, Algorithm 2. Details may be found in the work by [25]. Algorithm 5 shows the steps that need to be taken in order to find the system robust optimum for a constrained problem.

7.5. RESULTS

The algorithm is applied on one constrained benchmark problem and on an engineering case study. The benchmark problem used in this work was previously employed by the authors to demonstrate component level robust optimization [25]. BEGRO is tested on an engineering problem of an optical filter realized on a photonic integrated circuit. The problem is an interesting test case since the filter performance is very sensitive with respect to the fabrication variations in the geometry of the photonic integrated circuit.

7.5.1. NUMERICAL TEST PROBLEM

The following constrained test problem is used to demonstrate the algorithm [25],

$$\begin{aligned}
 f(\mathbf{x}_d, \mathbf{x}_e) &= 2x_{d1}x_{d5} + 3x_{d4}x_{d2} + x_{d5}x_{d3} + 5x_{d4}^2 + 5x_{d5}^2 - x_{d4}(x_{e4} - x_{e5} - 5) \\
 &\quad + x_{d5}(x_{e4} - x_{e5} + 3) + \sum_{i=1}^3 x_{ei}(x_{di}^2 - 1) - \sum_{i=1}^5 (x_{ei}^2), \\
 \text{s.t. } &5x_{d1} - x_{d2} + x_{d3} + x_{d4} - x_{d5} + x_{e1} - x_{e2} + x_{e3} + x_{e4} - x_{e5} \leq 0, \\
 &\mathbb{X}_d \in [-5, 5], \mathbb{X}_e \in [-3, 3].
 \end{aligned} \tag{7.28}$$

The problem consists of ten variables, all of which are present in both the objective and the constraint. While the constraint is a linear function, the objective is nonlinear.

For system level optimization, the problem is decomposed into expensive and cheap parts. We assume that the system level problem has a total of five expensive components. Four of these components are present in the objective while the fifth component belongs to the constraint. The system problem is defined as,

$$\begin{aligned}
 S[\mathbf{c}(\mathbf{x}_d, \mathbf{x}_e)] &= c_1 + c_2 + c_3 - c_4 - x_{d4}(x_{e4} - x_{e5} - 5) + x_{d5}(x_{e4} - x_{e5} + 3), \\
 \text{s.t. } &c_5 - x_{d5} - x_{e5} \leq 0, \\
 c_1(\mathbf{x}_d) &= 2x_{d1}x_{d5} + 3x_{d4}x_{d2}, \quad c_2(\mathbf{x}_d) = x_{d5}x_{d3} + 5x_{d4}^2 + 5x_{d5}^2, \\
 c_3(\mathbf{x}_d, \mathbf{x}_e) &= \sum_{i=1}^3 x_{ei}(x_{di}^2 - 1), \quad c_4(\mathbf{x}_e) = \sum_{i=1}^5 (x_{ei}^2), \\
 c_5(\mathbf{x}_d, \mathbf{x}_e) &= 5x_{d1} - x_{d2} + x_{d3} + x_{d4} + x_{e1} - x_{e2} + x_{e3} + x_{e4}, \\
 &\mathbb{X}_d \in [-5, 5], \mathbb{X}_e \in [-3, 3].
 \end{aligned} \tag{7.29}$$

The control variables, x_{d4} , x_{d5} and the environment variables x_{e4} , x_{e5} are present at both component level and system level. All the variables are also present in multiple components. The component c_2 has the fewest number of variables, i.e. 3. On the other hand, component c_5 , which is present in the constraint, is a function of eight variables. Some components are a function of both control variables and environment variables, e.g. c_3 , c_5 . Others are exclusively a function of control variables or environment variables, e.g. c_1 , c_2 , c_4 . The original problem, Equation (7.28), is decomposed into the system problem, Equation (7.29), in a manner such that all aspects of the system level algorithm may be tested.

BEGRO is applied on the decomposed problem, Equation (7.29). The initial component metamodel are constructed using LHS. Since the initial random sampling is non-deterministic, the algorithm is applied 20 times on the problem and the statistics of the

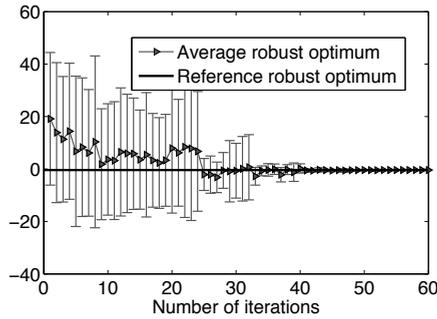


Figure 7.2: The robust optimum found on the approximate system response at each iteration of the algorithm is plotted. The error bars indicate the standard deviation around the average robust optimum at each iteration. The robust optimum found on the reference function is also plotted.

solution found are analyzed. The number of initial samples is different for each component and is given by $n_i = 10 \times$ number of dimensions of the i th component. Therefore, c_1 is initialized with only 40 samples, while c_5 is initialized with 80 samples. The total number of iterations of the algorithm is limited to 60. All the components are sampled at the respective infill sampling location at each iteration. The total number of samples used at termination by each component is $60 + 10 \times$ number of dimensions of the i th component.

The result found by BEGRO is compared against system level optimization of the problem, Equation 7.29, based on component metamodels constructed only using LHS. Once again, due to the non-deterministic sampling, the method is run 20 times. The LHS approach is also allowed the same total computational budget, i.e. the components are initialized with $n_i = 60 + 10 \times$ number of dimensions of the i th component. Finally, the solution is also compared against using component level Efficient Global Robust Optimization (EGRO) [25] on the original 10-dimensional problem, Equation (7.28).

Figure 7.2 shows the average robust optimum of the 20 BEGRO runs based on the optimal location found on $S_{\mathcal{X}}$ at each iteration. The objective value is found on the reference function as a post processing step. The errorbars indicate the standard deviation around the mean value. The reference robust optimum found on the reference function, Equation (7.28), without introducing metamodels is also plotted in Figure 7.2. The value of the reference robust optimum is -0.3866. In general, the approximate system response may also find robust optimal locations that underestimate the actual worst-case cost. Additionally, the solutions found in the beginning may also violate the constraint. The algorithm takes about 40 iterations before convergence is observed. For the first 25 iterations the standard deviation remains fairly constant. Thereafter it starts to drop. After a further 15 iterations, the mean robust optimum corresponds to the reference optimum and the standard deviation has also dropped significantly.

Table 7.1 shows the mean and standard deviation of the robust optimum found using the three methods, BEGRO, LHS and EGRO. As mentioned previously, BEGRO and LHS are allowed the same total computational budget. For both methods, \mathcal{K}_{c_5} , the metamodel with the highest number of samples, is allowed a total of 140 simulations. On

Table 7.1: Table shows the mean and standard deviation of the robust optimum found using the three methods, BEGRO, EGRO and LHS. The reference objective at the robust optimum is also given. The maximum evaluations for the metamodels in each method are also provided.

Optimum	Reference	BEGRO	LHS	EGRO
Mean	-0.3866	-0.397	-2.563	-0.395
Standard deviation	-	0.012	3.644	0.023
Constraint violations	-	0	0	0
Maximum metamodel evaluations	-	140	140	392

the other hand, since EGRO has to be applied directly on the original problem, Equation (7.28), it is allowed a much higher computational budget. The result shown in Table 7.1 for EGRO is based on a metamodel of the objective and constraint in Equation (7.28) built using 392 evaluations [25]. The system metamodel built for EGRO is relatively higher dimensional than the component metamodel for BEGRO and LHS. Therefore, after 140 iterations of EGRO, the solution is still very far from convergence. We allow EGRO to continue running to see how long it takes before reaching a result that is comparable with BEGRO.

Given the same computational budget BEGRO finds a solution that is much closer to the reference robust optimum than the average found via the LHS based approach. The standard deviation for BEGRO is also orders of magnitude lower. The results for the LHS approach seem to indicate that the best worst-case cost was underestimated in general. The fact that the mean solution of LHS is lower than the robust optimum found on the reference function also shows that the solution is quite inaccurate.

The accuracy of the average results of EGRO and BEGRO is almost the same. However, the highest number of component simulations in BEGRO was 140 for c_5 . On the other hand, the result of EGRO is based on 392 evaluations of the objective and constraints, Equation (7.28). The superior performance of BEGRO is to be expected since metamodels are made of low dimensional components. Finally, it is important to note that none of the methods give infeasible solutions. This seems to suggest that the linear constraint response is relatively simple to model given the computational budget.

7.5.2. ENGINEERING PROBLEM: ROBUST OPTIMIZATION OF AN OPTICAL FILTER

SERIAL RING RESONATOR OPERATION

A photonic integrated circuit (PIC) enables the confinement and propagation of light based on the principle of total internal reflection. The light travels within high refractive waveguides that are surrounded by material that has a lower refractive index. In this work, the photonic integrated circuit is based on TripleX based waveguides [26]. The waveguide consists of a high refractive index SiN layer buried in SiO₂ which has a lower refractive index.

The fabrication of the PICs requires subnanometer precision and is prone to manufacturing uncertainties. These uncertainties affect the geometry of the fabricated waveguides, often causing the width and thickness of the waveguide to dilate or shrink. This

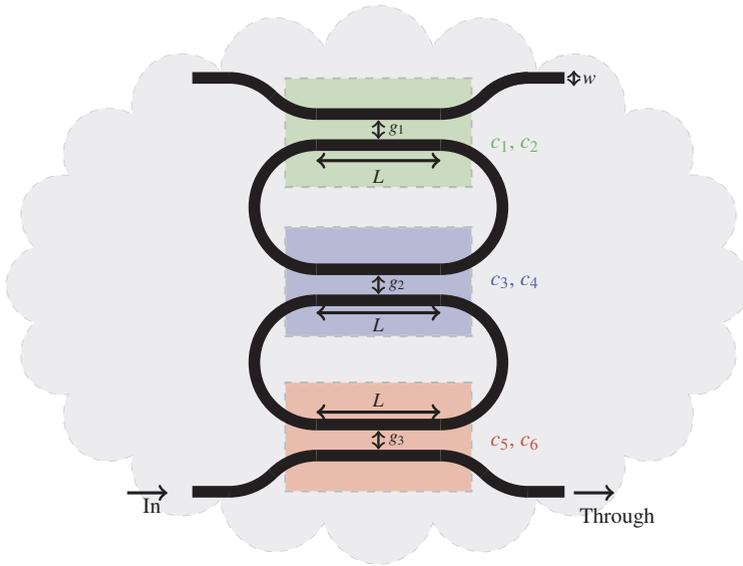


Figure 7.3: Top-view schematic of an integrated optical ring resonator.

change in geometry significantly impacts the optical performance of the PICs.

Amongst the multitude of applications that such photonic integrated circuits have, the optical filtering of light as a function of wavelength is especially important within optical telecommunication. Optical filters can be realized via multi-ring resonators. A schematic of a multi-ring resonators system is shown in Figure 7.3. The black optical paths indicate the waveguides through which light propagates.

When the waveguides are brought in close proximity to one another, any light present can partially or fully couple from one waveguide to the other. These sections are known as directional couplers. These directional couplers are enclosed by colored boxes in the illustration in Figure 7.3. The coupling between adjacent waveguides enables light, launched at the ‘In’ port, to propagate through the multi-ring resonator and partially exit at the Drop port. The remaining light exits at the ‘Through’ port. The coupling is not only a function of the wavelength of the input light but also of the gap and length of the waveguide coupling section.

Figure 7.4 shows the amount of coupled power as a function of the coupler length for a fixed gap between the waveguides in the coupler. The curve follows a sinusoidal trend. The length of waveguide needed for light to couple completely from one waveguide to the other and back to the original waveguide is known as L_π and is indicated in Figure 7.4. Some power is also coupled when the length of the coupling section itself is zero. In this case the light is being coupled by interaction between the adjacent waveguides leading to the coupled section. We refer to this contribution of the waveguide leads by P_{L0} . For a fixed gap, width and thickness, knowledge of the two parameters P_{L0} and L_π is enough to predict the coupled power for any coupler length. This is because the

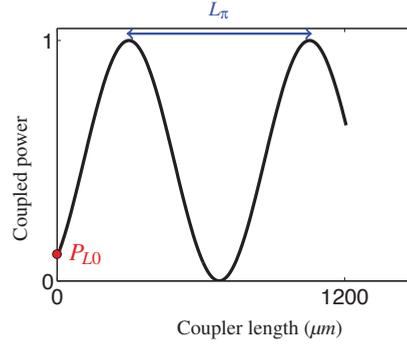


Figure 7.4: Power coupling ratio as a function of the coupler length for a certain coupler gap. This represents the response of the directional coupler section.

exchange of power between the waveguides is sinusoidal.

In this work we are interested in achieving the bandpass filter characteristic shown in Figure 7.5 at the Through port of the serial ring resonator. The optical performance of the resonator can actively be optimized by changing the gap between the waveguides, the width of the waveguides and the length of the coupling section. In this example, we consider the resonator geometry to be symmetric, i.e. $g_3 = g_1$. This means that there are only two variables, g_1 and g_2 , for the gap. The design variables of the problem are $\mathbf{x}_d \in [w \ g_1 \ g_2 \ L]$, where $w \in [1 \ 1.15]\mu m$, $L \in [0 \ 2400]\mu m$ and the gaps $g_1, g_2 \in [1 \ 1.3]\mu m$. The bandpass filter response must remain robust with respect to the parametric uncertainties, i.e. the width and thickness variations that could occur during the fabrication process. The parametric uncertainties $\mathbf{x}_e \in [\Delta w, \Delta t]$ reside in the sets $[-0.1 \ 0.1]\mu m$ and $[-0.003 \ 0.003]\mu m$ respectively.

To find the optimal robust design, BEGRO is also run on the serial ring resonator considering both fabrication uncertainties and simulation error. The calculation of P_{L0} is based on an approximate coupled mode approximation model. This computation is prone to error in simulation. On the other hand, we do not consider simulation error in L_π , since that is a much higher fidelity simulation. The simulation error in the problem is also treated as a parametric uncertainty. We assume a $\Delta f \in \pm 5\%$ uncertainty in the simulation of P_{L0} . The fact that we consider a symmetric structure for the serial ring resonator, in which $g_3 = g_1$, means that there are fewer possible combinations of component level simulation error against which the system has to be robust.

The robust optimization problem for the spectral response $H(n_f)$ at the ‘Through’ port may be written as,

$$\min_{\mathbf{x}_d \in \mathbb{X}_d} \max_{\mathbf{x}_e \in \mathbb{X}_e} \frac{1-b}{2} \|\bar{H}_{stop1}\|_p + b \left[1 - \|1 - \bar{H}_{pass}\|_p \right] + \frac{1-b}{2} \|\bar{H}_{stop2}\|_p, \quad (7.30)$$

where \bar{H}_{stop1} , \bar{H}_{pass} and \bar{H}_{stop2} represent the vector of responses for the normalized frequencies $n_f \in [0 \ 0.1]$, $n_f \in [0.2 \ 0.8]$ and $n_f \in [0.9 \ 1]$, respectively.

The objective is a weighed sum of the approximate maximum response in the stop

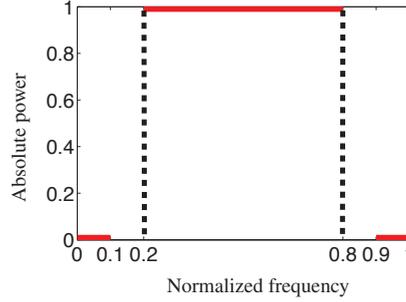


Figure 7.5: Desired band-pass filter spectral response (red line) of the serial ring resonator. No preference is specified in the intervals $[0.1, 0.2]$ and $[0.8, 0.9]$.

bands and the approximate minimum in the pass band. We choose $b = 0.6$ as the weight for the pass band contribution to the objective. The parameter p in the p-norm expression which approximates the maximum is given a value of 20. The p-norm is used instead of simply finding the maximum of the vector in each band so that the computation of the worst-case cost (inner maximization in \mathbb{X}_ρ) takes place on a continuous function. The outer minimization in Equation (7.30) could become nondifferentiable regardless of the choice of the objective itself, so taking the p-norm does not necessarily enable a smoother function for that case.

BEGRO

Computing the response of the directional coupler sections in the serial ring resonator for different geometry parameters requires expensive simulations. The response of each directional coupler, Figure 7.4, depends on the value of P_{L0} and L_π . Computation of P_{L0} and L_π requires expensive electromagnetic simulation for each configuration of the waveguide geometry and gap between waveguides. Figure 7.4 indicates that each directional coupler is built based on two expensive components.

A commercial electromagnetic solver, Phoenix Software [27], is used to simulate the components P_{L0} and L_π . Each simulation requires 10 minutes of computation time. Once these parameters are known for each directional coupler, the ‘Through’ port spectral response can be found via scattering matrix analysis. Details concerning the computation of the system response may be found in [28]. Since this evaluation is inexpensive, it can be performed at system level.

The serial ring resonator shown in Figure 7.3 consists of six components. Components c_1 , c_3 and c_5 give the initial power P_{L0} , while c_2 , c_4 and c_6 provide the beat length L_π for each coupler. Under the assumption that $g_3 = g_1$, the number of unique components reduces to 4, since c_5 and c_6 are then just copies of c_1 and c_2 , respectively.

BEGRO is applied on the serial ring resonator to estimate a robust optimum for the desired bandpass filter response given in Figure 7.5. We first consider the case where only fabrication uncertainties are present while there is no simulation error. Initial Kriging metamodels, \mathcal{K}_{c_1} and \mathcal{K}_{c_2} , are constructed for P_{L0} and L_π based on the design vari-

ables $[w \ g_1]$ and parametric uncertainties $[\Delta w \ \Delta t]$. 60 initial samples are chosen for this purpose using Latin Hypercube sampling. The algorithm is allowed a total computational budget of 180 expensive simulations for P_{L0} and L_π , excluding the initial 60 simulations used for space filling. At each iteration, the infill sampling criterion will provide new sampling locations for $[w \ g_1 \ g_2 \ L]$ and $[\Delta w \ \Delta t]$. This means that P_{L0} and L_π will be simulated twice for the two different combinations, $[w \ g_1 \ \Delta w \ \Delta t]$ and $[w \ g_2 \ \Delta w \ \Delta t]$, at each iteration. Therefore, the algorithm will run for 90 iterations, before the 180 expensive simulations are exhausted. The same setting is also used for robust optimization when considering simulation error as a parametric uncertainty.

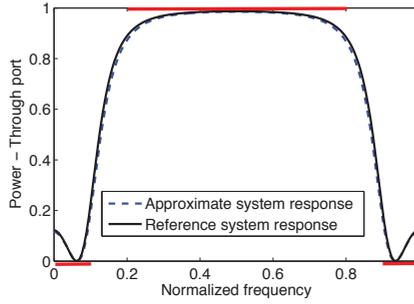


Figure 7.6: The approximate system response at the deterministic optimum is compared to the response found on the reference system based on the expensive simulation of P_{L0} and L_π .

7

\mathcal{K}_{c1} and \mathcal{K}_{c2} can basically be used to estimate the coupled power at a certain length for any directional coupler in the system since the domain of \mathbf{x}_d and \mathbf{x}_e is the same for all the directional couplers. Therefore, separate metamodels are not needed for each instance of the directional coupler in the serial ring resonator system. This also means that the multiple simulations performed at each iteration for P_{L0} and L_π can be used to improve \mathcal{K}_{c1} and \mathcal{K}_{c2} only, instead of constructing separate metamodels. This sit-

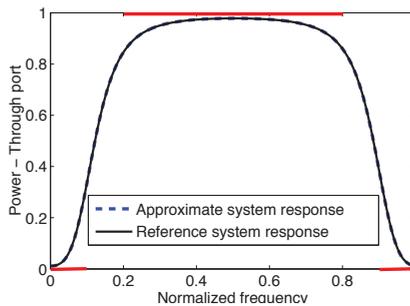


Figure 7.7: The approximate system response and the reference system response is plotted at the nominal location of the robust optimum (without considering simulation error).

uation where similar components are present in a given photonic integrated system is quite common. Therefore, using a single metamodel for multiple components further improves the efficiency.

The result of the application of BEGRO on the serial ring resonator is compared against deterministic optimization of the same problem using BEGO [17]. Since uncertainties are not considered in the deterministic problem, the component metamodels \mathcal{K}_{c1} and \mathcal{K}_{c2} for P_{L0} and L_{π} have to be constructed in the control variable design space \mathbb{X}_d only. The algorithm is therefore allowed a reduced total budget of 60 expensive simulations [17]. The serial ring resonator is considered to be symmetric for the deterministic optimization problem as well. Figure 7.6 shows the performance of the optimal deterministic design, evaluated using the approximate system response based on the Kriging component metamodels as well as with the reference system response. The plot suggests that the component metamodels are fairly accurate at the deterministic optimum so that the approximate and reference system responses are also quite similar. The filter attenuates the power in the stop-band while allowing most of the power to be present in the pass-band. Using a higher order ring resonator would enable even stronger attenuation in the stop-bands and better performance in the pass-bands.

Figure 7.7 shows the same comparison at the nominal location for the robust optimum found by BEGRO. Visually, the approximate and reference system response compare quite well. Once again, this indicates that the component metamodels \mathcal{K}_{c1} and \mathcal{K}_{c2} have a high fidelity in the neighborhood of the robust optimum. Furthermore, this suggests that the metamodels have been effectively sampled in the region of the robust optimum via the infill sampling criterion.

The performance of the deterministic and robust optimum at the respective worst-case location is shown in Figure 7.8. It is interesting to note that there is hardly any rejection of frequencies in the band stop regions for the deterministic optimum. Similarly, the performance in the pass-band has also worsened considerably. The deterioration in performance, when moving from the nominal location, Figure 7.6, to the worst-case location, Figure 7.8, is quite dramatic for the deterministic optimum. The numerical objective is 0.7545, where a value of 1 is theoretically the worst possible objective. In the case of the robust optimum, the filter response has also deteriorated, but not as much as it did for the deterministic optimum. Almost all the light is passed through in the pass-band. The rejection in the stop band is, however, not as strong anymore. The numerical objective is still a much lower value, 0.2226, than the worst-case cost for the deterministic optimum. Table 7.1 summarizes the comparison between the deterministic and the robust solution. The table also provides also provides the location at which the respective solutions were obtained.

BEGRO was also applied on the same problem with simulation error in the components c_2 and c_4 which give the beat length L_{π} for each coupler section. The simulation error in c_6 would be the same as the one in c_2 , since c_6 is a copy of c_2 when $g_3 = g_1$, Figure 7.3. The $\Delta f \in \pm 5\%$ simulation error is also treated as a parametric uncertainty. Even when including simulation error, the same robust optimum, as is given in Table 7.1 was found to be the most robust location. However, the worst-case cost now goes up to 0.246 from 0.2226 due to the inclusion of the uncertainty in simulation.

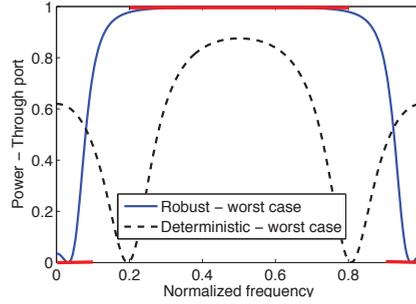


Figure 7.8: The system level response at the deterministic and the robust optimum (without considering simulation error), assuming that the worst-case fabrication takes place.

Table 7.2: The deterministic and robust optimum objective is compared at the nominal and worst-case locations. The values for Δw and Δt give the location at which the worst-case objective was found for each design. The last two columns provide the nominal and worst-case objective.

Optimum	w	g_1	g_2	g_3	L	Δw	Δt	Nominal	Worst-case
Deterministic	1.1250	1.1321	1.1122	1.1321	1897.7	-0.0268	-0.0027	0.0166	0.7545
Robust	1.0731	1.00	1.2715	1.00	130.299	-0.1	-0.003	0.0645	0.2226

7.6. CONCLUSION

We proposed a strategy for global robust optimization of systems involving computationally expensive components, that are independent in the sense that they do not exchange any coupling variables. The method was directed at problems involving bounded-but-unknown uncertainties, i.e. the bounds of the uncertainties were available, however the distribution of the uncertainties is unknown.

The independent component metamodells were constructed using Kriging and the approximate system response was obtained via the transformation of the cheap Kriging metamodells. A system level expected improvement and expected deterioration criterion was developed for the control variable and environment variable space, respectively. These criteria were based on a system level error estimate which was obtained via a linear system transformation of the respective errors of the components.

Since the component metamodells will usually have a much smaller dimension than the total number of dimensions of the system, only a few component simulations are needed to construct high fidelity surrogate responses. In addition, if the system has many components that are the same, a single metamodell can be used as a cheap replacement for all of them. These aspects mean that the proposed strategy can be used for optimizing systems with relatively large problem sizes.

The approach was run 20 times on a constrained benchmark problem with different initial samples found via LHS. The adaptive sampling scheme was allowed to run for 60 iterations on the problem. The average optimum found on the approximate system response corresponded well with the reference solution at termination. Significantly, the proposed method performed much better than the conventional strategy of con-

structuring a metamodel of the system and applying robust optimization on the resulting response. This was demonstrated by comparing the numerical results of the technique with Efficient Global Robust Optimization [16].

An engineering case study of a second order serial ring resonator affected by manufacturing uncertainties was employed to demonstrate the applicability of the technique in a practical setting. The objective was to achieve a certain bandpass optical filter performance. The optical performance was very sensitive to the variations in geometry caused by the imperfect fabrication process. The system could be decomposed into six components of which only two were unique. BEGRO was applied on this system and it was shown that the proposed adaptive sampling scheme enabled the algorithm to quickly reach the system robust optimum.

Since the method is based on the assumption that the component metamodels are constructed using black-box expensive responses, there is no restriction on the type of problem that can be tackled by the algorithm. The only caveat is that the system should involve independent low dimensional component models. The proposed method can be employed for robust optimization of this particular class of systems. For this purpose, pre-built libraries of initial component metamodels could be employed. The proposed optimal infill sampling criteria can then be used to refine the component models in relevant regions for system robust optimization for each specific design problem.

REFERENCES

- [1] J. R. R. A. Martins and A. B. Lambe, *Multidisciplinary Design Optimization: A Survey of Architectures*, AIAA Journal **51**, 2049 (2013).
- [2] W. Yao, X. Chen, W. Luo, M. van Tooren, and J. Guo, *Review of uncertainty-based multidisciplinary design optimization methods for aerospace vehicles*, Progress in Aerospace Sciences **47**, 450 (2011).
- [3] W. Hu, S. Azarm, and A. Almansoori, *New Approximation Assisted Multi-objective collaborative Robust Optimization (new AA-McRO) under interval uncertainty*, Structural and Multidisciplinary Optimization **47**, 19 (2012).
- [4] X. (Stacey) Gu, J. E. Renaud, and C. L. Penninger, *Implicit Uncertainty Propagation for Robust Collaborative Optimization*, Journal of Mechanical Design **128**, 1001 (2006).
- [5] M. Kokkolaras, Z. P. Mourelatos, and P. Y. Papalambros, *Design Optimization of Hierarchically Decomposed Multilevel Systems Under Uncertainty*, Journal of Mechanical Design **128**, 503 (2005).
- [6] A. Takagi, K. Jinguji, and M. Kawachi, *Wavelength characteristics of (2 times;2) optical channel-type directional couplers with symmetric or nonsymmetric coupling structures*, Lightwave Technology, Journal of **10**, 735 (1992).
- [7] L. B. Soldano and E. C. M. Pennings, *Optical multi-mode interference devices based on self-imaging: principles and applications*, Lightwave Technology, Journal of **13**, 615 (1995).

- [8] C. Madsen, *General IIR optical filter design for WDM applications using all-pass filters*, Journal of lightwave technology **18**, 860 (2000).
- [9] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust optimization* (Princeton University Press, 2009).
- [10] F. A. C. Viana, T. W. Simpson, V. Balabanov, and V. Toropov, *Metamodeling in Multidisciplinary Design Optimization: How Far Have We Really Come?* AIAA Journal **52**, 670 (2014).
- [11] T. Simpson, A. Booker, D. Ghosh, A. Giunta, P. Koch, and R.-J. Yang, *Approximation methods in multidisciplinary analysis and optimization: a panel discussion*, Structural and Multidisciplinary Optimization **27**, 302 (2004).
- [12] P. M. Zadeh, V. V. Toropov, and A. S. Wood, *Metamodel-based collaborative optimization framework*, Structural and Multidisciplinary Optimization **38**, 103 (2008).
- [13] W. Hu, M. Li, S. Azarm, and A. Almansoori, *Multi-Objective Robust Optimization Under Interval Uncertainty Using Online Approximation and Constraint Cuts*, Journal of Mechanical Design **133**, 061002 (2011).
- [14] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, *Design and analysis of computer experiments*, Statist. Sci. **4**, 409 (1989).
- [15] D. R. Jones, M. Schonlau, and W. J. Welch, *Efficient Global Optimization of Expensive Black-Box Functions*, Journal of Global Optimization **13**, 455 (1998).
- [16] S. Ur Rehman and M. Langelaar, *Efficient global robust optimization of unconstrained problems affected by parametric uncertainties*, Structural and Multidisciplinary Optimization, **1** (2015).
- [17] S. Ur Rehman and M. Langelaar, *Adaptive efficient global optimization of systems with independent components*, Structural and Multidisciplinary Optimization (submitted,).
- [18] A. I. J. Forrester and A. J. Keane, *Recent advances in surrogate-based optimization*, Progress in Aerospace Sciences **45**, 50 (2009).
- [19] G. G. Wang and S. Shan, *Review of Metamodeling Techniques in Support of Engineering Design Optimization*, Journal of Mechanical Design **129**, 370 (2007).
- [20] A. Forrester, S. Andras, and A. J. Keane, *Engineering design via surrogate modelling : a practical guide* (J. Wiley, 2008).
- [21] M. Schonlau and W. Welch, *Global optimization with nonparametric function fitting*, in *Proceedings of the Section on Physical and Engineering Sciences* (American Statistical Association, 1996) pp. 183–186.
- [22] B. M. Ayyub, *Risk Analysis in Engineering and Economics* (Chapman & Hall/CRC, Boca Raton, FL, 2003).

- [23] M. Schonlau, *Computer experiments and global optimization*, Ph.D. thesis, University of Waterloo (1997).
- [24] J. Parr, A. Forrester, A. J. Keane, and C. M. Holden, *Enhancing infill sampling criteria for surrogate-based constrained optimization*, Journal of Computational Methods in Sciences and Engineering **12**, 25 (2012).
- [25] S. Ur Rehman and M. Langelaar, *Expected improvement based infill sampling for global robust optimization of constrained problems*, Structural and Multidisciplinary Optimization (submitted,).
- [26] R. Heideman, M. Hoekman, and E. Schreuder, *TriPleX-Based Integrated Optical Ring Resonators for Lab-on-a-Chip and Environmental Detection*, Selected Topics in Quantum Electronics, IEEE Journal of **18**, 1583 (2012).
- [27] PhoeniX Software, *PhoeniX Software 4.8.4*, (2014).
- [28] O. S. Ahmed, M. A. Swillam, M. H. Bakr, and X. Li, *Efficient Design Optimization of Ring Resonator-Based Optical Filters*, Lightwave Technology, Journal of **29**, 2812 (2011).

III

INTEGRATED PHOTONICS

8

ROBUST OPTIMIZATION FOR INTEGRATED PHOTONIC SYSTEMS

8.1. INTRODUCTION

Integrated photonic devices and systems are prone to manufacturing uncertainties which are an unavoidable aspect of fabrication. If designers do not account for the geometrical variations that can arise in fabrication, the fabricated structure fails to perform according to the designed specifications. Design-for-Manufacturing (DfM) strategies for integrated photonics therefore have a potential to increase the overall yield and simultaneously reduce the cost of production. However, in order to perform this, information about the capability of the fabrication process is needed. Ideally, designers should have access to data related to the probability distribution of the uncertainties in fabrication. However, such probability data is usually classified and is not disclosed by foundries to external designers. In this case, designers often only know the tolerances of the fabrication process. In other words, the bounds on the fabrication uncertainties are known, but their distribution is unknown.

In the scenario that the uncertainties are bounded-but-unknown [1], robust optimization is an established approach to find a fault-tolerant design. Robust optimization involves finding the best worst-case performance. The design is optimized so that the best performance is achieved given that the worst-case uncertainty with respect to the performance metric is realized. The design found using this method is therefore not insensitive, but has a certain guaranteed minimum performance.

To determine the robust optimum, an iterative optimization process is required. An additional challenge in integrated photonic optimization is that the underlying electromagnetic simulation may be computationally expensive. Repeatedly changing the design parameters and rerunning the simulation to find the optimal design can therefore be prohibitively costly. In order to circumvent this problem, an inexpensive approximate model of the simulation can be constructed and the optimization can be performed on the cheap model. Amongst the available methods for mathematical modeling, Kriging

[2] is a strong candidate since it provides an estimator for the approximation error. Using these estimates, the cheap model, otherwise known as a metamodel, can adaptively be improved by simulating the integrated photonic device response in regions of the design domain that are relevant to robust optimization.

The described approximation approach can efficiently find the robust optimum of an integrated photonic device such as an MMI coupler [3, 4] or a single ring resonator [5]. But in order for the approach to be scalable it should also be able to produce a robust solution for large integrated photonic systems consisting of different components.

Research has been performed on finding tolerant designs for different integrated photonic devices [6–11]. However, most of these approaches have been focused on non-generic methods that only address a particular integrated photonic device. A scalable and generic approach for robust optimization of integrated photonic systems is still lacking.

In this work, we propose a system level robust optimization technique for efficiently identifying robust designs for integrated photonic systems. A cheap system model is constructed for this purpose based on mathematical models of the components. Since the approach is not based on a specific model, a robust optimum for any integrated photonic component or system can readily be found without altering the underlying method. The only restriction is that the structure of the system should be such, that the behavior of the components is independent from one another. This means that e.g. heaters that cause crosstalk between components cannot be included. Fortunately, for the majority of integrated photonic systems, this condition is met. The robust optimum found on the cheap system model should match the result on the reference simulation. To ensure this, the system response is iteratively improved by simulating the underlying components, using a combination of the system level error estimate and the predicted response, in areas that could potentially contain the system robust optimum. We employ a sound mathematical criterion to select the best locations in the design space for refinement, in order to minimize the computational effort of the process.

8

The proposed approach is suited to problems for which the system simulation is cheap and the component behavior is simpler to approximate than the system response. Systems with multiple identical components are especially strong candidates since a single metamodel can then replace the components. Once metamodels have been built for the components, the system is arbitrarily scalable at low computational cost. A library of pre-built component models (the initial samples used here) could be provided in a software package, or built by the user. These pre-built models only need to be refined for each specific case.

We showcase the method by performing robust optimization of optical filters, that can be seen as examples of integrated photonic systems consisting of several components. Second order and third order serial ring resonators based on single stripe TripleX technology are used for this purpose [12]. Kriging metamodels of the directional coupler sections of the resonators are constructed since simulating the directional coupler is computationally expensive. The suitability of the approach is demonstrated by comparing the robust solution found with the deterministic optimum, i.e. the optimum achieved when optimizing without taking fabrication uncertainties into account.

There has been previous work on optimization of ring resonators based optical filters

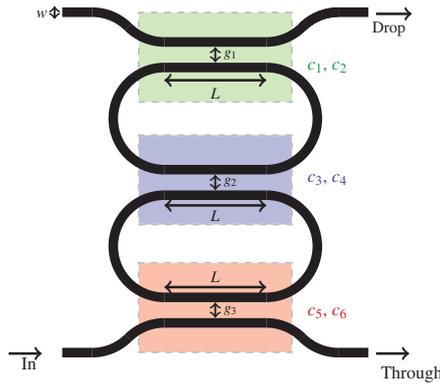


Figure 8.1: A second order serial ring resonator is illustrated. The width w , the vector of gaps \mathbf{g} and the length L are the design variables of the problem. The variations in width Δw and in thickness, Δt are the uncertainties with respect to which the design has to be robust.

[13–16]. Different approaches have been used for optimization. In [13], the placement of poles and zeros of the transfer function is optimized via trial and error. In [14], a perturbation based approach is employed to vary known mean coupling ratios in order to find the optimal design. However, these methods optimize the filter performance as a function of the coupling ratio of each directional coupler in the system. Optimization is not performed with respect to the geometrical parameters. Uncertainties in the geometry due to fabrication variations are therefore also not taken into account. In the present work, the filter is optimized directly as a function of the geometry, meanwhile the robustness with respect to the variations in geometry is also ensured. We have chosen serial ring resonators merely as demonstrators of the application of the proposed approach. The method can readily be applied on other integrated photonic systems as well as on a large class of engineering systems in general. For instance, once component metamodels are available for expensive to evaluate devices such as directional couplers, MMI couplers, large systems such as interferometers or optical add drop multiplexers consisting of many rings can be robustly optimized at low computational cost.

8.2. APPLICATION: SERIAL RING RESONATORS

In this work, we are interested in performing robust optimization of optical filters based on serial ring resonators. Fig.8.1 shows an illustration of a second-order serial ring resonator. The serial ring resonators are simulated using a single stripe TripleX waveguide [12] with designed thickness of 32nm. The directional couplers are extremely sensitive to variation at this thickness. This means that if the nominal performance is optimized then even slight variations in the geometry can cause the designed device to not operate as expected.

The design variables of the problem are the gaps, g_1 to g_n between the n directional couplers, the width of the waveguides and the length L of the directional couplers. The width $w \in [11.15]\mu m$, the gaps $g_1, g_2 \dots g_n \in [1 \ 1.3]\mu m$ and the length $L \in [0 \ 2400]\mu m$.

The width range is chosen such that the waveguide always remains single mode. The width and thickness variations caused by the imperfect fabrication process are denoted by $[\Delta w, \Delta t]$. For this process $\Delta w \in [-0.1 \ 0.1] \mu m$ and $\Delta t \in [-3 \ 3] nm$. The set of design variables (control variables) is denoted by \mathbf{x}_d , while the set of parametric uncertainties (environment variables) is represented by \mathbf{x}_e .

The filter performance should be robust with respect to the parametric uncertainties which impact the cross-sectional geometry, i.e. width and thickness variation. This involves finding the right combination of the design variables that leads to the most robust design.

Computing the response at the Through or Drop port basically involves simple linear algebra and matrix manipulation once the power coupling ratio is known for each coupler section [14]. Let P_{L0} represent the power coupling ratio when $L = 0 \mu m$. We denote the beat length by L_π . Computing the power coupling ratio of a directional coupler, given a length L and a certain geometry for the cross-section, can be time consuming, as computation of P_{L0} and L_π requires numerical simulation. A commercial electromagnetic solver, Phoenix Software [17], is used to simulate both quantities. A coupled mode theory model is employed to simulate P_{L0} . On the other hand, L_π is found using a mode solver. Both simulations require approximately 10 minutes.

Once P_{L0} and L_π are known for a given geometry, the power coupling ratio for any length L is cheap to compute. The scattering matrix analysis that follows, in order to find the serial ring resonator response is also not computationally expensive.

We therefore make a clear distinction between the computationally expensive and cheap parts of the system. We construct metamodels of the expensive components, i.e. response of P_{L0} and L_π , given the design variables and the parametric uncertainties. The power coupling ratio given by the combination of the cheap models is then used as an input to the scattering matrix analysis in order to get the system response for the serial ring resonator.

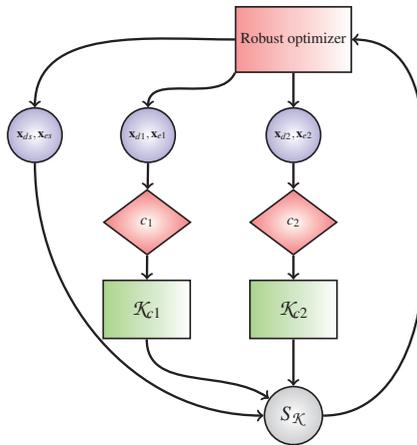


Figure 8.2: The process of robust optimization of the approximate system response based on Kriging models of expensive components is shown.

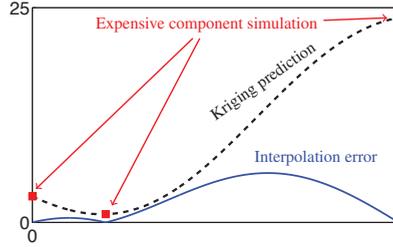


Figure 8.3: Kriging model of a one-dimensional function based on three samples. The predicted Kriging mean squared error is also shown in the plot. As expected, the predicted error is zero at the sample points.

Robust optimization can then be efficiently applied on the approximate system response. The robust optimum should converge to the solution that would have been found on the reference simulators. This convergence requires improvement of the cheap system response by adding more data points from the expensive simulation in strategically important regions until an initially specified budget for total simulations is exhausted. In what follows, we expand upon the robust optimization method and the proposed approach for adaptively improving the system response.

8.3. SYSTEM ROBUST OPTIMIZATION

Let $S_{\mathcal{K}}$ represent a system based on components c_1 to $c_{\mathcal{N}}$. Since the components are expensive to simulate, we construct Kriging metamodels \mathcal{K}_{c_1} to $\mathcal{K}_{c_{\mathcal{N}}}$ of the components based on a set of simulated responses. Robust optimization is applied on the approximate system response generated from the underlying Kriging metamodels.

Fig. 8.2 visually depicts the relationship between the design variables, \mathbf{x}_d , the parametric uncertainties, \mathbf{x}_e , and the components, system response. Since we construct metamodels for only P_{L0} and L_{π} , Fig. 8.2 shows only two component metamodels \mathcal{K}_{c_1} and \mathcal{K}_{c_2} . Once we have the cheap models for P_{L0} and L_{π} as a function of width, gap and thickness, the response for all the directional couplers in the serial ring resonator can be found since they share the same domain in the design variables $[w, g, t]$ and the uncertainties $[\Delta w, \Delta t]$. Fig. 8.2 shows that some variables and uncertainties ($\mathbf{x}_{c_s}, \mathbf{x}_{e_s}$) can directly impact the system response $S_{\mathcal{K}}$. For the serial ring resonator problem, the length L is a system level design variable since it does not impact the response of P_{L0} and L_{π} , but it has an influence on the system response.

The robust optimizer operates on the system response $S_{\mathcal{K}}$ and tries to find a relatively insensitive solution by optimizing the design variables. The system level robust optimization problem in general may be expressed as,

$$\min_{\mathbf{x}_d \in \mathbb{X}_d} \max_{\mathbf{x}_e \in \mathbb{X}_e} S_{\mathcal{K}}(\mathcal{K}_{c_1}(\mathbf{x}_{d1}, \mathbf{x}_{e1}), \mathcal{K}_{c_2}(\mathbf{x}_{d2}, \mathbf{x}_{e2}), \dots, \mathcal{K}_{c_{\mathcal{N}}}(\mathbf{x}_{d_{\mathcal{N}}}, \mathbf{x}_{e_{\mathcal{N}}}), \mathbf{x}_{d_s}, \mathbf{x}_{e_s}). \quad (8.1)$$

where \mathbf{x}_{d1} to $\mathbf{x}_{d_{\mathcal{N}}}$ are the design variables of component metamodel \mathcal{K}_{c_1} to $\mathcal{K}_{c_{\mathcal{N}}}$. The

parametric uncertainties \mathbf{x}_{e1} to \mathbf{x}_{eN} affect the component metamodels \mathcal{K}_{c1} to \mathcal{K}_{cN} , respectively. The design variables \mathbf{x}_{ds} and the parametric uncertainties \mathbf{x}_{es} directly affect the system response, Fig. 8.2. \mathbb{X}_d and \mathbb{X}_e are the domains for \mathbf{x}_d and \mathbf{x}_e , respectively. Equation (8.1) shows that the robust optimization problem is a nested optimization problem where the objective of the outer minimization itself involves an inner global maximization. This fact means that the efficient use of metamodeling techniques is essential to determine robust designs at affordable computational costs.

8.4. ADAPTIVE IMPROVEMENT OF APPROXIMATE SYSTEM

8.4.1. COMPONENT METAMODELS: KRIGING

Kriging is an interpolation technique with a statistical basis [2]. An important property of Kriging is that it provides an estimate for the interpolation error. Fig. 8.3 shows a Kriging metamodel of a one-dimensional function based on three samples of a reference function. The black dashed line is the predicted Kriging interpolation \hat{y} . The figure also shows the predicted interpolation error, s^2 , given by the solid blue line. The interpolation error is zero at the sample points and increases as the distance between the sample points increases.

The combination of the Kriging prediction \hat{y} and the interpolation error, s^2 , can be used to iteratively improve the metamodel so that the minimum of the expensive function is found efficiently. Jones *et al.* [18] devised such a method for adaptively improving the metamodel in regions of interest for optimization. The method assumes that the metamodel uncertainty in the response, \hat{y} , at any position \mathbf{x} in the domain can be modeled as a normal random variable with mean \hat{y} and variance s^2 , Fig. 8.4.

Fig. 8.5 shows this random variable superimposed on the Kriging prediction curve. The area shaded in pink quantifies the predicted probability of improvement over the current observed minimum, y_{min} , if an expensive simulation is performed for that location. If we take the first moment of area of the shaded region, we get the *expected* improvement over y_{min} . By maximizing the expected improvement (EI) criterion for the whole domain, a sampling location is found that provides the highest predicted improvement over y_{min} . Performing EI maximization over several iterations, with a new simulation point corresponding to the maximum EI value added at each iteration, enables the global minimum to be found efficiently.

8.4.2. SYSTEM LEVEL ROBUST EXPECTED IMPROVEMENT

The authors extended the EGO approach suggested by Jones *et al.* to the system level [19]. We proposed an approach for robust optimization of a system based on component metamodels, and verified it on different problems. A system level robust expected improvement criterion was derived which enabled iterative sampling of the expensive components such that the system robust optimum was found efficiently. Here we summarize the main steps of the method, for detailed derivation the reader is referred to [19].

To derive the system level robust EI criterion, a system level error estimate in the approximate system response $S_{\mathcal{N}}$ is needed. In order to find a system level error estimator s_{sys} , a linear Taylor series expansion of $S_{\mathcal{N}}$ was performed.

Let $r_{\mathcal{K}}^{sys}$ represent the best worst-case cost on the system response, determined using Equation (8.1). To improve over $r_{\mathcal{K}}^{sys}$ a location is sought that could potentially have a lower *worst-case* cost. Let $\hat{y}_{sys}^{max}(\mathbf{x}_d)$ represent the worst-case cost for a given value of \mathbf{x}_d ,

$$\hat{y}_{sys}^{max}(\mathbf{x}_d) = \max_{\mathbf{x}_e \in \mathbb{X}_e} S_{\mathcal{K}}. \quad (8.2)$$

The corresponding location in \mathbb{X}_e where the worst-case cost is obtained is given by \mathbf{x}_e^{max} .

The derived system level error estimator s_{sys} was used in combination with the system response, \hat{y}_{sys} , to give infill sampling criteria in the design variable range \mathbb{X}_d and parametric uncertainties range, \mathbb{X}_e . A system level robust expected improvement criterion was developed in \mathbb{X}_d to suggest locations with the highest expectation of improving over the current robust optimum $r_{\mathcal{K}}^{sys}$.

$$EI_d(\mathbf{x}_d) = (r_{\mathcal{K}}^{sys} - \hat{y}_{sys}^{max})\Phi\left(\frac{r_{\mathcal{K}}^{sys} - \hat{y}_{sys}^{max}}{s_{sys}^{max}}\right) + s_{sys}^{max}\phi\left(\frac{r_{\mathcal{K}}^{sys} - \hat{y}_{sys}^{max}}{s_{sys}^{max}}\right). \quad (8.3)$$

On the other had, a system level worst-case expected deterioration criterion was developed for the parametric uncertainty space \mathbb{X}_e which suggested locations with the highest expectation of deterioration in the worst-case system response at \mathbf{x}_d^{new} .

$$EI_e(\mathbf{x}_d^{new}, \mathbf{x}) = (\hat{y}_{sys} - g_{\mathcal{K}}^{sys})\Phi\left(\frac{\hat{y}_{sys} - g_{\mathcal{K}}^{sys}}{s_{sys}}\right) + s_{sys}\phi\left(\frac{\hat{y}_{sys} - g_{\mathcal{K}}^{sys}}{s_{sys}}\right). \quad (8.4)$$

The combination of EI_d and EI_e can be used to suggest a sampling location in \mathbb{X}_d and \mathbb{X}_e , respectively. To do this, the maximum for EI_d and EI_e in the respective domains \mathbb{X}_d and \mathbb{X}_e is found. This is the location at which the response is evaluated on the expensive simulation. New component metamodels are constructed with the augmented set of samples and responses. The process of maximizing EI_d , EI_e and sampling the expensive simulation is repeated until the total number of expensive simulations are exhausted. At this point, the location for the robust optimum, $r_{\mathcal{K}}$, found at the last iteration is returned as the final solution. Details related to the derivation and the actual algorithm may be found in [19].

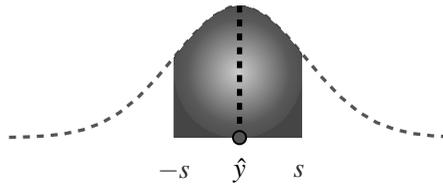


Figure 8.4: An example of a normally distributed random variable which models the uncertainty in the Kriging prediction \hat{y} for a given location x . The variance of the random variable is given by the Kriging mean squared error s^2 .

8.5. RESULTS

The algorithm is demonstrated on second order and third order TripleX based ring resonators. The objective is a bandpass filter response at the Through port. Let $H(n_f)$ represent the spectral response at the Through port. We normalize the frequency with respect to the free spectral range of the serial ring resonator. For the normalized frequency $n_f \in [0, 1]$, the aim is to achieve complete rejection in the stop-bands range $[0, 0.1]$, $[0.9, 1]$ and allow power to pass in the pass-band range $[0.2, 0.8]$. Strictly, a bandpass filter should ideally pass all frequencies in a certain range and reject frequencies outside that range. However, since we are considering only low (second and third) order filters in this work, the frequency ranges $[0.1, 0.2]$ and $[0.8, 0.9]$ are reserved for the slow roll-off.

The robust optimization problem may be written as,

$$\min_{w, g, L, \Delta w, \Delta t} \max \frac{1-b}{2} \|\tilde{H}_{stop1}\|_p + \quad (8.5)$$

$$b \left[1 - \|1 - \tilde{H}_{pass}\|_p \right] + \frac{1-b}{2} \|\tilde{H}_{stop2}\|_p,$$

where \tilde{H}_{stop1} , \tilde{H}_{pass} and \tilde{H}_{stop2} represent the vector of responses for the normalized frequencies $n_f \in [0, 0.1]$, $n_f \in [0.2, 0.8]$ and $n_f \in [0.9, 1]$, respectively. We take the p -norm of the vector of responses, \tilde{H}_{stop1} and \tilde{H}_{stop2} , in the stop bands. The p -norm approximates the maximum value for \tilde{H}_{stop1} and \tilde{H}_{stop2} in the respective stop band ranges. For the pass band, the p -norm is used to approximate the minimum value for \tilde{H}_{pass} in $n_f \in [0.2, 0.8]$. The sum found is dependent on the weight b . In this work, we choose $b = 0.6$ and $p = 20$. The objective in Equation (8.5) is basically a weighed sum of the approximate maximum in \tilde{H}_{stop1} , \tilde{H}_{stop2} and the approximate minimum in \tilde{H}_{pass} . The robust optimization involves finding the best worst-case cost of this weighed sum.

The robust optimum is compared against the optimal solution found when the uncertainties are not part of the optimization problem. Equation (8.6) shows the nominal

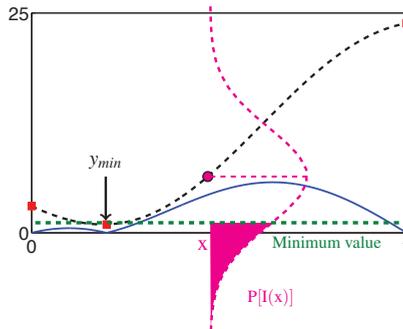


Figure 8.5: The probability of improvement over the minimum observed response y_{min} is shown for a certain location in the design domain.

optimization problem definition,

$$\min_{w, \mathbf{g}, L} \frac{1-b}{2} \|\bar{H}_{stop1}\|_p + b \left[1 - \|1 - \bar{H}_{pass}\|_p \right] + \frac{1-b}{2} \|\bar{H}_{stop2}\|_p. \quad (8.6)$$

In the above problem the weighed sum is simply minimized with respect to the design variables w, \mathbf{g}, L without considering the impact of the uncertainties.

The algorithm is demonstrated by applying it on a second order and third order serial ring resonator. The robust solution is compared against the deterministic optimum. The optimal locations found on the cheap system response are also fed into the expensive electromagnetic simulators as a post-processing step in order to verify the fidelity of the solution.

For deterministic optimization, it was assumed that the ring resonator structure is symmetric. This means that in the case of second order resonator $g_3 = g_1$. Similarly, for the third order resonator, $g_4 = g_1$ and $g_3 = g_2$. For robust optimization both the cases, one assuming symmetry and another without symmetry of the gaps, were considered. It was found that for both cases, the best worst-case objective obtained was relatively the same. Therefore, the greater flexibility of choosing unsymmetrical gap values does not automatically lead to a greater chance of a better solution. In this scenario, it makes sense to perform robust optimization using symmetric gaps, since this reduces the total number of design variables in the problem. In this work, the robust optimization results shown are based on symmetric resonators.

8.5.1. SECOND ORDER SERIAL RING RESONATOR

Robust optimization is applied on the cheap system response of the second order resonator. The approximate response is generated by applying scattering matrix analysis on the power coupling ratio for each directional coupler found via the component metamodels for P_{L0} and L_π . The robust optimization algorithm is started by constructing the initial component metamodels for P_{L0} and L_π . The metamodels are built based on 60 initial expensive simulations of the couple mode theory model (P_{L0}) and the mode solver (L_π). The locations for the design variables w, g and the uncertainties $[\Delta w \ \Delta t]$ is chosen in the combined design variable and uncertainties space. The initial locations are chosen via Latin Hypercube sampling (LHS) [20], a type of Design of Experiments. Since L is a system level design variable, it does not have to be sampled.

The algorithm is allowed a total computational budget of 240 expensive simulations for both P_{L0} and L_π . This means the method can run for 60 iterations, since three such simulations are run at each iteration for the three different gaps g_1, g_2 and g_3 .

A system level deterministic optimization algorithm [21] is applied on the problem for comparison with the robust solution. The approach is also based on adaptive improvement of component metamodels. Since uncertainties are not included in the problem definition in the deterministic case, the total number of variables is only limited to the design variables w, \mathbf{g} and L . A total computational budget of 60 expensive simulations is available. The initial metamodels for P_{L0} and L_π are constructed based on 10 locations for w and g chosen via LHS. Note that due to the lower dimensionality of the

Table 8.1: A comparison of the robust and nominal optima for the second order and the third order filters is given.

Optimum	w	g_1	g_2	g_3	g_4	L	Δw	Δt	Nominal	Worst-case
Nominal second order	1.1250	1.1321	1.1122	1.1321		1897.7	-0.0268	-0.0027	0.0166	0.7545
Robust second order	1.0731	1.00	1.2715	1.00		130.299	-0.0168	0.003	0.0645	0.2274
Nominal third order	1.0746	1.1583	1.1796	1.1796	1.1583	2390.9	0.1	0.0027	0.0089	0.8975
Robust third order	1.1356	1.0190	1.2990	1.2990	1.0190	230.573	0.1	-0.003	0.0255	0.1510

deterministic problem, fewer samples are needed compared to the robust case.

The approximate system response based on the component metamodels for P_{L0} and L_π is plotted in Fig. 8.6 at the deterministic optimum. The system response at the deterministic optimum based on simulation of P_{L0} and L_π on the actual simulator, Phoenix Software [17], is also plotted. As expected, the approximate system response is quite close to the reference solution.

The same comparison is plotted for the robust optimum at the nominal location. Once again, the solution found on the actual simulator is quite similar to the approximate system response. This shows that the component metamodels predict P_{L0} and L_π with high fidelity in the neighborhood of the robust optimum. Comparing Fig. 8.7 with Fig. 8.6 it may appear that the robust solution is a better solution at the nominal location than the deterministic optimum in Fig. 8.6. However, the numerical objective value for the deterministic optimum is lower than it is for the robust optimum since the highest value in both the stop bands is lower for the deterministic optimum than the corresponding highest value in the stop bands for the robust solution.

Fig. 8.8 shows the comparison of the deterministic (dashed black line) and the robust optimal solution (solid blue line) assuming that the worst-case fabricated structure is realized. The ideal band-pass response is indicated in red. The figure shows that, for the worst possible changes in ΔW and Δt , the filter performance for the deterministic optimum deteriorates dramatically. A significant portion of light is passing through in the stop bands and there is very little attenuation. Although, the filter still passes some light in the pass band, the performance is significantly worse compared to the performance at the nominal location, Fig. 8.6.

In comparison, the worst-case solution for the robust filter (solid blue line) gives much better performance in the pass band, since all the light is allowed to pass in the range of frequencies between $n_f \in [0.2 \ 0.8]$. The filter performance could be better since the frequencies in the stop band are not completely attenuated. The slow roll off means that a large amount of light is still being passed through in the regions of the stop bands that are closer to the pass band. However, it should be stressed that this is the worst possible filter performance that can be realized at the robust optimum assuming that structure is fabricated in a way that is most detrimental to the filter performance. For any other fabrication error in thickness and width, the performance would be better than the solution provided in the figure.

8.5.2. THIRD ORDER SERIAL RING RESONATOR

The deterministic and robust optimization algorithms are applied on a third order resonator as well. The same computational budget is allocated for both problems as was

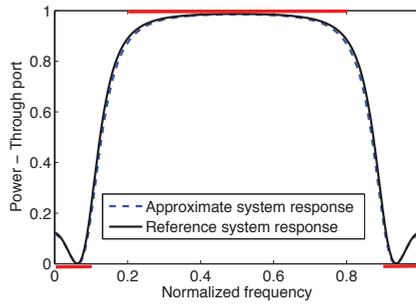


Figure 8.6: Comparison of the approximate system response and the reference system response is shown for the solution obtained by the deterministic optimization algorithm.

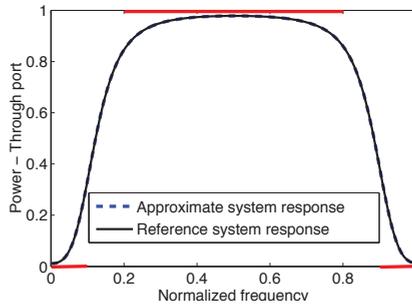


Figure 8.7: Comparison of the approximate system response and the reference system response is shown at the nominal location of the robust optimum.

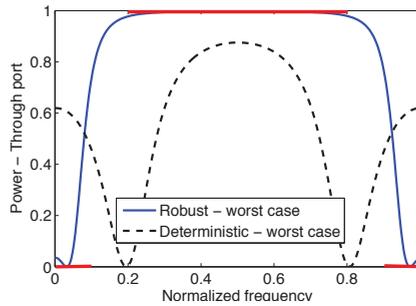


Figure 8.8: Spectral response at the Through port of the second order serial ring resonator for the deterministic and the robust optimum, assuming that the worst-case fabrication error is realized.

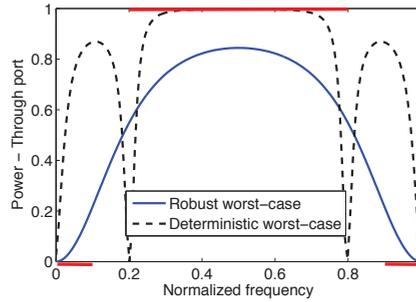


Figure 8.9: Spectral response at the Through port of the third order serial ring resonator for the deterministic and the robust optimum, assuming that the worst-case fabrication error is realized.

used for the second order resonator problem. We do not need to increase the computational budget since the underlying component metamodels are made for a single directional coupler. That directional coupler response can be reused for all the directional couplers in the system since all the couplers share the same design variables and uncertainties domain. The order of the resonator can therefore be increased arbitrarily without incurring high computational costs. This scalability at low cost is one of the primary attractions of the system based approach described in this work.

Fig. 8.9 compares the worst-case filter performance for the deterministic and robust optimum. There is hardly any rejection of frequencies in the stop bands for the deterministic optimum (dashed black line). The pass band performance is significantly better than the deterministic optimum for the second order resonator, Fig. 8.8. On the other hand, the worst-case filter response for the robust optimum shows much better attenuation of the light in the stop band. The performance for the robust optimum in the stop bands is also much better than the corresponding result for the robust optimum on the second order ring resonator, Fig. 8.8. However, the pass-band performance of the filter for the robust solution is far from ideal since quite a lot of power is lost.

Table 8.1 shows a numerical comparison of the second order and third order nominal and robust designs. The optimal design variable locations for w , g , L are given in Column 2 to Column 7. Column 8 and 9 provide the location for the fabrication uncertainties $[\Delta w \Delta t]$ at which the worst-case filter performance is found for the different optima. The last two columns give the numerical performance at the nominal and the worst-case location for the second and third order nominal and robust optimal solutions.

Turning our attention to the objective value at the nominal location, second last column in Table 8.1, we note that the nominal optimum provides a better (lower) solution for both the second and the third order resonators than the robust optimum. However, if the worst possible fabrication with respect to the objective were to occur, then the robust optimal solution deteriorates much less than the nominal solution for both the second and the third order ring resonators, last column. This indicates that, even if the robust optimum is nominally suboptimal, it performs much better in the worst-case than the nominal solution. As expected, the numerical solution for the robust optimum of the

third order filter is better than the robust solution for the second order filter. If higher order filters were robustly optimized, the best worst-case filter performance could further improve. Note that the same cannot be said for the deterministic optimum.

Columns 8 and 9 show the value for $[\Delta w \Delta t]$ at which the worst-case response was found. Apart from the worst-case location for the robust optimum of the third order ring resonator, all the other worst-case locations occur in the interior of the uncertainty set.

8.6. CONCLUSION

A robust optimization framework for efficiently designing manufacturable integrated photonic systems has been proposed in this work. The method is based on an iterative optimization strategy that optimizes an approximate system response based on mathematical modeling using Kriging. The approach is scalable since it depends on constructing mathematical models of integrated photonic components and using them in a system instead of building new inexpensive models of every new system that is considered. It was shown via an integrated photonic system example of second order and third order TripleX based serial ring resonators that the approach can efficiently and consistently find a robust design that is relatively insensitive to fabrication deviations. In this example, the robust design showed a lower nominal performance, but a significantly better worst-case performance. In practice, this would translate into substantially higher yields on integrated photonic systems optimized for robustness. Since the method is based on constructing metamodels of black-box components, the technique can readily be employed for efficient global robust optimization of any integrated photonic system.

REFERENCES

- [1] S. P. Gurav, K. Vervenne, H. J. Damveld, and A. van Keulen, *Bounded-but-Unknown Uncertainties in Design Optimization by combining the Multipoint Approximation Method and Design Sensitivities*, in *43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Denver, Colorado, Apr. 22-25, 2002*, AIAA-2002-1759 (2002).
- [2] J. Sacks, W. Welch, T. J. Mitchell, and H. P. Wynn, *Design and Analysis of Computer Experiments*, *Stat Sci* **4**, 409 (1989).
- [3] S. ur Rehman, M. Langelaar, and F. van Keulen, *Robust optimization of 2x2 multimode interference couplers with fabrication uncertainties*, *SPIE PhotonicWest*, 862711 (2013).
- [4] S. Rehman, M. Langelaar, and F. v. Keulen, *Robust Optimization of 2x2 Multimode Interference Coupler Affected by Parametric Uncertainties*, in *IEEE benelux Photonics* (2013).
- [5] S. Ur Rehman and M. Langelaar, *Efficient global robust optimization of unconstrained problems affected by parametric uncertainties*, *Structural and Multidisciplinary Optimization*, **1** (2015).

- [6] A. Melloni, G. Cusmai, and F. Morichetti, *Design on tolerances in integrated optics, in ECIO'08 Eindhoven - Proceedings of the 14th European Conference on Integrated Optics and Technical Exhibition, Contributed and Invited Papers* (2008) pp. 205–208.
- [7] C. Vázquez, A. Tapetado, J. Orcutt, H. C. Meng, and R. Ram, *Tolerance analysis for efficient MMI devices in silicon photonics*, (2014).
- [8] J. van der Tol, M. Felicetti, and M. K. Smit, *Increasing Tolerance in Passive Integrated Optical Polarization Converters*, *Lightwave Technology, Journal of* **30**, 2884 (2012).
- [9] K. Worhoff, P. V. Lambeck, and A. Driessen, *Design, tolerance analysis, and fabrication of silicon oxynitride based planar optical waveguides for communication devices*, *Lightwave Technology, Journal of* **17**, 1401 (1999).
- [10] D. Pérez-Galacho, R. Halir, A. Ortega-Moñux, C. Alonso-Ramos, R. Zhang, P. Runge, K. Janiak, H.-G. Bach, A. G. Steffan, and I. Molina-Fernández, *Integrated polarization beam splitter with relaxed fabrication tolerances*, *Opt. Express* **21**, 14146 (2013).
- [11] C. Alonso-Ramos, R. Halir, A. Ortega-Moñux, P. Cheben, L. Vivien, I. n. Molina-Fernández, D. Marris-Morini, S. Janz, D.-X. Xu, and J. Schmid, *A general approach for robust integrated polarization rotators*, (2013).
- [12] R. Heideman, M. Hoekman, and E. Schreuder, *TriPleX-Based Integrated Optical Ring Resonators for Lab-on-a-Chip and Environmental Detection*, *Selected Topics in Quantum Electronics, IEEE Journal of* **18**, 1583 (2012).
- [13] C. J. Kaalund and G.-D. Peng, *Pole-Zero Diagram Approach to the Design of Ring Resonator-Based Filters for Photonic Applications*, *J. Lightwave Technol.* **22**, 1548 (2004).
- [14] O. S. Ahmed, M. A. Swillam, M. H. Bakr, and X. Li, *Efficient Design Optimization of Ring Resonator-Based Optical Filters*, *Lightwave Technology, Journal of* **29**, 2812 (2011).
- [15] A. Melloni, *Synthesis of a parallel-coupled ring-resonator filter*, *Optics letters* **26**, 917 (2001).
- [16] C. Madsen, *General IIR optical filter design for WDM applications using all-pass filters*, *Journal of lightwave technology* **18**, 860 (2000).
- [17] PhoeniX Software, *PhoeniX Software 4.8.4*, (2014).
- [18] D. R. Jones, M. Schonlau, and W. J. Welch, *Efficient Global Optimization of Expensive Black-Box Functions*, *Journal of Global Optimization* **13**, 455 (1998).
- [19] S. Ur Rehman and M. Langelaar, *Infill sampling criterion for global robust optimization of systems with independent components*, *Engineering Optimization* (submitted).
- [20] M. Morris and T. J. Mitchell, *Exploratory designs for computational experiments*, *Journal of Statistical Planning and Inference* **43**, 381 (1995).

- [21] S. Ur Rehman and M. Langelaar, *Adaptive efficient global optimization of systems with independent components*, Structural and Multidisciplinary Optimization (submitted).

9

CONCLUSION AND FUTURE OUTLOOK

9.1. RECAP

In this thesis, we have striven to add to the body of work focused on efficient approaches for robust optimization of expensive components as well as deterministic and robust optimization of certain class of expensive to evaluate systems. The primary applications of this work were integrated photonic devices and systems affected by uncertainties. However, the methods proposed in this thesis are generic and can be applied to equivalent problems in other fields.

Deterministic and robust optimization of expensive to simulate problems can drain resources and consume a lot of time. In this work, we proposed application of optimization on cheap models of the expensive response. For this purpose, an iterative optimization strategy was employed which enabled adaptive improvement of the cheap model in areas of interest.

The metamodels in this work were constructed using Kriging. This choice was primarily motivated by the statistical basis of Kriging, which enables the interpolation error to be estimated. This interpolation error estimator was an important quantity in the derivation of the infill sampling criteria for all the component and system level problems shown in this work.

The most significant contribution of this thesis at device level was the derivation of sound infill sampling criteria for robust optimization of computationally expensive problems. Distinction was made between problems that were affected only by variations in the design variables or problem parameter and problems that were affected by both uncertainty types. For these different settings, we proposed strategies for robust optimization of unconstrained and constrained problems.

The infill sampling criteria in the design variables domain were based on the expectation of improvement over the best worst-case cost found on the metamodel response. A separate infill sampling criterion was developed for the parametric uncertainty do-

main. In this domain, the location with the highest expectation of deterioration in the worst-case cost was sought. Different criteria were suggested for unconstrained and constrained optimization problems. The component level algorithms were rigorously tested on several benchmark problems as well as on new test problems proposed by the author. The statistical performance of the proposed approaches was often orders of magnitude better than comparable techniques in terms of the number of expensive simulations needed to reach the global robust optimum. The algorithms were also used to find unconstrained and constrained robust optima of TripleX based single ring resonator problems. The ease of applicability of the techniques for global robust optimization of integrated photonic devices was thereby exhibited.

Efficient means of global deterministic optimization were developed for systems consisting of computationally expensive components. It was shown that for systems involving components that do not interact with each other, a viable system level expected improvement criterion can be derived for deterministic optimization. The system level efficient global optimization method discussed in Chapter 6 was the only algorithm in this thesis that targeted problems without uncertainties. To our knowledge, this was the first time that a system level expected improvement criterion was proposed for global optimization of systems with independent components. An equivalent expected improvement criterion for deterministic optimization existed at device level [1] and has extensively been used in literature, including in this work. But the lack of an equivalent system level strategy motivated the proposed algorithm. We demonstrated the method on unconstrained and constrained numerical problems. The relevance of the approach for optimization of integrated photonic systems was shown by applying it on a TripleX based serial ring resonator example. The objective of achieving a bandpass filter response was achieved using only a few expensive electromagnetic simulations.

The system level algorithm was extended to unconstrained and constrained systems involving uncertainties. Novel infill sampling criteria were developed which enabled efficient global system level robust optimization. A system level expected improvement criterion in the design domain and a system level expected deterioration criterion in the parametric uncertainty domain were suggested for this purpose. We applied the algorithm on the TripleX serial ring resonator example and demonstrated the benefit of including uncertainties in the problem formulation in terms of finding a robust design. In addition, simulation error was also included in the formulation by treating it as a parametric uncertainty.

Integrated photonic designers can use the suite of algorithms proposed in this work to perform efficient global robust optimization of components and systems. The framework has been developed in such a way that switching from one device to another should not require any change in strategy. Similarly, moving from device to system level should require minimal effort on the part of the designer.

Since the algorithms are demonstrated on black-box component and system level problems, they can be applied in several other engineering and non-engineering domains. The most important caveat that must be taken into account is that the framework is based on surrogate modeling techniques. Like all metamodel based methods, the approaches described in this work can only handle small problem sizes. This is especially more relevant to problems for which the robust optimum is sought, since robust opti-

mization is an innately expensive exercise. This effectively means that at component level the proposed techniques are not suitable for problems having more than ten to twenty design variables and uncertainties. The curse of dimensionality is mitigated in the case of the system level deterministic and robust optimization since metamodels are constructed at component level for those problems. The system level algorithms can therefore handle relatively larger problem sizes. However, the extent to which the problem size can increase such that the algorithms work well on the problem has not been extensively tested in this work. At system level, only a subclass of problems that do not have any interaction between components can be optimized using the algorithms proposed in this work. If there is any interaction between the components then the system has to be optimized using techniques developed in the field of multidisciplinary optimization [2].

9.2. CONCLUSION

The proposed algorithms should be employed for problems for which the simulation costs are significantly high compared to the internal computational requirements of the techniques. If the underlying device or system is not very expensive to simulate, then the process of applying the methods to find a robust optimum may in fact be quite slow. This is due to the fact that the internal global robust optimization performed on the cheap model on each iteration can also be computationally costly.

We can conclude from the multiple investigations in this work that infill sampling criteria are vastly superior in terms of enabling the robust optimum to be found in relatively few expensive simulations when compared to a space-filling based metamodel construction and subsequent robust optimization strategy. Designers can use pre-built libraries of initial metamodels constructed using a space-filling strategy. The proposed optimal infill sampling criteria can then be used to refine the component models in relevant regions for robust optimization for each specific design problem.

It was shown that efficiency gains can be made when treating implementation errors separately from parametric uncertainties. This was primarily because the implementation errors reside in the same dimension as the design variables while parametric uncertainties add to the total number of dimensions of the problem. Treating implementation errors as parametric uncertainties increases the total number of dimensions of the problem. This increases the cost of constructing a high fidelity metamodel.

For system level problems we chose an appropriate decomposition of the system model in terms of cheap and expensive parts. Metamodels were built for the expensive components in the model. This bi-level problem was robustly optimized. It was shown that appropriately selecting parts of the model so that they can be represented by metamodels and applying optimization on the resulting decomposed system can be significantly more efficient than employing optimization methods that treat the system as a single entity.

The comparison of the robust optimum with the deterministic solution for the chosen integrated photonic components and systems exhibited that the robust solution was relatively much less sensitive to uncertainties. The results indicated that the field of integrated photonics can greatly benefit from the use of a robust design approach. Using robust design strategies could significantly increase the yield of integrated photonic de-

vices and systems. The device level robust optimization techniques proposed in this thesis should enable robust designs to be obtained for integrated photonic components in an efficient manner. The system level approach was shown to be arbitrarily scalable since it was dependent on constructing mathematical models of integrated photonic components and employing them in a system instead of constructing new models of each new system. The generic and scalable nature of the system level algorithm should enable designers to efficiently apply the methods to different types of integrated photonic systems without significant additional implementation effort.

9.3. FUTURE OUTLOOK

There are several directions that could be taken to further refine the methods that have been described in this work. In what follows, we will expand upon both the obvious and more subtle improvements that could be made to the proposed techniques. The recommendations are divided in four main sections: 1. Kriging metamodel construction and optimization, 2. Enhanced strategies for differing simulation types, 3. Optimization of systems with interacting components and 4. Robust optimization - mitigating conservative results.

1. Kriging metamodel construction and optimization

The iterative robust optimization strategy proposed in this work requires a global robust optimum to be estimated on the cheap component or system response at each iteration. Even though the response on the metamodels is cheap to compute, determining the global best worst-case cost involves a nested min-max optimization which is an inherently expensive process. Methods and techniques to globally optimize radial basis functions in general, and Kriging models in particular, would vastly improve the efficiency of the proposed techniques. With regards to this, the fact that radial basis functions can basically be written as a difference of convex functions could be an exploitable property [3, 4]. Kriging models or radial basis functions are basically a weighted sum of basis functions whose individual minimum is known. The information of the individual minima and the weights attached to each basis function could be helpful in enabling intelligent starting points to be chosen for gradient based local optimization such that the global optimum is found. Another innovative research direction involving the use of a Gaussian transform to smoothen the radial basis function model in order to ease the search for the global optimum is also worth exploring further [5].

Kriging has been employed in this work for metamodel construction. The statistical approach of Kriging suffers from certain issues that also could hamper the performance of the proposed technique. For instance, the correlation matrix tends to become singular as points are added close to one another. Kriging also tends to underestimate the error in interpolation [6]. Furthermore, as the data sets for metamodel construction become larger, the Kriging correlation matrices occupy greater memory space and the speed of computing the metamodel response becomes slower. Improvements made to the underlying Kriging approach would directly benefit the proposed techniques. The methods proposed in this work are

not essentially tied to Kriging since any interpolation strategy that provides an interpolation error can be used in place of Kriging.

2. Enhanced strategies for differing simulation types

All the algorithms proposed in this work are based on the addition of only a single new expensive sample per iteration. It may be beneficial to extend the techniques to problems for which multiple expensive simulations can be run at the same time on different processors so that a parallel framework can take full advantage of the methods. A simple way to perform this for constrained problems would be to come up with Pareto fronts for the expected improvement in the objective and the probability of feasibility for the constraints and to sample at multiple Pareto optimal infill sampling locations. Similar work on this topic has been done in the context of deterministic optimization by Parr *et al.* [7, 8].

The proposed approaches assume that only the response of the black-box component is sampled at each iteration while the derivatives of the problem are not available. However, problems for which the gradient is cheaply available could benefit from a gradient enhanced adaptive optimization strategy [9, 10]. Extension of the techniques so that they can fully exploit potential availability of derivatives could be useful.

Simulations can also have variable fidelity. In this work, we have not considered this aspect. Simulation error has already been included in the proposed techniques as a parametric uncertainty. But extension of the algorithms so that they can harness multi-fidelity simulations to their advantage could also be a valuable addition [11]. Co-Kriging, which is a type of Kriging that can handle multi-fidelity samples could be a direction that can be explored in this context [12]. Space mapping is another strategy that has been widely used, especially in the electromagnetic domain, for problems based on multi-fidelity simulations [13]. Integrated photonic problems can also be simulated at different fidelity by changing the mesh size. Therefore, algorithms that use multi-fidelity simulations for efficient global robust optimization could directly benefit integrated photonics. When multi-fidelity simulations are employed then the error in the different simulation types will also be distinct. This aspect will have to be taken into account when performing robust optimization with respect to simulation error.

3. Optimization of systems with interacting components

The system level strategy proposed in this work can only handle systems in which the components do not interact with each other. While such a definition is sufficient for system optimization of integrated photonic problems, many engineering systems in general fall outside this category since they consist of dependent components. Optimization of systems with interacting components has traditionally been addressed within multidisciplinary optimization [2]. A means of bringing together the decades of research in multidisciplinary optimization [14, 15] with the system strategies discussed in this work could lead to a wider scope of application of the system algorithm. Deriving system level infill sampling criteria such as expected improvement for optimization of systems with expensive to evaluate inter-

acting components is a challenging problem. This is partially because the interaction between surrogates of the expensive components impacts the error estimate of each component metamodel in complex ways.

4. Robust optimization - mitigating conservative results

An important aspect of robust optimization is that it can often lead to conservative results. In this context, there are certain methods that could mitigate this problem. Amongst these techniques, globalized robust optimization [16] seems to be a promising approach. The method is used to find a solution that is robust on average by using smaller subsets within the larger uncertainty sets. Another interesting research direction is that of adjustable robust optimization, which involves making a clear distinction between variables whose values must be specified immediately i.e. 'here and now', and variables that can be adjusted later i.e. 'wait and see' [17]. This approach can be complemented with a folding horizon strategy where robust optimization is performed in steps as more information streams in concerning the problem such that the uncertainties in the problem reduce over time.

Adjustable robust optimization with folding horizon could prove to be an effective strategy in the case of integrated photonics because devices and systems are usually fabricated via a *multi-stage* process that can take several weeks. Within the fabrication process flow, the deposition of the waveguide layer at a designed thickness may take place early. The thickness of the layer can be measured at this early stage and robust devices and systems can then be designed given the measured thickness. In this manner, the thickness no longer remains an uncertainty with respect to which the problem has to be robust. Since the total number of uncertainties are reduced, a less conservative robust optimum can be found. A possible disadvantage in this scenario is that the designer has to produce the robust devices and systems in a short time frame so that the fabrication process is not delayed.

A suite of algorithms has been developed in this work for robust optimization of computationally expensive components and systems. By adopting the proposed adaptive sampling strategies, designers can produce robust devices and systems using limited computational and time resources. In particular, the proposed techniques should be able to address the design challenges that may surface when performing high volume production of integrated photonic devices and systems. The employed hierarchical system definition was broad enough to encompass a wide array of systems in integrated photonics as well as a subset of systems in general. Incorporating the most relevant suggestions for further research to the specific requirements of a particular discipline should also allow the algorithms to be suitable for a larger set of fields.

REFERENCES

- [1] D. R. Jones, M. Schonlau, and W. J. Welch, *Efficient Global Optimization of Expensive Black-Box Functions*, Journal of Global Optimization **13**, 455 (1998).
- [2] J. R. R. A. Martins and A. B. Lambe, *Multidisciplinary Design Optimization: A Survey of Architectures*, AIAA Journal **51**, 2049 (2013).

- [3] H. A. Le Thi, A. I. F. Vaz, and L. N. Vicente, *Optimizing radial basis functions by d.c. programming and its use in direct search for global derivative-free optimization*, *Top* **20**, 190 (2011).
- [4] T. Q. Phong, *An algorithm for solving general DC programming problems*, *Operations research letters* **15**, 73 (1994).
- [5] S. Pulkkinen, M. Makela, and N. Karmita, *A Continuation Approach to Global Minimization of Gaussian RBF Models*, (2011).
- [6] J. Kleijnen, W. van Beers, and I. van Nieuwenhuysse, *Expected improvement in efficient global optimization through bootstrapped kriging*, *Journal of Global Optimization* **54**, 59 (2012).
- [7] J. Parr, A. Forrester, a. J. Keane, and C. M. Holden, *Enhancing infill sampling criteria for surrogate-based constrained optimization*, *Journal of Computational Methods in Sciences and Engineering* **12**, 25 (2012).
- [8] J. M. Parr, A. J. Keane, A. I. Forrester, and C. M. Holden, *Infill sampling criteria for surrogate-based optimization with constraint handling*, *Engineering Optimization* **44**, 1147 (2012).
- [9] F. van Keulen and K. Vervenne, *Gradient-enhanced response surface building*, *Structural and Multidisciplinary Optimization* **27**, 337 (2004).
- [10] A. I. J. Forrester and A. J. Keane, *Recent advances in surrogate-based optimization*, *Progress in Aerospace Sciences* **45**, 50 (2009).
- [11] H. Shah, S. Hosder, S. Koziel, Y. A. Tesfahunegn, and L. Leifsson, *Multi-fidelity robust aerodynamic design optimization under mixed uncertainty*, *Aerospace Science and Technology* **45**, 17 (2015).
- [12] A. Forrester, S. Andras, and A. J. Keane, *Engineering design via surrogate modelling : a practical guide* (J. Wiley, 2008).
- [13] J. W. Bandler, Q. S. Cheng, S. A. Dakroury, A. S. Mohamed, M. H. Bakr, K. Madsen, and J. Sondergaard, *Space mapping: the state of the art*, *Microwave Theory and Techniques*, *IEEE Transactions on* **52**, 337 (2004).
- [14] P. N. Koch, D. Mavris, and F. Mistree, *Partitioned, Multilevel Response Surfaces for Modeling Complex Systems*, *AIAA Journal* **38**, 875 (2000).
- [15] W. Yao, X. Chen, Q. Ouyang, and M. van Tooren, *A surrogate based multistage-multilevel optimization procedure for multidisciplinary design optimization*, *Structural and Multidisciplinary Optimization* **45**, 559 (2011).
- [16] V. Gabrel, C. Murat, and A. Thiele, *Recent advances in robust optimization: An overview*, *European Journal of Operational Research* **235**, 471 (2014).
- [17] B. Gorissen, I. Yanikoglu, and D. D. Hertog, *Hints for practical robust optimization*, (2013).

SUMMARY

Design for manufacturing has become a key strategy for fulfilling the goal of achieving high yield and quality in production. The focus is on designing devices and system in such a way that, even if the manufacturing process inherently has certain defects, the manufactured product performs better than a certain minimal performance level. For this process to be successful, the designer needs access to the different ways in which defects can appear in the design during manufacturing and the extent to which they can impact the final design.

The amount of information available about these uncertainties largely governs the means by which and the extent to which the designer can create a robust design. If the uncertainties are bounded-but-unknown, robust optimization can be applied to find the best worst-case cost. Since robust optimization involves solving a nested min-max optimization problem, applying it directly on uncertain problems based on expensive computer simulations is prohibitively costly. Therefore an alternative strategy has to be employed in order to realize robust designs.

In this thesis, novel efficient global robust optimization techniques are proposed for finding robust designs for devices and systems based on expensive computer simulations. At device level, the computationally expensive problem is treated as a black-box. A cheap mathematical model of the expensive simulation is initially constructed using the Kriging interpolation method. This metamodel is adaptively sampled via novel infill sampling criteria that are based on the Kriging interpolation prediction and corresponding error estimate. The iterative strategy enables the robust optimum of the problem to be found efficiently. At each iteration, the derived infill sampling criterion in the design variables domain provides a new location that is most likely to give the highest improvement over the robust optimum. On the other hand, the infill sampling criterion in the uncertainties domain gives the location that is most likely to lead to the highest deterioration in the worst-case cost. Separate optimal approaches are devised for unconstrained and constrained problems affected by different uncertainty types.

A hierarchical structure is assumed for the problem at system level. This means that the responses of the components can be obtained independently of one another. Problems for which the component responses are expensive to simulate and the system evaluation is cheap are specifically targeted. For this problem setting, novel infill sampling criteria are derived for deterministic and robust optimization problems. For this purpose, cheap models are constructed via Kriging at component level. A linear approximation of the transformation that occurs from component to system level is used to derive a system level error estimator. This error estimator is used together with the cheap system level response in order to formulate the infill sampling criteria.

A primary application of the methods discussed in this work is robust optimization of integrated photonic devices and systems. The device level algorithms are applied on a TripleX based single ring resonator example. Similarly, at system level, TripleX based se-

rial ring resonators are employed to demonstrate the concept. The results at both device and system level indicate that the proposed robust optimization strategies can enable much higher yield for integrated photonic problems. Since the techniques are efficient, generic and scalable, they can be applied on disparate integrated photonic systems without incurring extra effort on the part of the designer.

The algorithms proposed in this work can be applied to a wide range of components and systems in engineering and non-engineering disciplines. This is due to the fact that very limited assumptions have been made concerning the problem setting, particularly at device level. Systems that have a hierarchical nature can readily be optimized, with and without uncertainty considerations, using the proposed strategies. Practitioners have to take into account, however, that the algorithms are based on metamodel construction at device level. Therefore, problems with a significant number of design variables and uncertainties at device level would be difficult to optimize using a limited number of simulations. The system level algorithms, on the other hand, can handle a much higher threshold on the number of design variables and uncertainties since metamodels have to be constructed at component level in that case.

SAMENVATTING

Ontwerpen voor fabricage (design for manufacturing) is uitgegroeid tot een belangrijke strategie om een hoge opbrengst te bereiken en de kwaliteit van de productie te waarborgen. De nadruk ligt op het op zodanige wijze ontwerpen van apparaten en systemen dat, zelfs indien het productieproces bepaalde fouten bevat, het vervaardigde product beter presteert dan een bepaald minimum prestatieniveau. Dit proces heeft alleen kans van slagen als de ontwerper kennis heeft van de verschillende manieren waarop defecten in het ontwerp kunnen ontstaan tijdens de fabricage en de mate waarin ze het uiteindelijke product kunnen beïnvloeden.

De hoeveelheid informatie die beschikbaar is over deze onzekerheden bepaalt grotendeels de wijze waarop en de mate waarin de ontwerper een robuust ontwerp kan maken. Indien de onzekerheden begrensd maar hun kansverdelingen onbekend zijn, kan robuuste optimalisatie worden toegepast om de beste worst-case cost te vinden. Daar robuuste optimalisatie inhoudt dat er een ingebed min-max optimalisatieprobleem opgelost dient te worden, is het direct toepassen ervan op problemen met onzekerheden gebaseerd op intensieve computersimulaties zeer tijdrovend en daarmee niet praktisch inzetbaar. Daarom moet er een alternatieve strategie worden toegepast om robuuste ontwerpen te verwezenlijken.

In dit proefschrift worden nieuwe efficiënte robuuste optimalisatietechnieken voorgesteld voor het vinden van robuuste ontwerpen voor componenten en systemen op basis van intensieve computersimulaties. Op componentniveau wordt het simulatiemodel beschouwd als een black-box. Allereerst wordt een goedkoop wiskundig model van de intensieve simulatie opgebouwd met de Kriging interpolatiemethode. Dit metamodel wordt adaptief getest via nieuw ontwikkelde infill testcriteria gebaseerd op de Kriging interpolatievoorspelling en de bijbehorende foutschatting. De iteratieve strategie maakt het mogelijk om het robuuste optimum van het probleem op efficiënte wijze te vinden. Bij elke iteratie geeft het infill testcriterium in het ontwerpvariabelen domein een nieuw ontwerp waarvan verwacht kan worden dat het de grootste verbetering van het robuuste optimum zal geven. Daarentegen zal het infill testcriterium in het onzekerheden domein het ontwerp selecteren dat hoogstwaarschijnlijk zal leiden tot de grootste verslechtering van de worst-case cost. Aparte optimale benaderingen zijn ontwikkeld voor problemen zonder en met beperkende nevenvoorwaarden die door verschillende soorten onzekerheden worden beïnvloed.

Voor problemen op systeemniveau wordt uitgegaan van een hiërarchische structuur. Dit betekent dat de reacties van de componenten onafhankelijk van elkaar kunnen worden verkregen. Speciale aandacht wordt besteed aan problemen waarvoor de componentreacties tijdrovend zijn om te simuleren en de systeemsimulatie goedkoop is. Voor deze probleemstelling zijn nieuwe infill testcriteria ontwikkeld voor deterministische en robuuste optimalisatieproblemen. Hierbij worden benaderingsmodellen opgesteld met behulp van Kriging op componentniveau. Er is een lineaire benadering gebruikt van de

transformatie die plaatsvindt van component- naar systeemniveau om de foutschatter op systeemniveau af te leiden. Deze foutschatter wordt samen met de benaderde componentreacties op systeemniveau gebruikt om de infill testcriteria te formuleren.

Een primaire toepassing van de methoden die in dit werk worden besproken is de robuuste optimalisatie van geïntegreerde optische componenten en systemen. De algoritmen op componentniveau worden toegepast op het voorbeeld van een enkele ring resonator gebaseerd op TripleX. Het concept wordt eveneens gedemonstreerd op systeemniveau, op Triplex-gebaseerde seriële ring resonatoren. De resultaten op zowel component- als systeemniveau geven aan dat de voorgestelde robuuste optimalisatie strategieën een veel beter rendement opleveren voor geïntegreerde optische problemen. Omdat de technieken efficiënt, schaalbaar en generiek zijn, kunnen ze worden toegepast op ongelijksoortige geïntegreerde optische systemen zonder extra inspanning van de ontwerper.

De in dit werk voorgestelde algoritmen kunnen worden toegepast op een breed scala van componenten en systemen in de techniek en in niet-technische disciplines. Dit vanwege het feit dat er zeer weinig veronderstellingen gemaakt zijn over de structuur van het probleem, vooral op componentniveau. Systemen van hiërarchische aard kunnen met de voorgestelde strategieën worden geoptimaliseerd, met en zonder onzekerheden. Bij toepassing dient er echter rekening mee te worden gehouden dat de algoritmen gebaseerd zijn op metamodel constructie op componentniveau. Om die reden zijn problemen met een groot aantal ontwerpvariabelen en onzekerheden op componentniveau moeilijk te optimaliseren met een beperkt aantal simulaties. De algoritmen op systeemniveau, daarentegen, kunnen een veel groter aantal ontwerpvariabelen en onzekerheden verwerken, omdat in dat geval metamodellen moeten worden opgesteld op componentniveau.

ACKNOWLEDGEMENTS

I would like to thank my promoter Fred van Keulen for mentoring and guiding me throughout the PhD process. Fred always pushed me to think critically and originally. He would go out of his way to help. A prime example of this was when he took care of all the arrangements such that I could present remotely at the World Congress on Structural and Multidisciplinary Optimization, (WCSMO-10), when I was not able to travel to the US. I am also grateful to him for affording me the flexibility to broadly define my PhD topic and for providing me the latitude to pursue research directions within that scope that most sparked my interest. This aspect, in particular, made the experience of working towards a PhD all the more enjoyable.

I am immensely grateful to my supervisor Matthijs Langelaar for investing his valuable time and energy into this project. Matthijs, I can't even count the number of times I knocked on your door and promptly marched in, declaring that it would only take five minutes. Those five minutes would invariably turn into an hour, but your enthusiasm never wavered. Thank you for giving such detailed feedback on all the articles. It has been a pleasure working with you.

This work would not have been possible without the financial support of STW. I would also like to take the opportunity to thank the committee members for reading and evaluating this work and for taking out their valuable time to attend the defense. The contribution of the project partners in this project is also much appreciated. I would like to especially thank Peter Harmsma from TNO for his help with the experiments. Thank you Peter for patiently attaching a new optical fiber every time I broke one! Thanks to Pim Muilwijk from TNO for helping build the measurement setup. I am grateful to Remco Stoffer from PhoeniX Software for the simulation support and expertise so readily and promptly provided. A word of thanks is due to Marcel Hoekman from LioniX for giving invaluable simulation and fabrication support. I would also like to express my thanks to Ronald Dekker from Xio Photonics for providing expertise on integrated optics. I am indebted to all of you for taking such an avid interest in this work over the last four years and for providing rich and useful feedback during and outside of the user meetings.

The relaxed and open atmosphere in the Structural Optimization and Mechanics group at TU Delft was, I felt, highly conducive to performing high quality research. For creating this positive and stimulating work environment I am thankful to all my colleagues: Marco, Evert, Alex, Anil, Deepak, Chi, Sasan, Floris, Gijs, Banafsheh, Rob, Salman, Marianne, Hans, Can, Alejandro and Matthijs.

It would be a disservice to forget my friends from Germany, some of whom visited me multiple times over the past four years. Osama, Ahad, Ahsan and Farooq, thanks for the great memories! I would also like to thank Akram, Musa'ab, Laiq, Imran, Shahzad, Adnan, Waqas, Mottaqiallah, Ershad, Faisal, Tabish, Nauman, Rajab for the wonderful

company they proved to be over the past four years. Your presence certainly enriched my experience of life in Delft.

I would like to express my deepest gratitude to my wife for the strength and steadfast support she has so readily offered over the last four years. It has been a privilege to share these moments with you. The PhD work never seemed like a source of stress or worry, for this I am grateful to you for your flexibility and understanding. The birth of our son brought with it fulfillment, perspective and a renewed will to strive and excel.

I am heavily indebted to the efforts of my parents, without whose tremendous support I would be nowhere near entering a PhD program, let alone completing it. Decades of commitment and sacrifice can neither be described in a few sentences nor can any recompense be offered for them. I am grateful to my father for always pushing me to pursue perfection and excellence. Likewise, I am extremely grateful to my mother for always believing in me. You have always remained a strong pillar of support for me.

LIST OF PUBLICATIONS

JOURNALS

1. S. Ur Rehman, M. Langelaar, and F. van Keulen, Efficient Kriging-based Robust Optimization of Unconstrained Problems, *Journal of Computational Science*, vol. 5, issue 6, pp. 872-881, Nov. 2014.
2. S. Ur Rehman, and M. Langelaar, Efficient global robust optimization of unconstrained problems affected by parametric uncertainties, *Structural and Multidisciplinary Optimization*, vol. 52, no. 2, pp. 319-336, May 2015.
3. S. Ur Rehman, and M. Langelaar, Efficient infill sampling for unconstrained robust optimization problems, *Engineering Optimization*, pp. 1-20, Nov. 2015.
4. S. Ur Rehman, and M. Langelaar, Expected improvement based infill sampling for global robust optimization of constrained problems, *submitted to Optimization and Engineering*, pp. 1-21, 2015.
5. S. Ur Rehman, and M. Langelaar, Adaptive efficient global optimization of systems with independent components, *submitted to Structural and Multidisciplinary Optimization*, pp. 1-17, 2015.
6. S. Ur Rehman, and M. Langelaar, Infill sampling criterion for global robust optimization of systems with independent components, *submitted to Engineering Optimization*, pp. 1-22, 2015.
7. S. Ur Rehman, and M. Langelaar, Generic scalable robust optimization of integrated photonic systems, *submitted to Journal of Lightwave Technology*, pp. 1-7, 2015.

CONFERENCE PROCEEDINGS

1. S. Ur Rehman, M. Langelaar, and F. van. Keulen, Robust Optimization of 2x2 Multimode Interference Coupler Affected by Parametric Uncertainties, in *18th Annual Symposium of the IEEE Photonics Benelux Chapter*, Eindhoven, the Netherlands, 2013.
2. S. Ur Rehman, M. Langelaar, and F. van Keulen, Robust multi-objective optimization of 2x2 multimode interference coupler using expected improvement, in *15th International Conference on Transparent Optical Networks (ICTON)*, Cartagena, Spain, 2013.

3. S. Ur Rehman, M. Langelaar, and F. van Keulen, Efficient Kriging-Based Robust Optimization of Unconstrained Problems, in *3rd International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH)*, Reykjavik, Iceland, 2013.
4. S. Ur Rehman, M. Langelaar, and F. van Keulen, Robust optimization of unconstrained problems using efficient response surface building strategy, in *World Congress on Structural and Multidisciplinary Optimization (WCSMO)*, Florida, USA, 2013.
5. S. Ur Rehman, M. Langelaar, and F. van Keulen, Robust optimization of 2x2 multi-mode interference couplers with fabrication uncertainties, in *Proc. SPIE 8627, Integrated Optics: Devices, Materials, and Technologies XVII*, 862713, San Francisco, USA, 2013.

CURRICULUM VITAE

Samee ur Rehman was born in 1987 in Lahore, Pakistan. He received his primary and secondary education from Aitchison College, Lahore. In 2009, he completed a B.Sc. in Electrical Engineering and Computer Science from Jacobs University in Germany. He received his M.Sc. degree in Electrical Engineering from the same institution in 2011. In 2010, he worked at the German Research Center for Artificial Intelligence (DFKI) in Bremen, Germany. Since October 2011, he is a Ph.D. candidate in the Structural Optimization and Mechanics group of the Precision and Microsystems Engineering department at Delft University of Technology, the Netherlands. Currently, he works as a Design Engineer at ASML, the Netherlands. His present research activities are focused on system optimization, deterministic and robust optimization, metamodelling, and their application to optics and electromagnetics.

