CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Civil Engineering
Department of Mechanics

Multi-Objective Reliability-Based Design Optimization using Approximation Techniques

DOCTORAL THESIS

Ing. Adéla Hlobilová

Doctoral study programme:	Civil Engineering
Branch of study:	Physical and Material Engineering
Doctoral thesis tutor:	Doc. Ing. Matěj Lepš, Ph.D.
Doctoral thesis tutor specialist:	Ing. Anna Kučerová, Ph.D.

Prague, 2020

CZECH TECHNICAL UNIVERSITY IN PRAGUE
Faculty of Civil Engineering
Thákurova 7, 166 29 Praha 6

DECLARATION

Ph.D. student's name:	Adéla Hlobilová
Title of the doctoral thesis:	Multi-Objective Reliability-Based Design Optimization using Approximation Techniques

I hereby declare that this doctoral thesis is my own work and effort written under the guidance of the tutors Matěj Lepš and Anna Kučerová.
All sources and other materials used have been quoted in the list of references.

The doctoral thesis was written in connection with research on the projects:
SGS12/027/OHK1/1T/11, SGS13/034/OHK1/1T/11, GAČR P105/12/1146,
SGS14/028/OHK1/1T/11, SGS15/030/OHK1/1T/11, GAČR 15-07299S,
SGS16/037/OHK1/1T/11, GAČR 16-11473Y, SGS17/042/OHK1/1T/11,
and SGS18/036/OHK1/1T/11.

In Prague on ....
signature

Acknowledgement

I would like to express my gratitude to my advisor doc. Ing. Matěj Lepš, Ph.D. for the continuous support of my doctoral study, for his patience, motivation, and immense knowledge. His guidance helped me throughout my research. I also want to thank my advisor specialist Ing. Anna Kučerová, Ph.D. for her encouragement, constructive comments and accepting me into her research group, where I could work on related topics.

In September 2013, I went for a research fellowship to the Vienna University of Technology for several months. I wish to express my sincere appreciation to my host Univ. Prof. Dipl.-Ing. Dr.techn. Christian Bucher who has devoted me all the time I needed and helped me to understand some stochastic, probabilistic and reliability phenomena. My fellowship at this university has been highly productive.

I also would like to thank prof. Dr. Ing. Bořek Patzák, doc. Dr. Ing. Daniel Rypl, Ing. Jan Sýkora, Ph.D., and doc. Ing. Vít Šmilauer, Ph.D., DSc. They were project leaders in short term student projects in which I actively participated throughout my doctoral studies.

I would like to thank my fellow doctoral students and friends, in particular, David, Eva, Eliška, Filip, Karel, Martin, and Jan. I spent a great time with you, not only in the office.

Last but not least, I wish to acknowledge the support and great love of my family, my husband Michal, and my parents. They kept supporting me regardless of the consequences, and this thesis would not have been possible without their help. Michal, I would never have finished my thesis without your substantial help. The last word goes for Jakub, my baby boy, who has been the light of my life over the previous two and a half years and who has given me the extra strength and motivation to get things done.

The work was supported by the Czech Science Foundation, projects Nos. P105/12/1146, 15-07299S, and 16-11473Y. Continued support by the Grant Agency of the Czech Technical University in Prague through student projects Nos. SGS12/027/OHK1/1T/11, SGS13/034/OHK1/1T/11, SGS14/028/OHK1/1T/11, SGS15/030/OHK1/1T/11, SGS16/037/OHK1/1T/11, SGS17/042/OHK1/1T/11, and SGS18/036/OHK1/1T/11 is also gratefully acknowledged.

Abstrakt

Spolehlivostní optimalizace konstrukcí (RBDO, z anglického Reliability-based design optimization) hledá kompromisní řešení mezi cenou a spolehlivostí systému. Tato optimalizační oblast je výpočetně velmi náročná zejména kvůli určení pravděpodobnosti poruchy. Mnoho výzkumných pracovníků proto hledá nové přístupy ke snížení výpočetních nároků RBDO, které lze rozdělit do dvou hlavních proudů. První proud využívá původní RBDO formulaci s dvěma cykly: vnější cyklus obstarává optimalizační část a vnitřní cyklus počítá pravděpodobnost poruchy pro každou kombinaci návrhových proměnných navrženou optimalizační rutinou. Výpočetní nároky pro spolehlivostní výpočet jsou sníženy například pokročilými simulačními technikami typu Monte Carlo, které využívají meta-modely. Druhá skupina RBDO přístupů minimalizuje výpočetní nároky reformulací přístupu se dvěma cykly do cyklu jednoho nejčastěji pomocí aproximace prvního řádu rezervy spolehlivosti (FORM, z anglického First-order reliability method) nebo převedením pravděpodobnostních omezujících podmínek na přibližné, nicméně deterministické podmínky, nebo obojím. Klasická vícekriteriální RBDO rozšiřuje jednokriteriální formulaci použitím více účelových funkcí se zachováním omezujících spolehlivostních podmínek. Tím pádem takovéto vyjádření neodpovídá na otázku, jaká je cena pro danou hladinu spolehlivosti. Tato práce proto definuje úlohu tak, aby na tuto otázku odpověděla. Tedy, aby byla výsledná řešení z Paretovy množiny indiferentním kompromisem mezi cenou a spolehlivostí. Hlavním cílem této práce je kromě zmíněné formulace úlohy i implementace metodologie, její optimalizace a validace pro vícekriteriální spolehlivostní optimalizaci konstrukcí s cenou a spolehlivostí definovanými v podobě účelových funkcí. Jelikož je tato úloha výpočetně náročná, dalšími cíli této práce je snížení výpočetních nároků ve třech klíčových místech navrhovaného algoritmu, a to ve výběru vhodného vícekriteriálního algoritmu, výběru vhodné spolehlivostní metody a náhrady původního modelu meta-modelem.

Navrhovaná RBDO metoda se dvěma cykly využívá simulační techniky založené na principu Monte Carlo s použitím meta-modelů vylepšených o optimalizovaný adaptivní návrh experimentu (DoE, z anglického Design of Experiments). Aktualizace DoE probíhá během RBDO, pomocí jíž se vyhledávají nejlepší řešení s ohledem na cenu a spolehlivost. Spolehlivost návrhu je vyhodnocena pokročilou simulační technikou na principu metody Monte Carlo, která vyhodnocuje meta-model místo původního modelu. Vhodné vzorky z této simulační techniky se následně využijí pro aktualizaci DoE. Pro sestavení vhodného meta-modelu pro RBDO výpočet totiž musí být návrhový prostor vhodně pokryt, což je zajištěno jednak rovnoměrným rozprostřením návrhových bodů a jednak umístěním bodů co nejblíže hyperploše mezi spolehlivou a nespolehlivou oblastí, tzv. hranici poruchy. První kritérium je zajištěno pomocí maximalizace minimální vzájemné vzdálenosti bodů návrhu. Jako druhé kritérium je použita minimalizace vzdálenosti návrhového bodu od hranice poruchy. Tato dvě kritéria vstupují do vícekriteriální optimalizace, která vyhodnocuje vzorky ze simulační metody a tím dohledává další návrhové body k zpřesnění meta-modelu tak, aby se choval jako původní model v nejzajímavějších oblastech. S každou generací evolučního algoritmu je tedy DoE obohaceno o body ze simulační techniky, které splňují podmínku rovnoměrného pokrytí návrhového prostoru a zároveň jsou blízko hranice spolehlivosti, následně ohodnocené pomocí originálního modelu. Pareto front z každé další generace je tedy s ohledem na spolehlivost přesnější, neboť je po každé aktualizaci DoE sestaven nový meta-model. Jelikož je návrhový prostor široký, v této práci využíváme kromě klasického meta-modelu s plnou Gramovou maticí sestaveného na celém návrhovém prostoru (GMM, z anglického global meta-model) i řídké globální meta-modely (SGMM, z anglického sparse global meta-model), sestavené na celém návrhovém pros-

toru pro každou generaci evolučního algoritmu, a lokální meta-modely (LMM), vytvořené pro každého jedince z každé generace evolučního algoritmu zvlášť. SGMM je založen na myšlence, že vzájemný vliv bodů DoE vymizí s jejich narůstající vzdáleností a tím pádem mají od sebe vzdálené vzorky na sebe navzájem malý vliv. Proto je místo plné Gramovy matice zahrnující spojení mezi všemi body DoE sestavena matice řídká, ve které je zohledněno pro každý bod DoE pouze okolí, které tento bod ovlivňuje. Všechny body mimo toto okolí jsou pro daný bod vynechány. Lokální meta-modely jsou sestaveny pouze v zájmové oblasti, která je menší než návrhový prostor, a tím pádem musí mít plnou Gramovou matici.

V rámci celé práce jsou porovnávány Pareto-fronty jako výsledná řešení z RBDO ze třech skupin aproximace na třech matematických příkladech a dvou jednoduchých konstrukcích. Příklady se od sebe odlišují svojí složitostí, jsou v nich zahrnuty různé stupně nelinearity ať už v modelu nebo v potřebné transformaci do normálního prostoru, spolehlivosti komponent i sériových systémů s odlišným počtem ploch odezvy, liší se i v různém počtu stochastických proměnných nebo v omezení návrhových domén. První skupina aproximace využívá pro výpočet předpodmíněnou metodu Monte Carlo, kde se za pomoci předpodmínění dosáhlo přibližně konstantního variačního koeficientu odhadu pravděpodobnosti poruchy. Aproximace je v této skupině zahrnuta pouze v meta-modelu. Druhá skupina aproximuje spolehlivost za pomoci šesti simulačních technik, jmenovitě Asymptotic sampling (asymptotické vzorkování), Importance sampling (generování vzorků blíže k oblasti poruchy), Subset simulation (simulování podmnožinami), klasické simulace Monte Carlo, Enhanced (vylepšené) simulace Monte Carlo a Scaled sigma sampling (samplování s váhováním sigem), a aproximační techniky prvního řádu FORM s využitím původního modelu. Aproximace je tedy pouze na úrovni výpočtu spolehlivosti. Třetí skupina kombinuje tři nejlepší simulační metody, a to Asymptotic sampling, Importance sampling a Subset simulation, se třemi meta-modely, tedy GMM, SGMM a LMM. Všechny výsledné Pareto-optimální fronty jsou ohodnoceny pomocí metrik a je pro ně určena chyba spolehlivosti tak, že se získaný Paretův set v návrhovém prostoru namapuje pomocí rekurentní simulace Monte Carlo s předepsaným variačním koeficientem odhadu pravděpodobnosti poruchy do prostoru účelových funkcí.

Z prezentovaných výsledků vyplývá, že pokud je funkce relativně hladká, pak jsou globální meta-modely s plnou, respektive i řídkou Gramovou maticí, vhodnou volbou náhrady původního modelu. Tento globální meta-model je i méně citlivý na volbu simulační metody. Pokud je ale návrhový prostor velký, existuje v něm více návrhových bodů nebo módů porušení a hranice poruchy je vysoce nelineární, pak jako náhrada originálního modelu lépe poslouží lokální meta-modely i s menším počtem bodů v DoE, tzn. nižším počtem evaluací původního modelu, v porovnání s modely globálními. Námi navrhovaná metoda využívající průběžně aktualizované meta-modely je přesnější a výpočetně méně náročná než RBDO se dvěma cykly využívající formulaci metody spolehlivostního indexu (RIA, z anglického Reliability index approach), tedy využití FORM pro výpočet spolehlivosti, který i pro méně přesné výsledky v porovnání s naší metodou potřebuje minimálně o dva řády více evaluací původního modelu. Pokročilé simulační metody kromě vyšší přesnosti při odhadu spolehlivosti s nižším počtem evaluací modelu také nabízí vhodnější body pro aktualizaci DoE, neboť lépe pokrývají oblast poruchy než klasická metoda Monte Carlo.

Abstract

Reliability-based design optimization (RBDO) searches for a trade-off between costs and safety under the assumption of uncertainties. This optimization area is computationally very expensive, especially due to an evaluation of the probability of failure. Many researchers seek novel techniques for computational cost reduction. The first stream utilizes the original RBDO double-loop formulation, which deals with an optimization task in the outer loop and assesses reliability in the inner loop. Advanced simulation techniques utilizing meta-models decrease demands for the reliability evaluation. The second stream minimizes the computational demands by reformulating the double-loop problem into a single-loop usually using first-order approximations (FORM) of a limit state function, by converting the probabilistic constraints to the approximated deterministic ones, or both. Classical multi-objective RBDO adds objectives into both mentioned formulations but keeps the reliability in the inequality constraints. Therefore, it does not answer the question of how much reliability costs. This thesis formulates the task such that the resulting Pareto front provides a dataset of compromising solutions concerning both costs and reliability to answer that question. The main objective of this thesis is to implement, optimize and validate a methodology for multi-objective reliability-based design optimization with costs and reliability in objective functions. Since this task is computationally demanding, partial objectives of this thesis are to reduce computational expenses in three key positions, namely in the selection of a suitable multi-objective algorithm and reliability assessment method and replacement of the original model with a meta-model.

The proposed double-loop RBDO method utilizes a meta-model-based Monte Carlo type approach which is enhanced by an optimized adaptive Design of (Computer) Experiments (DoE). The DoE update proceeds within a multi-objective evolutionary optimization algorithm, which searches for the best solutions in terms of costs and reliability. An advanced simulation technique based on Monte Carlo principles assesses the reliability evaluating the meta-model instead of the original model, and the DoE update uses suitable sampling points from this simulation technique. For quality meta-models, it is essential to have covered design space optimally with DoE points. For RBDO purposes, the DoE must meet two objectives. The first one is the space-filling property, in which the minimal interpoint distance is maximized. The second one is the distance to the limit state that divides the region into the failure and safe domain and therefore is significant for the reliability assessment; this objective is minimized. Both objectives are optimized to obtain the most accurate meta-model in the vicinity of the limit state with a minimal number of sampling points. Thus, within each generation of the evolutionary algorithm, additional points from the simulation technique optimized by space-filling and limit-state-proximity objectives enrich the DoE, which is subsequently evaluated using the original model. Therefore, the Pareto front from each subsequent generation is more accurate concerning reliability, as a new meta-model is assembled after each DoE update. Since the design space is wide, in addition to a classical global meta-model with a dense Gram matrix assembled for the entire design space (GMM), we use our novel methodology for sparse global meta-models (SGMM) assembled for each generation in the entire design space and local meta-models (LMM) constructed for each individual from each generation individually. The SGMM utilizes the fact that the mutual influence of the DoE points diminishes with their increasing distance, and thus the distant samples have a little impact on each other. Therefore, instead of a dense Gram matrix, including all links between all support points, each support point has its influence domain in the closest neighbourhood, and other points outside the influence domain are omitted. Local meta-models are constructed only in the influence domain having a dense

Gram matrix since this domain is smaller than the design space and therefore, all the links have to be included in the Gram matrix to maintain accuracy for that domain.

The presented thesis compares Pareto-fronts as final solutions from RBDO from three groups of approximations on three mathematical examples and the design on two simple structures. The examples differ in their complexity; they include different degrees of nonlinearity either in the model or in the necessary transformation into a normal space, reliability of components and series systems with a different number of limit state functions, they also differ in a number of stochastic variables or design constraints. The first group of approximations uses the preconditioned quasi-Monte Carlo method to assess the reliability, where the preconditioning guarantees an approximately constant coefficient of variation of the failure probability estimate. The approximation is included only in the meta-model. The second group approximates reliability using six simulation techniques, namely an Asymptotic sampling, Importance sampling, Subset simulation, classical Monte Carlo simulation, Enhanced Monte Carlo simulation and Scaled sigma sampling, and one approximation technique, namely First-order reliability method. The third group combines the three best simulation methods, namely an Asymptotic sampling, Importance sampling and Subset simulation, with three meta-models, i.e. GMM, SGMM, and LMM. Several performance measures and error indicators evaluate all resulting Pareto-optimal fronts; a recurrent Monte Carlo simulation with a prescribed coefficient of variation of the failure probability estimate maps the Pareto-optimal sets into the objective space to evaluate the reliability index error.

The presented results show that the global meta-models with a dense and sparse Gram matrix are a suitable choice to replace the original model, that is relatively smooth. The global meta-model is also less sensitive to the choice of the simulation method. However, if the design space is large, where more design points or failure modes exist, and the limit state is highly nonlinear, then local meta-models even with a smaller number of points in DoE serve better as a replacement for the original model. Our proposed method using adaptively updated meta-models is more accurate and computationally less demanding than double-loop RBDO using Reliability index approach, i.e. the use of FORM to evaluate the reliability. FORM needs at least two orders of magnitude more evaluations of the original model for less accurate results in comparison with our method. Advanced simulation techniques, in addition to better accuracy and precision in estimating structural reliability with a lower number of model evaluations, also offer more suitable points for DoE update, as they better cover the space around the limit state than the classical Monte Carlo method.

Contents

1 Introduction **1**

1.1 Background 1

1.2 Aims and objectives of the thesis 3

2 Reliability-based design optimization **5**

2.1 Methods for solving Reliability-based design optimization 5

2.1.1 Double-loop approach 5

2.1.2 Single-loop approach and Decoupled approach 9

3 Multi-objective optimization **10**

3.1 Non-dominated sorting genetic algorithm II (NSGA-II) 14

3.2 Performance measures 18

3.2.1 Metrics evaluating closeness to the Pareto front 18

3.2.2 Metrics evaluating diversity among non-dominated solutions 19

3.2.3 Metrics evaluating closeness and diversity 20

4 Structural reliability **22**

4.1 Introduction into stochastic variables 22

4.2 Probability of failure 26

4.3 Approximation techniques 29

4.3.1 First-Order Reliability Method (FORM) 29

4.4 Simulation techniques 35

4.4.1 Monte Carlo simulations (MC) 35

4.4.2 Importance sampling (IS) 40

4.4.3 Asymptotic sampling (AS) 42

4.4.4 Subset simulation (SS) 44

4.4.5 Enhanced Monte Carlo simulation (eMC) 49

4.4.6 Scaled sigma sampling (SSS) 52

5 Surrogate models **56**

5.1 Radial Basis Functions model 56

5.1.1 Global meta-models 59

5.1.2 Sparse global meta-models 59

5.1.3 Local meta-models 62

5.1.4 Implementation details 63

5.2 Kriging 72
5.3 Update of meta-models for reliability assessment 73
5.4 Updates for reliability-based design optimization 74
5.4.1 Updates for meta-models replacing an objective function in RBDO . . 74
5.4.2 Updates for meta-models replacing constraints functions in RBDO . . . 79

6 Proposed Reliability-based design optimization procedure 81
6.1 Adaptive multi-objective-optimization updating procedure 81
6.1.1 DoE updating independent of RBDO 81
6.1.2 DoE updating dependent on RBDO 83
6.2 Parallelization of Reliability-based Design Optimization using Surrogates . . . 89
6.2.1 Computational demands of RBDO 89

7 Numerical examples for the multi-objective reliability-based design optimization procedure 91
7.1 Example 1: Mathematical problem with a nonlinear limit state function 96
7.1.1 Comparison of meta-models and analytical model results together with a preconditioned quasi-Monte Carlo simulation 96
7.1.2 Results comparison of different reliability assessment techniques utilizing an analytical model 101
7.1.3 Results for approximation using meta-models and reliability assessment techniques 105
7.2 Example 2: Mathematical problem with a highly nonlinear series system reliability 110
7.2.1 Comparison of meta-models and analytical model results together with a preconditioned quasi-Monte Carlo simulation 111
7.2.2 Results comparison of different reliability assessment techniques utilizing an analytical model 114
7.2.3 Results for approximation using meta-models and reliability assessment techniques 119
7.3 Example 3: Mathematical problem with a series system containing three limit states 124
7.3.1 Comparison of meta-models and analytical model results together with a preconditioned quasi-Monte Carlo simulation 126
7.3.2 Results comparison of different reliability assessment techniques utilizing an analytical model 132
7.3.3 Results for approximation using meta-models and reliability assessment techniques 136
7.4 Example 4: A short column under oblique bending 140
7.4.1 Comparison of meta-models and analytical model results together with a preconditioned quasi-Monte Carlo simulation 142
7.4.2 Results comparison of different reliability assessment techniques utilizing an analytical model 147
7.4.3 Results for approximation using meta-models and reliability assessment techniques 151
7.5 Example 5: A 23-bar truss 156
7.5.1 Comparison of meta-models and analytical model results together with a preconditioned quasi-Monte Carlo simulation 158

7.5.2 Results comparison of different reliability assessment techniques utilizing an analytical model . . . 161
7.5.3 Results for approximation using meta-models and reliability assessment techniques . . . 165
7.6 Summary results for all examples . . . 170

8 Conclusions 178

9 Distributions for continuous variables 199
9.1 Uniform distribution . . . 199
9.2 Normal distribution . . . 200
9.3 Log-normal distribution . . . 201
9.4 Exponential distribution . . . 202
9.5 Gumbel distribution . . . 203
9.6 Weibull distribution . . . 205

10 Ultimate plastic state under bending and tension 208

11 Hardware and software parameters 210

12 Unpublished results for Examples 1 - 5 211
12.1 Box-plots for performance measures . . . 211
12.2 Unpublished combinations of simulation methods and meta-models . . . 219
12.3 Summary data . . . 224

Chapter 1

Introduction

Parts of this chapter are reproduced from the author's contributions [87, 149, 152].

1.1 Background

Optimization [19, 81, 132, 140, 160] and search methodologies [170] have become very popular for making products more desirable. A shape of the structure [119, 144, 153], topology [100, 209], reinforcement of concrete structures [56, 112, 114, 124], cross-sections [33, 52, 146], design of the concrete mix [20, 113], and many other properties can be optimized. Structural optimization [4, 14, 29, 35, 83, 179] is a process that seeks the best design under some predefined constraints. A deterministic model, i.e. model, which has identical outputs for a given input, is usually unrealistic due to uncertain inputs such as material properties, a structural topology, and loadings. Moreover, the deterministic optimization techniques can often lead to unacceptable results [94] since an optimal design with deterministic variables often terminates at a boundary between the failure domain and the safe domain. Even a small perturbation in inputs can lead to a fatal failure. For that reason, the model uncertainties are introduced; parameter uncertainties are associated with input data, whereas structural uncertainties express that a model need not clearly describe the physics of the problem [205]. YAO et al. [200] define uncertainty as an "incompleteness in knowledge and the inherent variability of the system and its environment". FORRESTER [74] mentioned that computational errors are mainly human-made (e.g. incorrectly entered boundary conditions) and systematic errors (e.g. a wrong model). Since computational experiments are deterministic, a random error is more common in experimental data. Another possible categorization of the sources of uncertainty is to consider *aleatoric uncertainty* (from the Latin *alea* with meaning rolling of dice) that is not influenceable by the experiment repetition and *epistemic uncertainty* (from Greek *episteme* that means knowledge) that disappears with better information of the problem [102, 200]. Uncertainties can be represented through interval bounds that is the vaguest definition; by membership functions which are used in fuzzy logic approaches; or by probability density functions that provide the best description of uncertainty [205].

An optimal design provides a small probability of failure assuming structural economy and reduction of the system variability to unexpected variations. These requirements categorize the optimization under uncertainty into two groups [94] as depicted in Figure 1.1. Economical design with high reliability is provided by *reliability-based design optimization* (sometimes re-

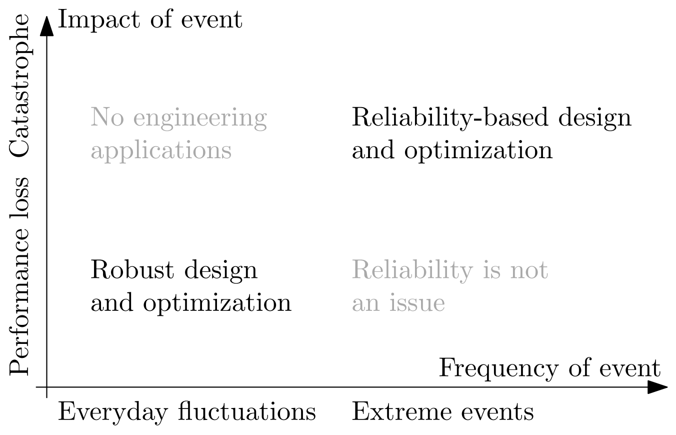

Impact of event
Performance loss
Catastrophe
No engineering applications
Reliability-based design and optimization
Robust design and optimization
Reliability is not an issue
Frequency of event
Everyday fluctuations
Extreme events

Figure 1.1: Uncertainty-based design domains, according to [94].

ferred just as *reliability-based optimization*) concentrating on worse-case scenarios that occur only in extreme events. *Robust design optimization* covers daily fluctuations minimizing the price as well as sensitivity to small changes in model inputs such as loading, structural parameters, and geometry. SCHUËLLER and JENSEN [173] include the third group of optimization under uncertainty - *model updating and system identification*; the goal is to reduce discrepancies which arise when the model prediction is compared with the test data [173].

Reliability-based design optimization (RBDO) minimizes an objective function (e.g. a structural weight, maximal displacement, benefits, construction costs or expected lifetime costs) with respect to deterministic constraints as well as probabilistic constraints evaluating a probability of failure. Optimization, as well as reliability assessment, require repetitive computation of a structural response with different settings of uncertain parameters and design variables; this is the most computationally expensive part of the problem. It is possible to evaluate a probability of failure analytically only for a particular type of problems (for instance, by Gauss quadrature approaches, Laplace Approximation approaches) which is limiting for RBDO applications. Exact computation is impractical [3]. *Approximation techniques* such as a First-order reliability method (FORM) or Second-order reliability method (SORM) and *simulation techniques* such as a crude Monte Carlo (MC) and variance reduction techniques are commonly used. FORM [25, 139, 156] is frequently preferred for its speed and only a few necessary evaluations of the performance measure, see e.g. [34, 55, 64]. The drawback is that the obtained reliability assessment is accurate only if the limit state function of the problem is a hyperplane and the variables are independent and normally distributed – the more nonlinear the problem, the more significant the error. A nonlinearity can also be hidden in the transformation from the non-normal space into the Gaussian space. If the problem is nonlinear, SORM [139] utilizing second-order derivatives is more precise since SORM respects the curvature around the most probable point. However, SORM is also computationally more expensive; fortunately, Prof. KARL BREITUNG [22] developed a second-order correction formula that decreases computational expenses.

Simulation techniques have better accuracy in predicting a probability of failure. However, they are much more time demanding due to repetitive evaluations of the virtual simulation, such as a finite element model. Monte Carlo simulation [131] itself is the most universal and robust method for the assessment of reliability [142]. However, it requires millions of samples for low probabilities of failure. Advanced simulation techniques, such as an Importance sampling [13, 98, 189] or Subset simulation [8, 189, 198], demand less computational effort while evaluating the performance function, which can still last quite a long time. Since the accuracy of the probability of failure is needed, reducing demands of simulation techniques is essential.

Some model of the original model with a very similar behaviour can replace the exact model to speed up the design process; those models of models are called *meta-models* or *surrogate models*. The original function is evaluated only in some so-called support points; any support point is a combination of input variables of the model. These support points form a *Design of the experiments* (DoE) [133]. Scientific experiments, as well as industrial, very often use a *uniform* Design of experiments [99] since it is robust and only a small number of support points are necessary to get a large amount of information about the relationship between the function inputs and outputs [117]. For some specific tasks, it is advantageous to use information concentrated in an important subdomain of the task. An initial small Design of experiments can detect this significant region. Subsequently, an *updating* procedure adds more support points there to improve the meta-model behaviour mimicking the original model. Initial meta-models do not have the same accuracy as the original models particularly in locations that are the most interesting for engineering design, e.g. a vicinity of the border between the safe and the failure domain called a *limit state*. Adaptive updating of meta-models can make a meta-model more accurate in those significant locations. Many types of meta-models are used in RBDO, namely Response surfaces [75, 116, 202, 208], Support vector machines [11, 12, 198], Kriging [59, 60], and Polynomial chaos expansion [36].

1.2 Aims and objectives of the thesis

This thesis aims to develop a methodology that provides fast and computationally simple solution of multi-objective reliability-based design optimization with cost and reliability defined in the objectives. In general, the double-loop approach of RBDO provides the most precise outputs; nevertheless, it is the most computationally expensive from all methodologies. If more than one objective function is considered, the result is a set of solutions defining the best trade-off between competing objectives, a Pareto-optimal front. The multi-objective formulation of the double-loop approach increases already high computational expenses. In 2012, the Department of Mechanics, CTU in Prague, where the author is doing her research, started to cooperate on joint projects with AIRBUS Defence & Space, Germany, that was funded under contract by ESA. The very first project was the Advanced Nozzle Extension Design Methodology (ANED-M) [42] in which the third work package was dedicated to the optimization of a nozzle extension geometry to minimize its weight and maximize its reliability concerning prescribed uncertain loading. The mentioned project was a part of a Future Launchers Preparatory Programme (FLPP) [67]. This thesis includes and extends the methodology that we developed in the framework of the project. Moreover, parts of this thesis were successfully employed in another joint project named Reliability Analysis and Life Prediction with Probabilistic methods (RALP) with AIRBUS D&S. One of the main issues of the formerly mentioned project was the price of reliability. However, the classical formulation of multi-objective RBDO has reliability defined only as a constraint inequality. Therefore, it is necessary to reformulate the classical formulation of MO-RBDO so that reliability becomes an objective function and the resulting Pareto-optimal front will provide compromise solutions concerning price and the just mentioned reliability.

The main objective of this thesis is to formulate, implement, optimize, and validate a methodology for multi-objective reliability-based design optimization. The most computationally expensive part is evaluating the reliability of the system whilst utilizing crude Monte Carlo simulation and the original complicated model (e.g. a finite element model). Therefore, the partial objectives are in a reduction of computational costs in these three significant areas:

1. a multi-objective optimization algorithm should converge with the fewest solutions as possible;

2. a fast evaluation of the probability of failure by advanced simulation techniques while maintaining acceptable result accuracy;

3. and effective and suitable meta-models mimicking the behaviour of the original (full) model in the critical regions.

The overall structure (see also Figure 1.2) of this thesis statement takes the form of eight chapters, including this introductory chapter. Chapter Two begins by laying out the Reliability-based design optimization concepts, and it describes the essential methodologies from the author's point of view. The third chapter concentrates on multi-objective optimization and its assessment metrics. The fourth chapter is concerned with an introduction to structural reliability and statistics and subsequently, to reliability assessment techniques for a sufficient probability of failure evaluation; this chapter covers a description of one approximation technique and six simulation methods including a well-known Monte Carlo simulation. The fifth chapter presents meta-models and their updates. We introduce there two novel approaches; one approach assembles meta-models on a specific segment of DoE, and the second approach modifies the design matrix of the meta-model. Updates of meta-models are classified according to the mode of the meta-model application. Note that the update of the meta-model differs in reliability assessment and RBDO areas. Chapter Six is the central part of this thesis, proposing a new multi-objective procedure utilizing advanced simulation methods, updating of the meta-models and possible ways of parallelization. The seventh chapter shows numerical studies for the proposed methodology. The last chapter concludes this work.

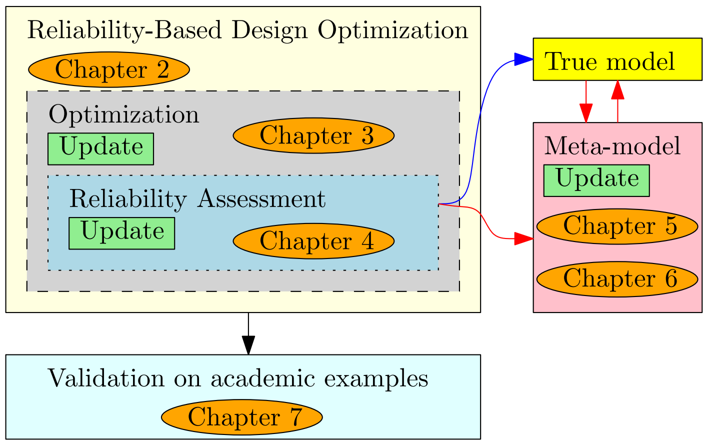


Figure 1.2: Double-loop RBDO formulation. Three different updates are available in RBDO. The first update is in the main optimization routine in the outer loop that improves the design. The second update is in the Reliability assessment with the meaning of the improvement of the probability of failure evaluation (e.g. adding new sample points). The third update is in the meta-model part where and update adds new DoE points into existing DoE.

Chapter 2

Reliability-based design optimization

Parts of this chapter are reproduced from the author's contributions [87, 89, 151].

A standard mathematical definition of RBDO is as follows

$$\min_{\mathbf{d}\in\mathbb{D}} \quad C(\mathbf{x},\mathbf{d}) \tag{2.1}$$

$$\text{s. t.} \quad H_i(\mathbf{d}) \leq 0,\ i = 1,\ldots,n_e \tag{2.2}$$

$$p_{F,j}(\mathbf{x},\mathbf{d}) \leq p_{F,j}^{tol},\ j = 1,\ldots,n_p. \tag{2.3}$$

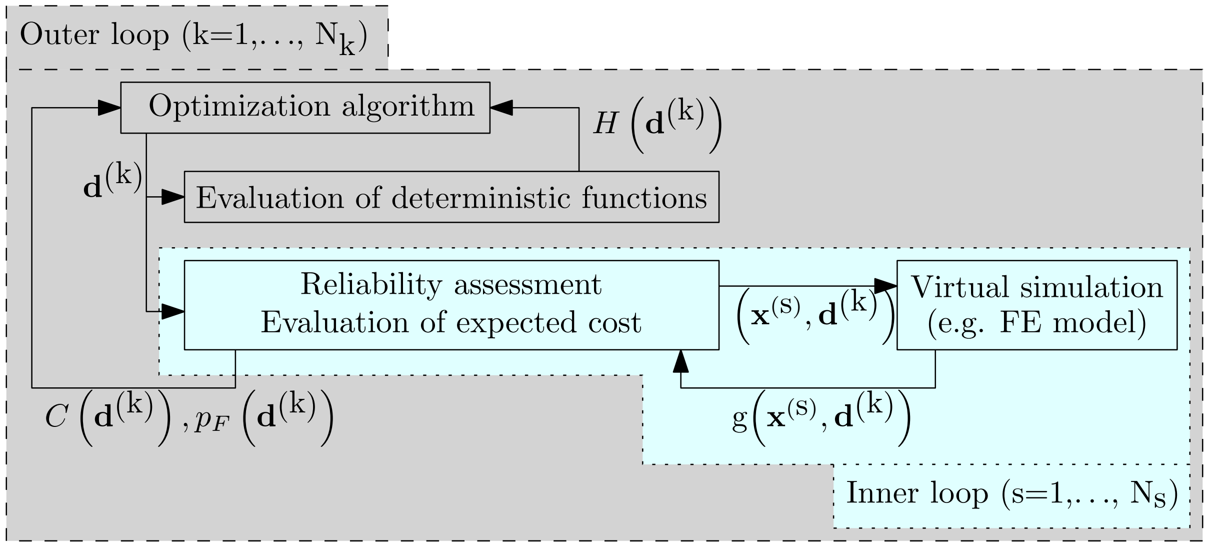
Optimal values of design variables arranged in a vector $\mathbf{d}$ minimize the cost function $C(\mathbf{x},\mathbf{d})$. The vector $\mathbf{d}$ contains deterministic variables or probability distribution parameters (e.g. the mean of random variables) and the elements of $\mathbf{d}$ are part of the design space $\mathbb{D}$. The cost function can be influenced only by deterministic values of $\mathbf{d}$ or by a combination of $\mathbf{d}$ and uncertain parameters arranged in a vector $\mathbf{x}$. All outputs values of constraints $H_i(\mathbf{d})$ has to be less than or equal to zero where i is from 1 to n_e and parameter n_e is equal to the total number of constraints. A probability of occurrence of j^{th} event $p_{F,j}(\mathbf{x},\mathbf{d})$ has to be less than or equal to a prescribed tolerable threshold $p_{F,j}^{tol}$. A number of events n_p is equal to one in the simplest cases.

2.1 Methods for solving Reliability-based design optimization

The methods for solving RBDO problems can be classified into three groups, namely *double-loop approaches* (also referred to as *two-level approaches*), *single-loop approaches* (also referred to as *mono-level approaches*), and *decoupling approaches* [3, 190].

2.1.1 Double-loop approach

A double-loop approach is the most direct approach and the most accurate method to solve RBDO problems [115]. It consists of two merged loops, where the outer loop solves the designing process, and the inner loop appraises the reliability evaluations. The optimization algorithm, be it a single- or a multi-objective technique, a mathematical programming or a stochastic optimization, proposes a combination of design variables values $\mathbf{d}^{(k)}$. Vector $\mathbf{d}^{(k)}$ is used for evaluation of deterministic functions $H(\mathbf{d}^{(k)})$ and, at the same time, it is passed to a reliability assessment routine to solve probability constraints and subsequently the cost function in one iteration k of the outer loop. The reliability assessment needs several virtual simulations runs of the limit state function to evaluate the probability of failure with different values of uncertain parameters $\mathbf{x}^{(s)}, s = 1, \ldots, N_s$. The reliability assessment, therefore, occurs in the inner loop with fixed design variables values $\mathbf{d}^{(k)}$. The algorithm runs the inner loop N_s times and the number of runs N_s differs for various methods. Parameter N_s can even differ for different k steps in the same algorithm. After running the entire inner loop for the fixed value $\mathbf{d}^{(k)}$, the algorithm evaluates whether the reliability condition $p_F\left(\mathbf{d}^{(k)}\right)$ is satisfied. The last step in one iteration of the outer loop is the expected cost evaluation $C\left(\mathbf{d}^{(k)}\right)$ and subsequent overall evaluation of the k^{th} iteration of the outer loop. The algorithm then suggest different values of for the design variables $\mathbf{d}^{(k+1)}$. The outer loop is run N_k times. The schematic representation of the double-loop approach is depicted in Figure 2.1. The double-loop approach may be divided into four main classes: (i) double-loop utilizing sampling techniques; (ii) Reliability index approach (RIA) that is similar to class (i) – the main difference is that RIA uses FORM; (iii) Performance measure approach which may be understood as an inverse task to RIA; and (iv) our proposed approach converting single-objective optimization into the multi-objective formulation.


Outer loop (k=1,..., N_k)
Optimization algorithm
$H\left(\mathbf{d}^{(k)}\right)$
$\mathbf{d}^{(k)}$
Evaluation of deterministic functions
Reliability assessment
Evaluation of expected cost
$\left(\mathbf{x}^{(S)}, \mathbf{d}^{(k)}\right)$
Virtual simulation
(e.g. FE model)
$C\left(\mathbf{d}^{(k)}\right), p_F\left(\mathbf{d}^{(k)}\right)$
$g\left(\mathbf{x}^{(S)}, \mathbf{d}^{(k)}\right)$
Inner loop (s=1,..., N_S)
Figure 2.1: Schematic representation of a double-loop RBDO problem inspired by [190].

Sampling-based approach

The described methodology utilizes a direct reliability assessment in the inner loop. Since the probabilistic constraint evaluation is the most expensive part, the essential process is to use a proper tool which has good accuracy and is computationally attainable. Crude Monte Carlo simulation is the most robust tool for all reliability assessment tasks, but it is also the most expensive one. Nevertheless, replacing the true function with a meta-model can reduce the computational time, e.g. as Monte Carlo utilizing Neural Networks [142]. Advanced simulation techniques in RBDO such as Importance sampling [13, 189], Subset simulation [189, 198] or Asymptotic sampling [147] require less computational effort than crude Monte Carlo simulation.

Reliability index approach

FORM is a very common method for solving this reliability task; AOUES and CHATEAUNEUF [3] and other authors as well (e.g. [187]) call the application of FORM the *Reliability index approach* (RIA). A j^{th} stochastic constraint (Equation 2.3) is transformed to

$$\beta_j(\mathbf{x}, \mathbf{d}) \geq \beta_j^{tol}, \tag{2.4}$$

where $\beta_j(\mathbf{x}, \mathbf{d})$ is a reliability index for prescribed design variables $\mathbf{d}$ and stochastic inputs $\mathbf{x}$ evaluated as

$$\beta_j(\mathbf{x}, \mathbf{d}) = -\Phi^{-1}\left(\int \cdots \int_{G(\mathbf{x},\mathbf{d}) \leq 0} f_X(\mathbf{x}) \mathrm{d}\mathbf{x} \right) \tag{2.5}$$

and β_j^{tol} is a minimum tolerable value. The index β represents the shortest distance between the origin of the standard normal space and the most probable point (MPP) as shown in Figure 2.2. Searching for the shortest distance is a nested optimization problem defined as

$$\min_{\mathbf{u}} \quad ||\mathbf{u}|| \tag{2.6}$$

$$\text{s. t.} \quad G(\mathbf{d}, \mathbf{x}) \leq 0, \tag{2.7}$$

where random variables $\mathbf{x}$ are mapped into the independent standard normal space $\mathbb{U}$ as a vector $\mathbf{u}$, e.g. by a Nataf's model [25, Chapter 2]. A lot of general-purpose optimization software makes use of the implementation of this approach [3].

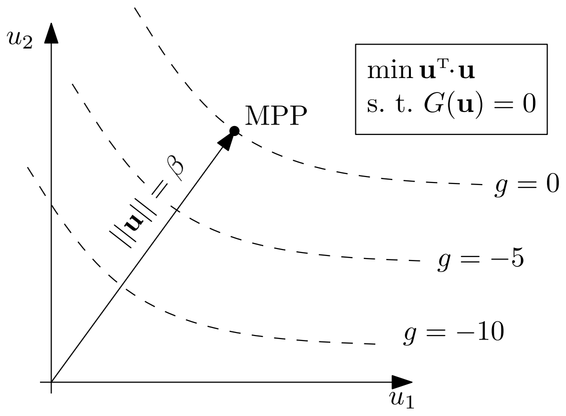

u_2
u_1
$\min \mathbf{u}^{\mathrm{T}} \cdot \mathbf{u}$
s. t. $G(\mathbf{u}) = 0$
MPP
$||\mathbf{u}|| = \beta$
$g = 0$
$g = -5$
$g = -10$

Figure 2.2: Forward reliability analysis [18]. The most probable point (MPP) in a standard normal space is the shortest distance between the origin and the limit state contour $g = 0$.

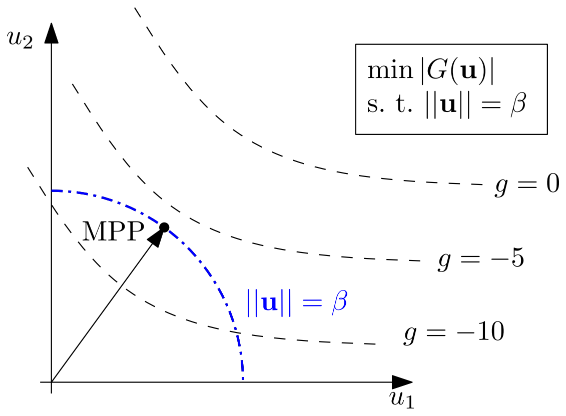


Figure 2.3: Probability measure approach [18]. MPP in a standard normal space lies on a circle with a perimeter β, and it has the smallest absolute value of the performance function.

Performance measure approach

TU et al. [187] formulated an inverse reliability assessment called a *Performance measure approach* (PMA) (also referred to as *inverse FORM* or *inverse reliability approach*) which seeks the threshold value $g^\star$ for the given probability of failure $\bar{p}_F$. PMA is the inverse method to the reliability index approach, where the probabilistic constraint is formulated as

$$G^{\star}(\mathbf{d},\mathbf{x}) \geq 0 \tag{2.8}$$

$$G^{\star}(\mathbf{d},\mathbf{x}) = F_G^{-1}\left(\int\limits_{G(\mathbf{x},\mathbf{d})\leq 0}\cdots\int f_X(\mathbf{x})\mathrm{d}\mathbf{x}\right) = F_{\mathbf{G}}^{-1}(\Phi(-\beta)) = g^{\star}. \tag{2.9}$$

Function F is a cumulative distribution function of a given distribution, and $G^{\star}$ is a probabilistic performance measure. The most probable failure point lies on a circle with the centre at the origin and the prescribed perimeter β in the standard normal space. The goal is to search for the minimum of the absolute value of the performance function on this constraint as depicted in Figure 2.3 as

$$\min_{\mathbf{u}} \quad G(\mathbf{d},\mathbf{x}) \tag{2.10}$$

$$\text{s. t.} \quad ||\mathbf{u}|| = \beta_j^{tol}. \tag{2.11}$$

This method is announced to be more robust and efficient than RIA [187] because the search direction is simply determined by exploring the spherical equality constraint [3].

Multi-objective approach

A single-objective problem can be reformulated into a multi-objective form[1] [147]. This approach is advantageous for several reasons. The most important reason is that we get an insight into the relationship between reliability and design cost. The trade-off between structural safety and the total costs is linear only in very special cases, and the whole Pareto front could help to identify a preferable solution. A single-objective optimization is usually not able to find all optima in a multimodal problem; thus, the multi-objective algorithm is employed to show the vicinity of the trade-off near the limit value of the prescribed reliability index. We prefer to maximize the reliability index obtained by the inverse cumulative distribution function of the standard normal distribution $\beta = \Phi^{-1}(1 - p_F)$ instead of minimizing the probability of failure due to the scaling issues. The deterministic constraint in Equation 2.14 remains the same as well as the threshold β_j^{tol} for the minimum acceptable structural reliability. However, the reliability constraints are optional in Equation 2.15; thus, the cutting off can be applied in the postprocessing stage. The mathematical formulation reads

$$\min_{\mathbf{d}\in\mathbb{D}} \quad C(\mathbf{d}) \tag{2.12}$$

$$\max_{\mathbf{d}\in\mathbb{D}} \quad \beta(\mathbf{x},\mathbf{d}) \tag{2.13}$$

$$\text{subject to} \quad h_i(\mathbf{d}) \leq 0,\ i = 1,\ldots,n_I, \tag{2.14}$$

$$\beta(\mathbf{x},\mathbf{d}) \leq \beta^{tol}. \tag{2.15}$$

[1]RBDO considers structural reliability to be an inequality constraint because the limits on which it is proposed are known. The basic recommended minimum value of the reliability index is 3.8 for reliability class RC2 and 4.3 for RC3 at 50 years reference period for the ultimate limit states [66, 155]. Outside the structural engineering area, such as industrial electronics [203, 204], or wind energy engineering [43], these limits do not have to be defined. Such type of optimization is called Reliability and Cost Optimization. The task is multi-objective in the basic definition but can be converted to single-objective, e.g. by using the weighted sum approach [70] or the epsilon-constraint method [170].

2.1.2 Single-loop approach and Decoupled approach

In a **single-loop approach**, some or all probabilistic constraints are replaced with the approximate deterministic constraints. For example, LI et al. [115] transform the limit state function into the standard normal space, and then they add to it a constant $\beta^{\star}$, which is equal to the minimum allowable reliability index. The constant $\beta^{\star}$ is just a distance which abridges the feasible region. This constraint is then wholly deterministic. Unfortunately, there is no free lunch. Since this method is an approximation in the sense of PMA, the first error is in an approximation involved in the FORM. The second error comes from the approximation of the shifted limit-state function. For the linear limit state functions, both approximation errors are equal to zero. For a spherical limit state function, only the first type of approximation error (in FORM) is involved. The error grows with the increasing nonlinearity of the problem as well as for higher ranges of failure probabilities.

A decoupled approach separates a reliability assessment and an optimization task. DU and CHEN [58] formulated a *Sequential optimization and reliability assessment method* (SORA); they shift boundaries of violated deterministic constraints to the feasible direction based on the reliability information obtained in the previous cycle [199]. In each cycle, the limit state function is used as a deterministic constraint, and the deterministic optimum is found on the boundary. After a location of the most probable point, the deterministic constraint is updated such that the most probable point lies on the deterministic boundary. The routine is repeated until the objective converges and the reliability requirement is achieved when all the shifting distances become zero [58].

Although these methods seem to be promising, it is necessary to note that they are based mainly on the FORM approximation. The methods can completely fail for highly nonlinear problems with a relatively low probability of failure and for multimodal as well as for multiple-failure-mode problems. These methods are frequently based on mathematical programming methods which can lead to premature convergence or completely fail in seeking the optimum. The selection of a particular starting point for those algorithms may affect the efficiency considerably as well [190].

Chapter 3

Multi-objective optimization

Parts of this chapter are reproduced from the author's contributions [147, 154].

Optimization is a process that seeks one or more feasible solutions within the bounds of possibility, e.g. familiarity with the problem and consecutively knowledge of the possible solutions, accessibility of the software or availability of the hardware and its computational time. The necessity is to know how to formulate the problem. As professor CHRISTIAN BUCHER has always asked the audience on his lectures: "Which is the cheapest bridge over the river?". Subsequently, he replies: "The cheapest bridge is no bridge.". The optimization itself has several different classifications:

1. according to a type of variables including *discrete variables* with a predefined set of concrete numbers or binary variables, and *continuous variables* from a whole domain of definition;

2. according to the extent of variables concerning *unconstrained* tasks with every possible value and *constrained* tasks that have to fulfil specific conditions;

3. according to the solvability of the problem, including *uni-modal problems*, in which the solution space contains only one global optimum, and *multi-modal problems* with several different local optima.

An optimization problem with only one objective function is called single-objective optimization. The goal is to find a combination of design variables for which the values of the objective function is at its minimum/maximum satisfying constraint equalities, inequalities, or both, within the predefined range of design variables. Maximization and minimization tasks are convertible to each other by multiplying the objective function by -1; this principle is called the duality. There exists a broad range of optimization algorithms dealing with single-objective optimization, which can be divided into deterministic methods and stochastic methods. *Deterministic optimization algorithms* always provide the same answer with the same number of objective function evaluations if the program is run multiple times within the same search-space, the same termination conditions, and the same starting point. These algorithms are based on strict sequences of mathematical principles concerning gradient-based techniques or combinatorial methods. Deterministic algorithms as types of solvers can be mistaken for optimization with deterministic variables. While the deterministic algorithms do not include the uncertainty in the optimization process, and therefore the results are reproducible, the optimization with deterministic variables is related to decision variables without concerning any probability. Stochastic optimization algorithms concerning heuristic and meta-heuristic algorithms are based on random search and making stochastic decisions. Therefore, the answer to this type of algorithm is different for different runs within the same conditions. Since the computers usually employ pseudo-random number generators, the answers for different runs may be identical with the same initial seed of the generator. This setting causes determinism of the stochastic algorithms [170].

In multi-objective optimization, the objectives can be conflicting or non-conflicting, and this phenomenon influences the final result. For two or more non-conflicting objectives, the set converges into one optimal solution. In this case, the objective functions are synergic, and the multi-objective optimization can help for faster convergence [194]. Within the formulation of multiple antagonistic objective functions, there exists not only one single optimal solution but a whole set of optimal non-dominating solutions. Solutions in this set are all indifferently good; each solution predominates in one objective and is worse in other objectives. The best optimal non-dominated solutions are projected into the decision variable space (or simple decision space) as a Pareto-optimal set (Pareto set for short), and into the objective space as a Pareto-optimal front (Pareto front in short). The objective space is created by mapping the decision space by objective functions. For each solution point $\mathbf{x}$, one point $\mathbf{z}$ in the objective space exists created by $\mathbf{z} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_n(\mathbf{x}))$, where f_i is i^{th} objective function. It is also necessary to note that not every algorithm can provide true Pareto-optimal solutions. Therefore we assume that all solutions gained by the meta-heuristic algorithms are only the approximation of the Pareto-optimal solutions. Moreover, in some cases, there is an infinite number of Pareto-optimal solutions.

The multi-objective problems are frequently transformed into a single-objective optimization via several possible methods. Weighted sum approach is a popular choice. In this procedure, as the name of the procedure suggests, the objectives are weighted by user-defined weights, and these weighted objectives are summed into a final single-objective function. A different set of weights provides a different optimal solution. The utilization of several sets of weights could provide the whole set of solutions for some types of tasks, e.g. for convex fronts. However, in case that a front of the objective values of the solution set is non-convex, the weighted sum approach cannot achieve all points on this front by changing the weights. Another popular method for transforming a multi-objective task into a single-objective one is ϵ-constraint method. Only one objective is selected here, and the rest of them is converted to constraints. It may be problematic that the solution depends on the chosen constraint limits.

The pure multi-objective optimization solvers can deal with all problems, including non-convex sets or discontinuous sets. Several interesting points in the Pareto-optimal front exist, and they are plotted in Figure 3.1. *Ideal objective vector* $\mathbf{z}^*$ is composed of optima $\mathbf{z}^{*(i)}$ of each i^{th} objective function in case, that they are individually optimized by a single-objective optimizer. Essentially, this vector is composed of the lower bounds of each objective function. Only in case, that objective functions are synergic or identical; this point is attainable by multi-objective optimizer; otherwise, this vector corresponds to a non-existent solution. The knowledge of this vector is good for normalization of the objective values for some type of solvers. *Utopian objective vector* $\mathbf{z}^{**}$ is a non-existent solution for any types of multi-objective problem formulation. It is composed of components that are marginally smaller than that of the ideal objective vector[1] [45]. Therefore,

$$\mathbf{z}^{**(i)} = \mathbf{z}^{*(i)} - \epsilon_i \tag{3.1}$$

[1]We assume minimization hereafter.

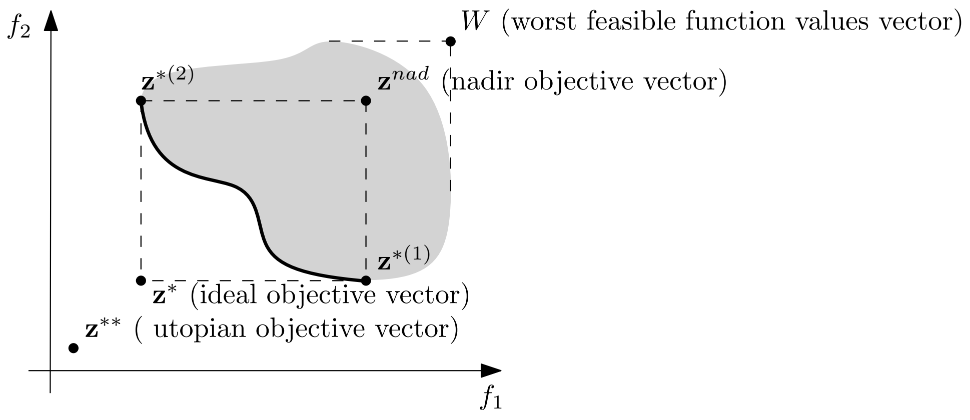

f_2
W (worst feasible function values vector)
$\mathbf{z}^{*(2)}$
$\mathbf{z}^{nad}$ (nadir objective vector)
$\mathbf{z}^{*(1)}$
$\mathbf{z}^{*}$ (ideal objective vector)
$\mathbf{z}^{**}$ (utopian objective vector)
f_1

Figure 3.1: The ideal, utopian, and nadir vector as referred in [45].

with $\epsilon_i > 0$ for all $i = 1, \ldots, M$, where M is a number of objective functions. *Nadir objective vector* $\mathbf{z}^{nad}$ is assembled by upper bounds of each objective in the whole Pareto-optimal set. This vector can be confused with a solution composed of the worst feasible function values in the entire search space W. Although the ideal point is usually easy to obtain, the nadir objective vector is hard to find since the knowledge of entire Pareto-optimal front is required. For 2D objective space, nadir point can be estimated as $\mathbf{z}^{nad} = (f_1(\mathbf{x}^{*(2)}), f_2(\mathbf{x}^{*(1)}))$ if $\mathbf{z}^{*(1)} = (f_1(\mathbf{x}^{*(1)}), f_2(\mathbf{x}^{*(1)}))$ and $\mathbf{z}^{*(2)} = (f_1(\mathbf{x}^{*(2)}), f_2(\mathbf{x}^{*(2)}))$. For some optimization algorithms, it is efficient to use the normalization of objective functions, that can be carried out by

$$f_i^{norm} = \frac{f_i - \mathbf{z}_i^*}{\mathbf{z}_i^{nad} - \mathbf{z}_i^*}. \tag{3.2}$$

Plenty of multi-objective optimizers utilize principles of domination. Every two solutions from the objective space obtained by mapping from the decision space have to be compared with each other. For a single-objective optimization, the comparison of two solutions is facile; in the case of minimization, $\mathbf{x}^{(1)}$ is better than $\mathbf{x}^{(2)}$ if $f(\mathbf{x}^{(1)}) < f(\mathbf{x}^{(2)})$. For more objective functions, seven cases can arise. We will now assume minimization of all objectives as being said above. In case 1, the solution $\mathbf{x}^{(1)}$ is strictly better than the solution $\mathbf{x}^{(2)}$, in other words, the solution $\mathbf{x}^{(1)}$ strictly dominates the solution $\mathbf{x}^{(2)}$ if $\mathbf{x}^{(1)}$ is better than $\mathbf{x}^{(2)}$ in all objectives, mathematically written $\mathbf{x}^{(1)} \prec\prec \mathbf{x}^{(2)}$. In case 2, $\mathbf{x}^{(1)}$ is not worse than $\mathbf{x}^{(2)}$ in all objectives and $\mathbf{x}^{(1)}$ is strictly better than $\mathbf{x}^{(2)}$ in at least one objective. This result means that $\mathbf{x}^{(1)}$ dominates $\mathbf{x}^{(2)}$, i.e. $\mathbf{x}^{(1)} \prec \mathbf{x}^{(2)}$. In case 3, $\mathbf{x}^{(1)}$ is not worse than $\mathbf{x}^{(2)}$ in all objectives, this event is called that $\mathbf{x}^{(1)}$ weakly dominates $\mathbf{x}^{(2)}$, mathematically written $\mathbf{x}^{(1)} \preceq \mathbf{x}^{(2)}$. In case 4, the solution $\mathbf{x}^{(1)}$ is incomparable or indifferent with $\mathbf{x}^{(2)}$ if neither $\mathbf{x}^{(1)}$ weakly dominates $\mathbf{x}^{(2)}$ or vice versa. These solutions are non-dominated to each other. In case 5, the solution $\mathbf{x}^{(1)}$ is weakly dominated by $\mathbf{x}^{(2)}$. Moreover, this event is opposite to case 4. Similarly, case 6 is opposite to case 2, i.e. $\mathbf{x}^{(1)}$ is dominated to $\mathbf{x}^{(2)}$. The last case 7 responds to the contrary of case 1, i.e. $\mathbf{x}^{(1)}$ is strictly dominated by $\mathbf{x}^{(2)}$. Figure 3.2 shows an illustrative example of used principles of domination. A bold polygonal, rectangular chain depicts the Pareto-optimal front. In some literature, the authors have connected the points on the Pareto-optimal front with a curve. However, there is no guarantee that any solution on the curve is Pareto-optimal or feasible. According to [71], it is safer to draw a boundary in objective space that is defined by an envelope of solutions that are verifiably dominated by those Pareto-optimal solutions. This envelope is called an *attainment surface*.

It is not always convenient to use minimization for all objectives. In case that objectives need to be maximized, the shape of the Pareto-optimal front changes. Figure 3.3 shows all possible combinations of minimization and maximization of two objective functions.

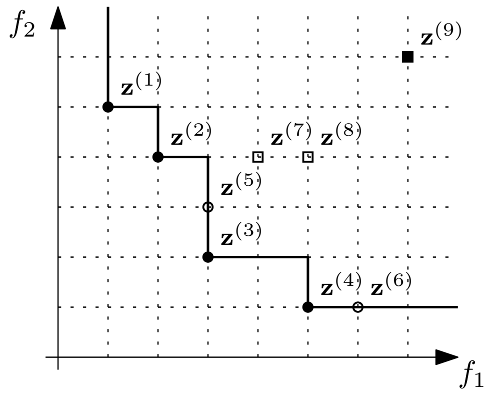


Figure 3.2: Examples of dominance relations of objective vectors for minimization of both objective functions. A bold polygonal chain depicts the Pareto-optimal front. Solutions $\mathbf{z}^{(1)}$,$\mathbf{z}^{(2)}$,$\mathbf{z}^{(3)}$, and $\mathbf{z}^{(4)}$ are non-dominated to each other; therefore, they create the Pareto-optimal front (case 4). Solution $\mathbf{z}^{(5)}$ is dominated by solution $\mathbf{z}^{(3)}$ as well as solution $\mathbf{z}^{(6)}$ is dominated by solution $\mathbf{z}^{(4)}$ (case 3). These points do not belong to the Pareto-optimal front. Solution $\mathbf{z}^{(7)}$ is dominated by solution $\mathbf{z}^{(2)}$ and strongly dominated by solutions $\mathbf{z}^{(3)}$ and $\mathbf{z}^{(5)}$. Solution $\mathbf{z}^{(8)}$ is dominated by solutions $\mathbf{z}^{(2)}$,$\mathbf{z}^{(4)}$, and $\mathbf{z}^{(7)}$; it is also strongly dominated by solutions $\mathbf{z}^{(3)}$ and $\mathbf{z}^{(5)}$. Solution $\mathbf{z}^{(9)}$ is strongly dominated by all other solutions.

If the optimization problem is complex, and the exact methods are not applicable, the meta-heuristic strategies [80] are convenient to use [206]. Simulated annealing [31, 101] , TABU search [78, 79], Evolutionary algorithms [63], and Swarm intelligence algorithms [57, 61] are just a few groups of algorithms belonging to these strategies. Evolutionary algorithms are prevalent algorithms that are inspired by evolutionary theory. They are well-liked for their simplicity, robustness, no need of the good initial estimation of a solution nor the derivatives or for a capability of finding the global optimum in the presence of several local optima. There exist several multi-objective versions of evolutionary algorithms, to name a few Pareto-archived evolution strategy (PAES) [103], Strength-Pareto evolutionary algorithm (SPEA) [213], improved SPEA (SPEA2) [212], Pareto envelope-based selection algorithm (PESA) [40], improved PESA (PESA-II) [39], Multi-objective evolutionary algorithm based on decomposition (MOEA/D) [207], Dynamic multi-objective evolutionary algorithm (DMOEA) [201], or Non-

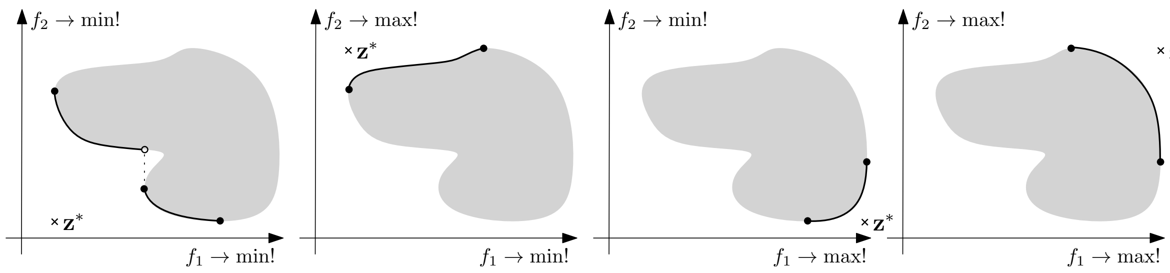


Figure 3.3: Examples of Pareto-optimal fronts for different cases of minimization, maximization, or both, of two objective functions. The bold continuous curve represents Pareto-optimal fronts; filled circles depict points belonging into these fronts. In contrast, the empty circle represents a point excluded from the Pareto-optimal front. Crosses represent ideal points $\mathbf{z}^*$ for all four cases.

dominated sorting genetic algorithm II (NSGA-II) [47]. For our purposes, we need a solver that does not use an archive. Since we will use an updated Design of Experiments and therefore, have an updated meta-model for each generation of the algorithm, the values of the objective functions will differ for identical individuals in different generations. The values of the objective functions obtained from meta-models will be more accurate with each next generation. When using the archive, we would have to recalculate the entire archive with every new generation, and that would be very computationally demanding. Unfortunately, PAES, SPEA, SPEA2, PESA, and PESA-II use an external population [106] that would have to be re-evaluated each time when DoE and subsequently the meta-model is updated. NSGA-II uses a different approach for keeping the elitist solution in the population instead of storing an external list of solutions. It is also well tested, efficient, popular, widely used, and it uses only a few parameters to be easily tuned [1, 37, 106]. According to Google Scholar, a freely available web search engine, indexing full text and metadata of academic publishing including results from databases Scopus and Web of Science, a group of NSGA algorithms is the most commonly used algorithm of the algorithms mentioned above. As of April 2020, NSGA, NSGA-II, and NSGA-III have 28,700 occurrences, SPEA, and SPEA 2 have 11,800 occurrences, PAES has 4,090 occurrences, MOEA/D has 3,480 occurrences, PESA, and PESA-II have 1,430 occurrences, and DMOEA has 274 occurrences.

Multi-objective optimization is usually efficient from two up to three objectives. If more than three objectives exist, it is advantageous to use many-objective optimization solvers. In comparison with the many-objective problems, the multi-objective problems are possible to visualize easily since 3D space need not any special visualization methods of multidimensional data such as a Scatter plot matrix, Bubble chart, Radial coordinate visualization, Parallel coordinates, and Heatmaps [188]. The proportion of non-dominated solutions in a randomly selected set of solutions is low if the maximum of the three objectives is defined. In many-objective problems, the number of non-dominated solutions in a randomly selected set of solutions exponentially grows with an increasing number of objectives [48]. Plenty of multi-objective optimization solvers emphasize non-dominated solutions in a population, and if plenty of solutions are non-dominated, it is hard to accommodate new solutions in a population if an elite-preserving algorithm is used [95]. The multi-objective optimization solvers use various types of diversity-preservation operators such as a crowding distance or a clustering operator; these operators become computationally intensive in a large dimensional space of objectives [48]. Several many-objective optimization solvers manage all the mentioned problems. If we consider evolutionary algorithms, we can name for example a Non-dominated sorting genetic algorithm III (NSGA-III) [48], Hypervolume estimation algorithm for multi-objective optimization (HypE) [10], Reference Vector Guided Evolutionary Algorithm (H-RVEA) [141], and Multi-objective evolutionary algorithm based on decomposition (MOEA/D) [207]. In some cases, the many-objective optimization problem often degenerates to a result with a low-dimensional Pareto-optimal front [48], and classical multi-objective algorithms are suitable to use.

3.1 Non-dominated sorting genetic algorithm II (NSGA-II)

Genetic algorithms (GAs) are a population-based method that creates an offspring population from a parental population by selection, crossover and mutation operators. While there is no guarantee of success, the offspring population is often better than the educated guess [1]. The first conception of a GA [93] was single-objective, which means that only one objective function can be optimized at one time resulting in the unique optimal set of solutions.

The *Non-dominated sorting genetic algorithm* (NSGA) [180] is a multi-objective genetic algorithm that utilizes a sorting according to ranks for emphasizing good points and niche method for maintaining stable sub-populations of good points. The second generation of this algorithm, the Non-dominated sorting genetic algorithm II (NSGA-II), was firstly published in [47]. It is an improved approach of NSGA where the main disadvantages were a high computational complexity of non-dominated sorting, lack of elitism, and need for specifying a sharing parameter for obtaining a wide variety of solutions. This newer version of the algorithm deals with all of these disadvantages to obtain a better solution much faster [49].

NSGA-II creates a random parental population at the beginning with N members. The original paper [47] does not define exactly any closer specification about this random population. We prefer to keep the uniformity of the design; therefore, we chose Halton sequences with a linear transformation to the original design space bounds since we have good experience with these sequences. However, any other quasi-uniform Design of Experiments or LHS would work well. If there exist constraints in the design space, it is necessary to use them even for the first parental population. The prescribed objective functions subsequently evaluate each member of the parental population; the objective functions transform a quasi-uniform distributed members in the design space into a non-uniform one in the objective space [38]. NSGA-II minimizes the fitness; therefore, if any of the objectives are maximized, it is necessary to transform it to minimization by multiplying its objective function values by minus one. The next offspring population with N members is created by using classical operators as a selection, a crossover and a mutation (see, e.g. [170] for more details). To select two parental members for two offspring individuals creation, NSGA-II uses a binary tournament selection with two criteria – a non-dominated sorting and a crowding distance. The tournament selection compares two members; if one member has a lower rank than the other, the member with lower rank is assigned into the mating pool. If two members have equal ranks and one has a higher crowding distance, the member in the lesser crowded region is placed into the mating pool. If two members have the same rank and the same crowding distance, one is chosen randomly to the mating pool. After the finished tournament selection, the mating pool contains N members. DEB et al. recommend using a Simulated binary crossover (SBX) [46] as a crossover operator in [49]. As a mutation operator, we use a standard Gaussian mutation with zero mean and σ_M standard

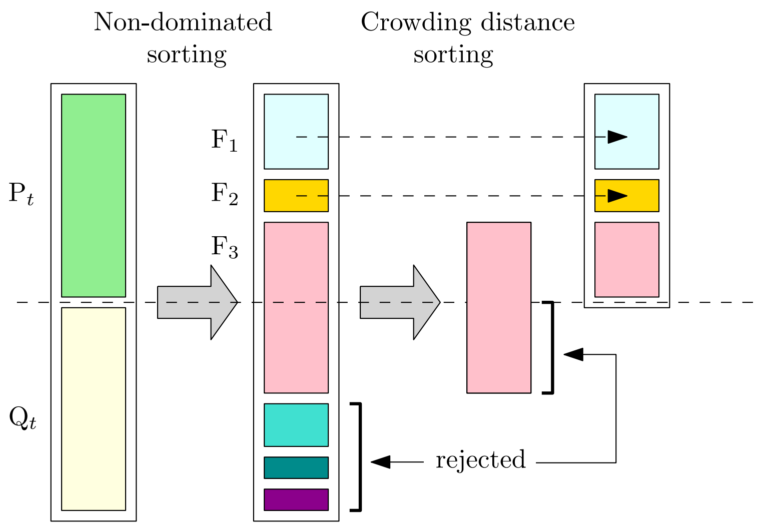


Figure 3.4: Schematic of the NSGA-II procedure.

deviation; σ_M is equal to 20% of the total difference between the upper and lower bound of each design variable. Each offspring is transformed from the design space into the objective space by the objective functions. NSGA-II is an elitist evolutionary algorithm preserving the elite members of the parental population. All parents (μ) and offspring (λ) individuals create a combined $(\mu + \lambda)$ population. The best members from $(\mu + \lambda)$ population are subsequently chosen to create the new parental population with N members. A non-dominated sorting assigns ranks to all members in $(\mu + \lambda)$ population, and all the slots for the next parental population are filled with the individuals sorted according to ranks from non-dominated sorting. The last included level is usually larger than the number of free slots and therefore can be accepted only partially. This situation is evident from Figure 3.4, where the third non-dominated front F_3 depicted with the pink colour does not fit into free slots. A crowding distance evaluates all potential candidates in this last but larger front. NSGA-II prefers diversity, and therefore the solutions that have larger crowding distance values are chosen to fill the free positions. An optimization run terminates after a predetermined number of generations.

SBX randomly choose two solutions from the mating pool to create two offspring individuals [16]

$$y_1, k = \frac{1}{2}[(1-\beta_k)x_{1,k} + (1+\beta_k)x_{2,k}], \tag{3.3}$$

$$y_2, k = \frac{1}{2}[(1+\beta_k)x_{1,k} + (1-\beta_k)x_{2,k}], \tag{3.4}$$

where individuals$x_{1,k}$ and $x_{2,k}$ are two randomly selected members from the mating pool and β_k is a sample from a random number generator having the density

$$p(\beta) = \begin{cases} \frac{1}{2}(\eta+1)\beta^{\eta}, & \text{if } 0 \le \beta \le 1, \\ \frac{1}{2}(\eta+1)\frac{1}{\beta^{\eta+2}}, & \text{if } \beta > 1, \end{cases} \tag{3.5}$$

where η is a distribution index – the higher, the closer created offspring are to parents. If a uniformly distributed random number $u(0,1)$ is used, this distribution can be obtained as

$$\beta(u) = \begin{cases} (2u)^{(1/(\eta+1))}, & \text{if } u(0,1) \le \frac{1}{2} \\ (2(1-u))^{-1/(\eta+1)}, & \text{if } u(0,1) > \frac{1}{2}. \end{cases} \tag{3.6}$$

Non-dominated sorting approach [47] assigns a rank to each individual; this rank is equal to its non-domination level – the smaller, the better. The first non-dominated level has rank equal to 1, the next-best level rank 2, and the last level l has rank l. Figure 3.5a) shows sorting into consecutive fronts, where each front has a different marker. The non-dominated sorting

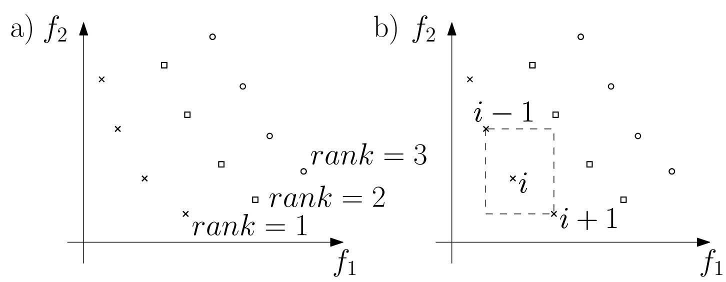


Figure 3.5: Calculations for NSGA-II (both objective functions have to be minimized) of a) Non-dominated rank and b) Crowding distance.

computes two characteristics for each member, first, the domination count n_x that represents the number of solutions dominating the solution $\mathbf{x}$, and second, the set of solutions s_x that the solution $\mathbf{x}$ dominates. The first non-dominated front F_1 has the domination count of all members equal to 0. This non-dominated front is removed from all fronts and the next front F_2 is located. Non-dominated sorting decreases the value in the domination count n_x by one for each dominated solution on the s_x list; this procedure is performed for each solution from the front F_1. Members in the front F_2 has the domination count n_x equal to zero and they are non-dominated to each other. The procedure terminates when all members are sorted into non-dominated fronts, or the needed number of members has its rank.

The non-dominated sorting approach solves constraint handling implicitly without having any penalty parameters and penalty constraints. If we compare two solutions, there exist three possible situations: one solution is feasible, and one is not; both solutions are feasible; both solutions are infeasible. The non-domination rank is always better for feasible solutions than for infeasible ones. If both solutions are feasible, the non-domination rank is better for the solution that dominates the other solution. If both solutions are infeasible, the non-domination rank is better for the solution with a smaller constraint violation. DEB et al. in [49] recommend to sum up all constraint violations of all constraints together. RAY et al. in [161] propose a non-domination check of constraint. The non-dominated sorting approach computes individually three ranks for each solution in the population – $R_{obj}$for the objective functions, R_{con} for constraints, and R_{com} for their combination. All feasible solutions having the best R_{com} rank are chosen to create the next population. If there are more free slots in the population, new individuals are selected from the remaining solutions by giving importance to the R_{obj} ranking in the selection operator and R_{con} in the crossover operator. According to [49], NSGA-II has worked better with their constraint handling approach; therefore, we use a summation of the constraints in our work.

A crowding distance is a metric that evaluates the density of solutions surrounding a particular solution in the population [47]. The solutions for which the crowding distance is calculated are sorted according to each objective function value in their ascending order of magnitude. For each boundary solution, i.e. the solution with the smallest and the largest function values, an infinite distance value is assigned to the crowding distance. For other solutions, a distance to the neighbouring solutions is evaluated for each objective function value. All the distances are summed for each solution

$$CD_i = \sum_{j=1}^{J} cd_{i,j}, \tag{3.7}$$

where CD_i is a crowding distance for the i^{th} solution, $cd_{i,j}$ is a partial crowding distance considering only the j^{th} objective and J is the number of objectives. We prefer to have normalized objective functions, and, therefore, we evaluate a distance between neighbouring solutions for one objective as

$$cd_{i,j} = \frac{|f_j(x_{i+1}) - f_j(x_{i-1})|}{f_j^{\max} - f_j^{\min}}. \tag{3.8}$$

Since we compare the solutions within only one front in one population, we do not need a nadir point, as stated in the general normalization in Equation 3.2, but the current maximum value $f_j^{\max}$ is sufficient. Figure 3.5b) shows the schema for the above equations, where a horizontal or a vertical distance indicated by a dashed line is a partial crowding distance $cd_{i,j}$ for the first and the second objective, respectively, and their sum represents the crowding distance CD_i for the i^{th} solution.

Several modifications of NSGA-II exists. An Omni-optimizer[51] differs from NSGA-II mainly in a modified domination principle, the so-called ε-domination. A decision space-based niching NSGA-II (DN-NSGA-II) [120] differs in two improvements, a niching method is used to create the mating pool, and a selection operator is altered. A Double-Niched Evolutionary Algorithm (DNEA) [123] uses a special environmental selection operator and a double-sharing function with two fine-tuning niche radius parameters each one for design space and objective space, respectively. An Improved NSGA-II Algorithm [68] combining NSGA-II and a tabu search improves the ability of local search.

3.2 Performance measures

There are two aims in multi-objective optimization; the first one is progressing towards the Pareto-optimal front; the second one is to maintain a diverse set of solutions in the non-dominated front. Since the commonly used algorithms are meta-heuristics, and therefore they utilize random numbers and operations, the resulting Pareto-optimal fronts and sets are not identical for several different runs. Therefore, it is necessary not only to compare results obtained from different multi-objective optimization algorithms but as well as for different stochastic runs. The final statement of the metric evaluation is not only a single value but an expected value of the given metric for the particular example and solver. Several categorizations of performance measures exist. One of them is a classification into unary and binary metrics. The *unary metrics* operate only with the one particular solution set, sometimes requiring some other external information. The unary metric is defined as the mapping from the solution set into the set of real numbers [104], that reflects a particular quality aspect. The *binary metrics* compare front A to front B and determine, which one is better. Further categorization is according to the goal, which is pursued by multi-objective optimization. There can be metrics evaluating the accuracy, diversity among the solutions, and their combination.

3.2.1 Metrics evaluating closeness to the Pareto front

This metric class follows a convergence of the set A to the Pareto-optimal set P^* as well as the distance between these two sets A and P^*. If the Pareto-optimal set is unknown, any other reference set can be used. The richer the Pareto-optimal set or the reference set, the more accurate the metric is.

Generational distance

This metric is binary. Therefore, it needs the Pareto-optimal front P^* or some reference front and the resulting approximation of the Pareto-optimal front A provided by the multi-objective optimizer. This metric represents a value of the distance between the front A from the front P^*, mathematically written as [45, 192]

$$G = \frac{\left(\sum_{i=1}^{n} d_i^p\right)^{1/p}}{n}. \tag{3.9}$$

For parameter p equal to 2, variable d_i represents Euclidean distance in the objective space between each vector of the front A and the nearest member of the front P^*; distances are evaluated for n numbers of vectors in the front A. The smaller the generational distance, the better the approximation of the Pareto-optimal front is. The best available solution is equal to zero; if this case occurs, the approximation of the Pareto-optimal front is identical to the true

Pareto-optimal front. According to [165], this metric was the second most used metric in EMO[2] conferences from 2005 to 2013 with 26 citations.

Two set coverage metric

The Two set coverage metric is binary; therefore, two solutions sets in the objective space are necessary, front A and front B. It calculates the proportion of solutions included in the front B, which are weakly dominated by solutions in the front A, mathematically written as

$$C(A,B) = \frac{|\{b \in B | \exists a \in A : a \preceq b\}|}{|B|}, \tag{3.10}$$

where operator $|\cdot|$ denotes the number of points in the set, and $\preceq$ is an operator for weak dominancy. If solutions in A weakly dominate all members of B, then the measure $C(A,B)$ equals to one. On the other side, if any member of the front the B is not weakly dominated by members in the front A, then the measure $C(A,B)$ equals to zero. This operator is not symmetric and therefore $C(A,B) \neq 1 - C(B,A)$. ECKART ZITZLER proposed this metric in his PhD thesis [211]. According to [165], this metric was the fourth most-used metric in EMO conferences from 2005 to 2013 (17 citations).

3.2.2 Metrics evaluating diversity among non-dominated solutions

This class of metrics observe the distribution of solutions and their spread. The optimal distribution is uniform in which all distances among solutions are similar; the optimal spread is as extensive as possible covering the whole attainable feasible front.

Spacing

The Spacing metric determines the distribution of vectors throughout the approximation of the Pareto-optimal front A. Since this metric operates only on one resulting front, the metric is unary. JASON R. SCHOTT proposed this metric in his master's thesis [171] as

$$s = \sqrt{\frac{1}{n}\sum_{i}^{n}\left(\bar{d} - d_i\right)^2}. \tag{3.11}$$

Parameter n denotes the number of vectors in the front A. Variable d_i is defined as a distance measure

$$d_i = \min_j \left(|f_1(\mathbf{x}^{(i)}) - f_1(\mathbf{x}^{(j)})| + |f_2(\mathbf{x}^{(i)}) - f_2(\mathbf{x}^{(j)})|\right). \tag{3.12}$$

Variable $\bar{d}$ represents a mean of all d_i. The smaller the spacing value s, the better the spacing between all vectors is. The ideal case is if s equals zero, which means that all members of the front A are equidistantly spaced. The original formulation is only for two objective functions. Reference [45] presents an extension into M objective functions:

$$d_i = \min_{j \wedge i \neq j} \sum_{m=1}^{M} |f_m(\mathbf{x}^{(i)}) - f_m(\mathbf{x}^{(j)})|. \tag{3.13}$$

[2]EMO is a bi-annual international conference series, dedicated to advances in the theory and practice of evolutionary multi-criterion optimization.

This distance measure is different from the Euclidean distance and represents the minimum value of the sum of absolute differences between objective functions evaluated in $\mathbf{x}^{(i)}$ and any other solution from the front A. This metric also does not consider all distances between all solutions stepwise, therefore if solutions are distributed in pairs, where points are close to each other but these pairs are distant from other pairs, the metric provides a skewed result. According to [165], this metric was the seventh most-used metric in EMO conferences from 2005 to 2013 with six citations.

Spread

The Spread metric is similar to the Spacing metric; however, it utilizes Euclidean distance instead of the Manhattan metric, and it employs some additional information about extreme possible solutions. This metric was proposed in [45, 50] as

$$\Delta = \frac{\sum_{m=1}^{M} d_m^e + \sum_{i=1}^{n-1} |d_i - \bar{d}|}{\sum_{m=1}^{M} d_m^e + (n-1)\bar{d}}, \tag{3.14}$$

where M equals to a number of objective functions, d^e is the Euclidean distance between extreme solutions and the boundary solutions of the approximation of the Pareto-optimal front A, d_i is a distance measure between neighbouring solutions, and n represents a number of solutions in the front A. The algorithm of this metric follows. First, all solutions are sorted according to one objective function values. Second, Euclidean distances between consecutive solutions are evaluated. Third, the average distance $\bar{d}$ from all distances d_i is computed. Fourth, extreme solutions fitting a curve parallel to that of the true Pareto-optimal front are obtained. Furthermore, fifth, the Euclidean distances between these extreme solutions and the boundary solutions of the front A are evaluated. The best available solution is that Δ is equal to zero, which occurs for uniformly distributed vectors. For worse distributed solutions, Δ-metric rises even above one. According to [165], this metric was the fourth most-used metric in EMO conferences from 2005 to 2013 (17 citations).

3.2.3 Metrics evaluating closeness and diversity

The combination of the classes mentioned above is possible. This class of metrics then provide information on convergence and spread.

Hypervolume

This metric evaluates the volume in the objective space of the solid that is bounded by attainable surface, and straight lines from a reference point R parallel to coordinate axes. Since it uses only information about the approximation of the Pareto-optimal set A, the metric is unary. However, additional information about the reference point is needed, that has to be defined by the user. The nadir objective vector $\mathbf{z}^{nad}$ or the worst feasible function values vector W can be used as the reference point. This metric value is therefore dependent on the choice of the reference point. According to [50], hypervolume is calculated by a union of all hypercubes that arise in the space

$$S = \text{volume}\left(\bigcup_{i=1}^{n} v_i\right), \tag{3.15}$$

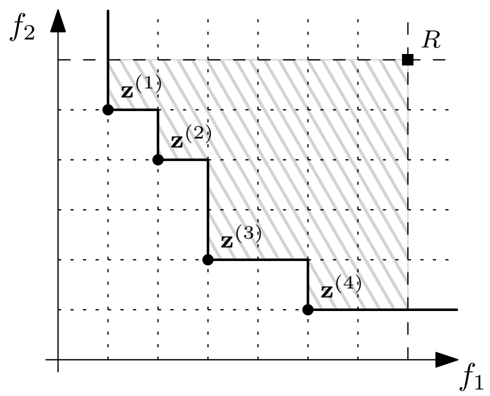


Figure 3.6: The hatched area has the meaning of hypervolume in the objective space with two objective functions for this illustrative example. Solutions $\mathbf{z}^{(1)}$, $\mathbf{z}^{(2)}$, $\mathbf{z}^{(3)}$, and $\mathbf{z}^{(4)}$ create the Pareto-optimal front and R represents the reference point. If every little square has an area equal to one, the hypervolume is equal to 21.

where v_i is the hypervolume of the i^{th} hypercube. Figure 3.6 shows the hypervolume in objective space with two objective functions. According to [165], this metric was the first most used metric in EMO conferences from 2005 to 2013 with 91 citations.

Coverage difference of two sets

This metric combines the Hypervolume metric and the coverage of two sets metric. ECKART ZITZLER proposed it in his PhD thesis [211]. This metric is binary; therefore, it needs information about two sets of decision vectors in the objective space - front A and front B, and a reference point R. The measure gives the information about the size of the space that is weakly dominated by the front A but not weakly dominated by the front B, mathematically written

$$D(A, B) = S(A + B) - S(B), \tag{3.16}$$

where $S(\cdot)$ is a Hypervolume metric described above, and $+$ operator serves for the union of two sets. This measure is not a symmetric operator, therefore $D(A, B) \neq D(B, A)$. The extreme case occurs for $D(A, B) = 0$, which means that the front A dominates the front B. The rising value of $D(A, B)$ indicates that the front A is dominated by the front B.

Chapter 4

Structural reliability

Parts of this chapter are reproduced from the author's contributions [87, 147, 150, 151, 152, 154].

4.1 Introduction into stochastic variables

Stochastic variables are variables with randomness. Random variables are denoted with uppercase letters (e.g. X); lowercase letters (e.g. x) are used for their realizations, i.e. possible values of the variable. They take various values x in the range $-\infty < x < \infty$. Hereafter, we assume that the *probability distribution* describes a random variable (e.g. $f_X(x)$). Variables can be discrete and continuous. For the discrete case, a variable can take only discrete values from a given list; a probability distribution is called a *probability function* or a *probability mass function*. For the continuous case, a variable can take any real value from a given range, $f_X(x)$ is called a *probability density function* (PDF). To determine the probability that the random variable X is lesser than some value x_1, a *cumulative distribution function* (CDF) is introduced

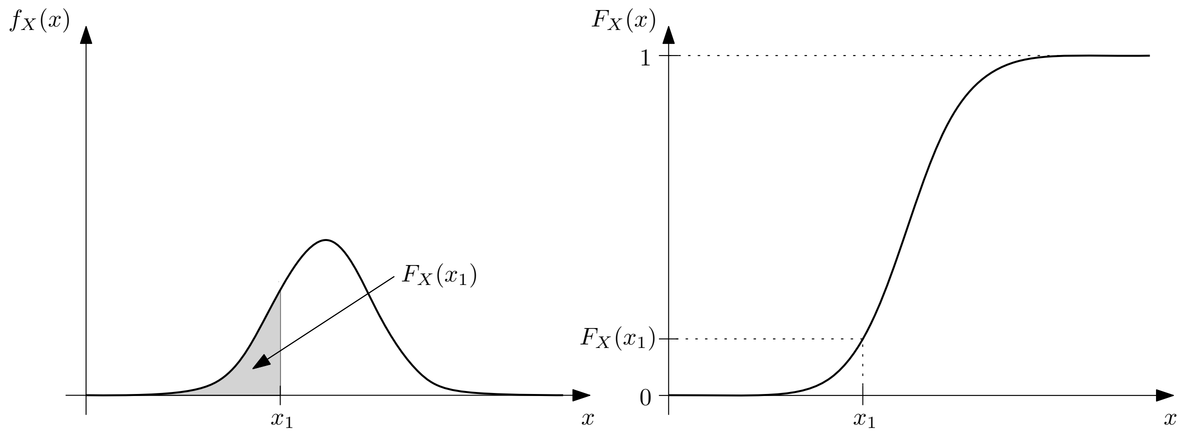


Figure 4.1: A probability density function (left) and a corresponding cumulative distribution function (right) of a random variable.

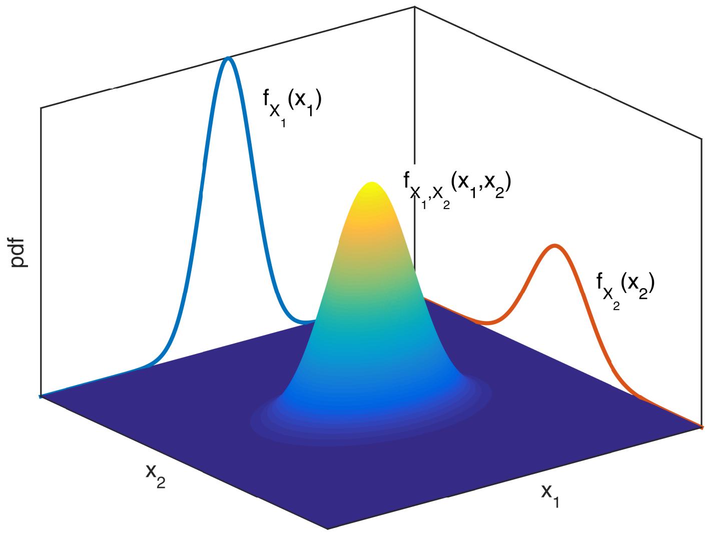


Figure 4.2: A joint probability density function for two random variables X_1 and X_2 and their corresponding marginal probability density functions $f_{X_1}(x_1)$ and $f_{X_2}(x_2)$

as an antiderivative of a function $f_X(x)$ denoted by e.g. $F_X(x)$

$$F_X(x) = \int_{-\infty}^{x} f_X(t)dt. \tag{4.1}$$

Figure 4.1 shows a probability density function of some random variable X on the left and the corresponding cumulative distribution function of the same variable on the right. The shaded area on the left figure corresponds to a value $F_X(x_1)$ of the cumulative distribution function.

Not only one random event can occur at the same time. If there exist two or more random events that happen concurrently, a joint probability density function describes the probability of its occurrence. For 2-dimensional random vector $\mathbf{X} = [X_1, X_2]$, the probability density function is denoted as $f_{X_1,X_2}(x_1, x_2)$, vectorially $f_\mathbf{X}(\mathbf{x})$. It is possible to express only the probability density of a random variable X_1 for all possible values of a random variable X_2 as a marginal probability density function $f_{X_1}(x_1)$ and vice versa. Figure 4.2 shows the joint probability density function for two random variables as well as their marginal probability density functions.

A variable is statistically independent if it is not affected by the probability of occurrence of the other variables. The joint probability density function of mutually independent random variables is their product

$$f_\mathbf{X}(\mathbf{x}) = \prod_{i=1}^{D} f_{X_i}(x_i). \tag{4.2}$$

Random variables can belong to various distributions. Sometimes, it is more convenient to work with uniform or normal distributions. Therefore, a transformation from one distribution to another one exists via equal values of cumulative distribution functions if variables are statistically independent. See Appendix 9 for more details about different distributions used in this thesis. Figure 4.3 shows transformations from the standard uniform space (Fig. 4.3 left) to the standard normal space (Fig. 4.3 middle) and from the standard normal space to an exponential space (Fig. 4.3 right). If a sample from the standard uniform distribution is given as v_1, its cumulative distribution function $F_V(v_1)$ is equal to the value v_1. This statement is, however,

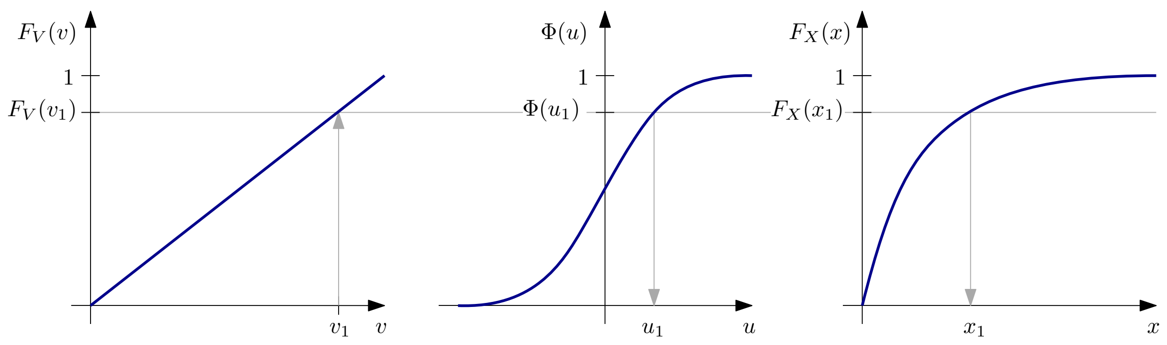


Figure 4.3: Cumulative distribution functions for the standard uniform distribution (left), standard normal distribution (middle), and exponential distribution (right). The horizontal grey line shows the same values of different CDFs. The downwards arrows show the transformation into the different distributions.

valid only in the standard uniform space, where v is from the interval $[0, 1]$. If any other statement about the used uniform space is not given in this thesis, the uniform space is considered as the standard uniform from the interval $[0, 1]$. The transformation from the uniform space into the standard normal space is given via the equality of values of their cumulative distribution functions. The cumulative distribution function for the $F_V(v_1) = \Phi(u_1)$ gives the value u_1

$$u_1 = \Phi^{-1}(v_1), \tag{4.3}$$

where $\Phi^{-1}(\cdot)$ denotes the inverse function of the standard normal CDF. Naturally, the transformation from the standard normal space into the uniform space is also possible via $v_1 = \Phi(u_1)$. The transformation to any other distribution is solved similarly. The sample has a given distribution, e.g. it is standard normally distributed, and it needs to be transformed into any other space. Therefore, the cumulative distribution function of the standard normal distribution is calculated and based on CDF equations, the inverse function of the new distribution is evaluated. This transformation is noted as $T^{-1}(\cdot)$ since

$$x_1 = F_X^{-1}(v_1). \tag{4.4}$$

The opposite transformation is possible, as well. If more random variables are considered in the space, and these variables are statistically independent, then each variable is transformed independently of other variables.

The statistical properties such as mean values or standard deviations can be estimated from probability density functions as well as from available samples. The *mean* (sometimes called the expected value or the average) measures the central tendency of the random variable described by the probability distribution. It is evaluated as a weighted average of every possible value that the random variable can take. In case that a variable is continuous, the mean value is calculated as

$$\mathrm{E}X = \mu_X = \int x f_X(x) dx. \tag{4.5}$$

For a discrete variable, the mean is defined as

$$\mathrm{E}X = \sum_{i=1}^{N} x^{(i)} f_X(x^{(i)}) \tag{4.6}$$

where N is the number of realizations of the random variable, and i is the index of a realization of the random variable. If a random variable is sampled and each sample has the same probability, then the arithmetic mean is a consistent and unbiased estimator for the mean value $\mathrm{E}X$ [25]

$$\mathrm{E}X = \frac{1}{N}\sum_{i=1}^{N} x^{(i)}. \tag{4.7}$$

The mean value is also called the first statistical moment since it represents a distance from the origin to the centroid of the probability density function identical to the first moment of area. Another attractive statistical property is the *median*, which separates the lower half of the sorted samples in ascending order from the higher half. It is calculated as a value of the random variable, where the cumulative distribution function has a value of 0.5, mathematically written as

$$\mathrm{median}(X) = \mathrm{m}(X) = F_X^{-1}(0.5). \tag{4.8}$$

For a discrete variable, the median is evaluated as $x_{(N+1)/2}$ for sorted samples in an ascending order or for attainable realizations of the random variable sorted identically if the number of realizations N is odd. For even N, the median is evaluated as

$$\mathrm{median}(X) = \frac{x_{N/2} + x_{(N/2)+1}}{2}, \tag{4.9}$$

for the attainable realizations sorted in ascending order. The median is also 50th percentile; the percentile divides the sorted data to hundredth, and it indicates which value of the random value realization is under the given percentage of all realizations. Other interesting percentiles are 25th and 75th percentile, which are called the first and the third quartile; the second quartile is the median value.

A *quantile function* $Q(\cdot)$ is an inverse cumulative distribution function associated with a probability distribution of a random variable. It provides the value p of a random variable such that the probability of the variable is less than or equal to that value that equals to the given probability, i.e.

$$x_p = F_X^{-1}(p) = Q(p) \mid \mathrm{Prob}[X \leq x_p] = F_X(x_p) = Q^{-1}(x_p) = p. \tag{4.10}$$

The median mentioned above, as well as quartiles and percentiles, are function values of a quantile function.

Whereas the mean value measures the central tendency, the *variance* measures the spread in the data around the mean. It is also called the second central moment of random variable X equal to $\mathrm{E}[(X - \mathrm{E}(X))^2]$. For the continuous random variable, the variance is calculated as

$$\mathrm{Var}X = \sigma_X^2 = \int (x - \mu_X)^2 f_X(x)dx. \tag{4.11}$$

In case that the random variable is discrete, the variance is evaluated as

$$\mathrm{Var}X = \sum_{i=1}^{N}(x^{(i)} - \mu_X)^2 f_X(x^{(i)}). \tag{4.12}$$

If a random variable is sampled and each sample has the same probability, then the sample variance is

$$\mathrm{Var}X = \frac{1}{N}\sum_{i=1}^{N}(x^{(i)} - \mu_X)^2. \tag{4.13}$$

It is sometimes necessary to use the same units for the measure of dispersion and the mean value. For that reason, a *standard deviation* is introduced as the squared root of the variance

$$\text{std}X = \sigma_X = \sqrt{\text{Var}X}. \tag{4.14}$$

Another remarkable measure for a dispersion of the probability distribution is a *coefficient of variation*, which is a standard deviation normalized by a mean value of variable X

$$\text{CoV}X = \frac{\sigma_X}{\mu_X}. \tag{4.15}$$

The coefficient of variation can be expressed as a percentage (in the range $[0, 100]$) or as a part of the whole (in the range $[0, 1]$). The higher the coefficient of variation, the more stochastic the event is.

Higher central statistical moments are also possible to evaluate, for the k th order the central statistical moment is

$$\text{E}[(X - \text{E}X)^k] = \int (x - \mu_X)^k f_X(x) dx. \tag{4.16}$$

From these centralized statistical moments, another interesting characterizations of the probability distributions proceed; well-known are skewness and kurtosis. The skewness is proportional to the third central moment

$$\text{s}X = \text{E}\left[\left(\frac{X - \mu_X}{\sigma_X}\right)^3\right]. \tag{4.17}$$

If the left tail of the probability density function is longer than the right tail, the skew is negative and vice versa, if the right tail of the probability density function is longer than the left tail, then the skew is positive. If the probability density function is symmetrical as for the normal distribution, the skew is zero. The kurtosis is proportional to the fourth central statistical moment

$$\text{Kurt}X = \text{E}\left[\left(\frac{X - \mu_X}{\sigma_X}\right)^4\right]. \tag{4.18}$$

4.2 Probability of failure

A probability of failure in an n-dimensional space of random variables $X_1 \ldots X_n$ can be computed as

$$p_F = \text{Prob}[G(\mathbf{X}) \leq 0] = \int \cdots \int_{G(\mathbf{X}) \leq 0} f_X(\mathbf{x}) \text{d}\mathbf{x}, \tag{4.19}$$

where $f_X(\mathbf{x})$ is a joint probability distribution function, $G(\mathbf{X})$ stands for a *limit state function* and $G(\mathbf{X}) \leq 0$ denotes the failure domain. A set of several realizations of the limit state function $y_i = G(x_i)$ is a random variable Y.

The main failure modes include static failure, fatigue failure, creep failure, corrosion failure, wear failure, and instability [159]. Those modes emerge separately or together in constraints. The *performance measure* used to define failure conditions is represented by, for example, stress, dynamic stability, temperature, fracture, buckling, displacements, stress intensity factors or eigenfrequencies [64, 205]. The difference between the response and the limit value of the admissible occurrence is called *limit state function* (LSF). The border that distinguishes the safe region and the failure region is called a *limit state* or *limit state surface*. These quantities are evaluated either analytically or by finite element models.

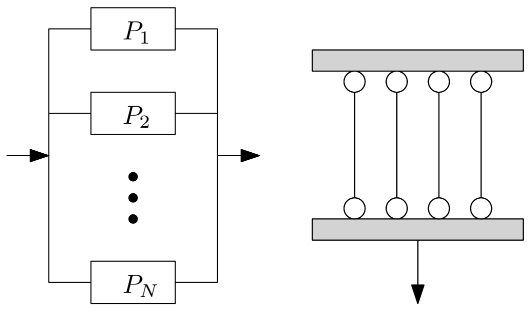


Figure 4.4: Parallel system problem

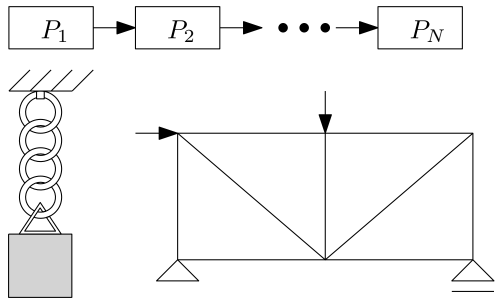


Figure 4.5: Series system problem

The failure probability expressed in Equation 4.19 is valid if only one limit state function is considered. This case usually signifies only one component in the system or only one failure mode. If more components are connected, or more modes of failure may occur, the system problems have to be taken into consideration. Fundamental connections of components are series system problems and parallel system problems. However, their combination in general system problems is possible. These general systems are subsequently dismantled into the series and parallel systems to evaluate their probability of failure.

If individual limit state functions for N components are defined as $G_1(\mathbf{x}), G_2(\mathbf{x}), \ldots, G_N(\mathbf{x})$ and the components are connected into the *parallel system*, the failure region is specified by $G_1(\mathbf{x}) \cap G_2(\mathbf{x}) \cap \cdots \cap G_N(\mathbf{x})$. In other words, the system fails if all components fail. Figure 4.4 on the left shows the general model for the parallel system, where each box P_i represents one component of the system. A typical example of the parallel problem is a structure of two panels interconnected with several independent rods having sufficient bearing capacity; the vertical arrow depicts the loading direction. For statistically independent components, the failure probability of the whole system is

$$p_F = \prod_{i=1}^{N} (p_{F_i}), \tag{4.20}$$

where p_{F_i} is the i^{th} component failure probability and N is a number of components. If a sampling methodology is used and unless otherwise is stated (e.g. for an Enhanced Monte Carlo simulation), it is possible to evaluate the series system probability as

$$p_F = \text{Prob}\left[\max[G_1(\mathbf{x}), G_2(\mathbf{x}), \ldots, G_N(\mathbf{x})] \leq 0\right] = \int \cdots \int_{\max[G_1(\mathbf{x}),G_2(\mathbf{x}),\ldots,G_N(\mathbf{x})]\leq 0} f_X(\mathbf{x})\mathrm{d}\mathbf{x}; \tag{4.21}$$

for a particular sample, all limit state functions are evaluated, the limit state function with the maximum value is selected, and this value is possible to take as the representative value for the assessment whether the system fails or not for the specific combination of parameters given by the sample.

The series system problem has the failure region defined as $G_1(\mathbf{x}) \cup G_2(\mathbf{x}) \cup \cdots \cup G_N(\mathbf{x})$ if the components are connected into the *series system* holding individual limit state functions defined as $G_1(\mathbf{x}), G_2(\mathbf{x}), \ldots, G_N(\mathbf{x})$. In other words, the system fails if even only a single component fails. The influence of the weakest element is put into effect. Figure 4.5 on the top shows the general model for the series system; each box P_i again represents one component of the system. Two typical examples of the series system are shown in the same figure underneath.

The left example is a vertical chain in tension loaded with a concrete block. The right example represents a statically determinate truss structure. If components P_i are mutually independent, then failure probabilities of those components p_{F_i} are also mutually independent, and the series system reliability is

$$p_F = 1 - \prod_{i=1}^{N}(1 - p_{F_i}). \tag{4.22}$$

As for the parallel system, if a sampling methodology is used and unless otherwise is stated (e.g. for an Enhanced Monte Carlo simulation), it is possible to evaluate the series system probability as

$$p_F = \text{Prob}\,[\min[G_1(\mathbf{x}), G_2(\mathbf{x}), \ldots, G_N(\mathbf{x})] \leq 0] = \int \cdots \int_{\min[G_1(\mathbf{x}),G_2(\mathbf{x}),\ldots,G_N(\mathbf{x})]\leq 0} f_X(\mathbf{x})\mathrm{d}\mathbf{x}; \tag{4.23}$$

again, for a particular sample, all limit state functions are evaluated, the limit state function with the minimum value is selected, and this value is possible to take as the representative value for the assessment whether the system fails or not for the specific combination of parameters given by the sample.

To show an evaluation of a failure probability with selected simulation techniques, let us first introduce a simple problem with a readily available solution and subsequently, apply the complicated methods on it. A probability of failure is easy to solve analytically only for some specific problems like a combination of a linear limit state function and normally distributed variables. The classical reliability task is evaluation of the probability of failure of a rod under tension with a capacity of R and a demand S depicted in Figure 4.6. The capacity is a normally

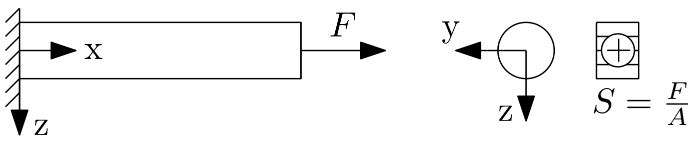


Figure 4.6: A rod under tension (Stress-strength model)

distributed random variable with a mean μ_R equal to 550 MPa and a standard deviation σ_R equal to 50 MPa. The demand is also a normally distributed random variable with a mean μ_S equal to 300 MPa and a standard deviation σ_S equal to 100 MPa. The limit state function is defined as the difference between the capacity R and the demand S

$$G(R, S) = R - S. \tag{4.24}$$

This difference is also called a *safety margin* denoted by Z. Since both variables are normally distributed, and the limit state function is linear, a mean μ_Z and a standard deviation σ_Z of the safety margin are given by

$$\mu_Z = \mu_R - \mu_S, \tag{4.25}$$

$$\sigma_Z = \sqrt{\sigma_R^2 + \sigma_Z^2}. \tag{4.26}$$

The reliability index β is then equal to a ratio of μ_Z to σ_Z, and the probability of failure is then a complement to a cumulative distribution function of the standard normal distribution

$$p_F = 1 - \Phi(\beta), \tag{4.27}$$

$$\beta = \frac{\mu_Z}{\sigma_Z}. \tag{4.28}$$

A reliability index of the stress-strength model defined above is equal to 2.2361, and an adequate probability of failure is equal to 0.0127.

The whole problem is depicted in Figure 4.7. Probability density functions for the capacity and demand as well as for a safety margin are shown in Figure 4.8.

4.3 Approximation techniques

The approximation techniques in reliability assessment are methods for obtaining an estimation of the failure probability by approximating the limit state function by a proper replacement. The main advantage of the approximation techniques is low computational demands in comparison with the simulation techniques discussed in the following section. However, not all reliability tasks can be solved by approximation techniques, since the limit state function is not properly substitutable. The most famous approximation technique is the First-Order Reliability Method [25, 86, 91, 139], which approximates the limit state function by a hyperplane in a design point in the standard normal space. Often, the first-order replacement is insufficient; therefore, the Second-Order Reliability Method [30, 54, 162] is applicable. This method approximates the limit state function by a quadratic function again in a design point in the standard normal space. Another possibility is to correct the First-Order Reliability Method approximation by the knowledge of the curvature of the limit state in the design point by Breitung's formula [22]. Another popular method is a Response surface method [26], which approximates the limit state function by a proper polynomial function.

4.3.1 First-Order Reliability Method (FORM)

The First-Order Reliability Method (FORM) is based on simplifying the probability density function by its transformation into the standard normal space and approximating the limit state function by the first-order Taylor expansion in the design point. The design point (also called the most probable point or the beta point) in the standard normal space is the point lying on the

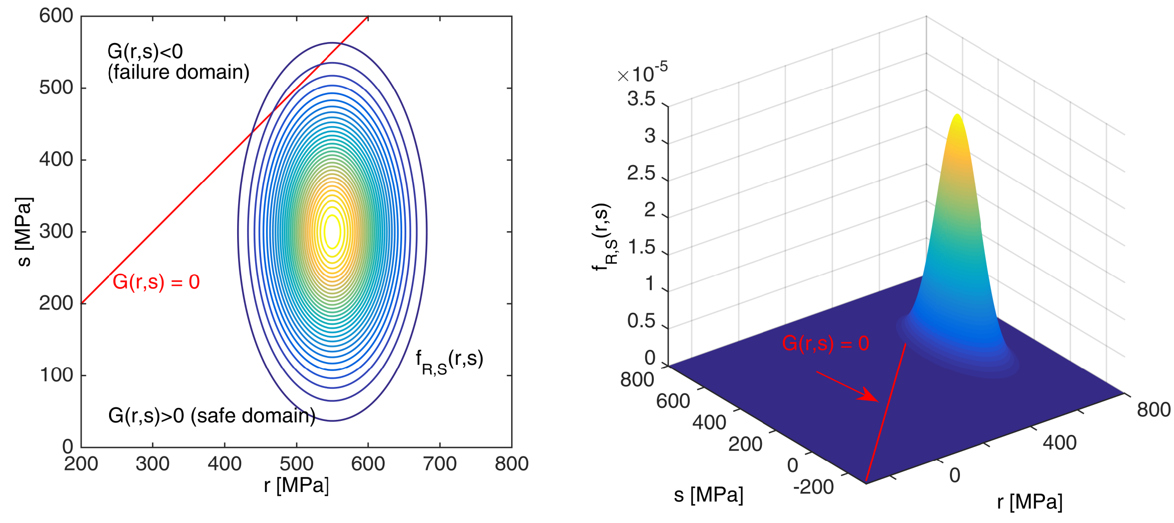


Figure 4.7: A rod under tension: Probability density function $f_{R,S}(r,s)$ and the limit state $G(r,s) = 0$. The left figure shows the contours of the PDF as well as the limit state dividing the space into the failure and safe region. 3D view of the problem is depicted on the right.

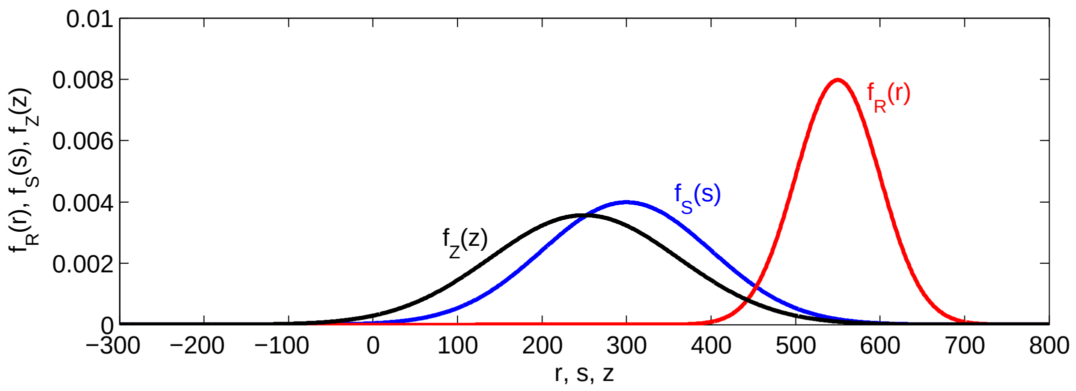


Figure 4.8: A rod under tension: Marginal probability density functions.

limit state with the shortest distance from the origin to the limit state. This minimum distance is equal to a reliability index β. HASOFER and LIND published the essentials of this method in the pioneering work [86] in the seventies. Later on, RACKWITZ and FIESSLER generalized the FORM in [157] for variables that belong to non-normal spaces [23].

The probability of failure is expressed as

$$p_F = \int_{G(\mathbf{X})\leq 0} f_X(\mathbf{x})\mathrm{d}\mathbf{x}, \tag{4.29}$$

$$p_F = \int_{\bar{G}(\mathbf{U})\leq 0} \phi_U(\mathbf{u})\mathrm{d}\mathbf{u}, \tag{4.30}$$

where Equation 4.29 represents the regular evaluation of the failure probability in the original $\mathbf{X}$-space and Equation 4.30 denotes the evaluation of the failure probability in the $\mathbf{U}$-space, which represents the standard normal space. The probability density function in the $\mathbf{X}$-space $f_X(\mathbf{x})$ is therefore transformed into the standard normal $\phi_U(\mathbf{u})$ and for that reason the limit state function $G(\mathbf{X})$ in the $\mathbf{X}$-space needs to be transformed into the standard normal space as well as

$$\bar{G}(\mathbf{U}) \equiv G(T^{-1}(\mathbf{U})). \tag{4.31}$$

The symbol $T^{-1}(\cdot)$ has the meaning of a transformation from the standard normal space into the original space; this transformation is described in Section 4.1 in more detail. The transformation of the limit state function can be made analytically by transforming random variables inside the function. However, if, e.g. some black-box evaluator is used, such as a finite element method, and the gradient is necessary to evaluate numerically, the transformation is possible to be made outside the evaluator. The numerical derivative is possible to evaluate by, e.g. forward, backward, or central difference formula. The central difference formula is the most accurate from the named ones; however, it uses a higher number of evaluations of the limit state function, specifically $(2D + 1)$ evaluations, where D is a number of dimensions of the problem. The backward and the forward difference formula uses only $D + 1$ evaluations of the limit state functions. Whenever the notation $\bar{G}(\mathbf{U})$ is used, function $G(\cdot)$ is understood as the transformed version of the limit state function and vice versa; the notation $G(\mathbf{X})$ means the original untransformed version of the limit state function. After transforming the random variables and the limit state function into the standard normal space, the first-order estimate of the probability failure is then obtained as

$$p_F = \int_{L(\mathbf{U})\leq 0} \phi_U(\mathbf{u})\mathrm{d}\mathbf{u}, \tag{4.32}$$

where $L(\cdot)$ is the linearized version of the limit state function $\bar{G}(\cdot)$.

The Taylor series expansion of an arbitrary one-dimensional function $f(x)$ in a point a is

$$f(x) \cong f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \dots, \tag{4.33}$$

$$f(x) \cong \sum_{n=0}^{\inf} \frac{f^{(n)}(a)}{n!}(x-a)^n, \tag{4.34}$$

where $f^{(n)}(a)$ is the n^{th} derivative of the function $f(x)$ evaluated at point a and $n!$ is the factorial of n. The multi-dimensional version of the Taylor series expansion is according to [163]

$$f(\mathbf{x}) \cong f(\mathbf{a}) + \sum_{i=1}^{D} \frac{\partial f(\mathbf{a})}{\partial x_i}(x_i - a_i) + \frac{1}{2!}\sum_{i=1}^{D}\sum_{j=1}^{D} \frac{\partial^2 f(\mathbf{a})}{\partial x_i \partial x_j}(x_i - a_i)(x_j - aj) + \dots \tag{4.35}$$

where D is a number of input $\mathbf{x}$ dimensions into the function $f(\mathbf{x})$. Vectorially written for the first three addends, the Taylor series in several variables are

$$f(\mathbf{x}) \cong f(\mathbf{a}) + \nabla f(\mathbf{a})^T(\mathbf{x}-\mathbf{a}) + \frac{1}{2}(\mathbf{x}-\mathbf{a})^T \nabla^2 f(\mathbf{a})(\mathbf{x}-\mathbf{a}), \tag{4.36}$$

where $\nabla f(\mathbf{a})$ is the gradient vector of first derivatives of $f(\cdot)$ calculated in the point $\mathbf{a}$ and $\nabla^2 f(\cdot)$ is the Hessian matrix composed of the second derivatives of $f(\mathbf{a})$ in the point $\mathbf{a}$. All vectors are considered as column vectors; The Hessian matrix is symmetrical on the neighbourhood $\mathcal{D}$ if the function $f(\cdot)$ is continuous on the neighbourhood $\mathcal{D}$. However, the first-order reliability method makes use of only the first two terms of Equation 4.36.

The linearization of the limit state function $\bar{G}(\mathbf{u})$ in the form of the first-order Taylor expansion in the design point $\mathbf{u}^*$ is therefore

$$\bar{G}(\mathbf{u}) \cong L(\mathbf{u}) = \bar{G}(\mathbf{u}^*) + \nabla \bar{G}(\mathbf{u}^*)^T(\mathbf{u}-\mathbf{u}^*). \tag{4.37}$$

The first addend of the right-hand side in Equation 4.37 vanishes, since the design point $\mathbf{u}^*$ lies on the limit state and therefore $\bar{G}(\mathbf{u}^*)$ is equal to zero. The gradient of $\bar{G}(\mathbf{u})$ at the design point $\mathbf{u}^*$ is arranged as

$$\nabla \bar{G}(\mathbf{u}^*) = \left(\frac{\partial \bar{G}(\mathbf{u})}{\partial u_1}, \frac{\partial \bar{G}(\mathbf{u})}{\partial u_2}, \dots, \frac{\partial \bar{G}(\mathbf{u})}{\partial u_D} \right)^T \Bigg|_{\mathbf{u}^*}. \tag{4.38}$$

It is a common custom in the first-order reliability method to express the gradient as its normalized negative version as

$$\alpha = -\frac{\nabla G(\mathbf{u}^*)}{||\nabla G(\mathbf{u}^*)||}.$$

By substitution Equation 4.3.1 into Equation 4.37, we get

$$L(\mathbf{u}) = -||\nabla \bar{G}(\mathbf{u}^*)||\alpha(\mathbf{u}-\mathbf{u}^*) = ||\nabla \bar{G}(\mathbf{u}^*)||(\alpha\mathbf{u}^* - \alpha\mathbf{u}). \tag{4.39}$$

In the right-hand side of Equation 4.39, the $\alpha\mathbf{u}^*$ is the shortest distance to the origin from the limit state and therefore, this distance is equal to the reliability index β

$$L(\mathbf{u}) = ||\nabla \bar{G}(\mathbf{u}^*)||(\beta - \alpha\mathbf{u}). \tag{4.40}$$

The first-order approximation of the failure probability is then

$$\hat{p_F} = \Phi(-\beta), \tag{4.41}$$

where $\Phi(\cdot)$ is a cumulative distribution function of the standard normal distribution.

The constrained optimization founds the design point

$$\mathbf{u}^* = \min ||\mathbf{u}||, \quad \text{s.t. } \bar{G}(\mathbf{u}) = 0. \tag{4.42}$$

For the searching for the design point, we can use the classical Hasofer-Lind-Rackwitz-Fiessler (HLRF) algorithm. We can consider a zero vector as a starting point $\mathbf{u}_1$. Next point is computed as

$$\mathbf{u}_{i+1} = \mathbf{u}_i + \mathbf{d}_i, \tag{4.43}$$

where the vector $\mathbf{d}_i$ is

$$\mathbf{d}_i = \left[\frac{G(\mathbf{u}_i)}{||\nabla G(\mathbf{u}_i)||} + \alpha_i^T \mathbf{u}_i\right] \alpha_i - \mathbf{u}_i. \tag{4.44}$$

The column vector α_i is the normalized negative gradient column vector at point i

$$\alpha_i = -\frac{\nabla G(\mathbf{u}_i)}{||\nabla G(\mathbf{u}_i)||}. \tag{4.45}$$

The algorithm is possible to stop when the length vector $\mathbf{d}_i$ is close to zero.

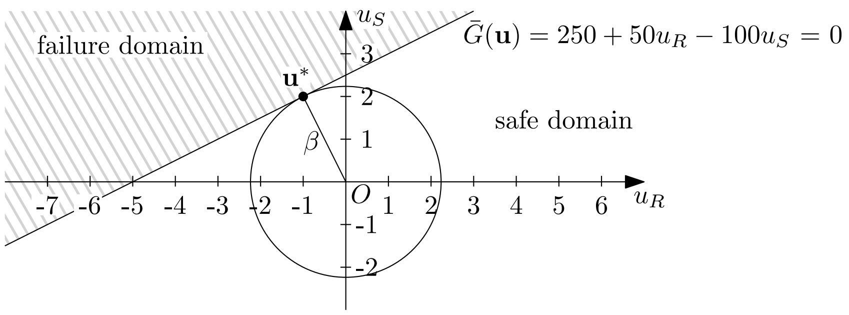


Figure 4.9: A rod under tension: the first-order reliability method. The two-dimensional space is composed of the capacity $R \sim N(550, 50)$ and the demand $S \sim N(300, 100)$, and it is transformed into the standard normal space. The design point $\mathbf{u}^*$ is [-1 2] and β is 2.23607.

Figure 4.9 shows the evaluation of the β-index for the example from Figure 4.6 with two variables: the capacity $R \sim N(550, 50)$ and the demand $S \sim N(300, 100)$. The variables are transformed into the standard normal space

$$R = 550 + 50u_R, \tag{4.46}$$
$$S = 300 - 100u_S, \tag{4.47}$$

The limit state function is transformed as well from $G(\mathbf{X}) = R - S$, where vector $\mathbf{X}$ is composed of $[R, S]$, into

$$\bar{G}(\mathbf{U}) = 250 + 50U_R - 100U_S, \tag{4.48}$$

where vector $\mathbf{U}$ is composed of $[U_R, U_S]$. Since the original limit state function is linear, and the problem is formulated in the normal space, the transformed problem into the standard normal space has the linear limit state function as well. Therefore, the evaluation of the reliability is exact, and the design point is found in one step. The second step serves for validation of the zero vector $\mathbf{d}_i$. All necessary variables are summarized in Table 4.1.

In the example mentioned above, the linearization of the limit state function in the standard normal space is not apparent because the problem is linear in itself even after transformation.

step i	1	2
$\mathbf{u}_i$	$[0,0]^T$	$[-1,2]^T$
$\bar{G}(\mathbf{u}_i)$	250	0
$\nabla\bar{G}(\mathbf{u}_i)$	$[50,-100]^T$	$[50,-100]^T$
$\lVert\nabla\bar{G}(\mathbf{u}_i)\rVert$	$\sqrt{12500}$	$\sqrt{12500}$
α_i	$[-0.447, 0.894]^T$	$[-0.447, 0.894]^T$
d_i	$[-1,2]^T$	$[0,0]^T$
β		2.23607

Table 4.1: A rod under tension: the first-order reliability method and the necessary evaluation of all variables in two steps.

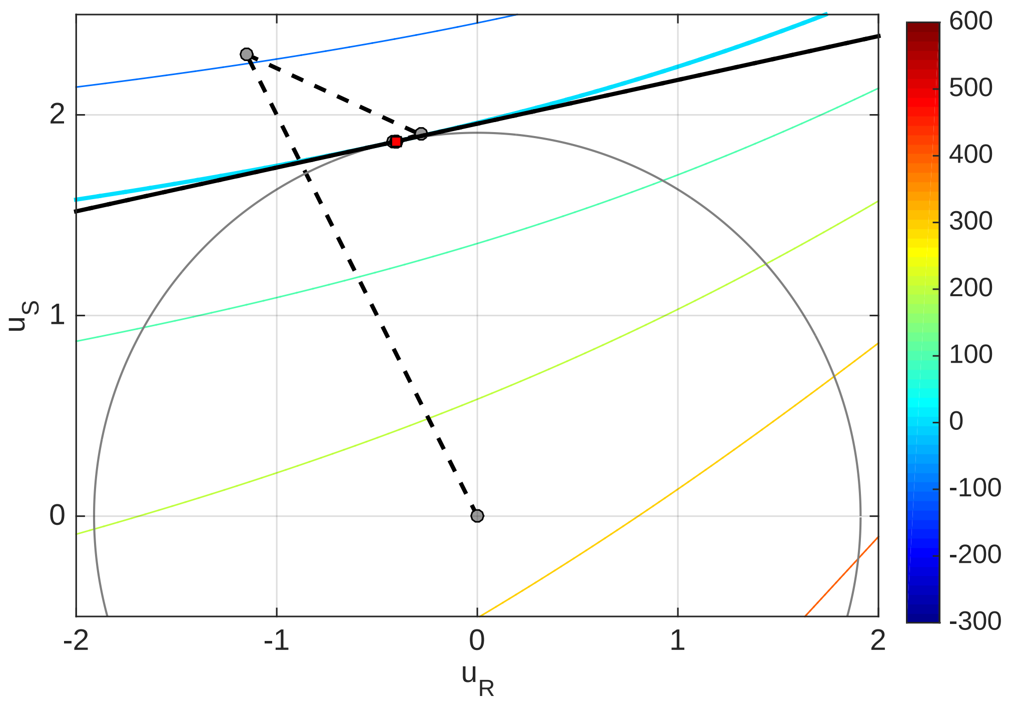


Figure 4.10: A rod under tension in Gumbel space: the first-order reliability method. The two-dimensional space is composed of the capacity $R \sim Gumbel(550, 50)$ and the demand $S \sim Gumbel(300, 100)$, and it is transformed into the standard normal space. The design point $\mathbf{u}^*$ is equal to $[-0.408, 1.867]^T$ and β corresponds with 1.91089.

Therefore, we change the variables from the normal space to the Gumbel space; however, means and standard deviations remain the same. Figure 4.10 shows the contours of the limit state function of the transformed problem into the standard normal space, which are plotted by coloured contours with thicker contour for the limit state. All steps in the FORM iterations are depicted by grey dots and dashed lines. The red square represents the final design point, and the linearized limit state function $L(\mathbf{u})$ in this design point is depicted by the black line for values equal to zero. The circle represents the shortest distance to the limit state, which perimeter is identical to the reliability index.

The system reliability is assessed by linearizing each limit state function $G_j(\mathbf{x})$ at a design point $\mathbf{u}_j$, $j = 1, 2, \ldots, N$, such that the tangent hyperplane approximates the surface $\beta_j - \alpha_j\mathbf{u}$

= 0, in which

$$\alpha_j = -\frac{\nabla G_j(\mathbf{u}_j^*)}{||\nabla G_j(\mathbf{u}_j^*)||}. \tag{4.49}$$

β_j represents a distance from the origin to the j^{th} hyperplane in the standard normal space and α_j is the j^{th} unit normal to the hyperplane [139]. The search for the design point differs in the series and parallel systems. The series system reliability requires the location of as many design points as the number of limit state functions with the same algorithm as for the component reliability in Equation 4.42, i.e.

$$\mathbf{u}_j^* = \min ||\mathbf{u}||, \quad \text{s.t. } \bar{G}_j(\mathbf{u}) = 0. \tag{4.50}$$

The parallel system reliability can utilize the same method of the design points search, or one joint design point can be searched by

$$\mathbf{u}^* = \min ||\mathbf{u}||, \quad \text{s.t. } \bar{G}_j(\mathbf{u}) \le 0, j = 1, \dots, N, \tag{4.51}$$

where multiple inequality constraints are evolved at the same time. The latter method provides a better approximation, but it is more complicated to use [139]. If we suppose that $v_j = \alpha_j \mathbf{u}, j = 1, 2, \dots, N$ are normal random variables with zero means, unit variances, and correlation coefficients $\rho_{kl} = \alpha_k \alpha_l^T, k, l = 1, 2, \dots, N$, it is valid for a series system that [139]

$$p_{F,series} = \text{Prob}\left[\bigcup_{j=1}^{N}(\beta_j \le v_j)\right] = 1 - \text{Prob}\left[\bigcap_{j=1}^{N}(v_j < \beta_j)\right] = 1 - \Phi_N(\mathbf{B}, \mathbf{R}), \tag{4.52}$$

where Φ_N is N-variate standard normal cumulative distribution function with argument $\mathbf{B} = [\beta_1, \beta_2, \dots, \beta_N]^T$ and a correlation matrix $\mathbf{R} = [\rho_{kl}]$; a correlation coefficient is evaluated as $\rho_{kl} = \alpha_k^T \alpha_l$. An estimate of the failure probability is calculated similarly [139]

$$p_{F,parallel} = \text{Prob}\left[\bigcap_{j=1}^{N}(\beta_j \le v_j)\right] = \text{Prob}\left[\bigcap_{j=1}^{N}(v_j < -\beta_j)\right] = \Phi_N(-\mathbf{B}, \mathbf{R}) \tag{4.53}$$

in case that the limit state functions are linearized one by one as in Equation 4.50. In the case of a joint design point found, the failure probability is evaluated as for the component system.

The First-Order Reliability Method needs low computational costs in comparison with sampling methods; however, it works only with some classes of reliability tasks. The FORM provides a wrong approximation of the failure probability for limit state functions in which limit state is strongly curved, especially around the design point. Unfortunately, it is not possible to predict if the approximation of the failure probability is underestimated or overestimated because the limit state is unknown in general. The error also dramatically increases with the growing number of dimensions of the problem. Other problematic tasks seem to be optimization problems with multiple local or global solutions. Frequently used Quadratic programming optimization methods would need several restarts to find the exact design point or can fail if the optimization task is not suitable for this type of mathematical programming solver. A different kind of optimization solvers such as heuristic methods suggests itself; however, these types of controlled sampling methods require plenty of evaluations, which increase computational costs with preserving disadvantages of the approximation methods based on the first-order approximation technique.

4.4 Simulation techniques

Simulation techniques are numerical methods that solve mathematical problems through random sampling. The mapping between the input design space and the output space represented by a limit state function is usually not known by an analytical model but only by an implicit model such as a finite element model, or a boundary element model. The only obtained value from an implicit model is the value of the model for a particular combination of input parameters. The other information such as derivatives needed for approximation techniques is reachable by a numerical derivation, e.g. by forward, backward, or central difference formula. These additional computations increase the number of evaluations of the original model especially for a higher number of variables which makes approximation techniques less effective with respect to low computational demands and the error of the failure probability prediction for nonlinear limit state functions. Therefore, the sampling methods, which require only the information about the value of the model, are applicable. With the development of computers in the forties, the sampling methods have gained their growth as well. Many simulation methods are based on Monte Carlo simulation, which was invented by METROPOLIS and ULAM and published in 1949 [131]. The original purpose of the Monte Carlo was a statistical approach for studying differential equations; however, due to its versatility, it extended into a field of all kinds of integro-differential equations that occur in many branches of the natural sciences [131]. The Monte Carlo method is a very robust tool. Nevertheless, it is also computationally very demanding, especially for solving reliability problems with small probabilities of failure. Therefore, several advanced sampling methods based on the Monte Carlo principle have been developed. These advanced methods reduce the variance of the probability failure estimator by several techniques. The sampling distribution can be modified (e.g. an Importance sampling, or Latin Hypercube sampling). The failure probability can be extrapolated with the expectation of its asymptotic behaviour (e.g. an Asymptotic sampling, Enhanced Monte Carlo simulation, or Scaled sigma sampling). Alternatively, the failure probability can be expressed as a product of larger conditional failure probabilities (e.g. a Subset simulation). In this section, we describe selected simulation techniques with a demonstration of their behaviour on the stress-strength model. All those mentioned methods are used for a reliability index evaluation in multi-objective reliability-based design optimization in Section 7.

4.4.1 Monte Carlo simulations (MC)

A Monte Carlo method is a numerical approach that solves equations from differential and integral calculus by experiments on random numbers. In structural reliability, the probability of failure is an essential measure for the safety of the engineering design. The failure probability from Equation 4.19 is possible to be rewritten via an indicator function $I_G(\mathbf{x})$ that is equal to one for a failure domain $G(\mathbf{X}) \leq 0$ and zero for a safe region $G(\mathbf{X}) > 0$, the failure probability is then

$$p_F = \int\limits_{-\infty}^{\infty} \cdots \int\limits_{-\infty}^{\infty} I_G(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}. \tag{4.54}$$

The probability of failure expressed in Equation 4.54 is an expected value of the indicator function [111]. The Monte Carlo method is a statistical sampling technique; the elemental principle is to generate samples as pseudo-random numbers from a given distribution, then to evaluate the value of the limit state function for all samples, and finally to assess how many samples felt into the failure domain. The ratio between the number of failures n_f and the number of all samples m denotes the estimator of the failure probability, which is

$$p_F \approx \hat{p}_F = \mathrm{E}[I_G(\mathbf{x})] = \frac{1}{m}\sum_{i=1}^{m} I_G(x^{(i)}) = \frac{n_f}{m}. \tag{4.55}$$

The approximate general reliability index β is an inverse cumulative distribution function of the standard normal distribution

$$\beta = \Phi^{-1}(1 - p_F). \tag{4.56}$$

The variance of the failure probability estimator can be computed as [111]

$$\mathrm{Var}\,\hat{p}_F = \mathrm{Var}\left[\frac{1}{m}\sum_{i=1}^{m} I_G(x^{(i)})\right] = \mathrm{Var}\left[\frac{1}{m^2}\sum_{i=1}^{m} I_G(x^{(i)})\right]. \tag{4.57}$$

Since the indicator function comes from independent outputs of the limit state function [111],

$$\mathrm{Var}\,\hat{p}_F = \frac{1}{m^2} m \mathrm{Var}\left[I_G(x^{(i)})\right], \tag{4.58}$$

the variance is defined as [167]

$$\mathrm{Var}X = \mathrm{E}[X^2] - (\mathrm{E}X)^2. \tag{4.59}$$

Therefore Equation 4.58 is possible to rewrite as

$$\mathrm{Var}\,\hat{p}_F = \frac{1}{m}\left(\mathrm{E}[I_G^2(x^{(i)})] - (\mathrm{E}[I_G(x^{(i)})])^2\right). \tag{4.60}$$

Since the indicator function is a binary number, then $\mathrm{E}[I_G^2(x^{(i)})] = \mathrm{E}[I_G(x^{(i)})]$ and $\hat{p}_F = \mathrm{E}[I_G(\mathbf{x})]$,

$$\mathrm{Var}\,\hat{p}_F = \frac{1}{m}(\hat{p}_F - \hat{p}_F^2) = \frac{\hat{p}_F(1 - \hat{p}_F)}{m}. \tag{4.61}$$

For $\hat{p}_F$ close to zero, the variance of the failure probability estimator is simplified as

$$\mathrm{Var}\,\hat{p}_F \approx \frac{\hat{p}_F}{m}. \tag{4.62}$$

The standard deviation of the failure probability estimator is according to Equation 4.62 and Equation 4.14

$$\sigma_{\hat{p}_F} = \sqrt{\frac{\hat{p}_f}{m}}. \tag{4.63}$$

This indicator $\sigma_{\hat{p}_F}$ can be understood as a measure for a range in which the possible realizations of the probability failure estimators $\hat{p}_F$ may fall from different Monte Carlo simulations of the same problem using the identical parameter setting.

A coefficient of variation is another remarkable measure. It is equal to a standard deviation normalized by a mean value and therefore for a Monte Carlo simulation failure probability estimator evaluated as

$$\mathrm{CoV}\,\hat{p}_F = \sqrt{\frac{1 - \hat{p}_F}{m \cdot \hat{p}_f}} \approx \sqrt{\frac{1}{m \cdot p_F}}|_{p_F \to 0}. \tag{4.64}$$

Confidence interval $(1 - \alpha) \cdot 100\%$ can also be employed for presenting an accuracy of the Monte Carlo method. The value of the probability failure estimator lies in this confidence interval with a statistical guarantee of $(1-\alpha)$ confidence; the level α is in the range $[0, 1]$. The larger

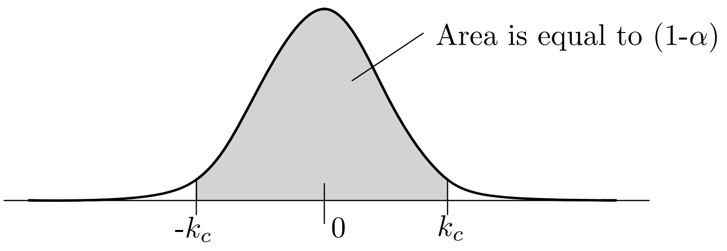


Figure 4.11: Confidence interval $(1-\alpha)\cdot 100\%$ representation together with parameter k_c.

confidence is required, the wider the confidence interval is. This confidence interval is based on the central limit theorem. The central limit theorem says that if we have a large number of independent and identically distributed random variables $X_1, \ldots, X_n$ with common mean μ and common variance σ^2, then their summation $X_1 + \cdots + X_n$ is approximately a normally distributed variable with mean $n\mu$ and variance $n\sigma^2$ [5]. Since the failure probability estimator of the Monte Carlo simulation is a summation of the indicator function over the failure samples as written in Equation 4.55, the estimator is approximately a normally distributed random variable. Each variable from a normal distribution can be transformed into the standard normal distribution. Therefore, for level α, it holds that

$$1-\alpha = \text{Prob}[-k_c \le Z \le k_c], \quad Z = \frac{p_F - \hat{p}_F}{\sigma_{\hat{p}_F}} \tag{4.65}$$

$$1-\alpha = \text{Prob}[\hat{p}_F - k_c\sigma_{\hat{p}_F} \le p_F \le \hat{p}_F + k_c\sigma_{\hat{p}_F}] \tag{4.66}$$

where Z belongs to the standard normal distribution which is substituted by a scaled probability of failure p_F into the standard normal distribution with mean $\hat{p}_F$ and standard deviation $\sigma_{\hat{p}_F}$; these estimates are known from sampling, therefore symbol $\hat{\cdot}$ is used. The parameter k_c is derived from the cumulative distribution function of a standard normal distribution; the probability density function of the normal distribution is symmetrical. Therefore the k_c value corresponds to

$$k_c = \Psi^{-1}\left(1 - \frac{\alpha}{2}\right). \tag{4.67}$$

Figure 4.11 depicts the graphical representation of the parameter k_c on the probability density function of the standard normal distribution. The area below the whole curve is equal to one since the maximum possible probability is equal to 1. Table 4.2 shows some typical confidence limit values $(1-\alpha)\cdot 100\%$ together with the parameter k_c. Thus, for example, 95% confidence interval is $(\hat{p}_F - 1.96\sigma_{\hat{p}_F}, \hat{p}_F + 1.96\sigma_{\hat{p}_F})$.

A crude Monte Carlo method utilizes a common *pseudorandom number generator* (PRNG). There exist plenty of generators, for instance, a very simple *linear congruential generator* [110] or the widely used *Mersenne twister* [130], which is currently used in MATLAB, Excel, Maple, Python, R, C++11, and in many other programs as a default pseudorandom number generator.

$(1-\alpha)\cdot 100\%$	k_c
90%	1.64
95%	1.96
99%	2.58

Table 4.2: Several k_c values for $(1-\alpha)\cdot 100\%$ confidence intervals.

The true randomness is usually not used in the computer programs, but if necessary in specific problems (e.g. cryptography or lotteries), it can come from, e.g. an atmospheric noise [158]. The difference between the stream of pseudorandom numbers and the stream of real random numbers is that PRNG generates a series of numbers that are deterministic and periodic. The period should be as large as it should not be noticed. For example, the Mersenne twister has the period $2^{19937} - 1$ [6]. A good PRNG is fast, and the stream of random numbers looks as similar to a truly random sequence of numbers in many statistical tests.

In the structural reliability, a Monte Carlo method is the most robust and in most cases, the most computationally demanding method for estimating the probability of failure. The number of samples needed in a crude Monte Carlo simulation is very high to obtain a precise approximation of p_F. A common rule of thumb recommends a number of samples from $10/p_F$ [24] over $100/p_F$ [186] up to $500/p_F$ samples. The nominator in the previous formula has a meaning of failed samples; the coefficient of variation of the Monte Carlo failure probability estimator is shown in Table 4.3 for several numbers of failed samples. Table 4.4 shows how many samples are needed for 10% coefficient of variation of the Monte Carlo failure probability estimator $\hat{p}_F$. In case that a number of samples in a crude Monte Carlo simulation is not sufficient, several techniques are available for the improvement of the probability of failure estimate, e.g.see [195, 197].

number of failed samples	1	10	100	500
C.o.V. of MC prediction	100%	31.62%	10.0%	4.47%

Table 4.3: Number of failed samples in Monte Carlo simulation for a needed coefficient of variation of Monte Carlo failure probability estimator $\hat{p}_F$

The *Latin Hypercube Sampling* exploits the division of the investigated space to N strata; thus, it covers the space more uniformly. While considering a standard uniform space, the interval $[0, 1]$ is therefore segmented into

$$[0, \frac{1}{N}), [\frac{1}{N}, \frac{2}{N}), \ldots, [\frac{N-1}{N}, 1], \tag{4.68}$$

where one value is chosen in each interval. Concerning this sample selection, several variants of LHS exist. Namely, median LHS, in which a sample is represented by a median from each interval and random LHS, in which a sample is generated randomly [193]. Subsequently, these samples are shuffled by generating several random permutations of N numbers from a series $1, 2, \ldots, N$ and by assigning samples to elements of the permutation. The best design is then

β	0	1	2	3	4
p_f	0.5	0.159	$2.28 \cdot 10^{-2}$	$1.35 \cdot 10^{-3}$	$3.17 \cdot 10^{-5}$
MC samples	200	629	$4.39 \cdot 10^{3}$	$7.41 \cdot 10^{4}$	$3.15 \cdot 10^{6}$
β	4.5	5	5.5	6	
p_f	$3.40 \cdot 10^{-6}$	$2.87 \cdot 10^{-7}$	$1.90 \cdot 10^{-8}$	$9.87 \cdot 10^{-10}$	
MC samples	$2.94 \cdot 10^{7}$	$3.49 \cdot 10^{8}$	$5.27 \cdot 10^{9}$	$1.01 \cdot 10^{11}$	

Table 4.4: Number of samples needed for a 10% coefficient of variation of Monte Carlo failure probability estimator p_F

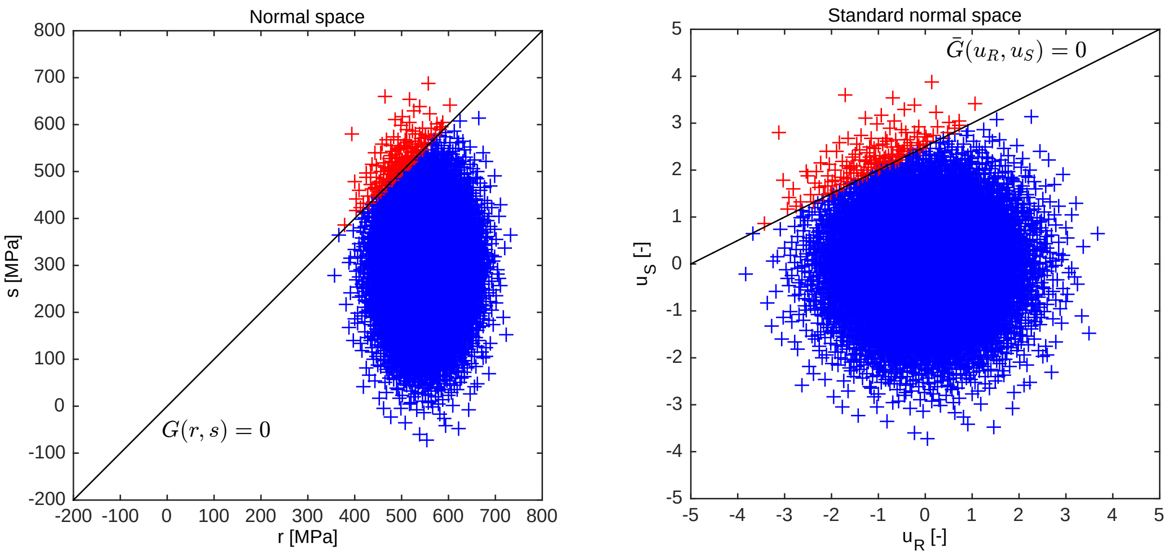


Figure 4.12: A rod under tension: 10^4 crude Monte Carlo samples of the capacity $R \sim N(550, 50)$ and the demand $S \sim N(300, 100)$. 126 samples (red crosses) are in the failure domain which means that p_F is equal to 0.0126. The corresponding reliability index β_{MC} is equal to 2.238.

chosen as the final Design of Experiments. The advantage of the LHS is that the Design of Experiments is more uniformly distributed than a pseudorandom sequence. Therefore samples are taken from the whole range of the distribution even with a relatively small sample size [193]. The disadvantage is that the sample size is not possible to determine as mentioned for a crude Monte Carlo simulation. However, the number of samples is typically less than the number of crude Monte Carlo samples. Another problem is that it is problematic to add new samples into the LHS Design of Experiments to maintain the uniformity of the design.

Another improving alternative to a crude Monte Carlo method is a *quasi-Monte Carlo method*. The quasi-MC uses *low-discrepancy sequences*, e.g. the one-dimensional van der Corput sequence [53], the Halton sequence [84] (i.e. an extension of the van der Corput sequence to higher dimensions), the Sobol sequence [177], the Faure sequence [69], or the Niederreiter sequence [138]. The discrepancy of a sequence is a measure of its uniformity. These sequences are n-tuples of samples that fill the n-dimensional space, they are a deterministic stream of numbers, but since they have a low discrepancy, they appear to be random. The convergence of a crude Monte Carlo is $\mathcal{O}(1/\sqrt{n})$ for n number of samples, while the convergence of quasi-Monte Carlo method is potentially close to $\mathcal{O}(1/n)$ [6]. These low-discrepancy sequences can fail in some statistical tests, and their projection in higher dimensions can have troubles with repetitive patterns [6]. Nevertheless, their generation is fast, and their properties are sufficient in many cases.

Figure 4.12 shows a crude Monte Carlo sampling for a rod under tension depicted in Figure 4.6. The left figure illustrates samples from the original distribution (the normal space) and the original limit state. The variables represented by samples are transformed into non-Gaussian components; first, the standard uniform space is sampled, and subsequently, samples are transformed into the normal space. The limit state function remains without any transformation. This procedure is useful for the black-box solvers, e.g. a finite element method, in which the transformation of the performance function would be complicated. The right figure shows samples in the standard normal space together with the transformed limit state $\bar{G}(u_R, u_S) = 0$. In this case, the limit state remains without any nonlinear transformation, and it is only shifted since there is only a linear relation between a normal distribution and a standard normal distribution. This procedure is preferred for analytical functions that are easily transformed from the original space to the standard normal space since the limit state function is transformed prior to the sampling. The samples are subsequently performed only in the standard normal space without any transformation to the original space.

4.4.2 Importance sampling (IS)

Importance sampling (IS) is a variance reduction method in a reliability assessment field. Probably the very first formulation of the idea behind the Importance sampling is in the article [98] of KAHN and HARRIS. HAMMERSLEY and HANDSCOMB described details of the technique in their monograph on Monte Carlo methods [85] in 1964. This method is beneficial for tasks with small failure probabilities, in which a crude Monte Carlo method would need an enormous number of samples to hit a failure region at least by one sample. An Importance sampling samples from a different distribution called an Importance sampling distribution (ISD) or a biasing density [181]. The goal is to sample in the important region to get more knowledge about interesting areas. In the best case of ISD choice, samples are located close to the failure region. For obtaining the approximated probability of failure, the weighting factor, which is equal to the ratio of the original PDF $f_{\mathbf{X}}(\mathbf{x})$ and the Importance sampling PDF $h_{\mathbf{Y}}(\mathbf{x})$, is used to scale the indicator function $I_G(\mathbf{x})$

$$p_F = \int\limits_{-\infty}^{\infty} \cdots \int\limits_{-\infty}^{\infty} I_G(\mathbf{x}) \frac{f_{\mathbf{X}}(\mathbf{x})}{h_{\mathbf{Y}}(\mathbf{x})} f_{\mathbf{Y}}(\mathbf{x}) d\mathbf{x}. \tag{4.69}$$

The above equation is possible to sample via a Monte Carlo simulation from the density $h_{\mathbf{Y}}(\mathbf{x})$ and then weight the indicator function by the evaluated weighting ratio

$$\hat{p}_F \approx \frac{1}{m} \sum_{i=1}^{m} \frac{f_{\mathbf{X}}(\mathbf{x})}{h_{\mathbf{Y}}(\mathbf{x})} I_G(\mathbf{x}), \tag{4.70}$$

where m is the total number of samples.

The Importance sampling probability density function $h_{\mathbf{Y}}(\mathbf{x})$ has to be picked very carefully. With a very poor choice of $h_{\mathbf{Y}}(\mathbf{x})$, the Importance sampling can be more computationally demanding than a classical Monte Carlo. The variance of the probability failure estimator of an Importance sampling method $\sigma^2_{\hat{p}_F}$ according to [65] is

$$\sigma^2_{\hat{p}_F} = \frac{1}{m-1} \left[\frac{1}{m} \sum_{i=1}^{m} \frac{f_{\mathbf{X}}(\mathbf{x})^2}{h_{\mathbf{Y}}(\mathbf{x})^2} I_G(\mathbf{x}) - p_F^2 \right]. \tag{4.71}$$

This formula gives the information about the optimal sampling density since it would be ideal if the variance of the probability failure estimator were minimal. Therefore, the optimal sampling density is [172]

$$h_{OPT}(\mathbf{x}) = \frac{f_{\mathbf{X}}(\mathbf{x})}{p_F} I_G(\mathbf{x}). \tag{4.72}$$

Unfortunately, this distribution requires the knowledge of the probability of failure, which is not known in advance. Several strategies to obtain the Importance sampling density are presented in the literature, namely scaling, translation, and exponential twisting of the original probability

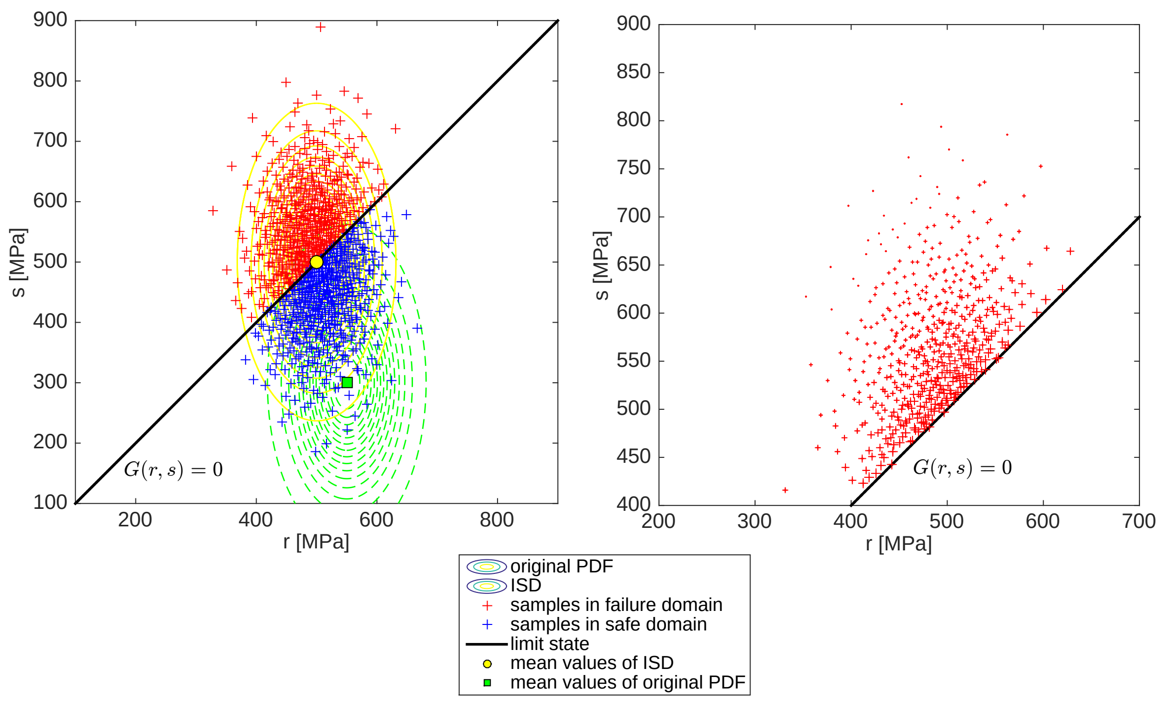


Figure 4.13: A rod under tension: the left figure shows 10^3 Importance sampling samples of the capacity $R \sim N(550,50)$ and the demand $S \sim N(300,100)$. Four hundred ninety-eight samples (red crosses) are located in the failure domain. The probability of failure is equal to 0.0125, and the corresponding reliability index β_{MC} is equal to 2.241. The coefficient of variation of the failure probability estimator is approximately 5.1 %. The original joint probability density function of the problem (green dashed contours) was translated into the design point $[DP_R, DP_S] = [500,500]$ MPa (yellow circle); the yellow solid contours depict it. The right figure shows the impact of the sample into the failure probability estimator from the different simulation.

density function [181], approaches based on kernel density estimators [2], on design points found by e.g. by optimization method [21] or adaptive sampling [27].

In this thesis, we use marginal ISDs having the same family of distribution as the original distribution. The standard deviations of the variables remain the same. The n-dimensional copula is translated into the design point $\mathbf{u}^*$, which is known by sequential quadratic programming in advance. The optimization task is carried out in standard normal space (SNS) as the minimization of the distance from SNS origin to the limit state $G(\cdot)=0$, which is the (n-1)-hyper-surface dividing the space into the failure region and the safe area

$$\mathbf{u}^* = \min\left(\sqrt{\mathbf{u}^T\mathbf{u}}\right) \tag{4.73}$$

$$\text{s.t.} \quad G(T^{-1}(\mathbf{u})) = 0. \tag{4.74}$$

Since the limit state function $G(\cdot)$ can be an implicit function and its transformation from the original space (OS) to the SNS is not straightforward, the transformation of each evaluated sample is carried out outside by isoprobabilistic transformation $T^{-1}(\cdot)$.

Figure 4.13 on the left shows an Importance sampling for a rod under tension depicted in Figure 4.6. The red crosses depict samples that belong to the failure domain, whereas the blue crosses are samples sampled in the safe domain. The original distribution function (the green dashed contours represents joint PDF) is shifted from the original means $[\mu_R, \mu_S] = [550,300]$ MPa (green square) into the design point $[DP_R, DP_S] = [500,500]$ MPa (yellow circle) to get more samples in the failure domain. The yellow solid contours represent joint Importance sampling density. The classical Monte Carlo simulation, that sample from the original distribution function, is depicted in Figure 4.12 in the previous section. The right figure shows only the samples from the failure domain (from a different simulation); the size of the crosses represent their relevance contribution into the failure probability estimator according to the weighting ratio. The indicator function $I_G(\mathbf{x})$ is equal to one in the failure domain and equal to zero in the safe domain (red crosses in Figure 4.13). The second case, therefore, does not contribute to the failure probability estimator in Equation 4.70 since this component of the summation is null.

The system reliability is assessed similarly as in the First-Order reliability method with the knowledge of the more precise reliability indices β_j from the Importance sampling. The N number of design points u_j^* is found for each limit state function $G_j, j = 1, 2, \ldots, N$ by

$$\mathbf{u_j^*} = \min\,(\sqrt{\mathbf{u}^T\mathbf{u}}) \tag{4.75}$$

$$\text{s.t.} \quad G_j(T^{-1}(\mathbf{u})) = 0. \tag{4.76}$$

The original PDF $f_{\mathbf{X}}(\mathbf{x})$ is then N-times translated into u_j^* design points to obtain N number of Importance sampling PDFs $h_{\mathbf{Y},j}(\mathbf{x})$. The N number of failure probabilities is then obtained via

$$\hat{p}_{F,j} \approx \frac{1}{m}\sum_{i=1}^{m}\frac{f_{\mathbf{X}}(\mathbf{x})}{h_{\mathbf{Y},j}(\mathbf{x})} I_{G_j}(\mathbf{x}), \tag{4.77}$$

together with corresponding reliability indices $\beta_j = \phi^{-1}(1 - \hat{p}_{F,j})$. The utilization of the maximum or the minimum β_j index for the series or the parallel system would be too optimistic because of the possible dependency of the limit state functions. We, therefore, use the same methodology as was mentioned in FORM. The necessary unit normal to the hyperplane α_j is evaluated via

$$\alpha_j = -\frac{\nabla G_j(\mathbf{u}_j^*)}{||\nabla G_j(\mathbf{u}_j^*)||}. \tag{4.78}$$

If we suppose that $v_j = \alpha_j \mathbf{u}, j = 1, 2, \ldots, N$ are normal random variables with zero means, unit variances, and correlation coefficients $\rho_{kl} = \alpha_k \alpha_l^T, k, l = 1, 2, \ldots, N$, it is valid that [139]

$$p_{F,series} = 1 - \Phi_N(\mathbf{B}, \mathbf{R}), \tag{4.79}$$

$$p_{F,parallel} = \Phi_N(-\mathbf{B}, \mathbf{R}), \tag{4.80}$$

where Φ_N is N-variate standard normal cumulative distribution function with argument $\mathbf{B} = [\beta_1, \beta_2, \ldots, \beta_N]^T$ and a correlation matrix $\mathbf{R} = [\rho_{kl}]$; a correlation coefficient is evaluated as $\rho_{kl} = \alpha_k^T \alpha_l$.

4.4.3 Asymptotic sampling (AS)

An Asymptotic sampling is a methodology that predicts a reliability index from an asymptotic behaviour of the probability of failure in an n-dimensional independent and identically distributed normal space [24, 175]. A principal idea is to sequentially scale random variables over the standard deviation σ to get more samples from a failure domain. An approximation of the asymptotic behaviour with original distributions gives the reliability index of the problem

$$\beta_{AS} = A\varphi + \frac{B}{\varphi}, \tag{4.81}$$

where φ denotes a scale factor that is expressed as $\varphi = \frac{1}{\sigma}$. Equation (4.81) can be written in terms of a scaled reliability index for better fitting purposes as

$$\frac{\beta_{AS}}{\varphi} = A + \frac{B}{\varphi^2}. \tag{4.82}$$

Coefficients A and B are obtained by a linear regression analysis through several so-called *support points*, see Figure 4.14.

Each support point represents an MC estimate of the reliability index for a specific value of the scale factor φ. The user determines the number of samples for one MC simulation m with the same σ's as well as the number of necessary samples belonging to the failure domain N_0 and the decreasing coefficient φ_d for the factor φ. If the number of failures N_f is higher than N_0, the reliability index $\beta_{AS,i}$ and the corresponding factor φ_i are stored as one support point. In another case, the factor φ_d decreases the factor φ without any support point storage. After gathering a sufficient number of support points K, the procedure stops, and coefficients A and B are obtained through linear regression of the model in Equation 4.82. If coefficients A and B are arranged into the vector $\mathbf{\Theta}$ as $[A, B]$, the vector $\mathbf{\Theta}$ can be calculated as

$$\mathbf{\Theta} = (\mathcal{A}^T\mathcal{A})^{-1}\mathcal{A}^T b, \tag{4.83}$$

where

$$\mathcal{A} = \begin{pmatrix} 1 & \varphi_1^{-2} \\ 1 & \varphi_2^{-2} \\ \vdots & \vdots \\ 1 & \varphi_K^{-2} \end{pmatrix}, \tag{4.84}$$

$$b = \begin{pmatrix} \beta_{AS,1}/\varphi_1 \\ \beta_{AS,2}/\varphi_2 \\ \vdots \\ \beta_{AS,K}/\varphi_K \end{pmatrix}. \tag{4.85}$$

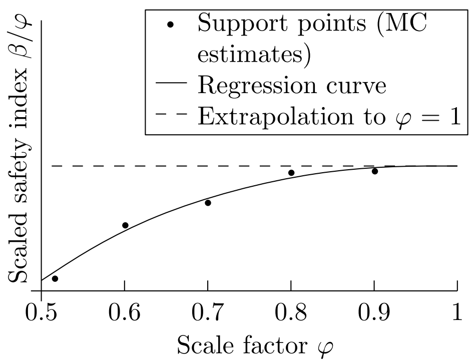
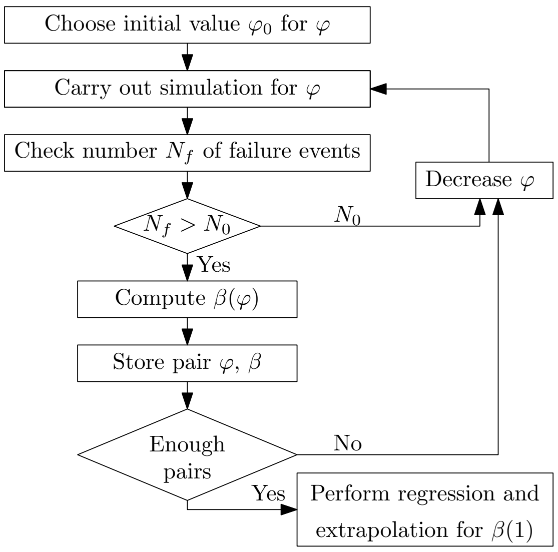


Figure 4.14: A basic concept of the Asymptotic sampling.



Choose initial value φ_0 for φ
Carry out simulation for φ
Check number N_f of failure events
Decrease φ
$N_f > N_0$
N_0
Yes
Compute $\beta(\varphi)$
Store pair φ, β
Enough pairs
No
Yes
Perform regression and extrapolation for $\beta(1)$

Figure 4.15: A flow chart of the Asymptotic sampling algorithm.

Summation of A and B then represents an estimated reliability index β_{AS} for unscaled random variables as an extrapolation of φ equal to 1. The whole routine is depicted in the flow chart in Figure 4.15.

Figure 4.16 shows sequential Asymptotic sampling populations for a rod under tension depicted in Figure 4.6. The setting is as follows: there have to be at least 10 samples in the failure domain (N_0); every Monte Carlo simulation has 1024 samples (m); the initial coefficient φ is equal to 0.9; the decreasing factor φ_d is equal to 0.8 due to graphical purposes; the number of sample points K (combinations of saved β_i and φ_i) is equal to 5; and Halton sequences are used to generate Monte Carlo population.

The next iteration is wider than previous ones. The difference in the Monte Carlo result and the Asymptotic sampling result is due to the unsuitable coefficient setting that was performed for the sake of graphical output. Increasing m to 2048 and φ_d to 0.9, the reliability index β_{AS} is equal to 2.11 and the probability of failure p_F equals to 0.0174. It can be seen that the disadvantage of this method is in the sensitivity to coefficients. The analysis of Asymptotic sampling parameters for the truss structure bridge, which is used later in this thesis statement is presented in [148].

4.4.4 Subset simulation (SS)

A Subset simulation [8] is based on a formulation of the failure event F as an intersection of M nested intermediate events $F_M \subset F_{M-1} \subset \cdots \subset F_2 \subset F_1$, in which $F_M = F$ is the failure

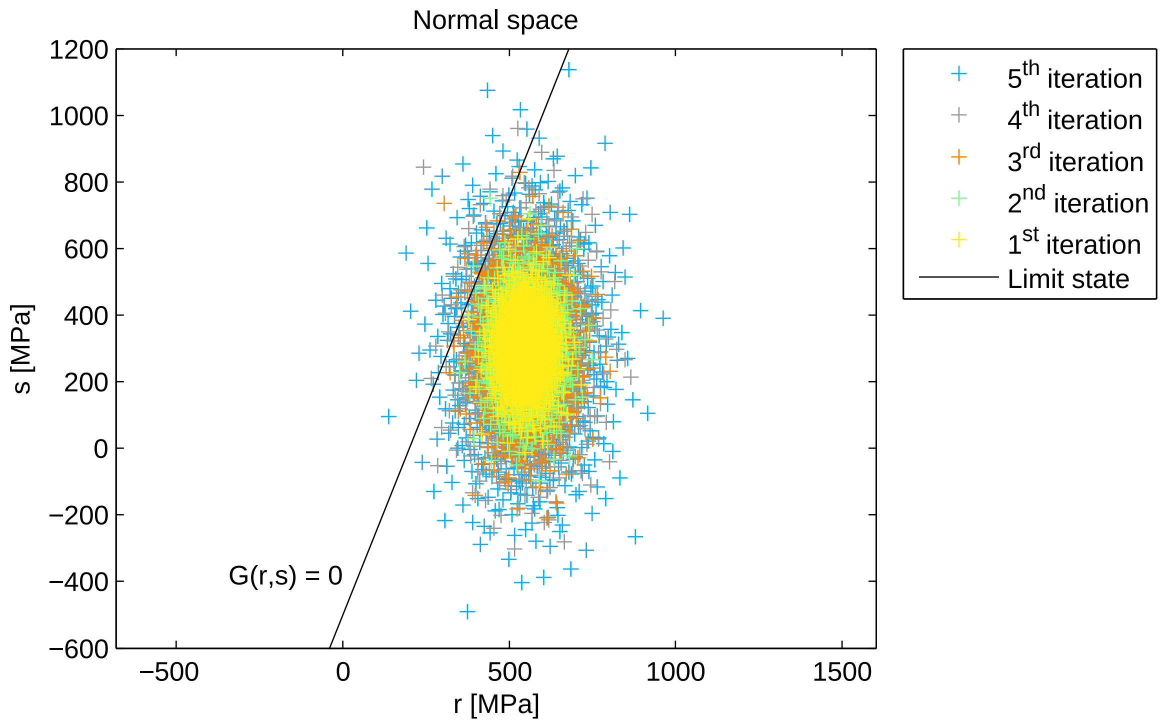


Figure 4.16: A rod under tension: Asymptotic sampling samples of the capacity $R \sim N(550, 50)$ and the demand $S \sim N(300, 100)$. The limit state $G(r, s) = 0$ divides the space to the failure region (on the left from the limit state) and the safe domain (on the right from the limit state). β_{AS} is equal to 1.94 and the probability of failure p_F is approximately equal to 0.0261.

event

$$F = \cap_{i=1}^{M} F_i. \tag{4.86}$$

Therefore, the rare event problem is reformulated into a series of more frequent events that are easier to solve. An estimation of the probability of failure is evaluated as a product of conditional probabilities

$$\begin{aligned} p_F &= \text{Prob}[F] = \text{Prob}[\cap_{i=1}^{M} F_i] = \text{Prob}[F_M|F_{M-1}] \cdot \text{Prob}[F_{M-1}] = \\ &= \text{Prob}[F_1] \cdot \prod_{i=2}^{M} \text{Prob}[F_i|F_{i-1}] \end{aligned} \tag{4.87}$$

Each intermediate event is defined as

$$F_i = \mathbf{x} : G(\mathbf{x}) \leq y_i^*, \quad i = 1, \dots, M, \tag{4.88}$$

where y_i^* is a series of threshold values of the structural response. Unfortunately, this series cannot be determined in advance but has to be found adaptively during the sequential evaluation of intermediate failure probabilities.

The failure probability of conditional level 0 (i.e. unconditional) Prob$[F_1]$ is evaluated by a classical Monte Carlo method with hundreds or a few thousands of samples N. The user sets the parameter N, and it remains the same during the whole Subset simulation. Samples are generated in the standard normal space $\mathbf{U}$ described by a joint probability density function (PDF) $\varphi_d(\mathbf{u})$ obtained as $\prod_{d=1}^{D} \varphi(u_d)$, where $\varphi(\cdot)$ is a standard normal PDF and d is a coordinate out of the total number of coordinates D. Samples in $\mathbf{U}$-space are transformed afterwards, e.g. by the Rosenblatt transformation [166] to the target $\mathbf{X}$-space, where the family and statistical moments are known, and therefore a value of the limit state function can be evaluated as $y = G(T^{-1}(\mathbf{u})) = G(\mathbf{x})$. The $\mathbf{X}$-space is used only for limit state function evaluation; the Subset simulation is carried out in the $\mathbf{U}$-space. The intermediate failure event F_1 is defined as $F_1 = \{\mathbf{u} : G(T^{-1}(\mathbf{u})) < y_1^*\}$, where the level probability $p_{F,1} =$ Prob$[F_1]$ is chosen in the interval $p_{F,i} \in (0,1)$, and it remains the same for all levels. If already simulated samples are sorted in ascending order according to their value of the limit state function, a threshold value y_1^* is $p_{F,i}$-quantile of the N sampled system response. The threshold value y_1^* can be estimated as $y_0^{(N_c)}$ [7] or as in reference [214]

$$y_1^* = \frac{y_0^{(N_c)} + y_0^{(N_c+1)}}{2}, \tag{4.89}$$

where the subscript 0 signifies that samples y correspond to the 0-level, superscript denotes the number of a sample in the sorted set of samples and N_c is equal to the level probability $p_{F,1}$ multiplied by the number of samples N.

The failure probability in the first intermediate domain is estimated via sampling from a conditional distribution $\varphi_d(\mathbf{u}|F_1)$, e.g. by Markov chain Monte Carlo (MCMC) with modified Metropolis algorithm. The domain F_1 is characterized as $\{\mathbf{u} : G(T^{-1}(\mathbf{u})) \leq y_1^*\}$ and the threshold value y_1^* is known from the previous level. All samples $y_0^{(1)}$ to $y_0^{(N_c)}$ belong to F_1 domain and they are used as seeds for the Markov chains. Thus, the remainder of the samples at the first level is generated. Seeds can be used or discarded; both approaches are common in literature. However, AU and WANG [9] claim that seeds should be discarded after their usage since their rejection reduces the correlation between samples at different simulation levels. Each chain has N_s samples equal to $\frac{1}{p_{F,i}}$ or $(\frac{1}{p_{F,i}} - 1)$ if seeds are discarded or kept, respectively. Note, that N_s times N_c amounts to the number of samples in one level N.

Markov chain Monte Carlo samples can be generated with various algorithms; a modified Metropolis algorithm is a suitable sampling approach specially designed for sampling from conditional distributions such as $\varphi_d(\mathbf{u}|F_i)$. The seed $\mathbf{u}_0^{(1)}$ belongs to a conditional distribution $\varphi_d(\mathbf{u}|F_1)$, a modified Metropolis algorithm generates the next sample $\bar{u}$ in the following manner. A proposal sample η is generated from a chosen proposal distribution with the symmetric property, that has a mean value equal to $\mathbf{u}_0^{(1)}$ and standard deviation to σ. This distribution can be uniform, normal, triangular, or other symmetrical. The standard deviation can be estimated by expert opinion or by evaluation of standard deviations of all seeds in level $(i-1)$ in all dimensions d. An acceptance ratio r_d is evaluated for each coordinate d as

$$r_d = \frac{\varphi(\eta_d)}{\varphi(u_d)}, \tag{4.90}$$

where $\varphi(\cdot)$ is a PDF of the standard normal distribution since whole Subset simulation is executed in a standard normal space, the numerator is a PDF value of the standard normal distribution of the d^{th}-dimension of proposal sample, and the denominator is a PDF value of the standard normal distribution of the previous Markov chain step. If a randomly generated number from the standard uniform distribution $rand()$ is lesser than $\min(1, r_d)$, then η_d is accepted as a candidate solution ξ_d, otherwise ξ_d is equal to the previous value u_d. The candidate solution ξ has to be located in F_1 for the first level, or F_i for the i^{th} level in general, i.e. $F_1 = \{\xi : G(\xi) \leq y_1^*\}$ for the first level. If a case that ξ is located outside F_1, the next state of Markov chain Monte Carlo is identical with the previous state, i.e. $\bar{u} = u$. State $\bar{u}$ is then a next state of the Markov chain, and it is the next mean value of the proposal distribution. Since the seed of the Markov chain is incontrovertibly located in the desired region where we want to sample, there is no burn-in phase, and the Markov chain is stationary [143].

The next intermediate levels of i are simulated in the same manner. Seeds for Markov chains are taken from the previous level as N_c sorted samples according to their values of the limit state function. The rest of the samples is generated from a conditional distribution $\varphi_d(\mathbf{u}|F_i)$. Samples are again sorted; the threshold value y_{i+1}^* is detected such that the level probability $p_{F,i}$ is equal to the prescribed value. The last level M is reached if the threshold value y_{i+1}^* change its sign compared to previous levels. The estimation of the failure probability is then

$$p_F = p_{F,1} \cdot p_{F,i}^{(M-1)} \cdot \frac{1}{N} \sum_{j=1}^{N} I_{F_i}(\mathbf{u}_j), \tag{4.91}$$

where $p_{F,1}$ stands for the intermediate failure probability of conditional level 0 carried out by a Monte Carlo, $p_{F,i}$ is for the intermediate failure probability of conditional levels 1 to $(M-1)$ executed by a Markov chain Monte Carlo; note that usually $p_{F,1}$ and $p_{F,i}$ are set to the same number. The last factor in Equation 4.91 evaluates the intermediate failure probability of the last level M, where $I_{F_i}(\cdot)$ is an indicator function equal to 1 in case of failure and 0 otherwise. The whole algorithm described above is depicted in Figure 4.17.

Setting the level probability $p_{F,i}$ close to zero provides less number of intermediate levels, but the number of samples N would have to be enormous to get the required precision and vice versa, the level probability $p_{F,i}$ close to one follows in a large number of intermediate levels. The level probability and the number of samples have to be therefore set carefully such that the number of chains $N_c = p_{F,i} \cdot N$ and the number of samples in one chain $N_s = \frac{1}{p_{F,i}}$ are positive whole numbers. The value $p_{F,i}$ equal to 0.1 is a relatively common recommendation in literature, however wider setting is possible as in AU et al. in [7], who recommend to set $p_{F,i}$ in interval $0.1 \sim 0.2$ or even to a higher number in ZUEV et al. in [215], who suggests

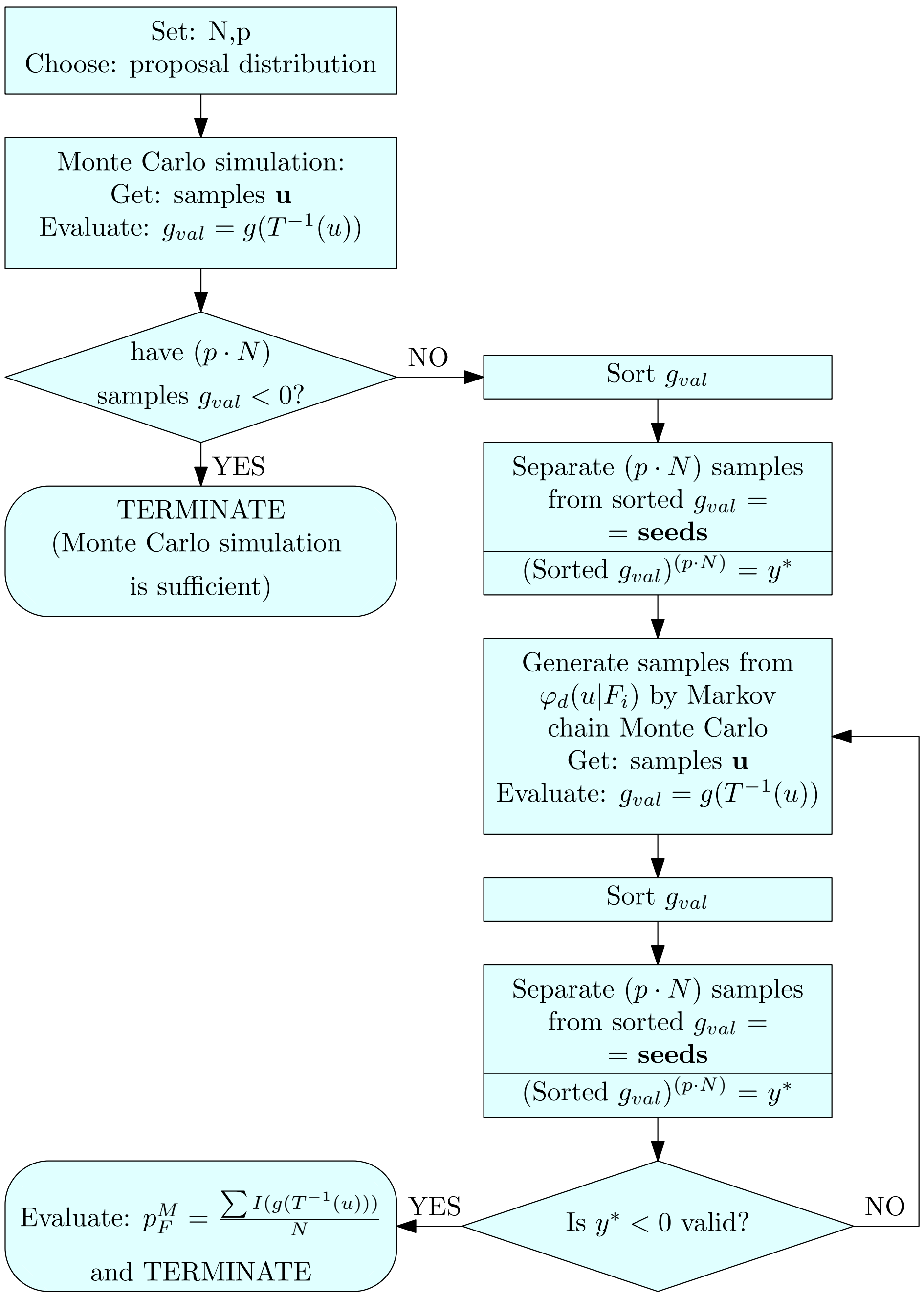

Set: N,p
Choose: proposal distribution
Monte Carlo simulation:
Get: samples $\mathbf{u}$
Evaluate: $g_{val} = g(T^{-1}(u))$
have $(p \cdot N)$ samples $g_{val} < 0$?
NO
YES
TERMINATE (Monte Carlo simulation is sufficient)
Sort g_{val}
Separate $(p \cdot N)$ samples from sorted g_{val} = = **seeds**
(Sorted $g_{val})^{(p \cdot N)} = y^*$
Generate samples from $\varphi_d(u|F_i)$ by Markov chain Monte Carlo
Get: samples $\mathbf{u}$
Evaluate: $g_{val} = g(T^{-1}(u))$
Sort g_{val}
Separate $(p \cdot N)$ samples from sorted g_{val} = = **seeds**
(Sorted $g_{val})^{(p \cdot N)} = y^*$
Is $y^* < 0$ valid?
YES
NO
Evaluate: $p_F^M = \frac{\sum I(g(T^{-1}(u)))}{N}$ and TERMINATE

Figure 4.17: A flow chart of the Subset Simulation algorithm

setting $p_{F,i}$ in an interval of $0.1 \sim 0.3$. The analysis of Subset simulation parameters for series system of mathematical functions and the truss structure bridge, which is used later in this thesis statement, is presented in [90].

Figure 4.18 shows a Subset simulation for a rod under tension depicted in Figure 4.6; the capacity is $R \sim N(550, 50)$ and the demand is $S \sim N(300, 100)$, the limit state function is equal to subtraction the demand and the capacity. In this example, we set the number of samples in one step N equal to 5000 and the level probability $p_{F,i}$ equal to 0.2. The proposal distribution in a modified Metropolis algorithm is chosen as a normal distribution. The standard deviation σ of the proposal distribution is evaluated as standard deviations of all seeds in level $(i-1)$ in all

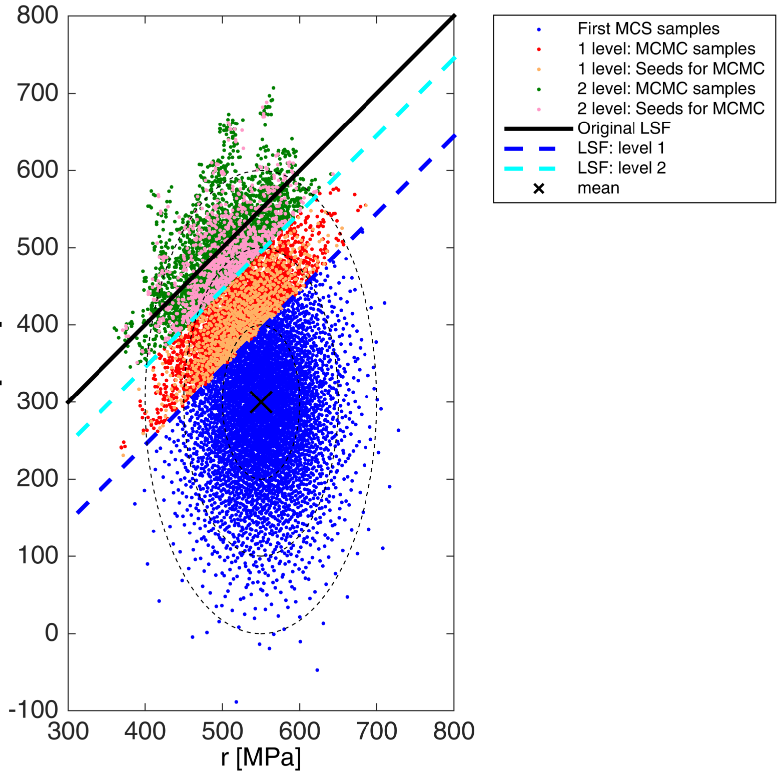


Figure 4.18: A rod under tension: Subset simulation samples of the capacity $R \sim N(550, 50)$ and the demand $S \sim N(300, 100)$. The limit state $G(r, s) = 0$ divides the space to the failure region and the safe domain. β_{AS} is equal to 1.94 and the probability of failure p_F is approximately equal to 0.0261.

dimensions d. A classical Monte Carlo simulation generates the level 0, which is depicted by blue dots. The blue dashed line represents the shifted limit state function for the first level. The seeds (orange dots) are separated from samples in level 0, and Markov chains are subsequently generated (red dots). The next level is evaluated. The seeds (pink dots) are separated anew from the samples in the previous set, and Markov chains (green dots) are generated afterwards anew as well. The cyan dashed line represents the shifted limit state function for the second level. In the next hypothetical step, the shifted limit state function would appear on the other size than the limit state functions for the level 1 and 2 in comparison with the original limit state function (solid black line), which means that the algorithm terminates. The black cross represents the mean of the original joint distribution function, which is depicted by dashed black contours. The failure probability estimator is equal to 0.0137, and subsequent reliability index is equal to 2.206.

The previous example, the stress-strength model, is linear. We tested our implementation on the nonlinear models as well. We found the main problem in the instability of the selection of the subsequent random walks directions. A Subset simulation method proceeds in the steepest descent direction. A fixed portion of samples is taken from samples generated in the previous step and sorted in the ascending order serves for initial seeds in a random walk. Since a pseudo-random generator sampler does not always hit all possible directions to all most probable failure points, some crucial regions can be omitted. Figure 4.19 illustrates this behaviour and shows two independent runs of a Subset simulation with the same setting.

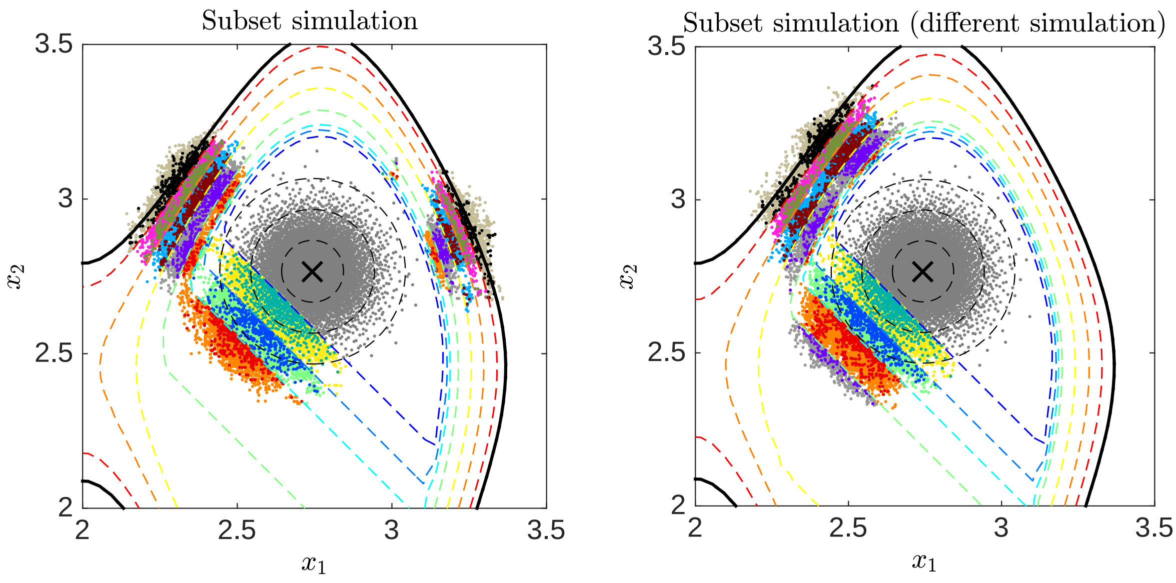


Figure 4.19: A Subset simulation shows instability in the selection of sampling directions. Colours of samples serve for differentiation of levels as well as with their seeds. Coloured dashed contours distinguish different $y^\star_m$ threshold levels. Each Subset simulation assessment comprised of eight levels. Both simulations have the same mean values of variables, $\mu_{X1} = 2.74$ and $\mu_{X2} = 2.71$ (the black cross). The bold solid line is the limit state, black dashed circles mark the mean values plus one, two, and three standard deviations, respectively.

4.4.5 Enhanced Monte Carlo simulation (eMC)

The Enhanced Monte Carlo simulation is based on the idea of sequentially shifting the limit state function to obtain more responses in the failure domain. This novelty idea was brought by ARVID NAESS et al. in [136]. The original problem is defined as the evaluation of the failure probability using

$$p_F = \text{Prob}[G(\mathbf{X}) \leq 0], \tag{4.92}$$

where $G(\cdot)$ is the limit state function and $G(\cdot) \leq 0$ denotes failure. The limit state function G (the original paper uses a symbol M for the limit state function) is extended into the parameterized class of limit state functions as

$$G(\mathbf{X}, \lambda) = G(\mathbf{X}) - (1 - \lambda)\text{E}[G(\mathbf{X})], \tag{4.93}$$

where λ is a scaling factor, and $\text{E}[\cdot]$ is an expectation operator. The scaling factor λ is from an interval $[0, 1]$. The limit state function remains unchanged for λ equal to one, and this case is intended to extrapolate since it corresponds to Equation 4.92. With the decreasing value of the scaling factor λ, the failure region is larger. The expectation operator $\text{E}[\cdot]$ is generally unknown a priori. If it cannot be calculated analytically, it is estimated by Monte Carlo simulation as the mean value of limit state function responses.

The original probability of failure in Equation 4.92 is therefore reformulated into

$$p_F(\lambda) = \text{Prob}[G(\mathbf{X}, \lambda) \leq 0]. \tag{4.94}$$

Practically, only one Monte Carlo simulation is performed, and several scaling factors from the interval $[0, 1]$ are utilized for Equation 4.93 to obtain several probabilities of failure for different

λ values as

$$\hat{p}_F(\lambda) = \frac{N_f(\lambda)}{N}, \tag{4.95}$$

where $\hat{p}_F(\lambda)$ is an estimator of the scaled probability of failure corresponding with λ, $N_f(\lambda)$ is a number of samples in the failure domain for corresponding λ and N is the total number of samples in the Monte Carlo simulation. For extrapolating the probability of failure $\hat{p}_F(\lambda = 1)$, pairs $\hat{p}_F(\lambda)$ and λ are fitted by simplified regression function

$$\hat{p}_F(\lambda) \approx q \exp[-a(\lambda - b)^c], \tag{4.96}$$

where parameters q, a, b, c are obtained by Levenberg-Marquardt least-squares optimization method.

The coefficient of variation of the estimator in Equation 4.95 is

$$\text{CoV}[\hat{p}_F(\lambda)] = \sqrt{\frac{1 - \hat{p}_F(\lambda)}{\hat{p}_F(\lambda)N}}. \tag{4.97}$$

NAESS et al. in [136] recommend using the approximation of the 95% confidence interval for the value $\hat{p}_F(\lambda)$ as $CI_{0.95(\lambda)} = (C^-(\lambda), C^+(\lambda))$, where

$$C^{\pm}(\lambda) = \hat{p}_F(\lambda)(1 \pm 1.96 \cdot \text{CoV}[\hat{p}_F(\lambda)]). \tag{4.98}$$

The estimation of the parameters q, a, b, c is then carried out by minimizing the mean square error function with respect to all four parameters in the logarithmic formulation

$$F(q, a, b, c) = \sum_{j=1}^{M} w_j \left(\log \hat{p}_F(\lambda_j) - \log q + a(\lambda_j - b)^c\right)^2, \tag{4.99}$$

where λ_j contains a set of all valid scaling parameters with their total number of M together with corresponding estimators of the failure probability $\hat{p}_F(\lambda_j)$ and w_j are weight factors. NAESS et al. in [136] recommend using weights as

$$w_j = (\log C^+(\lambda) - C^-(\lambda))^{-2}. \tag{4.100}$$

The failure probability of interest is then obtained by setting λ to one

$$p_F(1) \approx \hat{q} \exp[-\hat{a}(1 - \hat{b})^{\hat{c}}] \tag{4.101}$$

for estimated parameters $\hat{q}$, $\hat{a}$, $\hat{b}$, and $\hat{c}$.

Even though according to authors, the Levenberg-Marquardt method generally works well, they recommend a simplification of Equation 4.99 by fixing parameters b and c and evaluating parameters a^* and q^*. The parameter a^* is obtained via

$$a^*(b, c) = -\frac{\sum_{j=1}^{M} w_j (x_j - \bar{x})(y_j - \bar{y})}{\sum_{j=1}^{M} w_j (x_j - \bar{x})^2}, \tag{4.102}$$

where $y_j = \log \hat{p}_F(\lambda_j)$, $x_j = (\lambda_j - b)^c$, $\bar{x} = \sum_{j=1}^{M} x_j / M$, and $\bar{y} = \sum_{j=1}^{M} y_j / M$. The logarithm of q^* is then evaluated by

$$\log q^*(b, c) = \bar{y} + a^*(b, c)\bar{x}. \tag{4.103}$$

Authors use the Levenberg-Marquardt method on the function $\tilde{F}(b, c) = F(q^*(b, c), a^*(b, c), b, c)$ to find the optimal values of parameters b^* and c^*. The corresponding parameters a^* and q^* are then calculated from Equations 4.102 and 4.103.

We have to remark, that we could not use the Levenberg-Marquardt least-squares optimization algorithm for searching for the parameters. Instead of it, we used the Trust region reflective algorithm since the Levenberg-Marquart least-squares optimization does not use the bounds for the inputs. However, if the b parameter is lesser than any λ_j, the x_j part of the simplification becomes a complex number since the mantissa $(\lambda_j - b)$ of the exponential function is a negative number, and this negative number is then raised to the power of c, which is a real number. We also had to restrict the parameters from below. If the algorithm found the optimum with the parameters in the order of magnitude larger than 3, the probability of failure becomes NaN. It seems that the minimized function $\tilde{F}(b, c)$ has several local optima or the problem is multimodal in our testing benchmarks, especially in Example 4 in Section 7.

Figure 4.20 shows an illustrative example with a rod under tension depicted in Figure 4.6. The first row of images presents the samples in the random $\mathbf{X}$-space; the red dots are samples in the failure domain; the blue dots are samples in the safe domain. The green line shows the original unscaled limit state $G(x) = 0$. The first image in this row depicts the original Monte Carlo simulation with corresponding λ equal to one. The rest of images in this row show

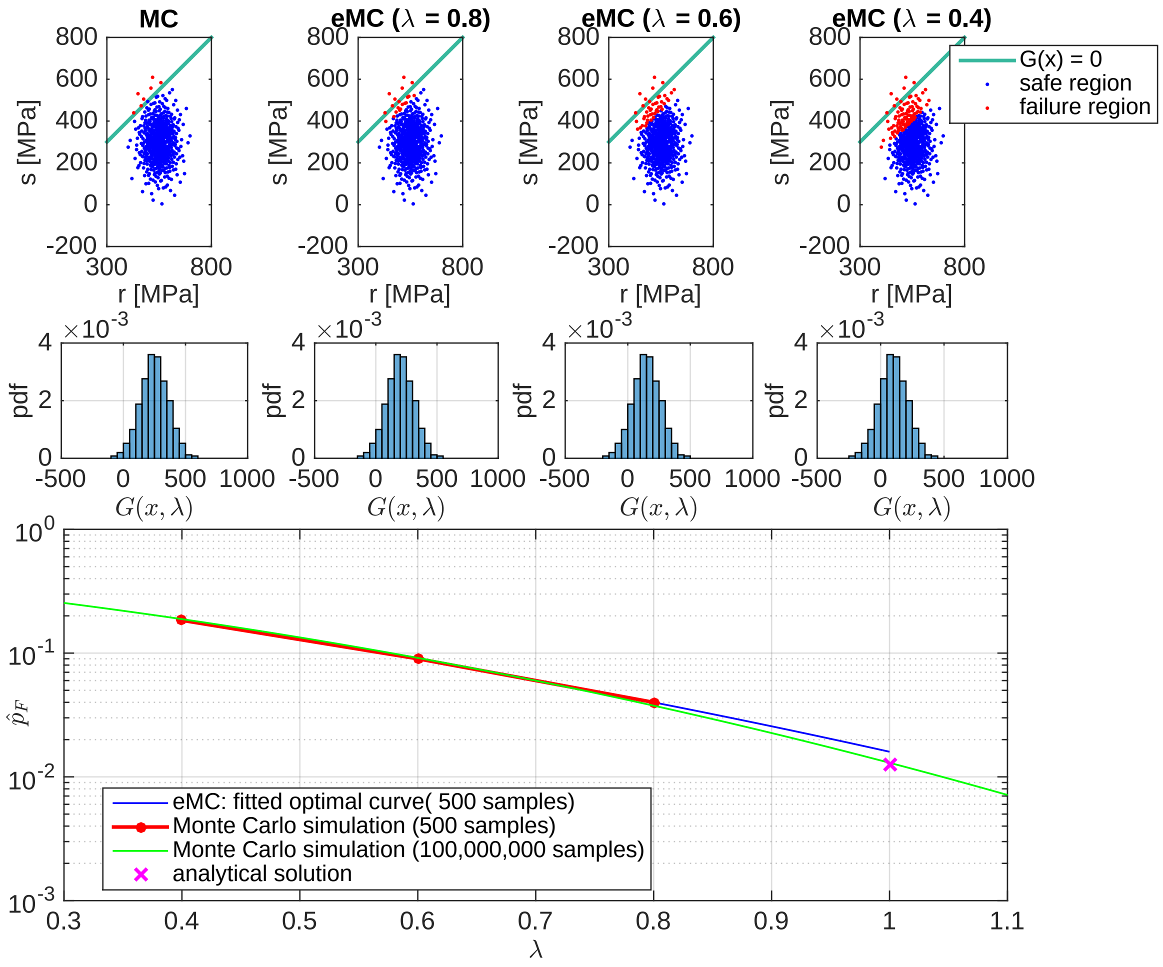


Figure 4.20: A rod under tension: Enhanced Monte Carlo simulation of the problem with two random variables - the capacity $R \sim N(550, 50)$ and the demand $S \sim N(300, 100)$.

samples with scaled limit state function values; the samples in the random $\mathbf{X}$-space remain unchanged since the scaling is carried out only in the space of the limit state function responses. The second row of images shows the histogram of the limit state function responses normalized into the probability density function appearance. These histograms correspond to samples and λ factors in the first row of images. The histograms have the same shape; their modification is carried out as translation closer to the failure domain, that is mathematically expressed as $G(\mathbf{x}) \leq 0$. The bottom image shows a graph with a scaling factor λ and the corresponding probability of failure $\hat{p}_F(\lambda)$. The red points show the pairs of $[\lambda,\hat{p}_F(\lambda)]$ from images above. The blue curve represents a fitted curve from Equation 4.96, where the value with λ equal to one is an approximation of the desired probability of failure. The green curve shows the Monte Carlo simulation with a very high number of samples, and therefore it should be very close to the correct solution. The magenta cross shows the analytical solution. The parameters $[\hat{q}, \hat{a}, \hat{b}, \hat{c}]$ are approximated as $[4.25 \cdot 10^{-18}, -19.75, 13.85, 24.20]$ and therefore the approximation of the probability of failure $\hat{p}_F$ is equal to $1.595 \cdot 10^{-2}$. The number of samples in the Monte Carlo simulation is equal to 500.

This methodology is also capable of solving system reliability analysis, as was published in the pioneering paper [136]. The extended parameterized class of limit state functions in Equation 4.93 becomes

$$G_j(\mathbf{X}, \lambda) = G_j(\mathbf{X}) - (1 - \lambda)\mathrm{E}[G_j(\mathbf{X})], \tag{4.104}$$

where j-index corresponds to the j^{th} limit state function. The probability of failure for a series system is then

$$p_F(\lambda) = \text{Prob}\left[\bigcup_{j=1}^{n_p}(G_j(\lambda) \leq 0)\right]. \tag{4.105}$$

In contrast, the probability of failure for a parallel system becomes

$$p_F(\lambda) = \text{Prob}\left[\bigcap_{j=1}^{n_p}(G_j(\lambda) \leq 0)\right]; \tag{4.106}$$

in both system reliability problems, n_p is a number of limit state functions. The combination of parallel-series systems is accomplishable as well.

Asymptotic sampling developed by Christian Bucher is based on a similar concept of extrapolation of the probability of failure. Both methods were published in the same year – 2009. The difference between the Enhanced Monte Carlo simulation and the Asymptotic sampling is that the formerly mentioned method eMC scales only the responses of the limit state function, the latter mentioned method AS scales the random variables. The Enhanced Monte Carlo, therefore, simulates only one Monte Carlo simulation and scales the responses of the limit state function in the post-process. The Asymptotic sampling needs several sequential simulations since it scales the inputs into the limit state function as the pre-process.

4.4.6 Scaled sigma sampling (SSS)

The Scaled sigma sampling methodology is based on fitting the relation of traditional numerical integration for failure probability evaluation by sequential Importance sampling with increasing standard deviations. SUN et al. published it in [185] as the novel methodology. The original problem of the failure probability is defined as

$$p_F = \text{Prob}[\mathbf{x} \in F] = \int_F f(\mathbf{x})d\mathbf{x}, \tag{4.107}$$

which means that the failure probability p_F is defined as the probability of events $\mathbf{x}$ that lie in the failure domain F. This occurrence is also possible to express through the integration of the probability density function $f(\mathbf{x})$ over the failure domain F. The probability of failure is theoretically possible to evaluate by traditional numerical integration, which divides the D-dimensional random $\mathbf{X}$-space into a union of disjoint hypercubes. For each hypercube i, a representative point $\mathbf{x}^{(i)}$ is selected inside the hypercube. If a volume of the hypercube i is ω_i, then the probability of failure is possible to approximate as

$$p_F \approx \sum_{\mathbf{x}^{(i)} \in F} f(\mathbf{x}^{(i)})\omega_i. \tag{4.108}$$

Introducing the indicator function $\mathbf{I}_F(\mathbf{x}^{(i)})$ equal to 1 if the $\mathbf{x}^{(i)}$ is in the failure domain F and equal to 0 otherwise; the probability of failure p_F is approximately

$$p_F \approx \sum_{i=1}^{N} \mathbf{I}_F(\mathbf{x}^{(i)}) f(\mathbf{x}^{(i)})\omega_i, \tag{4.109}$$

where N is a number of hypercubes. If all variables are divided into the equal number of intervals n, the number of hypercubes is equal to n^D. This traditional numerical integration is, therefore, unaffordable for a large number of variables. SUN et al. in [185], therefore, suggest approximating this relation with a linear regression through a sequential Monte Carlo simulation with an increasing standard deviation. They use a probability density function of a joint normal distribution with zero mean and equal standard deviation for all variables σ

$$h(\mathbf{x}) = \frac{1}{(\sigma\sqrt{2\pi})^D} \exp\left[\frac{-||\mathbf{x}||^2}{2\sigma^2}\right]. \tag{4.110}$$

Operator $||\cdot||$ denotes the Euclidean norm. Substitution this probability density function into Equation 4.108 and assuming that all volumes of hypercubes ω_i are identical ω one obtain

$$p_{F,h} \approx \frac{\omega}{(\sigma\sqrt{2\pi})^D} \sum_{\mathbf{x}^{(i)} \in F} \exp\left[\frac{-||\mathbf{x}^{(i)}||^2}{2\sigma^2}\right]. \tag{4.111}$$

The subscript h in $p_{F,h}$ denotes the substitution of the probability density function $h(\cdot)$ into Equation 4.108. Calculating the logarithm of both sides in Equation 4.111 and approximating log-sum-exp operator as the maximum we get

$$\log p_{F,h} \approx \log \frac{\omega}{(2\pi)^{\frac{D}{2}}} - D\log\sigma + \max_{\mathbf{x}^{(i)} \in F} \left[\frac{-||\mathbf{x}^{(i)}||^2}{2\sigma^2}\right] \tag{4.112}$$

The previous equation is possible to substitute with α, β, and γ parameters to obtain

$$\log p_{F,h} \approx \alpha + \beta \log\sigma + \frac{\gamma}{\sigma^2} \tag{4.113}$$

$$\alpha = \log \frac{\omega}{(2\pi)^{\frac{D}{2}}} \tag{4.114}$$

$$\beta = -D \tag{4.115}$$

$$\gamma = \max_{\mathbf{x}^{(i)} \in F} \left[-\frac{||x^{(i)}||^2}{2}\right]. \tag{4.116}$$

The above equation shows the relation between the scaled failure probability $\log p_{F,h}$ and the standard deviation σ, which is possible to understand as a scaling factor. The search for parameters α, β, and γ is, however, problematic due to the lack of knowledge about the failure region. Therefore, SUN et al. in [185] fit the analytical model in Equation 4.113 by linear regression for different scaling factors σ and corresponding scaled failure probabilities $\log p_{F,h}$. The original probability of failure p_F is then approximated as extrapolation

$$p_F \approx \exp[\alpha + \gamma]. \tag{4.117}$$

The maximum likelihood estimation fits the model in Equation 4.113. The number of sequential Monte Carlo simulations is set in advance as Q as well as the values of the standard deviations σ_q for $q = 1, 2, \dots, Q$. For each σ_q, a probability density function $h(\mathbf{x})$ is sampled with setting $\sigma = \sigma_q$ and with N_q samples. The limit state function is evaluated N_q times for each q Monte Carlo simulation and the probability of failure is estimated as

$$p^{MC}_{F,h,q} = \frac{1}{N_q} \sum_{n=1}^{N_q} \mathbf{I}[\mathbf{x}^{(n)}]. \tag{4.118}$$

For each estimation of $p^{MC}_{F,h,q}$ a variance of the estimator is evaluated

$$\text{Var}\, p^{MC}_{F,h,q} = \frac{1}{N_q} p^{MC}_{F,h,q}(1 - p^{MC}_{F,h,q}). \tag{4.119}$$

With this information, the maximum likelihood estimation of parameters $[\alpha, \beta, \gamma]$ arranged into a vector $\mathbf{\Theta}$ is estimated from relation

$$\mathbf{\Theta} = (\mathbf{A^T \Sigma_h^{-1} A})^{-1} \mathbf{A^T \Sigma_h^{-1}} \log \mathbf{P^{MC}_{F,h}}. \tag{4.120}$$

The matrix $\mathbf{A}$ contains standard deviations used as scaling factors of Monte Carlo simulations arranged as

$$\mathbf{A} = \begin{bmatrix} 1 & \log \sigma_1 & \sigma_1^{-2} \\ \vdots & \vdots & \vdots \\ 1 & \log \sigma_Q & \sigma_Q^{-2} \end{bmatrix}, \tag{4.121}$$

the covariance matrix $\mathbf{\Sigma}_g$ is arranged as

$$\mathbf{\Sigma}_g = \text{diag}\left[\frac{\text{Var}\, p^{MC}_{F,h,1}}{(p^{MC}_{F,h,1})^2}, \dots, \frac{\text{Var}\, p^{MC}_{F,h,Q}}{(p^{MC}_{F,h,Q})^2} \right], \tag{4.122}$$

and the logarithm random variable $\log \mathbf{P^{MC}_h}$ is adjusted as

$$\log \mathbf{P^{MC}_{F,h}} = \left[\mathbf{p^{MC}_{F,h,1}}, \dots, \mathbf{p^{MC}_{F,h,Q}} \right]^{\mathbf{T}}. \tag{4.123}$$

This approach is quite similar to the Asymptotic sampling, which also extrapolates the probability of failure from several Monte Carlo simulations with different scaling factors projected into the standard deviations. The difference is in the model of extrapolation and its fitting. An Asymptotic sampling utilizes only reliability indices and the scaling factors from Monte Carlo simulations with a sufficient amount of samples in the failure domain. These pairs create the support points for the model in Equation 4.82, and subsequently, Asymptotic sampling parameters are predicted by a linear regression model written in Equation 4.83. While the Scaled sigma sampling methodology utilizes more information from Monte Carlo simulations, not only the probability of failure and the scaling factors as in the Asymptotic sampling but also a variance of the failure probability estimator is included into the fitting of the model in Equation 4.113. The estimators $\log \mathbf{P}^{\mathbf{MC}}_{\mathbf{F,h}}$ are therefore weighted by the inverse of the covariance matrix $\mathbf{\Sigma}_h$. This means that the large variance of the estimator $\log \mathbf{P}^{\mathbf{MC}}_{\mathbf{F,h}}$ reduces its influence to parameter fitting in $\mathbf{\Theta}$.

The probability of failure was evaluated for the illustrative example as well as for the other reliability assessment methods; a rod under tension depicted in Figure 4.6. The number of all structural analyses was set to 5,000, and the scaling factor σ_q was set to [2,3,4,5,6]. Figure 4.21 (left) shows all sequential sampling sets (called generations 1 - 5) differentiated from each other with different colours. Each sampling set is sampled via a Monte Carlo simulation with different scaling factors. The problem was sampled in the standard normal space and subsequently transformed into the target space for evaluation of the limit state function purposes as well as for the illustrative purposes. The teal line represents the limit state dividing the space into the safe and the failure region. Figure 4.21 (right) shows the pairs of scaling factor σ_q together with log value of the failure probability for each generation $\log p^{MC}_{F,h,q}$. Those pairs were interposed by an approximation curve (blue curve), and the approximation of the failure probability estimator is read for the scaling factor σ_q equal to 1. The probability of failure estimator is equal to $6.49 \cdot 10^{-3}$, and the corresponding reliability index is equal to 2.484 for this particular simulation.

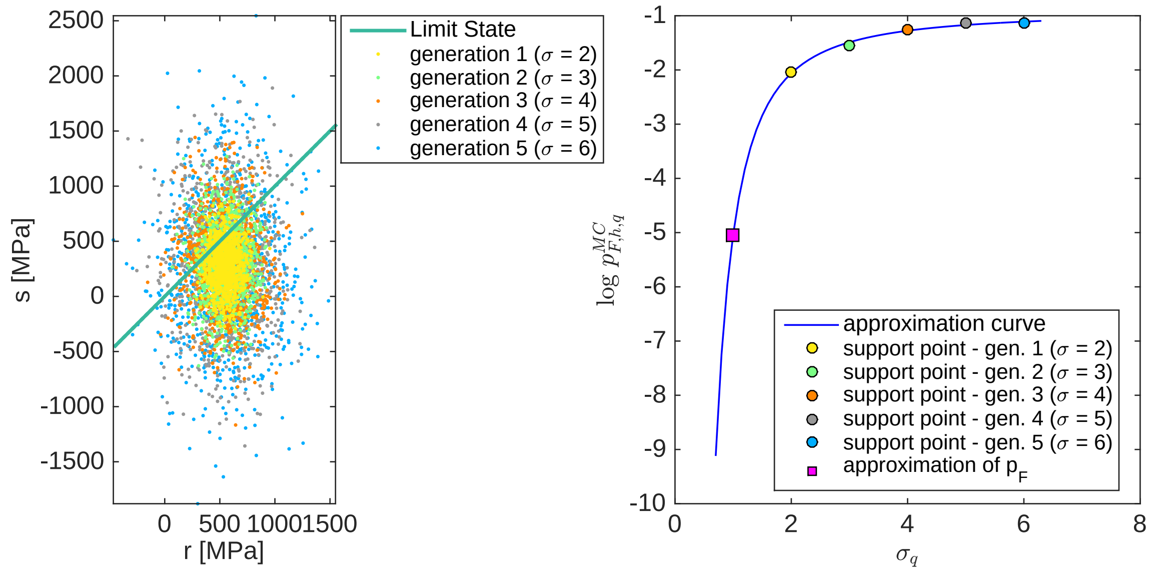


Figure 4.21: A rod under tension: Scaled sigma sampling of the problem stress-strength model with two random variables - the capacity $R \sim N(550, 50)$ and the demand $S \sim N(300, 100)$. The probability of failure estimator is equal to $6.49 \cdot 10^{-3}$ and the corresponding reliability index is equal to 2.484. The maximum likelihood estimation of parameters $[\alpha, \beta, \gamma]$ arranged in the vector $\mathbf{\Theta}$ is equal to $[-1.1430; 0.0792; -3.8945]$.

Chapter 5

Surrogate models

Parts of this chapter are reproduced from the author's contributions [87, 88, 89, 134, 149, 151, 154].

Surrogate models can be divided into two fundamental parts: (i) *non-interpolating models* minimizing thesum of squares errors from some predetermined functional form (e.g. *polynomial surfaces*) and (ii) *interpolating models* intersecting all support points based on an idea of a linear combination of some basis functions. Interpolating models can contain fixed basis functions (e.g. *thin-plate splines* or *multiquadrics*) in Radial basis functions model or basis functions to be tuned, e.g. *Gaussian basis in Radial basis functions model* or *Kriging* [96].

5.1 Radial Basis Functions model

A Radial Basis Functions model (RBF) approximates a complicated but relatively smooth true function or produces an estimate to an unknown function from a set of input data. The approximation is made by evaluation of easier basis functions using input dataset, multiplying these results by weight coefficients and summing them together. For noise-free data, the model is typically of the form

$$\hat{y}(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\psi} = \sum_{i=1}^{n_c} w_i \psi(||\mathbf{x} - \mathbf{c}^{(i)}||), \tag{5.1}$$

where $\mathbf{w}$ is a weighting vector, $\boldsymbol{\psi}$ is a vector of length n_c holding evaluated basis functions on Euclidean distances between the prediction $\mathbf{x}$ and centres of basis functions $\mathbf{c}$. The Euclidean norm is evaluated as

$$||r|| = \sqrt{(\sum_{i=1}^{n_e} r_i^2)} \tag{5.2}$$

for a vector r with n_e elements. Other metrics are possible; however, the Euclidean metric is quite often used.

The basis functions are symmetrical due to the Euclidean norm [28] and centred on a set of support points [74]. Basis functions can be fixed or parametric (to be tuned) and are chosen by the user according to the type of the original function. Table 5.1 and Figure 5.1 show frequently used functions. An optimization algorithm can found parameters of the bases.

Fixed bases	
linear	$\psi(r) = r$
cubic	$\psi(r) = r^3$
thin plate spline	$\psi(r) = r^2 \ln[r]$
Parametric bases	
Gaussian	$\psi(r) = \exp\left[\frac{-r^2}{\sigma^2}\right]$
Hardy multiquadrics	$\psi(r) = \sqrt{r^2+\sigma^2}$
Inverse multiquadrics	$\psi(r) = \frac{1}{\sqrt{r^2+\sigma^2}}$
Inverse quadrics	$\psi(r) = \frac{1}{r^2+\sigma^2}$
C^0 Matérn	$\psi(r) = \exp[-\sigma r]$
C^2 Matérn	$\psi(r) = (1+\sigma r)\exp[-\sigma r]$
C^4 Matérn	$\psi(r) = (3+3\sigma r+(\sigma r)^2)\exp[-\sigma r]$

Table 5.1: Radial basis functions according to [73], [74], [82], [145], [164]

A Gaussian radial basis function is one of the most common choices; however, several variants of a definition of Gaussian basis exist. The difference is mainly in the definition of a scaling parameter σ (in some literature called shape parameter). Other variants are for example $\exp\left[\frac{-r^2}{2\sigma^2}\right]$, $\exp\left[f-(\sigma r)^2\right]$ or $\exp\left[-\sigma r^2\right]$. For the Gaussian basis defined in Table 5.1, Figure 5.2 plots different scaling parameters σ. According to [137], the approximation of parameter σ can be computed as

$$\sigma = \sqrt{\frac{d_{\max}}{\sqrt[n_d]{n_d \cdot n_c}}} \tag{5.3}$$

where $d_{\max}$ is a maximum distance among the data and n_d is the number of dimensions equivalent to the number of variables. This formula represents how large would be a space around each point in DoE if the points in DoE had been optimally uniformly distributed.

Centres of basis functions can be identical to the Design of Experiments $\mathbf{c}^{(i)} = \mathbf{x}^{(i)}$. This DoE is then used as a set of training data. It is necessary to evaluate the original function $y(\mathbf{x})$ in the training dataset, assemble DoE to the Gram matrix $\mathbf{\Psi}$ and solve the linear system of

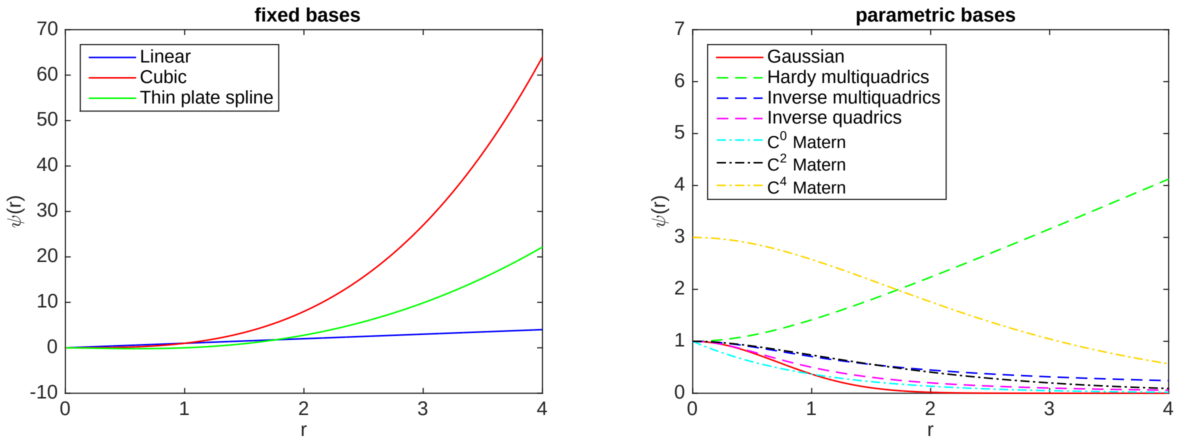


Figure 5.1: Commonly used radial basis functions listed in Table 5.1.

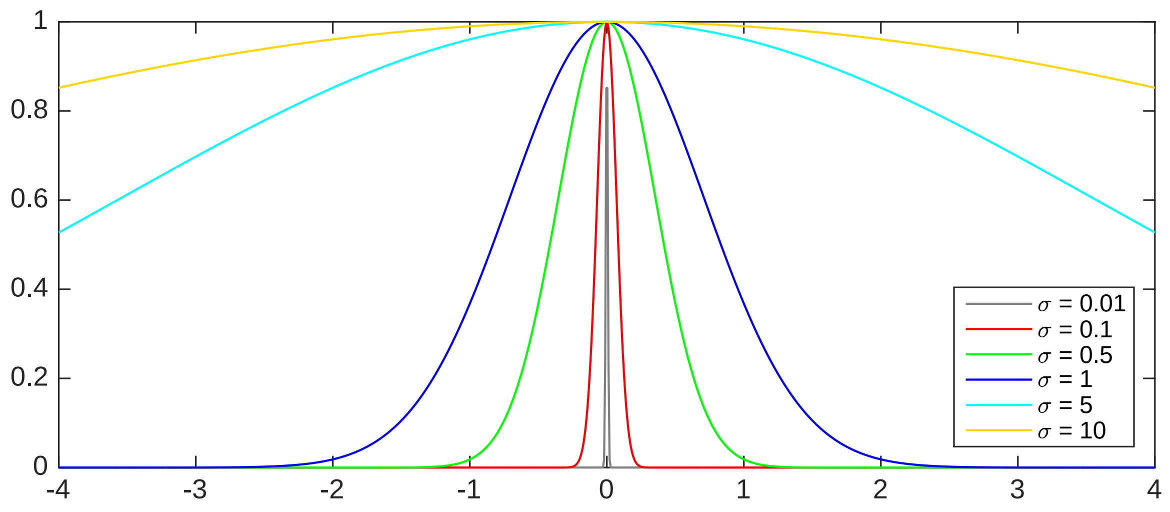


Figure 5.2: Gaussian radial basis functions with different scaling parameter setting.

equations

$$\sum_{i=1}^{n_c} w_i \psi(||\mathbf{x}^{(j)} - \mathbf{x}^{(i)}||) = y(\mathbf{x}^{(j)}), \quad j = 1, \ldots, n_c. \tag{5.4}$$

to obtain parameters in a weighting vector. Equation 5.4 can be rewritten in a matrix form as

$$\mathbf{\Psi}\mathbf{w} = \mathbf{y} \tag{5.5}$$

where each element of the Gram matrix $\mathbf{\Psi}$ is $\Psi_{i,j} = \psi(||\mathbf{x}^{(i)} - \mathbf{x}^{(j)}||)$, where $i, j = 1, \ldots, n_c$ and n_c is a number of points in the DoE, fully rewritten as

$$\mathbf{\Psi} = \begin{pmatrix} \psi(||\mathbf{x}^{(1)} - \mathbf{x}^{(1)}||) & \psi(||\mathbf{x}^{(1)} - \mathbf{x}^{(2)}||) & \cdots & \psi(||\mathbf{x}^{(1)} - \mathbf{x}^{(n_c)}||) \\ \psi(||\mathbf{x}^{(2)} - \mathbf{x}^{(1)}||) & \psi(||\mathbf{x}^{(2)} - \mathbf{x}^{(2)}||) & \cdots & \psi(||\mathbf{x}^{(2)} - \mathbf{x}^{(n_c)}||) \\ \vdots & \vdots & \ddots & \vdots \\ \psi(||\mathbf{x}^{(n_c)} - \mathbf{x}^{(1)}||) & \psi(||\mathbf{x}^{(n_c)} - \mathbf{x}^{(2)}||) & \cdots & \psi(||\mathbf{x}^{(n_c)} - \mathbf{x}^{(n_c)}||) \end{pmatrix}. \tag{5.6}$$

This Gram matrix is symmetrical and positive definite, e.g. for the Gaussian basis functions or the inverse multiquadrics [74, 191]. Therefore, Cholesky decomposition is advantageous to use for small and moderate problems. Iterative methods can solve large-scale problems. The Gram matrix is also possible to precondition. It is also worth mentioning, that if points in the training dataset are very close to each other, it can cause ill-conditioning of the Gram matrix and problems with Cholesky decomposition [74].

Radial basis functions model can be represented as a network structure with one hidden layer; the example of the representation is depicted in Figure 5.3. The input layer contains the data $\bar{\mathbf{x}}^{(1)}, \ldots, \bar{\mathbf{x}}^{(J)}$, where the prediction of the meta-model is desired. The number of the data is identical to the number of nodes in this input layer. The data is then distributed to the next hidden layer, and it comprises evaluated basis functions for the Euclidean distances between the data for the response prediction and the centres of basis functions, where the meta-model is trained. The number of nodes is identical to the number of centres. Each of the nodes contains a scalar value for one particular node from the input layer. The evaluated basis functions are passed into the output layer via weighted connections, and the weighted values of basis functions are summed to the final prediction of the meta-model $\hat{y}^{(1)}, \ldots, \hat{y}^{(J)}$. The output layer contains as many nodes as the input layer and is independent of the number of nodes in

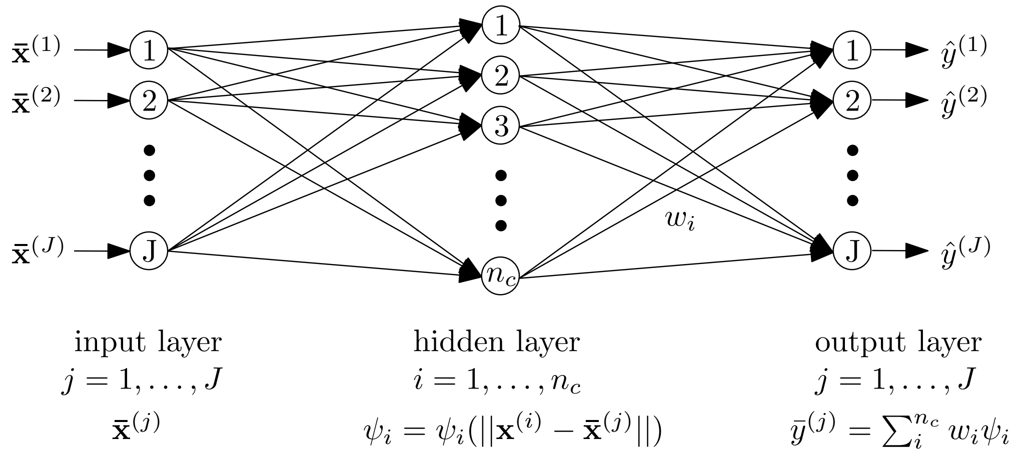


Figure 5.3: Diagram showing the Radial basis function model as a network structure.

the hidden layer, which corresponds to the number of centres. In case that the training data are identical to the centres of the bases, i.e. the number of centres is identical to the number of the data where the meta-model is trained, then the number of nodes in the hidden layer is identical to the number of nodes in the input/output layer, i.e. n_c is equal to J.

5.1.1 Global meta-models

The first model that we have used is the classical model that contains all information from all DoE points. This model satisfies everything that is described in the previous section. Therefore, it comprises links between all support points, where the links are represented by elements in a dense Gram matrix. Figure 5.4 shows a 2-dimensional illustrative example with 20 points in DoE, the image on the left depicts this DoE, and the right part represents the visualisation of the sparsity pattern of the Gram matrix constructed from the DoE. The large green dot represents one construction sample. In the dense global meta-model, the whole DoE creates the influence domain for all points; therefore the influence domain covers all points in DoE (magenta circles).

5.1.2 Sparse global meta-models

The influence of the point diminishes with the increasing distance from the point. This means that the distant construction samples have a little impact on one another. If the domain is wide, as it is for the optimization tasks, the global meta-model would need a large number of points in DoE to describe the whole space and the dense matrix would have large memory demands. Therefore, instead of utilizing all links between all support points, each support point has its influence domain in the closest neighbourhood, and other points outside the influence domain are omitted. The resulting Gram matrix is, therefore, sparse and should be faster to solve.

Two possible algorithms exist for finding the influence domain. The first one is a *range search algorithm*; this algorithm selects all points lying inside the sphere with a perimeter r_s and the centre of the sphere in the query point Q identical to the i^{th} support point. The second one is a *k-nearest neighbour algorithm* (*k*-NN), a frequently used tool for classification. The closest point from DoE to the query point Q is called the nearest neighbour $nn(Q)$. For this point holds [196]

$$nn(Q) = P \in DoE \mid \forall P' \in DoE : ||P - Q||_2 \leq ||P' - Q||_2, \tag{5.7}$$

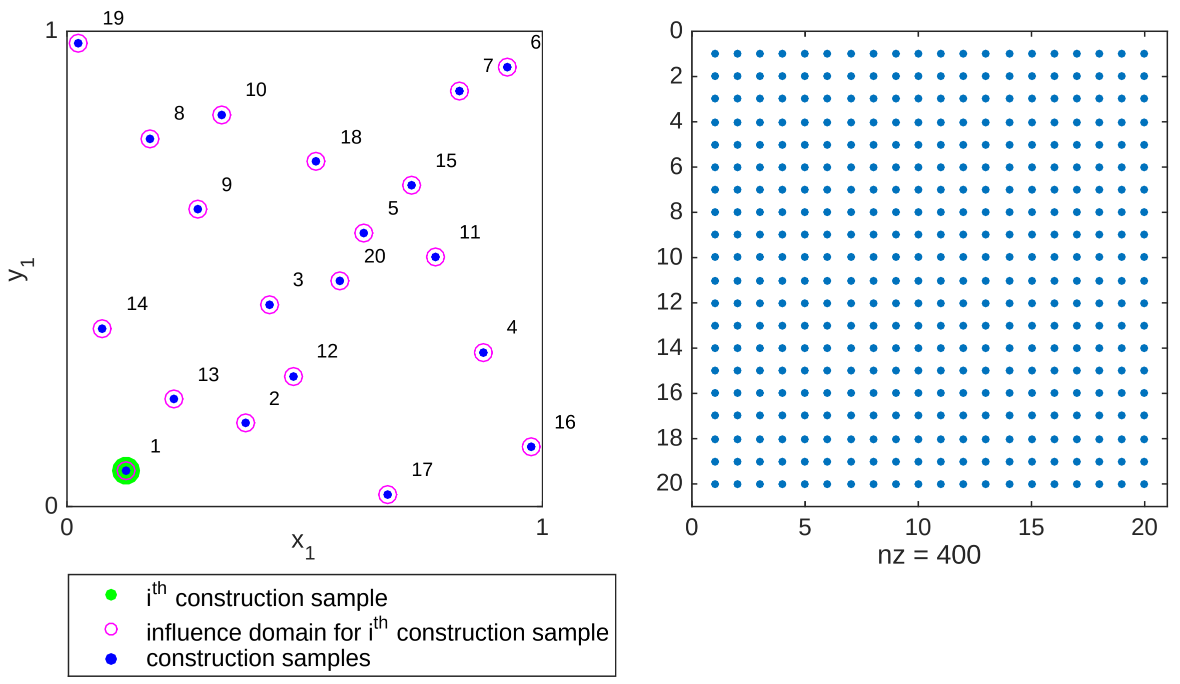


Figure 5.4: Influence domain for classical RBF and the sparsity pattern visualization of a Gram matrix for a global meta-model. Abbreviations: nz - nonzero elements.

where P is the nearest neighbour, and P' is any point from DoE. The distance is then evaluated as

$$nn^{dist}(Q) = ||nn(Q) - Q||_2. \tag{5.8}$$

The *k*-NN algorithm finds *k*-closest points in the DoE to the query point Q, that is again identical to the i^{th} support point. Algorithms for the range search as well as for the *k*-NN algorithm use the same algorithms in MATLAB according to [128]. A *kd-tree* is used if DoE is not sparse, the dimension is less than or equal to 10, and the metric is Minkowski

$$d = \sqrt[p]{\sum_{j=1}^{n_d} |x_j - y_j|^p}, \tag{5.9}$$

where d is the distance, n_d is a dimensionality of the problem, x is a point from DoE and y is a query point. Special cases of Minkowski metric are the City Block metric (also called the Manhattan metric) with p equal to 1, the Euclidean metric with p equal to 2 and the Chebychev metric with p equal to ∞. If DoE is sparse, the dimension of the problem is greater than ten, and the metric is different from the Minkowski, the *exhaustive search* is used. The exhaustive search evaluates the distances from the query point to each point in DoE, the list of points is sorted in ascending order according to the distances, the k number of points is selected, and the list with selected points is returned as a result. The dimensionality greater than 10 is critical to the similarity searches methods [196], tree-based algorithms are even slower than a brute-force search in which the exhaustive search belongs to as well.

In our work, we use the radial basis functions model with Gaussian basis. The maximal range r_s taken into account as influential is possible to relate to the scaling parameter σ. The basis $\exp\left[\frac{-r^2}{\sigma^2}\right]$ can be identified with the weight of the point. Therefore, the influential order x of this weight is considered as in the relation

$$10^{-x} = \exp\left[\frac{-r^2}{\sigma^2}\right]. \tag{5.10}$$

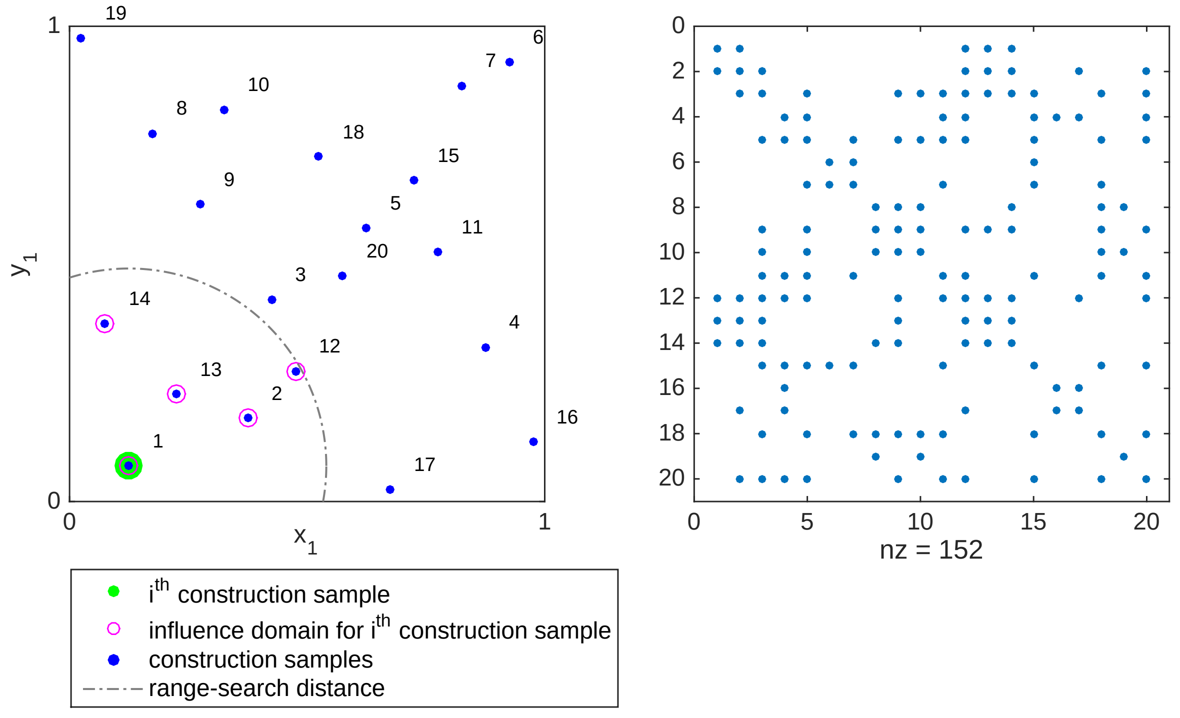


Figure 5.5: Influence domain for sparse global RBF using a range search algorithm and sparsity pattern visualization of a Gram matrix for a sparse global meta-model. Abbreviations: nz - nonzero elements.

After some easy math, we obtain

$$-x\ln[10] = \frac{-r^2}{\sigma^2}. \tag{5.11}$$

The influential distance r is made identical to the perimeter of the sphere r_s as

$$r_s \approx \sigma\sqrt{2.3x}. \tag{5.12}$$

In practice, if we want to consider as influential distances given to seven decimal places as nonzero, we set x equal to seven. All other distances will be neglected and therefore considered in the Gram matrix as zero if the matrix is dense or not considered at all if the matrix is sparse.

Figure 5.5 on the left shows the same DoE as Figure 5.4 with identical construction samples. The right part of this figure is dedicated to the sparse Gram matrix. For one particular construction sample, the influential perimeter of the sphere is evaluated, and the range-algorithm determines points inside this sphere. These points are influential for the i^{th} construction sample. Then, the next point of the DoE is selected and another influential set of samples is selected for the same influential perimeter. The algorithm terminates when all design samples have their influential domains. Specifically, Figure 5.5 depicts the 1st construction sample (the large green dot) with influence domain depicted by the grey dot-and-dash circle and defined by points $\mathcal{D}_1 \in (1, 2, 12, 13, 14)$ represented by magenta circles, which appears in the sparse Gram matrix by filling the first row and the first column in these positions. Other distances are omitted. Note, that the range search algorithm variant of the sparse Gram matrix provides a symmetrical and positive definite matrix which is decomposable by the Cholesky decomposition. Another interesting aspect is that the number of points differs in the influence domain for each construction point. If the influential perimeter is set too narrow, the influence domain contains only the

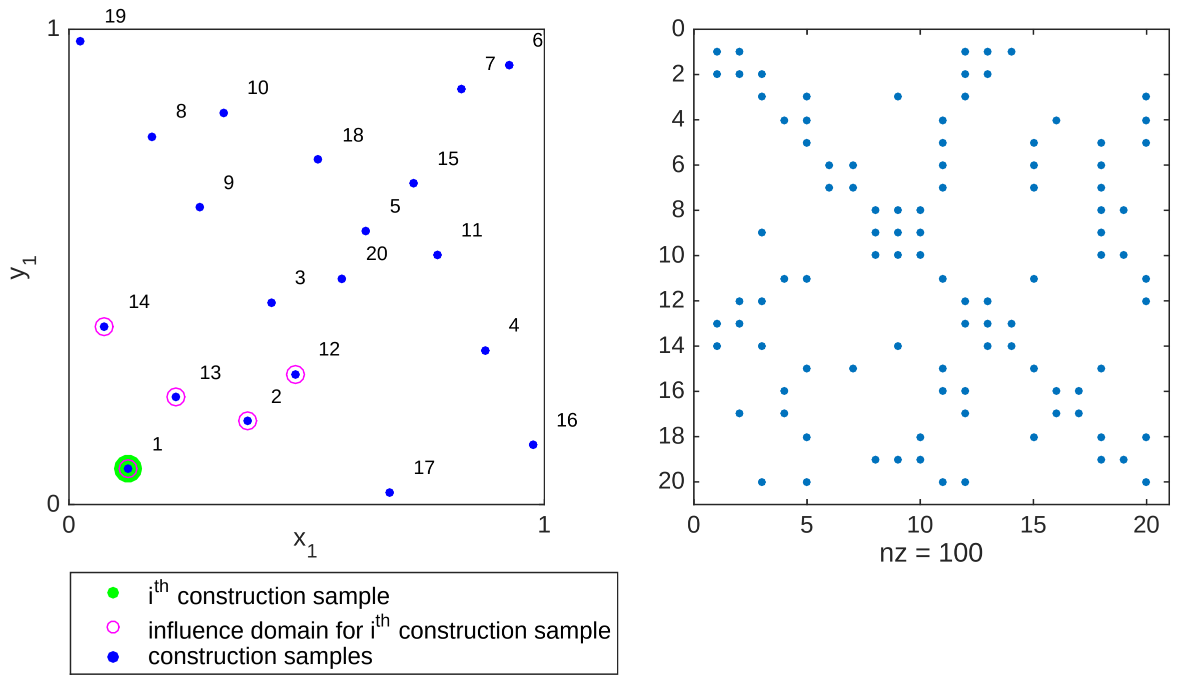


Figure 5.6: Influence domain for sparse global RBF using a *k*-NN algorithm and sparsity pattern visualization of a Gram matrix for a sparse global meta-model. Abbreviations: nz - nonzero elements.

particular construction point. On the other side, if the perimeter is set to the distance between the most distant points, all points will be placed in the influence domain, and the Gram matrix will be dense.

Figure 5.6 shows the selection of the points in the influence domain for a *k*-NN algorithm. This algorithm always keeps k number of points in each influence domain for each construction point. This provides an advantage that the number of filled positions in the Gram matrix is known in advance and do not vary. On the other side, the matrix is not always symmetrical, as apparent from the right side of Figure 5.6. To show the difference between Gram matrices constructed using the range search and *k*-NN algorithm, we selected the number of influence points equal to five. This number of points is identical to the number of influence points for the first construction sample in the example with the range search algorithm in Figure 5.5. These points have an impact on the sparse Gram matrix by filling the first row in these positions. Other distances are omitted. Note, that the first column is not filled identically as the first row and therefore the matrix is not symmetrical. The same phenomenon may occur in other rows and columns.

5.1.3 Local meta-models

Local meta-models are constructed only in the influence domain. They use a similar idea as sparse meta-models; however, the idea about omitting the non-interesting domain goes further. As being said, the domain for the optimization is extensive. The area for the reliability assessment around the potential optimum is much narrower. Therefore, local meta-models are constructed only with samples in this area in contrast to sparse global meta-models that utilize the influence area only for i^{th} construction sample. The Gram matrix is smaller in contrast to a global meta-model and dense in contrast to a sparse global meta-model. Specifically for Figure 5.7 on the left, if the first point of a DoE (green filled circle) is a potential op-

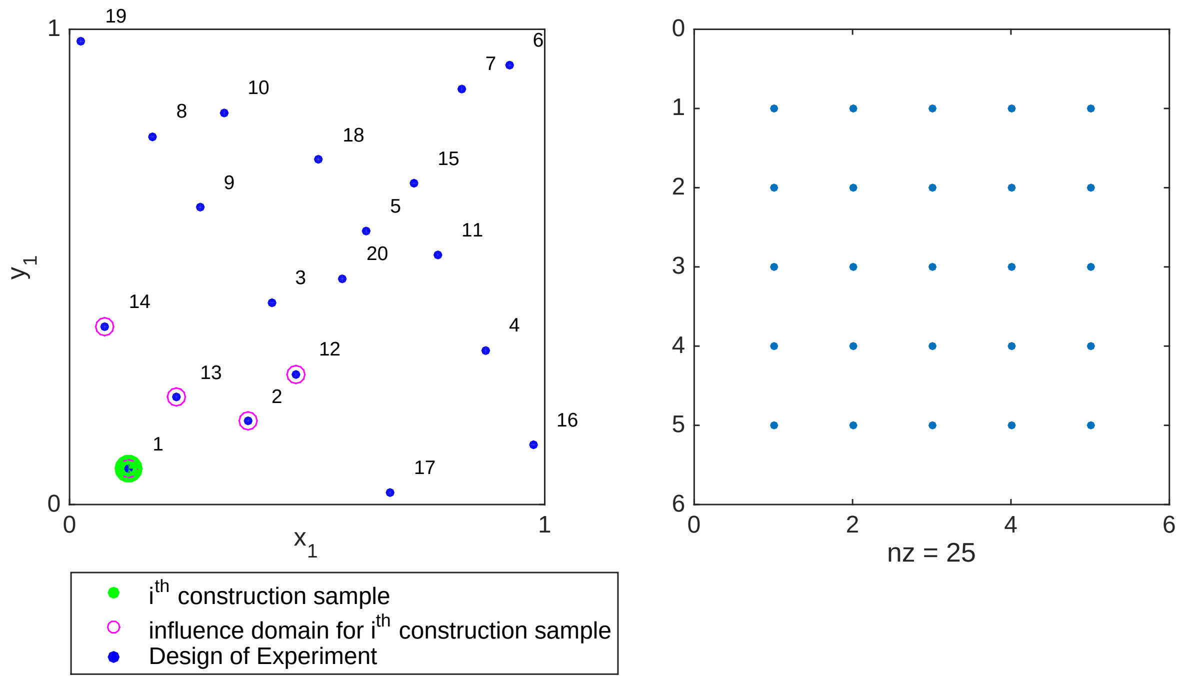


Figure 5.7: Influence domain for local RBF using a k-NN algorithm and the and sparsity pattern visualization of a Gram matrix for a local meta-model. Abbreviations: nz - nonzero elements.

timum, the influence domain for whole local meta-model is chosen by a k-NN algorithm as $\mathcal{D}_1 \in (1, 2, 12, 13, 14)$ (the magenta circles). Other samples of DoE are omitted (only blue dots without any other specification). After this selection of influence domain, the local meta-model is constructed in the same manner as the dense global meta-model. Therefore, the Gram matrix has only five rows and five columns for this illustrative example. For a different potential optimum, it is necessary to choose its influence domain by a k-NN algorithm and construct a different local meta-model. The advantage of local meta-models is that the Gram matrix is smaller than for a global meta-model and the solving of the linear equation system is fast, and it has fewer memory demands. The disadvantage is that the span of the influence domain has to be selected in advance and the meta-model needs to extrapolate for the choice of a too narrow area.

5.1.4 Implementation details

Since the meta-model construction and its evaluation takes a significant portion of evaluation time, we had to optimize the implementation. The steps of the procedure (pseudocode) are as follow:

1. Find a Design of Experiments as a dataset $\mathbf{X}$ for a meta-model construction and evaluate the original function $y(\mathbf{x})$ in the dataset $\mathbf{X}$.

2. For the selected basis function, assemble the Gram matrix $\mathbf{\Psi}$ for centres of basis functions identical to the dataset $\mathbf{X}$.

3. Evaluate weights $\mathbf{w} = \mathbf{\Psi}^{-1}\mathbf{y}$.

4. Find a Design of Experiments $\mathbf{Z}$ in which the meta-model should predict the model behaviour.

5. Evaluate the response of the meta-model by $\hat{y}(\mathbf{z}) = \sum_{i=1}^{n_c} w_i \psi(||\mathbf{z} - \mathbf{c}^{(i)}||)$.

For a complicated original model, the most time-consuming part is its evaluation in the dataset $\mathbf{X}$. If the parameter of the Gaussian basis functions is estimated by Equation 5.3, the second most time-consuming part is a calculation of the interpoint distances, which are requisite in Step 2 as well as in Step 5 of the pseudocode above. The next place regarding the consumed computational time is for the linear equation solving for weight evaluation in Step 3.

Interpoint distances

Assume a Design of Experiments as a dataset for a meta-model construction stored in a matrix $\mathbf{X}$ and a set of samples for a meta-model prediction stored in a matrix $\mathbf{Z}$, both in d-dimensional space. We need to compute the distances between any point in $\mathbf{X}$ and any other point in $\mathbf{Z}$, or vice-versa. The basic and the most straightforward evaluation of these interpoint (pairwise) distances is via three embedded for-loops. The outer loop runs over the number of points in $\mathbf{X}$ (line 5-9 in the following code), the inner loop runs over the number of points in $\mathbf{Z}$ (line 6-8), and the third hidden loop is vectorized via dimensions of the problem (line 7). Unfortunately, this code is slow because of the mentioned for-loops.

Fundamental evaluation of interpoint distance via three for-loops

```
1  function distMat_2 = distance2_basic(X,Z)
2    numInX = size(X,1); % number of points in X
3    numInY = size(Z,1); % number of points in Z
4    distMat_2 = zeros(numInX,numInZ); % preallocation
5    for i = 1:numInX
6      for j = 1:numInZ
7        distMat_2(i,j) = sum((X(i,:)-Z(j,:)).^2);
8      end
9    end
10 end
```

The interpoint distances are evaluated in the meta-model assembly and subsequent meta-model evaluation. Since we use Gaussian parametric bases, where Euclidean distances are raised to the power of two, see Table 5.1, the output matrix containing distances are squared in all compared distance functions.

Method A: MATLAB standard `pdist2` algorithm Method A uses a MATLAB algorithm `distMat = pdist2(X,Z)`, which returns a matrix `distMat` with the Euclidean distances between each pair of observations in the matrix $\mathbf{X}$ and $\mathbf{Z}$, where rows correspond to observations and columns correspond to variables [129]. The algorithm itself is unfortunately inaccessible due to conversion of an m-file into a mex-file, and therefore whole evaluation routine is hidden in `pdist2mex`. This mex-file version of `pdist2` has been used since MATLAB version R2010a.

distance2_A.m file

```
function distMat_2 = distance2_A(X,Z)
  numInX=size(X,1); % number of points in X
  numInY=size(Z,1); % number of points in Z
  distMat = zeros(numInX,numInZ); % preallocation
  distMat = pdist2(X,Z); % MATLAB pairwise distance calculation
  distMat_2 = distMat.*distMat;
end
```

Method B: `bsxfun` version Method B was proposed in [182]. Here, $\mathbf{x}_i$ is used as a vector in $\mathbf{X}$ and $\mathbf{z}_j$ as a vector in $\mathbf{Z}$. The squared distance d_{ij} between $\mathbf{x}_i$ and $\mathbf{z}_j$ is

$$d_{ij}^2 = \|\mathbf{x}_i - \mathbf{y}_j\|^2 = \|\mathbf{x}_i\|^2 + \|\mathbf{z}_j\|^2 - 2 < \mathbf{x}_i, \mathbf{z}_j >, \tag{5.13}$$

where $< \cdot, \cdot >$ is a dot product. The squared distance d_{ij}^2 can be considered as an entry of the matrix D^2. Therefore, the formula in matrix form can be written as:

$$D^2 = \bar{\mathbf{x}}1' + 1\bar{\mathbf{z}}' - 2\mathbf{X}'\mathbf{Z} \tag{5.14}$$

where, $\bar{\mathbf{x}}$ is a column vector of squared norms of all vectors in $\mathbf{X}$. Eq. 5.14 can be easily implemented in MATLAB just by one line, see line 5 of the following code. The MATLAB function `C = bsxfun(fun,A,B)` applies the element-wise binary operation specified by the function handle `fun` to arrays `A` and `B`.

distance2_B.m file

```
1  function distMat_2 = distance2_B(X,Z)
2    numInX=size(X,1);  % number of points in X
3    numInY=size(Z,1);  % number of points in Z
4    distMat_2 = zeros(numInX,numInZ);   % preallocation
5    distMat_2 = bsxfun(@plus,dot(X,X,2),dot(Z,Z,2)')-2*(X*Z');
6  end
```

Method C: two helper matrices This approach was proposed in [125] as the fastest among other comparable algorithms. It is based on a matrix multiplication (line 15 in the following code) of two matrices that are composed of addends from Equation 5.13 stepwise for all dimensions via for-loop as

$$\begin{array}{lcl} \texttt{helpA} & = & [\ \mathbf{1}, \quad -2\cdot\mathbf{X}_1, \quad \mathbf{X}_1^2, \quad \ldots \quad \mathbf{1}, \quad -2\cdot\mathbf{X}_n, \quad \mathbf{X}_n^2 \ ], \\ \texttt{helpB} & = & [\ \mathbf{Z}_1^2, \quad \mathbf{Z}_1, \quad \mathbf{1}, \quad \ldots \quad \mathbf{Z}_n^2, \quad \mathbf{Z}_n, \quad \mathbf{1} \ ], \end{array}$$

where a lower index in matrices $\mathbf{X}$ and $\mathbf{Z}$ means the particular dimension.

distance2_C.m file

```
1  function distMat_2 = distance2_C(X,Z)
2    numDim=size(Z,2); % number of dimensions
3    numInX=size(X,1); % number of points in X
4    numInY=size(Z,1); % number of points in Z
5
6    helpA = zeros(numInX,3*numDim); % preallocation
7    helpB = zeros(numInZ,3*numDim); % preallocation
8    for idx = 1:numDim
9      helpA(:,3*idx-2:3*idx) = ...
10             [ones(numInX,1), -2*X(:,idx), X(:,idx).^2 ];
11     helpB(:,3*idx-2:3*idx) = ...
12             [Z(:,idx).^2 ,  Z(:,idx), ones(numInZ,1)];
13   end
14   distMat_2 = zeros(numInX,numInZ); % preallocation
15   distMat_2 = helpA * helpB';
16 end
```

Method D: one for-loop over DoE This method is a slightly modified version with two for-loops. Here, the only for-loop runs over the number of samples in X, the rest of the code is vectorized. Another version of line 6 is via `bsxfun` as already used in Method B as
`d = sum(bsxfun(@minus,X(i,:),DoE).^2,2);`
however, we obtained faster results with the proposed code.

distance2_D.m file

```
function distMat_2 = distance2_D(X,Z)
  numInX=size(X,1);  % number of points in X
  numInY=size(Z,1);  % number of points in Z
  distMat_2 = zeros(numInX,numInZ); % preallocation
  for i=1:numInX
    distMat_2(i,:) = sum((ones(numInZ,1)*X(i,:)-Z).^2,2);
  end
end
```

Method E: one for-loop over MC Samples The method E is slightly modified version of the method D. Here, the only for-loop runs over the number of samples in Z, the rest of the code is vectorized using `repmat` function that repeats a copy of an array.

distance2_E.m file

```
function distMat_2 = distance2_E(X,Z)
  numInX=size(X,1); % number of points in X
  numInY=size(Z,1); % number of points in Z
  distMat_2 = zeros(numInX,numInZ); % preallocation
  for i=1:numInZ
    distMat_2(:,i) = sum((X-repmat(Z(i,:),numInX,1)).^2,2);
  end
end
```

Parallelization We implemented a parallelization of presented five methods A – E in MATLAB environment via a `parfor` loop technology. Hardware and software parameters are specified in Table 11.1. The testing methodology setup has studied the influence of the number of CPUs and the size of the batched data that are sent at once to CPUs on the performance of the individual algorithms. A million random samples are evaluated for the distance calculation in consecutive batches containing 100, $1,000$, $10,000$, and $100,000$ points, respectively. For the sake of statistics, all data are averages out of 10 independent runs.

Obtained times are listed in Table 5.2. The first column describes the number of used CPUs; a zero number is used for a serial version with a for-loop, in which MATLAB utilizes more than one CPU by itself; one number is used for a parallel version using a single CPU. From the first sight, methods A, B and C are faster than methods D and E; the method C is being the fastest almost irrespective to the number of CPUs and batch sizes.

Figure 5.8 presents data for method A in a graphical format. The trends of other methods in terms of obtained times are very similar. The deterioration caused by sending too much data is visible in the case of $100,000$ sample batches. Figure 5.9 highlights the same trend showing the speed-up for all four batch sizes. While using smaller batches, linear speed-up is obtained, reaching value above 6 for 8 CPUs. Figure 5.9 also shows an interesting point that the MATLAB serial version using classical for-loop is able to use more processing units without user intervention.

Another interesting issue is the bad performance of method E with small batches. Unfortunately, MATLAB does not allow memory profiling and, therefore, it is difficult to deduce the cause of the problem. We assume that MATLAB needs some additional memory for ten thousand small batches that runs out of free memory. Except method C, all other methods need the same amount of memory. Only method C uses two additional helper matrices.

CPUs	Batch	A	B	C	D	E
0	$1{\cdot}10^5$	26,2	18,3	15,4	60,8	24,5
	$1{\cdot}10^4$	30,2	25,0	15,1	60,5	34,3
	$1{\cdot}10^3$	30,3	24,9	19,0	51,3	62,9
	$1{\cdot}10^2$	30,2	18,3	26,1	52,3	275,2
1	$1{\cdot}10^5$	68,9	63,2	61,9	103,8	85,7
	$1{\cdot}10^4$	68,7	65,7	60,2	104,8	84,9
	$1{\cdot}10^3$	65,0	62,0	57,8	91,8	101,7
	$1{\cdot}10^2$	68,5	59,8	58,1	82,9	294,8
2	$1{\cdot}10^5$	35,8	33,8	32,3	55,1	44,1
	$1{\cdot}10^4$	36,0	33,1	30,8	53,6	43,9
	$1{\cdot}10^3$	34,6	31,4	31,3	48,2	52,9
	$1{\cdot}10^2$	35,5	31,7	30,7	43,2	151,8
3	$1{\cdot}10^5$	28,2	26,3	25,3	42,1	34,8
	$1{\cdot}10^4$	24,5	23,0	21,1	36,4	29,6
	$1{\cdot}10^3$	23,7	21,9	21,5	32,6	36,2
	$1{\cdot}10^2$	25,3	23,0	21,8	29,8	102,8
4	$1{\cdot}10^5$	21,7	21,2	19,8	32,2	27,0
	$1{\cdot}10^4$	18,7	17,8	16,4	27,9	22,9
	$1{\cdot}10^3$	18,4	17,1	16,6	24,8	27,5
	$1{\cdot}10^2$	19,3	17,8	16,7	22,8	78,3
5	$1{\cdot}10^5$	15,5	24,8	14,5	22,7	20,4
	$1{\cdot}10^4$	15,5	14,7	13,6	22,9	18,9
	$1{\cdot}10^3$	15,3	14,4	13,8	20,6	22,8
	$1{\cdot}10^2$	16,3	15,0	14,0	19,1	65,7
6	$1{\cdot}10^5$	16,8	82,0	14,3	22,3	22,2
	$1{\cdot}10^4$	13,5	13,1	12,0	20,0	16,6
	$1{\cdot}10^3$	13,0	12,5	12,0	17,5	19,4
	$1{\cdot}10^2$	13,9	13,0	12,0	16,3	55,5
7	$1{\cdot}10^5$	80,3	95,4	149,4	110,1	50,7
	$1{\cdot}10^4$	11,5	11,5	10,4	17,0	14,2
	$1{\cdot}10^3$	11,3	11,0	10,6	15,2	16,9
	$1{\cdot}10^2$	12,2	11,4	10,5	14,1	47,8
8	$1{\cdot}10^5$	137,7	310,3	260,4	225,5	88,3
	$1{\cdot}10^4$	10,4	10,5	9,4	15,3	12,7
	$1{\cdot}10^3$	10,0	9,8	9,4	13,5	14,9
	$1{\cdot}10^2$	10,8	10,2	9,5	12,7	42,0

Table 5.2: Obtained times in seconds for a different number of CPUs and size of batches for five implemented algorithms A – E.

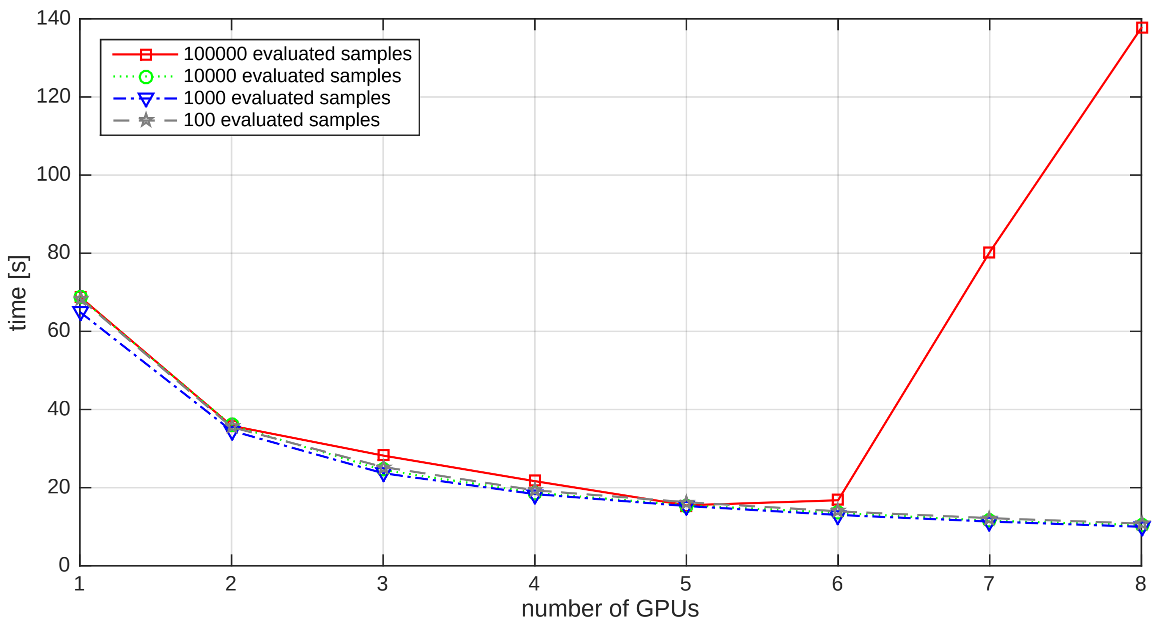


Figure 5.8: Obtained times for method A and four sizes of batch data.

Comparison of sparse and dense RBF models

The Gram matrix of the sparse global meta-model is assembled in a slightly different manner than the Gram matrix of the dense model, and therefore, not all of the interpoint distances algorithms from Section 5.1.4 can be used. We made the comparison of the Gram matrices assembling on `Distance2_d.m` algorithm version. Both models require `X` matrix with the centres of bases identical to the training points, `sigma` parameter as the scaling factor and `lambda` parameter for a regularization. The output is the `rbfs` variable containing the Gram matrix. Op-

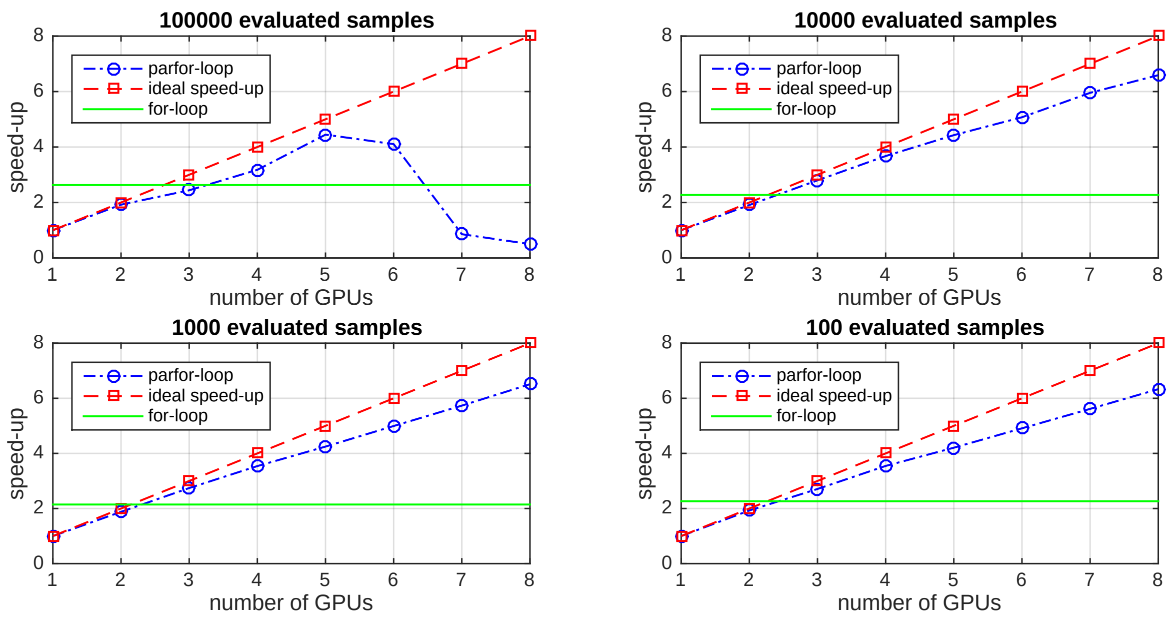


Figure 5.9: Speed-up for method A for four sizes of batch data.

timization founds the σ parameter or its approximation can be calculated by Equation 5.3. The regularization factor λ prevents a matrix to be only semidefinite but to become positive definite. Since the Gram matrix contains the Euclidean distances between points passed into the radial basis functions and the distances on the diagonal are zeros in case that the centres are identical to the training points, the whole diagonal is zero and the matrix is therefore only semidefinite. The small positive constant λ is then added to diagonal entries of the Gram matrix Ψ as $\Psi + \lambda I$. The eigenvalues are thereafter only positive and the Gram matrix becomes definite.

assembly_Gram_matrix function

```matlab
function rbfs = assembly_Gram_matrix(X,sigma,lambda)
% inputs: X ... Design of Experiments corresponding to centres of bases,
%                      number of rows represents a number of points,
%                      number of columns represents a number of dimensions
%          sigma ... scaling factor
%          lambda ... regularization factor
% output: rbfs ... Gram matrix
numInX=size(X,1); % number of centres
rbfs = zeros(numInX,numInX);

for i=1:numInX
    delta = sum(bsxfun(@minus,X(i,:),X).^2,2)'; % squared Euclid. dist.
    rbfs(i,:) = exp( -delta./sigma); % Gaussian basis
    rbfs(i,i) = rbfs(i,i) + lambda; % regularization
end
```

The assembling of the sparse Gram matrix utilizing k-NN algorithm is in the following algorithm. The training points `X`, the scaling factor `sigma`, and the regularization factor `lambda` have the same meaning as in the previous code. Besides, it is necessary to define the number of nearest points in the influence domain `K`. A k-NN algorithm is used as the built-in function from MATLAB and its 'Statistics and Machine Learning Toolbox'. The first input is a matrix, the second input is a matrix of query points, and the third input defines that the fourth input is a positive integer specifying the number of nearest neighbours `K` in the first input for each point in the second input of this function. The Gram matrix is sparse, and therefore it is necessary to save the position of the filled element by its row and column index and its value. The memory is preallocated for row indices in variable `sparse_x`, for the column indices in variable `sparse_y` and values of the element in variable `sparse_vals`. The number of nonzero points is known in advance because of the k-NN algorithm and the fixed number of points in the influence domain. The matrix is assembled and then converted into the sparse matrix by the MATLAB command `sparse(i,j,v)` from triplets `i`, `j`, and `v` such that `S(i(k),j(k))=v(k)`. The last line performs regularization of the Gram matrix to ensure this matrix is positive definite.

assembly_sparse_Gram_matrix function

```matlab
function rbfs = assembly_sparse_Gram_matrix(X,sigma,lambda,K)
% inputs: X ... Design of Experiments corresponding to centres of bases,
%                      a number of rows represents number of points,
%                      a number of columns represents number of dimensions
%          sigma ... scaling factor
%          lambda ... regularization factor
%          K ... number of points in the influence domain
% output: rbfs ... sparse Gram matrix
numInX=size(X,1); % number of centres
points_id = knnsearch(X,X,'k',K); % map for the nearest points

% allocation of memory for sparse matrix
sparse_x = zeros(1,K*numInX);
```

```
sparse_y = zeros(1,K*numInX);
sparse_vals = zeros(1,K*numInX);

for i=1:numInX
    delta = sum(bsxfun(@minus,X(i,:),X(points_id(i,:),:)).^2,2)';
    % delta ... squared Euclidean distances
    rbfs_loc = exp( -delta./sigma); % Gaussian basis
    indx = i*ones(1,length(points_id(i,:)));
    sparse_x(1,(i-1)*K+1:i*K) = indx; % id of filled row positions
    sparse_y(1,(i-1)*K+1:i*K) = points_id(i,:); % id of filled col. pos.
    sparse_vals(1,(i-1)*K+1:i*K) = rbfs_loc; % values of filled positions
end

rbfs = sparse(sparse_x,sparse_y,sparse_vals); % sparse matrix assembly
rbfs = rbfs + sparse(1:numInX,1:numInX,lambda*ones(1,numInX));
```

Figure 5.10 shows the memory requirements for sparse and dense Gram matrices. The influence region contains 10% of the points from the whole domain in the nearest neighbourhood. The comparison is made for hundred, thousand, and ten thousands of construction samples. The memory requirements for hundred of samples are larger for a sparse Gram matrix than for the dense Gram matrix. For other cases are memory requirements for a sparse Gram matrix lesser than for the dense Gram matrix. The darkest shade of green is for hundred construction samples; the last point of this curve represents the dense matrix since the influence domain also contains a hundred of samples. The sparse container requires more space for storing the dense matrix than the regular storage of a variable.

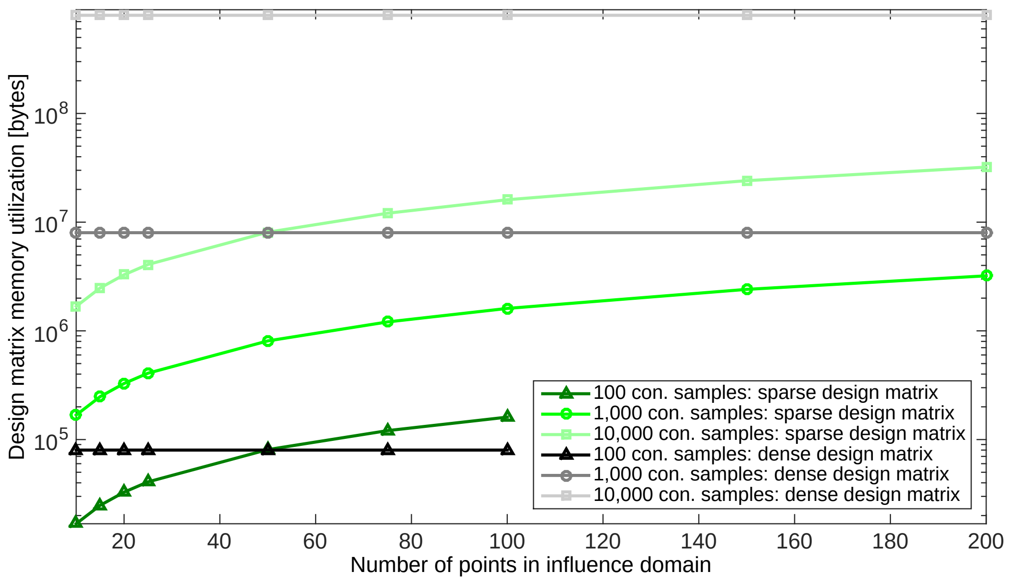


Figure 5.10: Memory requirements for the sparse Gram matrix and dense Gram matrix. 100, 1,000, and 10,000 construction samples were used. Dense Gram matrix is fully filled, while the sparse Gram matrix utilizes influence region occupying the number of points on the x-axis from the whole domain.

Solvers of the linear equation system

Sparse and dense RBF models were compared to achieve the best time savings. Three designs of experiments were generated with a hundred, thousand, and ten thousand construction samples. We chose a Cholesky decomposition to solve a system of linear equations with the dense Gram matrix, while a conjugate gradient squared method (`cgs`) and the MATLAB `mldivide` function (namely the Unsymmetric MultiFrontal PACKage) were used for the sparse Gram matrix.

Figure 5.11 shows the time spent on evaluation of weights by solving the system of linear equations. Blue lines are for conjugate gradient squared method, the red lines are for Unsymmetric MultiFrontal PACKage, and the green lines are for the Cholesky decomposition on the dense matrix. The darkest shades are for hundred points, the middle shades for thousand points, and the lightest shades for ten thousands of construction points. The horizontal axis represents the number of points in the influence domain for the sparse Gram matrix. It is efficient to use sparse Gram matrices together with Unsymmetric MultiFrontal PACKage only for hundred of construction samples, designs of a larger number of construction samples are more efficient to solve by sparse matrices with a conjugate gradient squared method. Evaluation of weights with the Cholesky decomposition was most computationally intensive in comparison with the other solvers for the same number of construction samples.

Since the size of the influence domain does not affect only the computing time but also the precision of the result, we measured the root-mean-square error (RMSE) of the response $\hat{y}$ obtained from the dense Gram matrix between the sparse solver and the Cholesky decomposition on the same testing dataset represented in Figure 5.12. Above fifty points in the influence domain, the Unsymmetric MultiFrontal PACKage provides the same results as the Cholesky decomposition and the RMSE is zero. The conjugate gradient squared method was faster but the precision is worse in comparison with the Unsymmetric MultiFrontal PACKage; nevertheless, 4% error is acceptable for thousand and ten thousand of construction points with more than 20 points in the influence domain. The RMSE would probably be zero for a large number of steps but at the expense of the computational time.

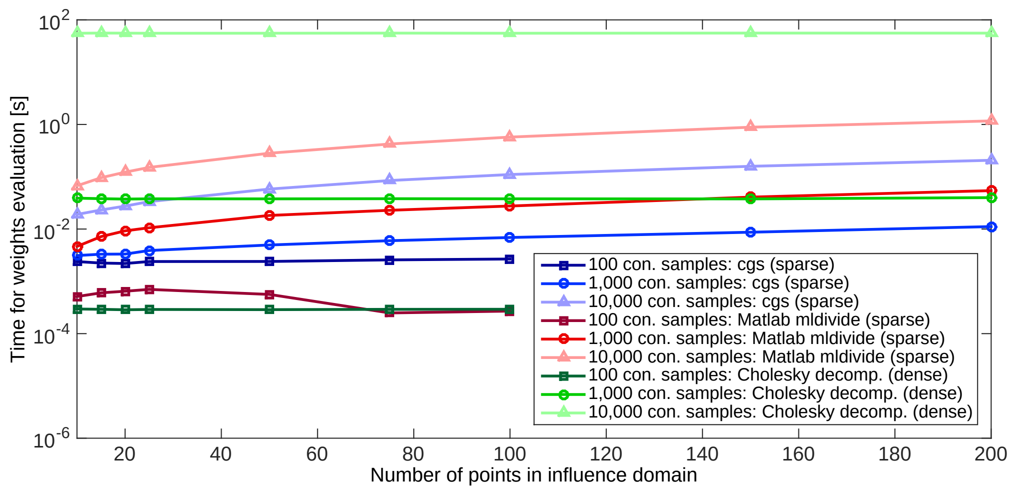


Figure 5.11: Time for RBF weights evaluation.

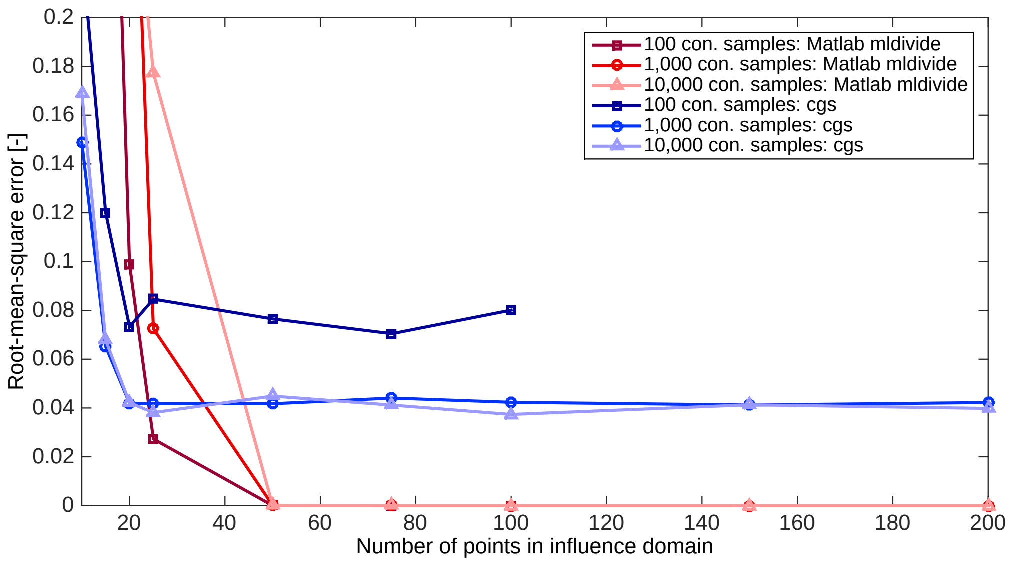


Figure 5.12: RMSE between dense and sparse RBF.

5.2 Kriging

Kriging is very often called the *DACE stochastic process* model where DACE is an abbreviation for *Design and Analysis of Computer Experiments*. In different research areas, this process has several names. It is called e.g. *Kriging* [41] in mathematical geology and *Bayesian global optimization* [97, 210] or *Random function approach* [15] in global optimization. Kriging was originally developed by the South African mining engineer D.G. KRIGE in the early fifties. In the 1960s, the French mathematician G. MATHERON [126] gave theoretical foundations to this method and named the method after KRIGE [74].

The value of a response function $y(x^{(i)})$ at a point $x^{(i)}$ is derivable as a linear combination of functions loaded by an error, i.e.

$$y(x^{(i)}) = \sum_{h} \beta_h f(x_h^{(i)}) + \epsilon^{(i)}, \quad i = 1, \dots, n, \tag{5.15}$$

where β_hs are unknown coefficients to be estimated, $f(x_h^{(i)})$ are (non)linear functions, h is a number of functions to be combined, and n is a number of points. Kriging is working on the assumption that computational experiments are deterministic and they are affected by a dependent error [74], which can be pictured as the left-out terms in x. The closer any two points, the more related (*correlated*) errors are.

To consider the distance between two points, the weighted distance function $d(x^{(i)}, x^{(j)})$ is presented

$$d(x^{(i)}, x^{(j)}) = \sum_{h=1}^{k} \theta_h \left| x_h^{(i)} - x_h^{(j)} \right|^{p_h}, \quad \theta_h \geq 0, p_h \in [1, 2], \tag{5.16}$$

where k is a number of variables and θ_hs and p_hs are correlation parameters; θ_h measures the sensitivity of a function to the change of the variable x_h ('*width* parameter') and p_h measures smoothness of the function ('*smoothness* parameter'). Some literature refers to use p_h equal to 2 and fit only θ_h parameters [176]. In some cases, only one parameter θ_h for the whole process is also recommended [176].

The correlation between two errors $\epsilon(x^{(i)})$ and $\epsilon(x^{(j)})$ can be defined as

$$\mathrm{Corr}\left[\epsilon(x^{(i)}),\epsilon(x^{(j)})\right]=\exp\left[-d(x^{(i)},x^{(j)})\right]. \tag{5.17}$$

This metric is called the Gaussian correlation function (or anisotropic generalized exponential model [59]), and it is the most used in literature. If point $x^{(i)}$ and $x^{(j)}$ are close to each other, the correlation is approaching one and reversely, the correlation is near zero for distant points.

The correlation between points is so powerful tool that the regression part (the first term) in Equation 5.15 can be replaced by a constant μ,

$$y(x^{(i)})=\mu+\epsilon(x^{(i)}), \tag{5.18}$$

which is a mean of a stochastic process and the error $\epsilon(x^{(i)})$ (stationary Gaussian process [59]) is normally distributed zero-mean random variable with auto-covariance matrix $\mathbf{C_{XX}}$

$$\mathbf{C_{XX}}=\sigma^2\mathbf{R}. \tag{5.19}$$

Each element of the correlation matrix R_{ij}[1] is created by Corr $\left[\epsilon(x^{(i)}),\epsilon(x^{(j)})\right]$ as in Equation 5.17. Another possibility is to use linear or other polynomial functions instead of the first term in Equation 5.15 [176]; however, only the constant μ term is used in the following work.

Now, we have $2k+2$ unknown parameters μ, σ, $p_1,\ldots,p_k$ and $\theta_1,\ldots,\theta_k$. Maximizing the likelihood function L is one way how to estimate them. Other methods are variographic analysis or Bayesian estimation [59].

The predictor is defined as [169]

$$\hat{y}(x^*)=\hat{\mu}+\mathbf{r}^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-\mathbf{1}\hat{\mu}), \tag{5.20}$$

where $\mathbf{r}$ is a correlation vector between a new point x^* where the response is predicted and the data used for meta-model construction.

The mean squared error (MSE) of the predictor is

$$s^2(x^*)=\sigma^2\left[1-\mathbf{r}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{r}+\frac{(1-\mathbf{1}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{r})^2}{\mathbf{1}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{1}}\right] \tag{5.21}$$

and the standard error is the squared root of MSE

$$s(x^*)=\sqrt{s^2(x^*)}. \tag{5.22}$$

5.3 Update of meta-models for reliability assessment

The type of a meta-model update depends on a field of the model application. Despite appearing similar, a reliability assessment and reliability optimization require a different updating approach. Consider the evaluation of the failure probability p_F in an D-dimensional space of random variables $X_1,\ldots,X_D$ as a multiple integral of a joint probability density function $f_{\mathbf{X}}(\mathbf{x})$ over the failure domain $G(\mathbf{X})\leq 0$. In the standard normal space, the Hasofer-Lind reliability index β can be geometrically understood as the shortest connecting line between the origin and the *most probable failure point* (MPFP) also called the design point. For improving the behaviour of the meta-model for the reliability assessment, the update is the most essential around this MPFP, because this region contributes most to the total failure probability. BUCHER and BOURGUND [26] came with this idea of updating the response surface utilizing second-order polynomials.

[1]The correlation matrix is symmetrical, thus it is possible to assembly just the half of it and use a Cholesky decomposition instead of inverse matrix enumeration.

5.4 Updates for reliability-based design optimization

Meta-models can replace both the cost function and the deterministic and probabilistic constraints. Nevertheless, there is a difference in their consecutive update. If the meta-model is used to compute the values of the objective function, an improvement of the meta-model is necessary for the vicinity of the best-so-far optimum found to allow for convergence to the global optimum. This idea is adopted from global optimization. Several strategies are used, namely *minimizing a response surface* [96], *minimizing a statistical lower bound* [96], *maximizing the probability of improvement* [107, 96], *maximizing the expected improvement* [97] or *goal-seeking* [96]. On the other hand, if a meta-model is utilized for the constraint replacement, some contour (e.g. a limit state) has to be approximated. Algorithms as the *Efficient Global Reliability Analysis* (EGRA) [17], modified *Active Kriging + Monte Carlo Simulation* [62] or *Meta-model-based importance sampling* [59] can be effectively employed. The latter algorithms utilize some special meta-models features that are not available for all types of meta-models. The generic update can be carried out through placing new support points in a vicinity of the limit state still ensuring the uniform space-filling criterion, e.g. by the MiniMax metric which leads to the multi-objective optimization [135]. Although the replacement of the probabilistic constraint by a meta-model seems to be a similar problem as in the reliability assessment, the layout is different due to its repeated evaluation for different designs. The limit state function can be understood as a collection of the MPFPs for different design variables combinations, and thus the meta-model updating only in the one MPFP vicinity is not sufficient, but the vicinity of the whole contour needs to be updated.

5.4.1 Updates for meta-models replacing an objective function in RBDO

Minimizing a response surface approach

Minimizing a response surface approach [96] is independent of a used meta-model type. New support points are added sequentially into the best-so-far optimum found on the meta-model. In the case of multi-modal problems, this method can converge prematurely in a local optimum or fail in the worst possible case. Additional points in DoE for the non-interpolating meta-models with a fixed number of degrees of freedom need not help at all, and this method can be misleading in finding any optima.

Figure 5.13 shows a function $y(x) = (6x-2)^2 \sin(12x-4)$ which is replaced by a quadratic surface and the next support point is adaptively added into the response surface minimum. Nor the fourth iteration cannot improve the response surface behaviour for the low polynomial degree. The behaviour of interpolating surrogate model is much better than for a non-interpolating meta-model which is shown in Figure 5.14, but the quality of the meta-model highly depends on the initial Design of Experiments. Since the new support point is added to the meta-model minimum, the true minimum can be missed as depicted in Figure 5.15.

Minimizing a statistical lower bound

Minimizing a statistical lower bound [74], [97] utilizes the ability of Kriging to compute MSE of a prediction. Uncertainty in the prediction is lesser (low MSE) in areas with a higher concentration of support points and reversely. The statistical lower bound is then evaluated as

$$LB(\mathbf{x}) = \hat{y}(\mathbf{x}) - A_{LB}s(\mathbf{x}) \tag{5.23}$$

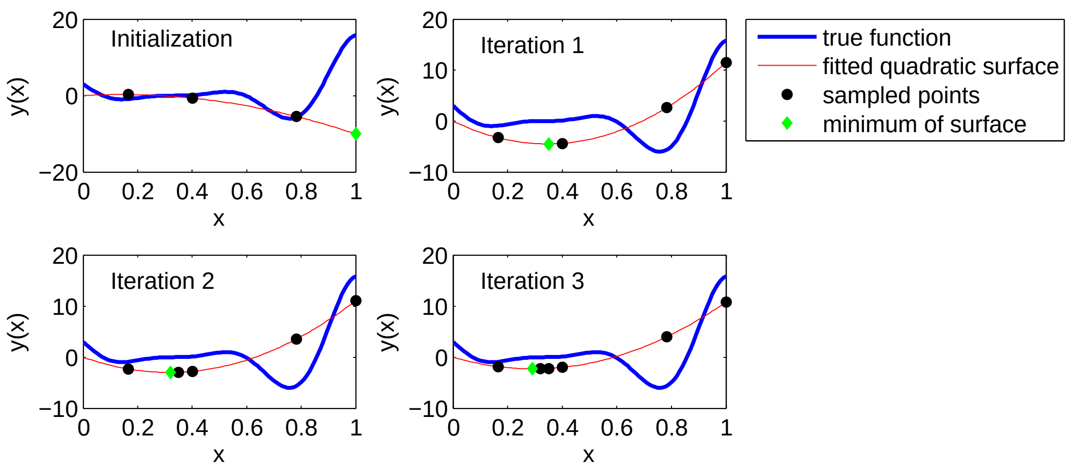


Figure 5.13: Minimizing a response surface approach with a quadratic surface.

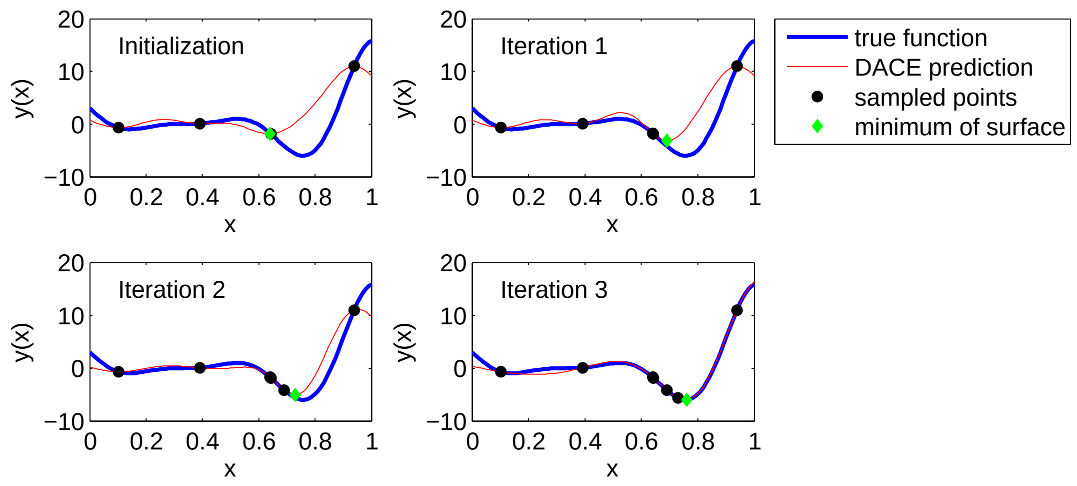


Figure 5.14: Minimizing a response surface approach - Kriging. The behaviour is much better than by a quadratic surface in Figure 5.13.

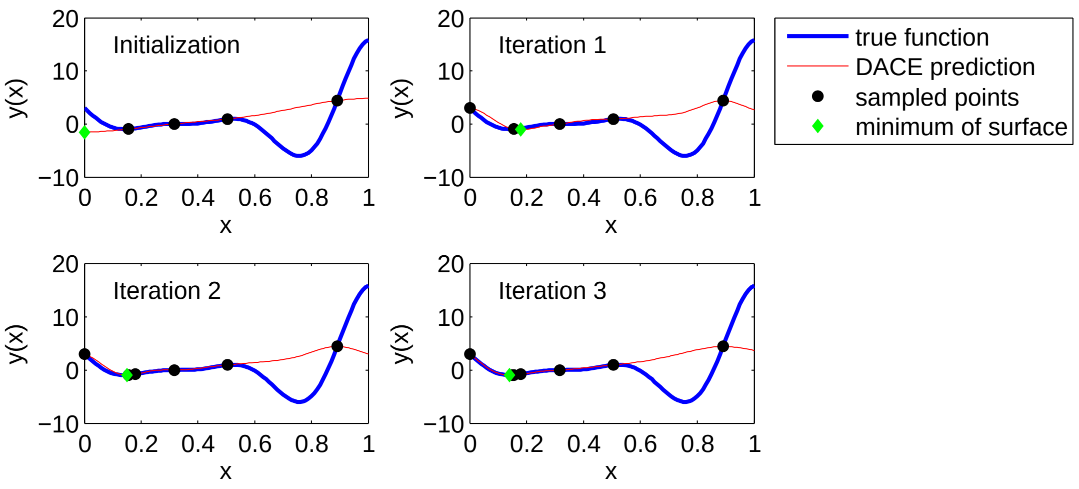


Figure 5.15: Minimizing a response surface approach with a Kriging meta-model. The minimum of the true function is missed due to the inappropriate initial Design of Experiments.

where $\hat{y}(\mathbf{x})$ is a Kriging prediction, $s(\mathbf{x})$ is a Kriging error, and A_{LB} is a constant that influences the search of areas to be improved. The closer is A_{LB} to zero, the more exploitation is involved in the minimum of the predictor because the Kriging prediction predominates in Equation 5.23. DACE prediction itself is a lower bound with A_{LB} equal to zero. The more A_{LB} approaches infinity, the more exploration is participated because of the prevailing the second addend in Equation 5.23. Figure 5.16 shows several statistical lower bounds with different settings of the parameter A_{LB}. FORRESTER [74] chooses the parameter manually according to the behaviour of the function. JONES [97] sets different parallel A_{LB} levels and clusters new gained support points, however, this approach is relatively time consuming. Figure 5.17 depicts sequential updates with A_{LB} equal to 2.

Maximizing the probability of improvement

Maximizing the probability of improvement [107], [97] also employees the knowledge of the Kriging prediction error. The output y^* at the point x^* not identical to any support point is not known, therefore y^* can be modelled as a random normal variable Y with a mean equal to the Kriging prediction $\hat{y}(\mathbf{x})$ and a standard deviation s equal to the Kriging standard error. To find

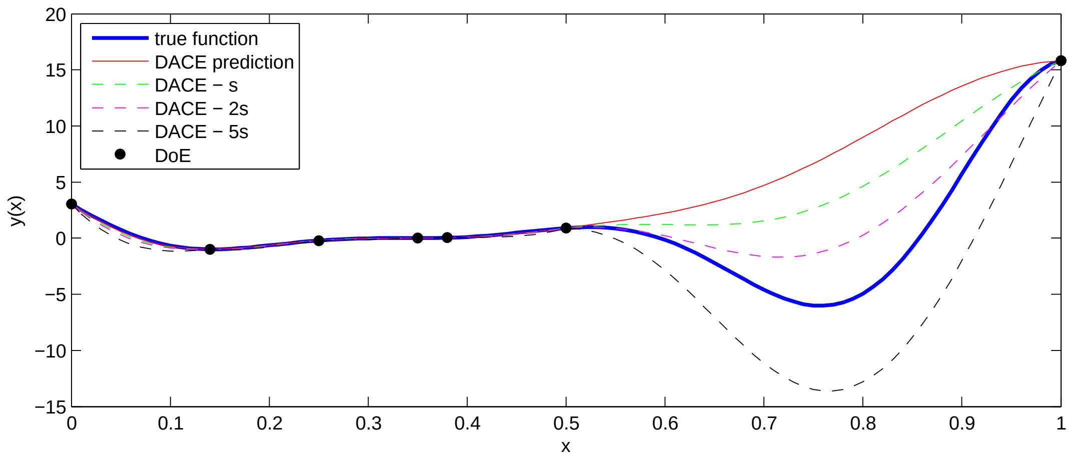


Figure 5.16: Minimizing a statistical lower bound. Several statistical lower bounds are shown.

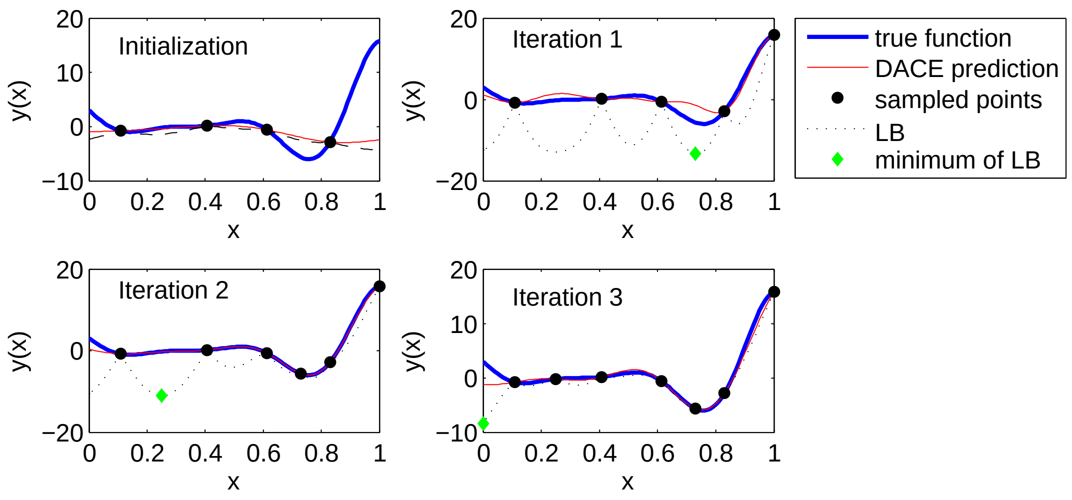


Figure 5.17: Minimizing a statistical lower bound. The parameter A_{LB} is set to 2.

the global optimum, the minimum value $y_{\min}$ of the true function evaluated on initial DoE is determined, and the probability of improvement is maximized across the whole domain of x. The improvement of the meta-model at a point $x^{\star}$ is therefore $I = \max(f_{\min} - Y, 0)$ and the probability of improvement is [74]

$$P[I(x)] = \frac{1}{s\sqrt{2\pi}} \int_{-\infty}^{0} \frac{(I - \hat{y}(x))^2}{2s^2} \mathrm{d}I. \tag{5.24}$$

This idea was first introduced by KUSHNER for 1D problems [107] and extended to higher dimensions, e.g. by [210]. Figure 5.18 shows a graphical interpretation of the probability of improvement in an illustrative example.

Later on, JONES in [96] extended the idea that there is a target improvement T to achieve instead of a basic improvement I. The improvement is achieved when the target T is greater than the uncertain output Y. The target T is slightly lesser than the $y_{\min}$ and it can be evaluated as $T = y_{\min} - A_{PI}|y_{\min}|$ where A_{PI} is a constant defining the distance between the best so far optimum and the target T. For example, $T = y_{\min} - 0.25|y_{\min}|$ means that the improvement of at least 25% is searched. This method is quite sensitive due to a setting of the T. Higher T is good for exploring (a global search); however, the convergence is slow. Lower T helps exploit (a local search), but there is a premature local convergence threat. JONES in [96] proposes to set up the target T to several levels and find several support points in each iteration. Those points can be clustered to reduce computational demands. This procedure is possible to be run in parallel.

The probability of improvement $P[I(\mathbf{x})]$ is

$$\P[I(x)] = \frac{1}{s\sqrt{2\pi}} \int_{-\infty}^{T} \exp \frac{(y - \hat{y}(x))^2}{2s^2} \mathrm{d}y \Bigg|_{\mathrm{NS}\to\mathrm{SNS}} = \tag{5.25}$$

$$= \int_{-\infty}^{\frac{T-\hat{y}(x)}{s}} \exp\left[-\frac{y'^2}{2}\right] dy' = \Phi\left(\frac{T - \hat{y}(x)}{s}\right). \tag{5.26}$$

The target T remains the same for the whole range of $\mathbf{x}$. The standard error s as well as a Kriging prediction $\hat{y}(x)$ differs for each value of $\mathbf{x}$. New support points are sequentially added into the maximum value of $P[I(\mathbf{x})]$ until the $P[I(\mathbf{x})]$ approximates zero. $P[I(\mathbf{x})]$ shows the location of the maximum improvement but not its amount. Therefore, the better criterion is the expected improvement function. Figure 5.19 shows an illustrative example of the maximizing the probability of improvement on the function $y(x) = (6x-2)^2 \sin(12x-4)$ with 7 iterations.

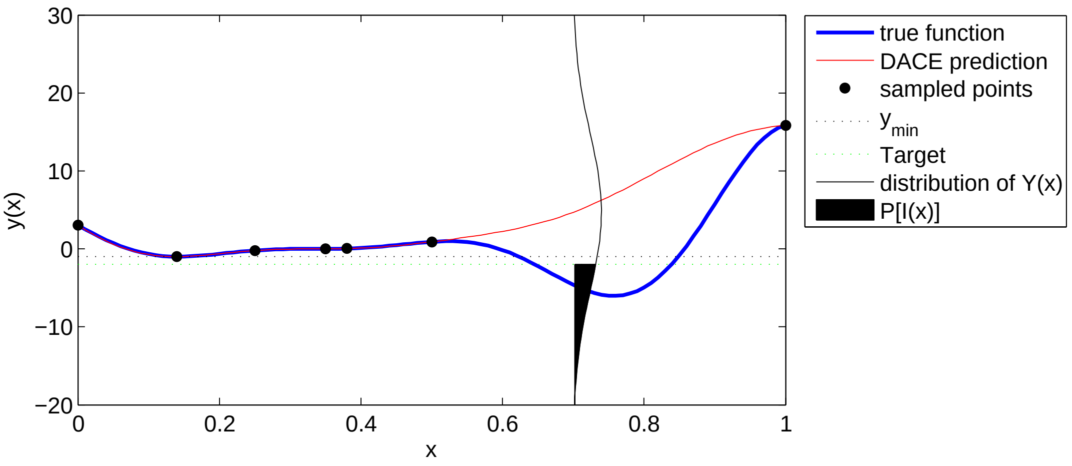


Figure 5.18: A graphical interpretation of the probability of improvement.

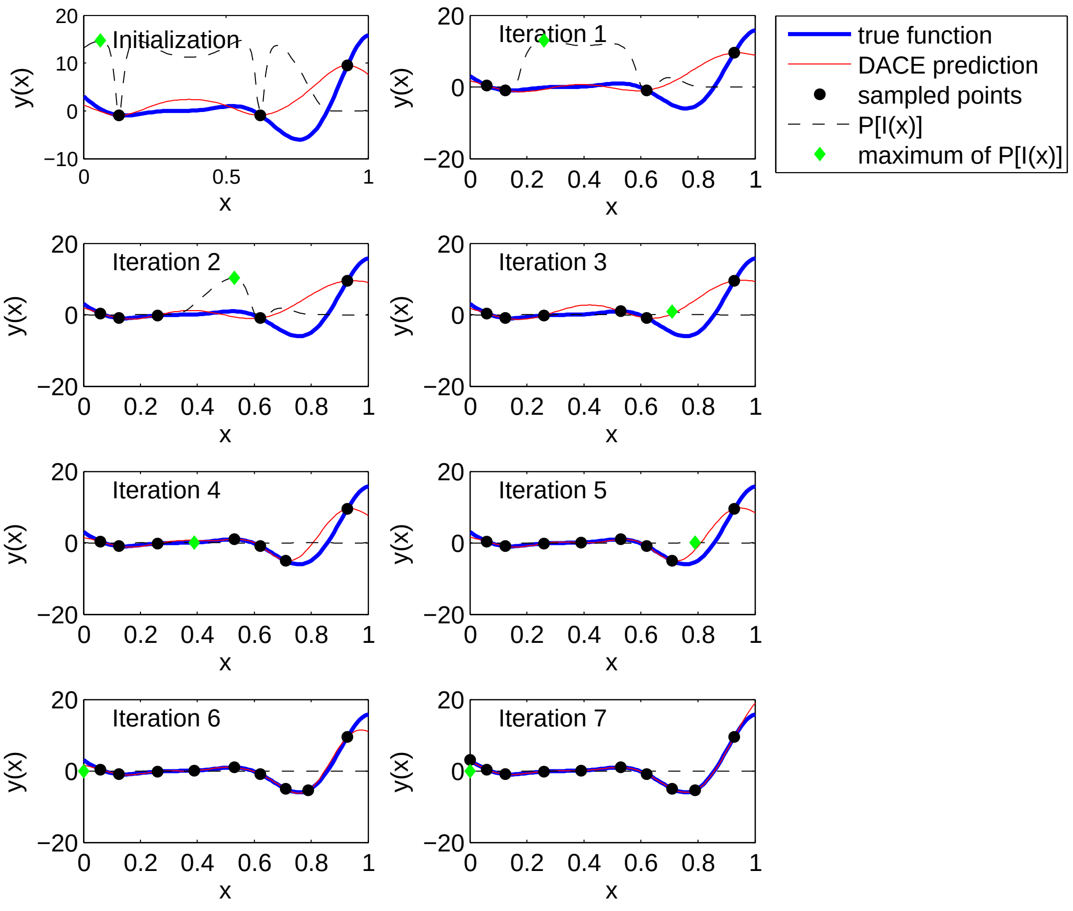


Figure 5.19: Maximizing the probability of improvement.

Maximizing the expected improvement function

Maximizing the expected improvement function (EIF) [97] is based on the idea of the probability of improvement. EIF is an expected value of $P[I(x)]$ defined as

$$\mathrm{E}[I(x)] = \mathrm{E}[\max(f_{\min} - Y, 0)] = (f_{\min} - \hat{y})\Phi\left(\frac{f_{\min} - \hat{y}}{s}\right) + s\phi\left(\frac{f_{\min} - \hat{y}}{s}\right). \tag{5.27}$$

The maximum value of EIF localizes the next point that should be added into the surrogate model to make it more accurate. Proposed EIF makes a balance between a local and a global search. The local search is focused on an improvement of the minimum local vicinity, and the global search concentrates primarily on unknown areas exploration where the standard error of the predictor has the maximum value. This approach is sequential, and maximization of EIF is done repeatedly until the maximum of EIF is greater than some prescribed value. JONES in [97] uses the Branch and Bound method for the optimization of EIF in *Efficient Global Optimization* (EGO); however, this approach is quite often too expensive to run to final convergence [17]. Figure 5.20 shows an illustrative example of the maximizing the expected improvement function on the function $y(x) = (6x - 2)^2 \sin(12x - 4)$ with 7 iterations.

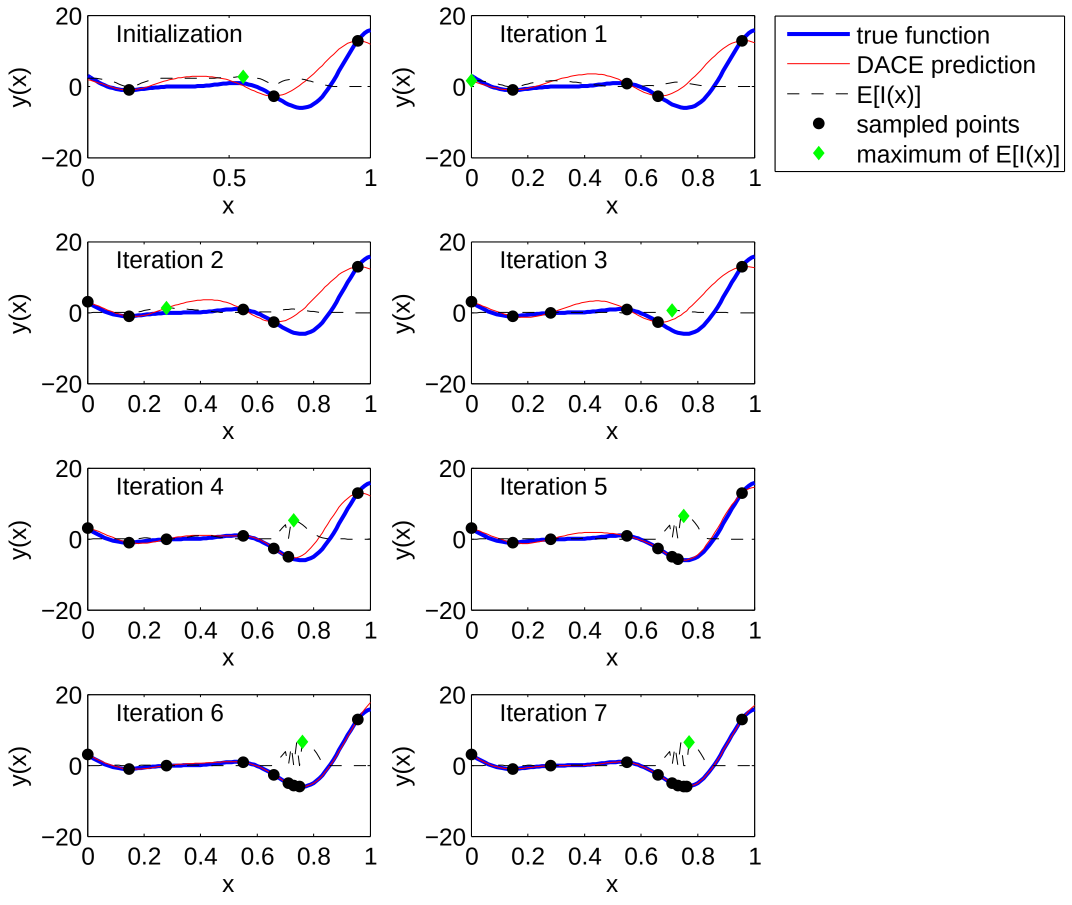


Figure 5.20: Maximizing the expected improvement function.

5.4.2 Updates for meta-models replacing constraints functions in RBDO

Efficient Global Reliability Analysis

BICHON et al. in [17] use an adaptive update of Kriging together with an adaptive importance sampling (AIS). An update of a meta-model is inspired by JONES et al. in [97]. Instead of EIF, *expected feasibility function* (EFF) is utilized. Since an improvement of the meta-model is not used for global minimization but for the reliability assessment, the bound dividing the domain into the safe and the failure region has to be more accurate. An equality constraint for a reliability assessment is defined as $G(\mathbf{u}) = \bar{z}$ where $\bar{z}$ is a threshold value. EFF is therefore integrated over a region $\bar{z} \pm \varepsilon$

$$\mathrm{E}F[\hat{G}(\mathbf{u})] = \int_{\bar{z}-\epsilon}^{\bar{z}+\epsilon} [\epsilon - |\bar{z} - G|] f_{\bar{G}} \,\mathrm{dG} \tag{5.28}$$

$$\begin{aligned} \mathrm{E}F[\hat{G}(\mathbf{u})] \quad = \quad & (\mu_G - \bar{z}) \left[2\Phi\left(\frac{\bar{z}-\mu_G}{\sigma_G}\right) - \Phi\left(\frac{z^- - \mu_G}{\sigma_G}\right) - \Phi\left(\frac{z^+ - \mu_G}{\sigma_G}\right) \right] \\ & - \sigma_G \left[2\phi\left(\frac{\bar{z}-\mu_G}{\sigma_G}\right) - \phi\left(\frac{z^- - \mu_G}{\sigma_G}\right) - \phi\left(\frac{z^+ - \mu_G}{\sigma_G}\right) \right] \\ & + \epsilon \left[\Phi\left(\frac{z^+ - \mu_G}{\sigma_G}\right) - \Phi\left(\frac{z^- - \mu_G}{\sigma_G}\right) \right], \end{aligned} \tag{5.29}$$

where ϵ is proportional to the standard deviation of the Gaussian process model predictor and z^- and z^+ are shorter notations of $\bar{z} \mp \epsilon$. All z^-, z^+, μ_G, σ_G and ϵ^2 are functions of the location $\mathbf{u}$, whereas $\bar{z}$ is a constant. *Efficient Global Reliability Analysis* (EGRA) first generates a small number of support points (at least to define the quadratic polynomial) and computes true function values in those support points for building the Gaussian process model. Those samples are recommended to cover the design space uniformly over the bounds $\pm 5\sigma$. The point that maximizes the EFF is located by DIRECT algorithm and the true function is calculated. EFF is maximized and support points are added into the meta-model repeatedly until the stopping criterion in the form of the prescribed maximum EFF value is not fulfilled.

Active Kriging + Monte Carlo Simulation

Active Kriging + Monte Carlo Simulation [62] creates and updates a meta-model only on a generated Monte Carlo (MC) population and not on any other points. At the beginning, few support points are chosen from the whole MC population by e.g. *k-means clustering*. Subsequently, support points for an update are obtained by minimizing the so-called *Learning function* on whole MC population,

$$U(\mathbf{x}) = \frac{|\hat{G}(\mathbf{x})|}{s^2(\mathbf{x})},$$

where $|\hat{G}(\mathbf{x})|$ is a predicted value by a meta-model and $s^2(\mathbf{x})$ is Kriging variance. Since the MC population is generated only for one combination of design variables, this method in its unmodified version works only for reliability assessment. Nevertheless, several options are available to extend it for RBDO. First, a new meta-model can be trained for each combination of design variables proposed by RBD optimizer, however, this approach can be quite time-consuming. Second, a meta-model is kept for all design variables combinations, and just new points from new MC populations are added to make the meta-model more precise across the whole design space. Third, the MC population can be widened just for meta-model improvement purposes.

[2] ϵ is recommended to be equal to $2\sigma_G$.

Chapter 6

Proposed Reliability-based design optimization procedure

Parts of this chapter are reproduced from the author's contributions [87, 89, 135, 151].

The double-loop Reliability-based design optimization (RBDO) approach utilizing reliability assessment methods based on a quasi-Monte Carlo simulation provides the most precise solutions among other formulations. Even though several advanced simulation techniques can assess the reliability task, the computational demands are still high while evaluating the failure probability of the original model. Therefore, we have constructed a meta-model to accelerate the reliability assessment. The first meta-model assembled on the DoE that is only space-filling is not precise enough; therefore, we have proposed an adaptive multi-objective optimization updating procedure. In our previous work [135], a meta-model was updated independently of an optimization routine. This procedure, described in the following section, turned up to be accurate but computationally very demanding since the meta-model was updated in the whole design space and not only in the desired target space. Therefore, we included the updating of the DoE (and subsequently the meta-model) inside the optimization routine utilizing data from the simulation techniques evaluating the reliability. Since we work with several formulations of the meta-models, more specifically, the local and the global meta-models, we propose two different routines for the updating procedure in the following text.

6.1 Adaptive multi-objective-optimization updating procedure

6.1.1 DoE updating independent of RBDO

This procedure uses a multi-objective double-loop optimization for locating additional support points regardless of the meta-model type [135]. The achieved quality of the meta-model so far controls the outer loop, and the inner loop contains the optimization method for discovering the proper points to be added to DoE, which is subsequently used to assemble new meta-model. Those points should be far from already added points; the miniMax metric controls this space-filling criterion as the first objective. All adaptively added points should lie on the limit state of the performance function to improve the quality in this region. The meta-model evaluates this

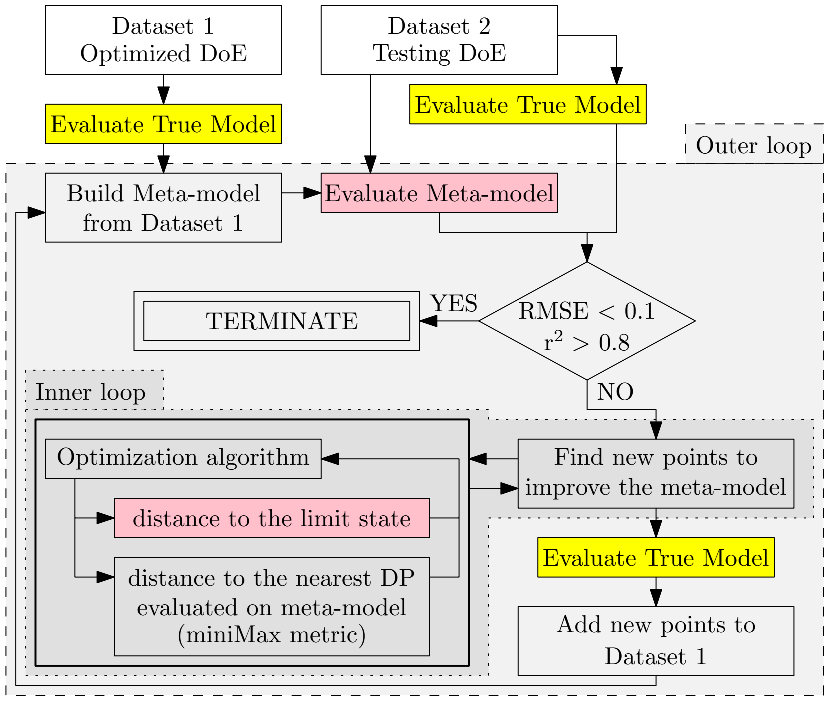

Dataset 1
Optimized DoE
Dataset 2
Testing DoE
Evaluate True Model
Evaluate True Model
Outer loop
Build Meta-model from Dataset 1
Evaluate Meta-model
TERMINATE
YES
RMSE < 0.1
r² > 0.8
NO
Inner loop
Optimization algorithm
distance to the limit state
distance to the nearest DP evaluated on meta-model (miniMax metric)
Find new points to improve the meta-model
Evaluate True Model
Add new points to Dataset 1

Figure 6.1: A flow chart of the adaptive meta-model updating.

second objective. After several inner loops of the multi-objective optimization algorithm[1], the methodology examines and clusters the best points and the true model[2] evaluates a response in these points to extend the dataset, that is used for assembling of an improved meta-model. After several outer loops, when the stopping criterion is fulfilled, the final meta-model is established.

[1]We use the Non-dominated sorting genetic algorithm II that is briefly outlined in Section 3.1.

[2]We are aware that the "true" model still does not describe the reality of the real-life problem exactly in all cases, however, we assume that this description is sufficient for our needs.

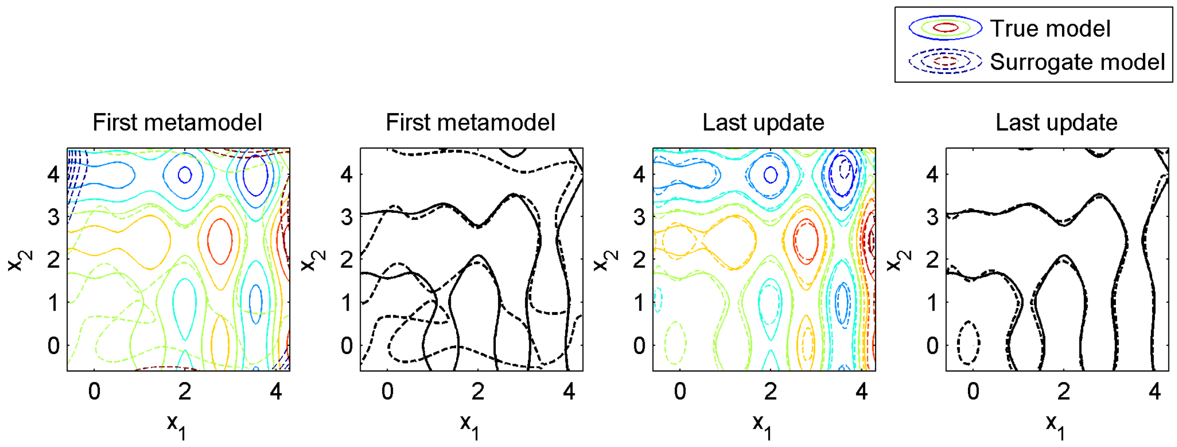


Figure 6.2: Comparison of the contour plots of the true model (the first two plots on the left) and the first meta-model without an update and the last updated meta-model (the last two plots on the right). Even subplots show all contours of a model whereas odd subplots show only the zero contour of the function (the limit state). The solid line is used for the real model while dashed line serves for meta-models drawings.

In the simplest case, the procedure is stopped after the finite number of outer loops. In the best case, a different dataset exists only for testing of quality and computing a Root-mean-square error (RMSE) and a *Coefficient of determination* (r^2). Figure 6.1 depicts the flow chart of this adaptive meta-model updating procedure. Figure 6.2 shows the initial meta-model trained for one limit state function in Example 2 described in Chapter 7. We used 40 iterations/updating of the original DoE. The updating procedure added approximately 220 points.

6.1.2 DoE updating dependent on RBDO

Global meta-models employed

Global meta-models are advantageous to their coverage of the whole design space, contrarily, disadvantageous for the memory demands and a large number of necessary added points into the DoE by the meta-model update. Figure 6.3 depicts the whole optimization algorithm proposing the candidate design solutions in the outer loop (the grey box). The inner loop (the light blue box in the same figure) is for the reliability assessment. For the illustration purposes, we used Monte Carlo simulation, but any other reliability assessment method is adequate. The pink boxes represent the steps for the meta-model assembly and the subsequent update. The yellow boxes depict the evaluation of the original response model.

First of all, we construct the primary Design of Experiments $\mathcal{Q}$. In our previous work [154], we have found out that Halton sequences [105] are efficient, fast, and well distributed. Thus, we have used them for every uniform design in this thesis. The true response function $\mathrm{g}(\cdot)$, that is computationally very demanding in real-life optimization problems (e.g. FEM simulation), evaluates responses $\mathcal{G}$ for DoE $\mathcal{Q}$. Therefore, the DoE should be relatively sparse to save up the computational time. This dataset is saved as Dataset 1 $[\mathcal{Q}, \mathcal{G}]$ for the following meta-modelling. The optimizer proposes one individual $\mathbf{d}^{(k)}$ or a set of individuals (in case of an evolutionary algorithm) for each generation k. Every individual has the meaning of the mean of a stochastic design variable, or it is a deterministic design variable. For each individual, deterministic functions, as well as the reliability assessment, are necessary to evaluate. Deterministic functions are not computationally demanding in most cases, and if so, they are calculated only once for each individual. However, the reliability assessment slows down the computational speed of the whole process. Since we use a sampling approach for the failure probability estimation, we evaluate the response function repeatedly. Therefore, a surrogate model substitutes the response function, and the meta-model is uniquely assembled by $\mathscr{S}(\cdot)$ for each generation as $\overline{g}(\cdot)$. Since we use $(\mu + \lambda)$ genetic algorithm[3], the Monte Carlo simulation using newly assembled meta-model has to evaluate the whole population including the elitist individuals that survived from the previous generations. This update of the reliability results ensures that the reliability results are comparable for all individuals in the mating pool.

This algorithm slightly differs for a sparse global meta-model (SGMM) and a dense global meta-model (GMM). If we employ an ordinary global meta-model, all points from the dataset $\mathcal{Q}$ are used to construct a surrogate for the original limit state function as described in Section 5.1.1 and 5.1.4. The sparse global meta-models support points have its influence domain in the closest neighbourhood, and other points outside the influence domain are omitted as described in Section 5.1.2 and 5.1.4. Therefore, a range search algorithm $\mathcal{R}(\cdot)$ selects only the most interesting points for each support point from the dataset $\mathcal{Q}$. The information about the

[3] $(\mu + \lambda)$ means that individuals for the next generation are selected from the mating pool created by parents (μ) as well as offspring (λ) from the actual generation. We chose the Non-dominated sorting genetic algorithm II as the optimization algorithm, which is described in Section 3.1.

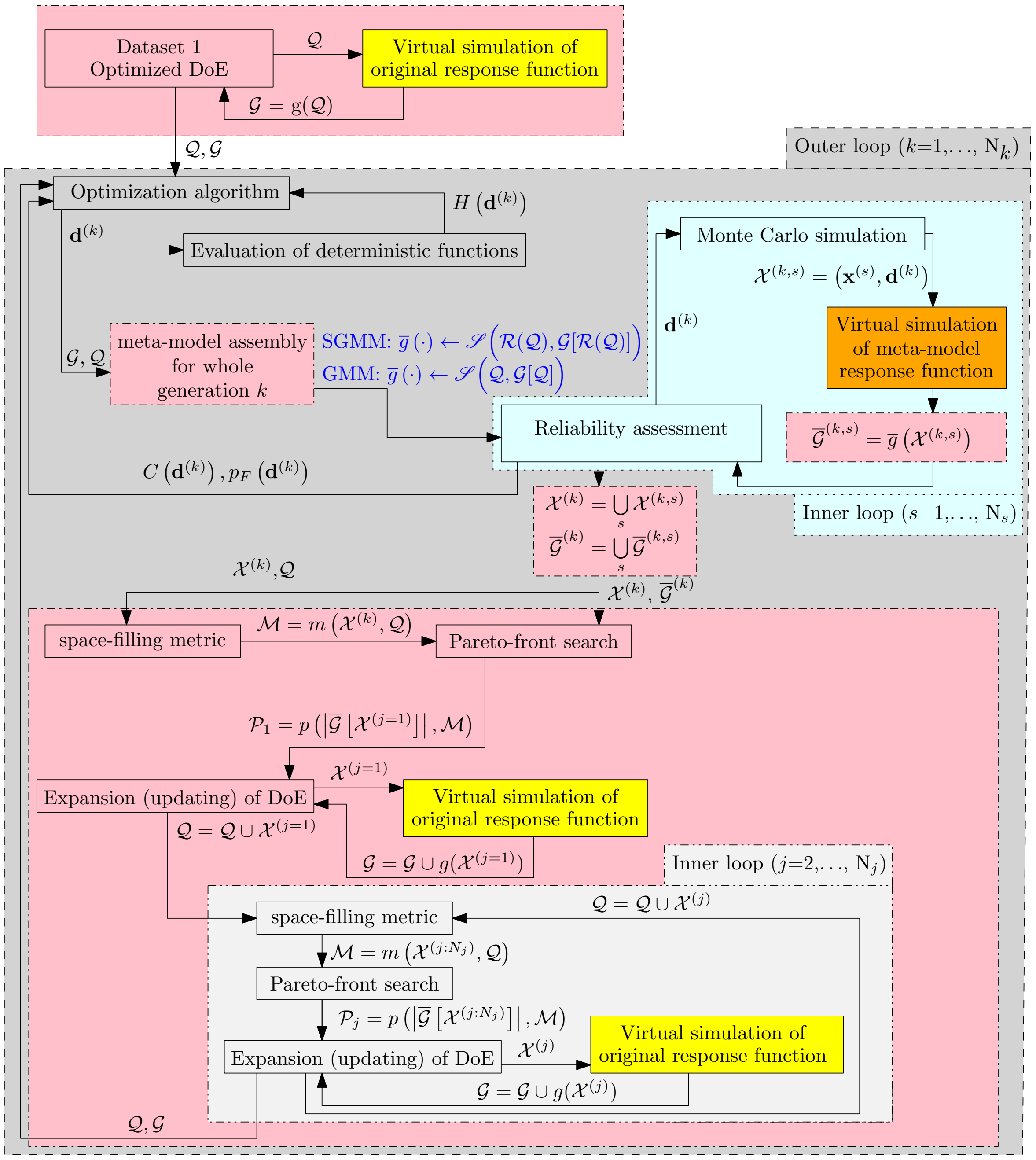

Dataset 1
Optimized DoE
$\mathcal{Q}$
Virtual simulation of original response function
$\mathcal{G} = \mathrm{g}(\mathcal{Q})$
$\mathcal{Q}, \mathcal{G}$
Outer loop (k=1,…, N$_k$)
Optimization algorithm
$H\left(\mathbf{d}^{(k)}\right)$
$\mathbf{d}^{(k)}$
Evaluation of deterministic functions
Monte Carlo simulation
$\mathcal{X}^{(k,s)} = \left(\mathbf{x}^{(s)}, \mathbf{d}^{(k)}\right)$
$\mathbf{d}^{(k)}$
Virtual simulation of meta-model response function
$\mathcal{G}, \mathcal{Q}$
meta-model assembly for whole generation k
SGMM: $\overline{g}(\cdot) \leftarrow \mathscr{S}\left(\mathcal{R}(\mathcal{Q}), \mathcal{G}[\mathcal{R}(\mathcal{Q})]\right)$
GMM: $\overline{g}(\cdot) \leftarrow \mathscr{S}\left(\mathcal{Q}, \mathcal{G}[\mathcal{Q}]\right)$
Reliability assessment
$\overline{\mathcal{G}}^{(k,s)} = \overline{g}\left(\mathcal{X}^{(k,s)}\right)$
$C\left(\mathbf{d}^{(k)}\right), p_F\left(\mathbf{d}^{(k)}\right)$
$\mathcal{X}^{(k)} = \bigcup_s \mathcal{X}^{(k,s)}$
$\overline{\mathcal{G}}^{(k)} = \bigcup_s \overline{\mathcal{G}}^{(k,s)}$
Inner loop (s=1,…, N$_s$)
$\mathcal{X}^{(k)}, \mathcal{Q}$
$\mathcal{X}^{(k)}, \overline{\mathcal{G}}^{(k)}$
space-filling metric
$\mathcal{M} = m\left(\mathcal{X}^{(k)}, \mathcal{Q}\right)$
Pareto-front search
$\mathcal{P}_1 = p\left(\left|\overline{\mathcal{G}}\left[\mathcal{X}^{(j=1)}\right]\right|, \mathcal{M}\right)$
Expansion (updating) of DoE
$\mathcal{X}^{(j=1)}$
Virtual simulation of original response function
$\mathcal{Q} = \mathcal{Q} \cup \mathcal{X}^{(j=1)}$
$\mathcal{G} = \mathcal{G} \cup g(\mathcal{X}^{(j=1)})$
Inner loop (j=2,…, N$_j$)
$\mathcal{Q} = \mathcal{Q} \cup \mathcal{X}^{(j)}$
space-filling metric
$\mathcal{M} = m\left(\mathcal{X}^{(j:N_j)}, \mathcal{Q}\right)$
Pareto-front search
$\mathcal{P}_j = p\left(\left|\overline{\mathcal{G}}\left[\mathcal{X}^{(j:N_j)}\right]\right|, \mathcal{M}\right)$
Expansion (updating) of DoE
$\mathcal{X}^{(j)}$
Virtual simulation of original response function
$\mathcal{G} = \mathcal{G} \cup g(\mathcal{X}^{(j)})$
$\mathcal{Q}, \mathcal{G}$

Figure 6.3: Schematic representation of a double-loop RBDO problem extended to the global meta-models update.

individual from the evolutionary algorithm does not influence the meta-model. The estimation of the probability of failure $p_F(\mathbf{d}^{(k)})$ as well as a value of the cost function $C(\mathbf{d}^{(k)})$ are given back to the optimization algorithm.

The reliability simulation method samples around the design variable and the reliability assessment needs the response in these samples. Generally, the accuracy of the meta-model is increased by the addition of new samples into the DoE $\mathcal{Q}$ including their true responses of the original model. This response is more reliable than the meta-model response. Thus, new samples have to be added to DoE into the vicinity of the limit state; this is the first criterion of our updating procedure. We already know the meta-model responses of the Monte Carlo samples, and therefore, we choose those new updating points from the Monte Carlo sample set. Every single sample from the reliability sampling method and its response value are therefore stored in $\mathcal{X}^{(k)}$ and $\overline{\mathcal{G}}^{(k)}$, respectively. These new samples should also be situated into the area with the deficient knowledge to uncover more possible limit states. For this second criterion – the space-filling metric, the closest distances m are evaluated between the samples from the Monte Carlo simulation and all points in DoE $\mathcal{Q}$ and subsequently stored in $\mathcal{M}$. These two criteria are antagonistic; therefore, we select the Pareto front with the best indifferent solutions. The updating procedure adds points from the Pareto set (the corresponding coordinates in the design space for the Pareto front) to the DoE one by one. The first point of the Pareto set $\mathcal{P}_1$ is placed into the final DoE $\mathcal{Q}$ and the true model evaluates its response $g(\cdot)$. Since the DoE changed, the space-filling metric has to be recalculated. As a result, the second criterion for the update also changed, and therefore, the updating procedure has to find the new Pareto front $\mathcal{P}_j$. If the j^{th} point of the $\mathcal{P}_1$ is still in the Pareto front $\mathcal{P}_j$, it is embedded into the DoE $\mathcal{Q}$ and the original model evaluates the true response for this sample. Therefore, the procedure extends the DoE in every k^{th} step of the optimization algorithm or every generation of the genetic algorithm, respectively.

Local meta-models employed

This methodology proceeds from the updating of the global meta-model inside the optimization described in the previous section. The green dash-dot-dotted boxes and the blue captions and arrows in Figure 6.4 highlight the differences between the updating of the global meta-model and updating of the Design of Experiments for the local meta-models with subsequent model assembly. The beginning of the procedure is identical, first of all, we construct the primary Design of Experiments $\mathcal{Q}$ and the true response function g$(\cdot)$ evaluates this DoE. This dataset is saved as Dataset 1 $[\mathcal{Q}, \mathcal{G}]$. The optimization algorithm in the outer loop (depicted by a grey background) nominates a candidate solution $d^{(k)}$ for an optimum. The deterministic and stochastic functions have to be evaluated for this candidate $d^{(k)}$. Since deterministic functions are usually facile to evaluate and only one evaluation is necessary for one candidate, the original function does not have any surrogate. A sampling method evaluates the stochastic function. Figure 6.4 mentions a Monte Carlo simulation as a sampling method, but any other simulation method, such as an Asymptotic sampling or Subset simulation, is viable to use. The response function $g(\cdot)$ would be necessary to be evaluated N_s times here. Any approximation method, such as a First Order Reliability Method, can reduce the evaluation requirements for the stochastic function; however, the approximation can carry the computational errors into reliability. The surrogate model $\bar{g}(\cdot)$, that is uniquely assembled by $\mathscr{S}(\cdot)$ for each individual $\mathbf{d}^{(k)}$, substitutes the response function $g(\cdot)$ due to its repetitive evaluation. Thus every individual has its meta-model and individuals do not share them. The meta-model is constructed from DoE dataset points $\mathcal{Q}$ that are nearby the individual $\mathbf{d}^{(k)}$; the selection of the dataset is labelled as $\mathbf{Q}^{\mathbf{d}^{(k)}}$. We use the *k-nearest neighbours algorithm*[4] [77] for the proximity assessment with the prescribed number of points in the influence domain. The *k*-NN algorithm is labelled as $\mathcal{K}(\mathbf{a}, \mathbf{B})$, where $\mathbf{a}$ is a point or a set of points for which the nearest neighbour has to be found from a set of query points $\mathbf{B}$. We found out that one-tenth of the all DoE points is sufficient but this setting varies from problem to problem. However, the tenth is a good initial guess. The estimation of the probability of failure $p_F(\mathbf{d}^{(k)})$, as well as a value of the cost function $C(\mathbf{d}^{(k)})$, are given back to the optimization algorithm.

Some samples provided by the sampling methodology for reliability assessment updates the DoE held in $\mathcal{Q}$. In every step k, the reliability simulation method samples around the design

[4]We use *k*-NN algorithm implemented in MATLAB.

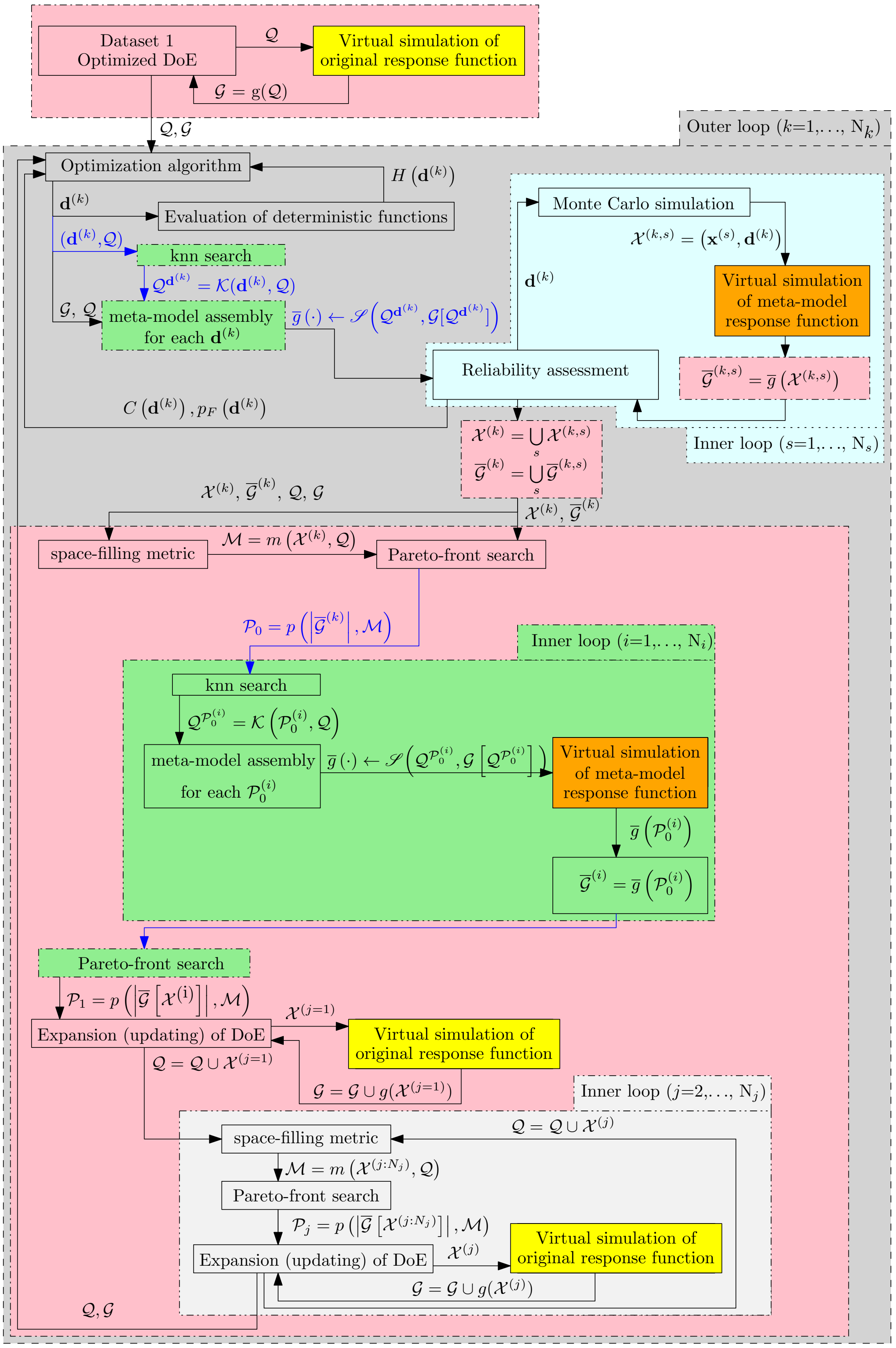

Dataset 1
Optimized DoE
$\mathcal{Q}$
Virtual simulation of original response function
$\mathcal{G} = g(\mathcal{Q})$
$\mathcal{Q}, \mathcal{G}$
Outer loop (k=1,..., N$_k$)
Optimization algorithm
$H\left(\mathbf{d}^{(k)}\right)$
$\mathbf{d}^{(k)}$
Evaluation of deterministic functions
$(\mathbf{d}^{(k)}, \mathcal{Q})$
knn search
$\mathcal{Q}^{\mathbf{d}^{(k)}} = \mathcal{K}(\mathbf{d}^{(k)}, \mathcal{Q})$
$\mathcal{G}, \mathcal{Q}$
meta-model assembly for each $\mathbf{d}^{(k)}$
$\overline{g}(\cdot) \leftarrow \mathscr{S}\left(\mathcal{Q}^{\mathbf{d}^{(k)}}, \mathcal{G}[\mathcal{Q}^{\mathbf{d}^{(k)}}]\right)$
Monte Carlo simulation
$\mathcal{X}^{(k,s)} = \left(\mathbf{x}^{(s)}, \mathbf{d}^{(k)}\right)$
$\mathbf{d}^{(k)}$
Virtual simulation of meta-model response function
$\overline{\mathcal{G}}^{(k,s)} = \overline{g}\left(\mathcal{X}^{(k,s)}\right)$
Reliability assessment
$C\left(\mathbf{d}^{(k)}\right), p_F\left(\mathbf{d}^{(k)}\right)$
$\mathcal{X}^{(k)} = \bigcup_s \mathcal{X}^{(k,s)}$
$\overline{\mathcal{G}}^{(k)} = \bigcup_s \overline{\mathcal{G}}^{(k,s)}$
Inner loop (s=1,..., N$_s$)
$\mathcal{X}^{(k)}, \overline{\mathcal{G}}^{(k)}, \mathcal{Q}, \mathcal{G}$
$\mathcal{X}^{(k)}, \overline{\mathcal{G}}^{(k)}$
space-filling metric
$\mathcal{M} = m\left(\mathcal{X}^{(k)}, \mathcal{Q}\right)$
Pareto-front search
$\mathcal{P}_0 = p\left(\left|\overline{\mathcal{G}}^{(k)}\right|, \mathcal{M}\right)$
Inner loop (i=1,..., N$_i$)
knn search
$\mathcal{Q}^{\mathcal{P}_0^{(i)}} = \mathcal{K}\left(\mathcal{P}_0^{(i)}, \mathcal{Q}\right)$
meta-model assembly for each $\mathcal{P}_0^{(i)}$
$\overline{g}(\cdot) \leftarrow \mathscr{S}\left(\mathcal{Q}^{\mathcal{P}_0^{(i)}}, \mathcal{G}\left[\mathcal{Q}^{\mathcal{P}_0^{(i)}}\right]\right)$
Virtual simulation of meta-model response function
$\overline{g}\left(\mathcal{P}_0^{(i)}\right)$
$\overline{\mathcal{G}}^{(i)} = \overline{g}\left(\mathcal{P}_0^{(i)}\right)$
Pareto-front search
$\mathcal{P}_1 = p\left(\left|\overline{\mathcal{G}}\left[\mathcal{X}^{(i)}\right]\right|, \mathcal{M}\right)$
$\mathcal{X}^{(j=1)}$
Expansion (updating) of DoE
Virtual simulation of original response function
$\mathcal{Q} = \mathcal{Q} \cup \mathcal{X}^{(j=1)}$
$\mathcal{G} = \mathcal{G} \cup g(\mathcal{X}^{(j=1)})$
Inner loop (j=2,..., N$_j$)
$\mathcal{Q} = \mathcal{Q} \cup \mathcal{X}^{(j)}$
space-filling metric
$\mathcal{M} = m\left(\mathcal{X}^{(j:N_j)}, \mathcal{Q}\right)$
Pareto-front search
$\mathcal{P}_j = p\left(\left|\overline{\mathcal{G}}\left[\mathcal{X}^{(j:N_j)}\right]\right|, \mathcal{M}\right)$
Expansion (updating) of DoE
$\mathcal{X}^{(j)}$
Virtual simulation of original response function
$\mathcal{G} = \mathcal{G} \cup g(\mathcal{X}^{(j)})$
$\mathcal{Q}, \mathcal{G}$

Figure 6.4: Schematic representation of a double-loop RBDO problem extended to the local meta-models update.

variable $\mathbf{d}^{(k)}$ with N_s points $\mathcal{X}^{(k,s)}$; a union of these points creates a set $\mathcal{X}^{(k)}$ together with the meta-model responses set $\bar{\mathcal{G}}^{(k)}$. As in the updating procedure of the global meta-model, there are two criteria for selection of new points into DoE stored in the dataset $\mathcal{Q}$. The first criterion is the space-filling metric $m(\cdot,\cdot)$, which represents the minimal distances between $\mathcal{X}^{(k)}$ and $\mathcal{Q}$ stored in the set $\mathcal{M}$. The second criterion is the difference between the response of the meta-model value and the limit state $\left|\mathcal{G}^{(k)}\right|$. The first criterion focuses on the distances in the design space and the second criterion aims at the distances in the objective space. Pareto-optimality conditions select the Pareto front $\mathcal{P}_0$ using these two criteria for all points in $\mathcal{X}^{(k)}$. We assume that the meta-model is more accurate in the middle of the influence domain than at its edge. Therefore, the procedure assembles a new meta-model for each point in the Pareto set to make sure that the limit state prediction value is the most accurate value we can get from the meta-model at a given time. The influencing area of the local meta-model is hereby centred around each point in the Pareto set. These meta-models are subsequently used to get a better meta-model response for the Pareto set. Since we have N_i number of points in the Pareto set corresponding to the Pareto front $\mathcal{P}_0$, the updating procedure has to assemble N_i meta-models in the inner loop (green rectangle) to get more precise meta-model responses. Subsequently, the new Pareto front has to be found again as $\mathcal{P}_1$. If the responses of the former local meta-models have the same accuracy as the responses of the later meta-models, the Pareto set outgoing from the later Pareto-optimality rescreening will contain the same number of samples as the previous one. The last step identical to the updating of the global meta-model is the sequential addition of new points from the Pareto set into the dataset $\mathcal{Q}$. The updating procedure adds the first point of the Pareto set into the DoE, together with its response to the original model. Since the DoE changed, the space-filling metric has to be recalculated among points in new DoE and the remaining points in the Pareto set. The new Pareto front is recalculated, and the next point from the Pareto set is added into the DoE together with the response of the original model. This sequential extension of the dataset [$\mathcal{Q}$,$\mathcal{G}$] runs in another inner loop with N_j steps.

Figure 6.5 shows an illustrative example of the presented updating procedure utilizing local meta-models. Since the updating procedure is independent of the reliability assessment method, we selected Asymptotic sampling as a representative. Image a) presents the analytical model of the limit state function depicted with coloured contours, and blue dots have the meaning of the initial DoE. The orange contour illustrates the limit state; the positive numbers represent the safe space, and, conversely, negative LSF values indicate the failure region. The initial DoE is space-filling; we used quasi-random sequences to design it. For illustrative reasons, we used only four individuals in a population of the evolutionary algorithm. The green triangles represent those individuals in image b). For these individuals representing the mean values of stochastic design variables, the algorithm calculates their reliability using Asymptotic sampling. Asymptotic sampling samples are plotted in different colours for each individual. In each Asymptotic sampling, a single simulation step is colour-coded representing a single step of sequential Monte Carlo simulation. Asymptotic sampling calls a local meta-model of the limit state function to evaluate the indicator function. Each individual has its unique local meta-model assembled, using support points of DoE around the individual. One of the local meta-models is depicted in image c) by solid contours, the black box represents the bounds where the meta-model interpolates, and the red circles show the selected support points. Naturally, the local meta-model may also extrapolate beyond these limits; however, the output values will no longer be as accurate as inside these bounds if the original model is highly nonlinear. Two criteria evaluate each sample from Asymptotic sampling, the already calculated response value of the meta-model and the distance to the nearest point from the DoE. Image e) shows these criteria, where each colour of the dot is for a different Asymptotic sampling representa-

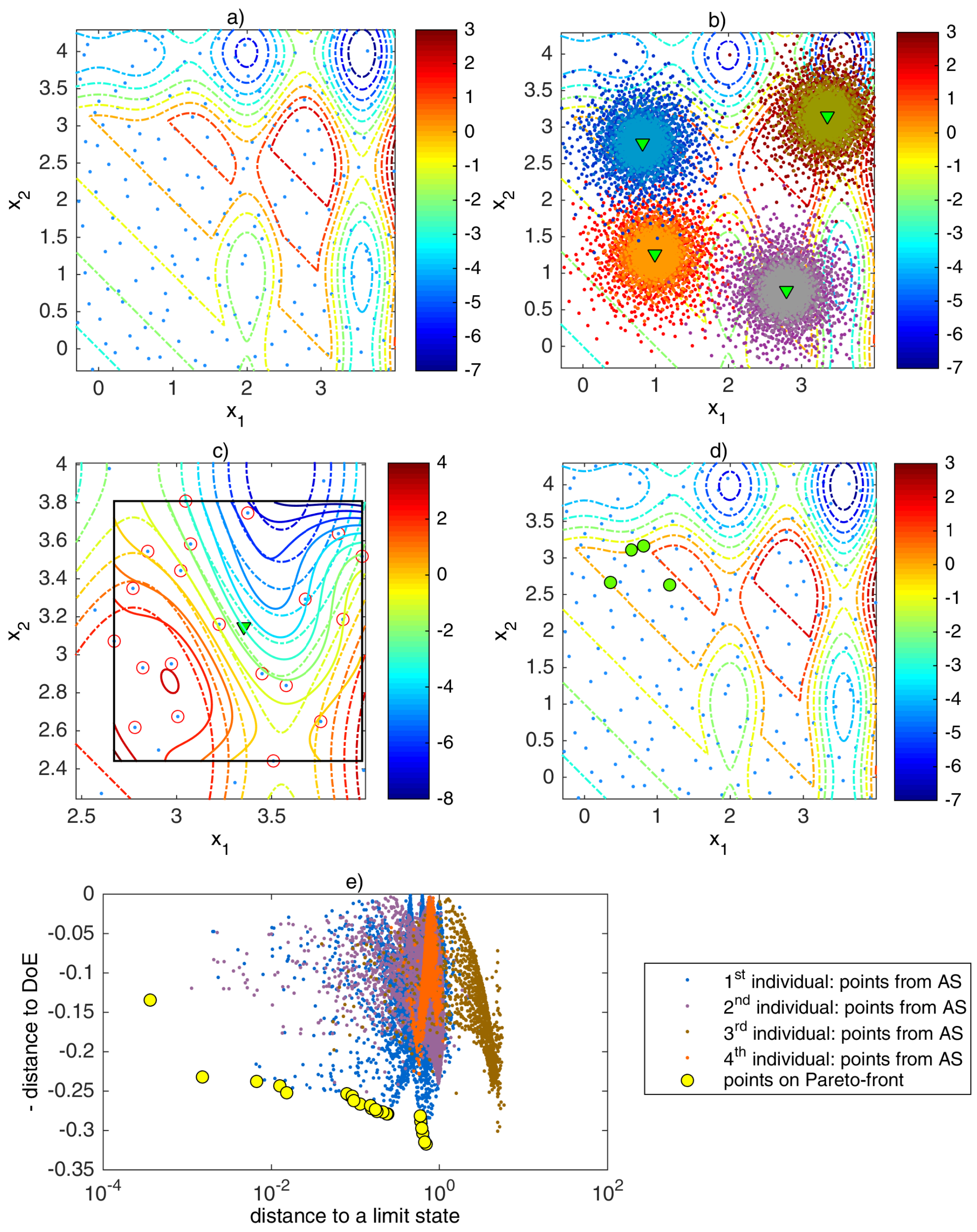


Figure 6.5: An illustrative example of the updating procedure using local meta-models. Image a) represents original LSF (dash-dotted contours) and initial DoE (blue dots). Image b) shows four individuals (green triangles) of an evolutionary algorithm with Asymptotic sampling samples (dots). Image c) is devoted to one local meta-model (solid contours) with support points (red circles) in the influence domain assembled for one individual of an evolutionary algorithm (green triangle), the black box represents the meta-model interpolation domain. Image d) shows new support points (green circles) in DoE (blue dots). Image e) depicts two criteria (a distance to a limit state and a distance to the nearest DoE support point) for decision making of DoE potential candidates.

tion. The yellow circles show the points on the Pareto front that are candidates for the DoE update. For each yellow point on Pareto front, the new meta-model is assembled, and the distance to the limit state is recalculated. Image d) demonstrates the resulting support points that update the DoE.

6.2 Parallelization of Reliability-based Design Optimization using Surrogates

For an outer optimization loop, we use Non-dominated sorting genetic algorithm II [49] as a basic multi-objective algorithm with a simulated binary crossover operator, a Gaussian mutation operator, and a tournament selection operator; probabilities of the operator utilization for creating offspring populations are 0.9, 0.5 and 1, respectively. The number of individuals was set to 50, and the number of total generations was set to 30. For the inner loop, we use the pure Monte Carlo method to provide unified results for the application of surrogates.

β	0	1	2	3	4	4.5
p_f	0.5	0.159	$2.28{\cdot}10^{-2}$	$1.35{\cdot}10^{-3}$	$3.17{\cdot}10^{-5}$	$3.40{\cdot}10^{-6}$
MC samples	200	629	$4.39{\cdot}10^{3}$	$7.41{\cdot}10^{4}$	$3.15{\cdot}10^{6}$	$2.94{\cdot}10^{7}$

Table 6.1: Number of samples needed for 10% coefficient of variation of Monte Carlo failure probability estimator p_f

6.2.1 Computational demands of RBDO

The most challenging part of the RBDO is the assessment of reliability. Table 6.2 presents basic settings of our double-looped approach together with computational demands in terms of time and the number of meta-model evaluations. Hardware and software details are listed in Table 11.1. Besides time, the last line shows the most demanding part of the RBDO benchmark problem, where almost ten billion evaluations of the meta-model are needed. This is in correspondence with Table 6.1, which presents a minimum number of samples for particular probabilities of failures and 10% coefficient of variation for Monte Carlo sampling procedure for our range of reliability indices.

Taking into account that data presented in Table 6.2 are just for the simple 2D benchmark, the application of parallelization is inevitable for real-world examples. There are two main possibilities, how to parallelize double-looped RBDO:

1. **Parallelization of the optimization procedure:** Since we are dealing with the multi-objective version of RBDO, evolutionary algorithms, using a set of possible solutions usually called a "population", are the best choice for optimization algorithms here. Therefore, the evaluation of the population can be done in parallel using a well-known master-slave paradigm, where each individual is one computing unit, and CPU workers are enumerating the reliability for each of them. The main advantage is the simplicity of the formulation; however, since each solution can differ in orders of magnitude in terms of probability of failure, the computation demands are highly unbalanced. Therefore, bad load-balancing can be expected, and particular attention must be paid to synchronization issues.

number of CPUs	8
number of individuals	50
number of generations	30
elapsed time [sec]	29,171
elapsed time [hours]	8.10
Number of samples in initial DoE	50
Number of added samples to DoE during optimization	280
Number of analytical limit state function evaluations	330
Number of meta-models built for optimization purposes	1,500
Number of meta-models' evaluations	9,179,723,956

Table 6.2: Typical computational demands of RBDO with meta-models.

2. **Parallelization of the reliability assessment:** Here, all requests for reliability assessment are collected for all individuals in the population, hence forming a huge dataset for objective/constraints evaluation for all Monte Carlo samples. This set is then evaluated in a parallel fashion using a classical "parameters sweep" parallel paradigm. Here, linear speed-up can be achieved because the load is uniformly spaced along with available computing units. However, the data are sent in batches, and the size of these packages must be carefully chosen not to bee too small or too big.

The second approach was implemented in the MATLAB environment. Profiling has shown that almost 20% of the time is spent on the solution of the system of equations 5.1; however, more than 70% of the time was spent on the evaluation of the interpoint distances. Therefore, we concentrated mainly on this issue in code optimization, and several implementation details are described in Section 5.1.4.

Chapter 7

Numerical examples for the multi-objective reliability-based design optimization procedure

This chapter presents a selected set of examples to validate the proposed methodology described in Chapter 6. Every example is introduced together with a short description of its history and already published results that authors found for single-objective optimization.

A Non-dominated sorting genetic algorithm II (NSGA-II) was used for multi-objective optimization with a setting 50 individuals in each generation, 30 offspring generations, 10% probability of mutation, 20% distance between the upper bound and the lower bound for the design variables as the standard deviation of the Gaussian mutation, 90% probability of crossover, and 100% probability of selection. Since each new generation of the genetic algorithm provides a new update to DoE, a reliability assessment method recalculates reliability indices of a parental population with a new meta-model for the selection purposes. The Pareto front is constrained by the bounds [0,4.5] since this interval is the most interesting for the structural optimization from our point of view. The same constraint would be equal to bounds $[3.398 \cdot 10^{-6}, 0.5]$ if recalculated to the failure probabilities. The failure probability bounds are reverse to the reliability index bounds since the higher the probability of failure, the lower the reliability index is. The dependence is given via the cumulative distribution function of the standard normal distribution $p_F = \Phi(-\beta)$ as depicted in Figure 7.1.

The global meta-models are assembled for the whole generation of the genetic algorithm; we used one initial generation and 30 subsequent generations; therefore, 31 meta-models were built for the whole designing process. Local meta-models were assembled for each individual from all 50 individuals across all 31 generations resulting in $50 + 30 \times (50 + 50) = 3050$ meta-models assembled during optimization designing process. If we compare the parental population with the offspring, the reliability indices have to be recalculated with new meta-models since the next generation DoE contains newly added update support points, and the meta-model is more accurate than for the previous generation. Therefore, each generation 2-31 needs 100 reliability indices evaluations, which means 100 local meta-models assembly or 100 evaluations of the reliability assessment using dense or sparse global meta-models in one generation. For local meta-models, we build additional meta-models for each potential new sample in DoE to ensure that this sample is truly needed and the response is not distorted, e.g. because of the extrapolation. We set the number of support point in initial DoE to 200

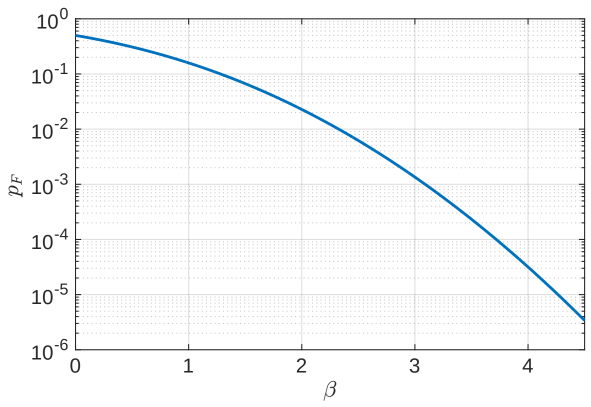


Figure 7.1: The dependence between the probability of failure p_F and the reliability index β.

samples for MO-RBDO using local meta-models and 50 samples for MO-RBDO using the global meta-model and the sparse global meta-model. Both global meta-models need smaller initial DoE than local meta-models since they are assembled for the whole design domain. Local meta-models need a larger portion of an initial DoE since they are covering a smaller area and therefore need more information around each potential design candidate. The number of support points added to DoE by the updating procedure is not restricted for any meta-model.

To assess the quality of the obtained solutions, we use (i) visual evaluation of the resulting recalculated Pareto sets and Pareto fronts, and (ii) numerical evaluation using a) metrics from Subsection 3.2, b) errors of the achieved solutions using meta-models, simulation methods or approximation techniques and c) a number of discarded solutions due to $\pm\infty$ in the reliability index after recalculation. The used performance measures are unary metrics (a Hypervolume, Spacing, and Spread) working with the recalculated Pareto fronts and binary metrics (a Generational distance, Two set coverage metric, and Coverage difference of two sets) comparing the recalculated Pareto fronts with a superior Pareto front. The reliability index error is equal to the absolute value of the difference between the solution from the original Pareto front and the recalculated front. If the recalculated value is equal to $\pm\infty$, we discarded this result, and the corresponding solution is excluded from the error evaluation. Since each solution from the original Pareto front has a potential error of the reliability index error, and we have several independent results available for each method, we evaluate four types of error indicators. Therefore, on each original Pareto front, we evaluate the corresponding error indicator

$$|\beta_{MC_{\mathrm{CoV}<0.05}} - \beta_{\mathrm{oPF}}|, \tag{7.1}$$

and we average these errors for all independent Pareto fronts. The value $\beta_{MC_{\mathrm{CoV}<0.05}}$ represents the corrected and therefore accurate and precise reliability index and β_{oPF} is the original value. We corrected the original Pareto fronts with an analytical model and a quasi-Monte Carlo simulation with recurring addition of sampling data to get the coefficient of variation lesser than 5% since we suppose that this combination provides the most precise results of the reliability index. First, we calculate the average minimum error to see if the method can achieve at least one good result, and in the case of worse results, it, therefore, fails only in some specific areas. We also evaluate the average maximum error to find the weakest point of the methodology. Finally, we evaluate the average mean error and the average standard deviation of the reliability index error for each methodology. These two error indicators provide information about the quality of the method in terms of accuracy and precision. Figure 7.2 shows the four options that

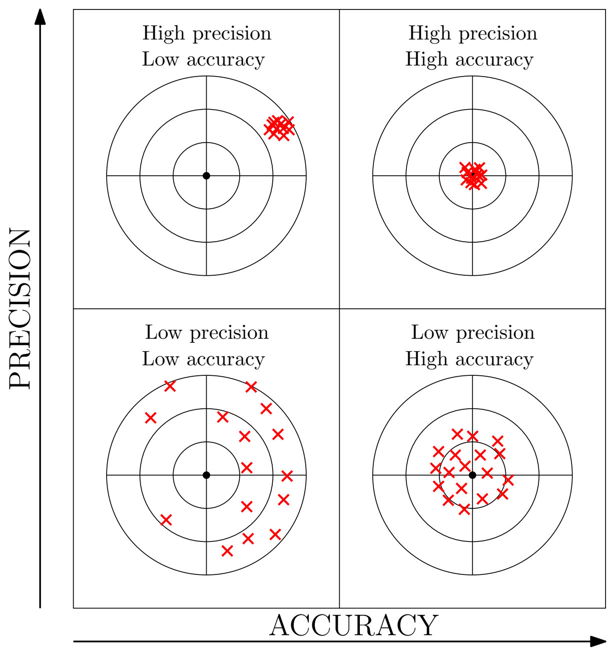


Figure 7.2: Accuracy and precision

can arise from the results. The mean value affects accuracy, and the standard deviation affects precision. If the mean value is high, and the standard deviation is low, then the result is precise but has low accuracy. If the mean is low, but the standard deviation is high, then the result is accurate but has low precision. If there are small or large mean values and standard deviations of the error, respectively, then we speak of a result with high or low accuracy and precision, respectively.

Throughout the whole chapter, we work with four types of datasets. The first dataset consists of the *original Pareto fronts* and Pareto sets obtained directly from MO-RBDO algorithms. However, these Pareto fronts are potentially erroneous because of the approximation technique - the meta-model, the reliability assessment method, or both. The *recalculated fronts* together with the original Pareto sets comprise the second dataset, we corrected those values as described for $\beta_{MC_{\mathrm{CoV}<0.05}}$ value. The recalculated fronts may not be Pareto efficient because the pure recalculation does not use any Pareto efficiency conditions but only the quasi-Monte Carlo simulation. Therefore, the recalculated fronts are re-sorted by using the Pareto efficiency conditions and the solutions with the best rank create *recalculated Pareto fronts*. The third dataset consists of the original Pareto sets and the recalculated Pareto fronts. The *superior Pareto set* and *superior Pareto front* create the fourth dataset. Because we do not have the Pareto-optimal solutions, we created their approximation. We merged all the Pareto fronts from several runs of MO-RBDO utilizing an analytical model and a preconditioned quasi-Monte Carlo simulation, and the Pareto efficiency conditions selected the superior Pareto front as the approximation of the Pareto-optimal front. The superior Pareto set corresponds with the superior Pareto front by mapping with the reliability assessment method and the cost function.

Since we use a uniform random design plans for a Design of Experiments for a meta-model assembly, each variable in DoE needs interval bounds. We work with three types of variables: deterministic design variables, stochastic variables, and stochastic design variables. Deterministic design variables do not bring any uncertainty into the problem, and therefore the interval bounds for optimization are the same as the interval bounds for DoE. Stochastic variables are not bounded (e.g. normally and Gumbel distributed random variables), one-sidedly bounded (e.g. log-normally, exponentially, and Weibull distributed random variables on the interval $(0, \infty)$), or bounded (e.g. a uniformly distributed random variable on the interval $[a, b]$). Therefore, we need to set up the bounds for variables with a domain of definition with open intervals and one-sided intervals. Otherwise, the Design of Experiments would be unnecessarily wide, and therefore it would need a lot more samples to get the same details as if the variable is artificially bounded. We use 0.15^{th} and 99.85^{th} percentiles as the lower and the upper bound for each random variable. These percentiles correspond approximately to ± 3 standard deviations of a standard normal distribution. Technically, we sample from a standard uniform distribution and this sample plan is then transformed into the uniform distribution with 0.15^{th} and 99.85^{th} percentiles bounds. Stochastic design variables combine the designing intervals defined for mean values of random variables together with uncertainty. The bounds are specified as the extension for the stochastic variables; the lower bound is 0.15^{th} percentile of a random variable with a minimum allowed mean value and given standard deviation and the upper bound is 99.85^{th} percentile of a random variable with a maximum permissible mean value and given standard deviation.

We present our results of the multi-objective reliability-based design optimization (MO-RBDO) utilizing the analytical model and three types of meta-models, namely global meta-models, sparse global meta-models, and local meta-models described in Sections 5.1.1, 5.1.2, and 5.1.3. To compare the behaviour of these three meta-models and the analytical model, a quasi-Monte Carlo method was used for the reliability assessment. It would be computationally very demanding to use a fixed number of samples for each reliability assessment in MO-RBDO. Therefore, we estimated the probability of failure of the individual candidate solution with the Subset simulation (SS) with a level probability $p_{F,i}$ equal to 0.1, the number of samples N in one level to 5,000, and the proposal distribution as normal with the standard deviation equal to standard deviations of all seeds in all dimensions d. Subsequently, the probability of failure p_F was recalculated with a quasi-Monte Carlo simulation with a number of samples equal to $100/p_F^{SS}$. With this preconditioning, we saved the computational resources for high failure probabilities where only a low number of samples is necessary. Due to the preconditioning by the Subset simulation, we distinguish this method as a preconditioned quasi-Monte Carlo simulation in the following text from the classical Monte Carlo simulation with a fixed number of evaluations for each individual. Every MO-RBDO with each model was run several times to obtain data for statistics.

As the second part of the published results, we compare several reliability assessment techniques for the multi-objective reliability-based design optimization utilizing only an analytical model. We use these results to minimize the error of the model and to concentrate on the errors in the reliability assessment technique. We use several reliability assessment techniques, namely a First-order reliability method (FORM), classical quasi-Monte Carlo simulation without any preconditioning (MC), Importance sampling (IS), Asymptotic sampling (AS), Subset simulation (SS), Enhanced Monte Carlo simulation (eMC), and Scaled Sigma Sampling (SSS). Sections 4.3 and 4.4 describe all the reliability assessment methods. We set the simulation techniques to get approximately the same number of limit state function evaluations during the MO-RBDO run; the First-order reliability method is the approximation method, and the larger number of limit state function evaluations does not satisfy the more precise failure probability estimates. A First-order reliability method employs a MATLAB `fmincon()` function utilizing an Active-set algorithm to find the design point. A quasi-Monte Carlo simulation used in this results section differs from the preconditioned quasi-Monte Carlo simulation from the first results section used with meta-models. We simply fix the number of samples of Halton sequences to 30,000 samples regardless of a failure probability. An Importance sampling utilizes a translation of the original PDF from mean values into the design point; an Active-set algorithm founds that design point via a MATLAB `fmincon()` function. The number of samples for each individual is equal to 30,000 in case of component reliability, and 30,000 samples divided by the number of limit state functions in case of system reliability, which is 30,000 samples in total per one individual reliability assessment. An Asymptotic sampling utilizes m number of samples in each quasi-Monte Carlo simulation equal to 2,048 samples per one level, an initial coefficient φ equal to 1, a decreasing factor φ_d equal to 0.95, N_0 equal to 10 representing failure samples quantity - if at least 10 samples are located in the failure domain, an Asymptotic sampling stores this step as a support point, and K number of support points equal to 10. A Subset simulation uses N equal to 5,000 samples in each level of probability $p_{F,i} = 0.1$. We chose a normal distribution as a proposal distribution with a standard deviation evaluated as a standard deviation of all seeds in level $(i-1)$ in all dimensions. An Enhanced Monte Carlo simulation utilizes 30,000 samples and λ is set to 100 values uniformly distributed in the interval $[0.1, 0.9]$; we also had to use a different optimization algorithm for a parameter search, namely a Trust region reflective algorithm instead of the Levenberg-Marquardt method used by authors since we had to use parameter bounds to stay in the real numbers in a domain of definition. A Scale sigma sampling uses five levels of Monte Carlo simulations with a σ parameter equal to $2, 3, \ldots, 6$, and 30,000 number of samples in total.

As the last part of the results, we combined the computationally inexpensive reliability assessment methods with meta-models to have the approximation in both parts of the algorithm. We selected the two best meta-models according to the first part of the results and the three best reliability assessment methods according to the second part of the results to obtain the best combination of approximation techniques for each numerical benchmark. As the indicator of quality, we selected the mean and the standard deviation of the error since we prefer having as much accurate and precise results as possible. Although these selected indicators are non-conflicting, we can compare them with Pareto efficiency conditions to find out the best algorithm. Therefore, we assigned a rank to each reliability assessment method for every example, and we summed those ranks for each reliability assessment method. Table 7.1 on the left presents the results. As the best reliability assessment method, we selected an Importance sampling, Subset simulation, and Asymptotic sampling. Tables and figures summarizing the worst meta-models with the selected reliability assessment methods for each example are published in Appendix 12. We use those results for the overall evaluation of achieved results in Section 7.6.

Example	AS	eMC	FORM	IS	MC	SS	SSS	GMM	LMM	SGMM
1	4	5	2	1	2	3	6	2	3	1
2	1	3	5	4	2	1	4	1	3	2
3	1	5	4	2	3	3	3	1	2	3
4	2	4	3	1	3	2	3	2	1	1
5	3	4	3	2	4	1	3	3	1	2
Sum	**11**	**21**	**17**	**10**	**14**	**10**	**19**	**9**	**10**	**9**
Placings	**3**	**7**	**5**	**1-2**	**4**	**1-2**	**6**	**1-2**	**3**	**1-2**

Table 7.1: Sorting to fronts according to Pareto efficiency conditions with a mean and standard deviation of the error criteria. Reliability assessment techniques and meta-models are compared independently.

7.1 Example 1: Mathematical problem with a nonlinear limit state function

This problem considers two deterministic design variables d_1 and d_2 and two stochastic variables X_1 and X_2; the deterministic design variables contribute only to the cost function while both deterministic design variables, as well as stochastic variables, are comprised in the limit state function.

$$\begin{aligned}
\min \quad & C(\mathbf{d}) = d_1^2 + d_2^2 && (7.2)\\
\max \quad & \beta(\mathbf{x},\mathbf{d}) = -\Phi^{-1}\big(\text{Prob}\big[g(\mathbf{x},\mathbf{d}) \le 0\big]\big) && (7.3)\\
& g(\mathbf{x},\mathbf{d}) = \frac{1}{5} d_1 d_2 x_2^2 - x_1 && (7.4)\\
& 0 \le d_{1,2} \le 15. && (7.5)
\end{aligned}$$

Stochastic variables X_1 and X_2 have a normal distribution with the mean value μ_{X_1} equals to 5 and μ_{X_2} equals to 3. Both variables have a coefficient of variation equal to 0.3 and they are statistically independent. For practical purposes, we limit the reliability index β to the range $0 \le \beta \le 4.5$.

AOUES and CHATEAUNEUF published this example as a single-objective reliability optimization problem in [3]. Their formulation is to minimize the cost function subject to the probability constrain bounded by the target failure probability $p_F^T = 0.01$ corresponding to the target reliability index $\beta^T = 2.32$ and the same constrain as in Equation 7.5.

$$\begin{aligned}
\min \quad & C(\mathbf{d}) = d_1^2 + d_2^2 && (7.6)\\
s.t. \quad & \text{Prob}\Big[\frac{1}{5} d_1 d_2 x_2^2 - x_1 \le 0\Big] \le p_F^T && (7.7)\\
& 0 \le d_{1,2} \le 15. && (7.8)
\end{aligned}$$

Paper [3] shows the single-objective optimum as $\mathbf{d}^* = [d_1, d_2] = [5.65, 5.65]$. We run the quasi-Monte Carlo simulation with 10^7 samples for this particular optimum and we obtained the reliability index β equal to 2.3363 and the failure probability $9.74 \cdot 10^{-3}$, which satisfies the constraint. The optimum cost $C(\mathbf{d}^*)$ is equal to 63.845 according to the authors and 63.88 according to the substitution to Equation 7.3. The small inaccuracy is in the rounding of the optimum values $\mathbf{d}^*$.

7.1.1 Comparison of meta-models and analytical model results together with a preconditioned quasi-Monte Carlo simulation

To study the behaviour of the meta-models, we used a preconditioned quasi-Monte Carlo simulation as a reliability assessment method as the most stable one. We run all the simulations on a desktop computer with hardware and software settings defined in Table 11.1. Table 7.2 shows the elapsed time and average values of simulations for one optimization run. Among the meta-models, the global meta-model was the slowest in utilization with MO-RBDO if the code was run parallel on eight threads. In the serial run, times are almost comparable among the meta-models. Since the analytical model does not use the meta-model assembly, there is no need to evaluate any samples in the Design of Experiments (DoE) and responses of the model in DoE. On the other hand, all the reliability assessment uses the analytical model. Therefore,

	AM	LMM	GMM	SGMM
elapsed time [hours] (1 core)	0.05	0.87	1.00	0.98
elapsed time [hours] (8 cores)	0.02	0.13	0.54	0.16
initial DoE	0	200	50	50
added samples	0	187	323	316
analytical g(x)-calls	$1.11 \cdot 10^8$	387	373	366
MM built for opt.	0	3050	31	31
MM built for update	0	672	0	0
MM-calls	0	$1.43 \cdot 10^8$	$9.06 \cdot 10^7$	$1.24 \cdot 10^8$

Table 7.2: Example 1: Comparison of statistics for RBDO utilizing quasi-Monte Carlo simulation with an analytical model (AM), local meta-model (LMM), global meta-model (GMM), and sparse global meta-model (SGMM).

all the responses are hidden in the table item "analytical g(x)-calls". The global meta-model and the sparse global meta-model need the same number of samples in the initial DoE since they are assembled for the whole design domain; local meta-models need a larger portion of an initial DoE since they are covering a smaller area and therefore need more information around each potential design candidate. On the other hand, the number of added samples into the existing DoE was larger for both versions of the global meta-models, the dense as well as the sparse versions, respectively. Thus the number of analytical g(x)-calls in all versions of meta-models is comparable. If we compare the number of meta-models evaluations and the number of analytical model evaluations for reliability purposes, the results are comparably the same.

Figure 7.3 depicts all the final Pareto sets and Pareto fronts from all runs used for the statistics above. Superior Pareto fronts dominate the single-objective optimum from reference [3] (see Figure 7.40). The global meta-model, as well as the sparse global meta-model, show similar behaviour as the analytical model. The Pareto sets are slightly wider compared to the analytical model, but the Pareto fronts are almost identical among the global meta-models and the analytical model in all runs. The local meta-models work well at the early stage of the Pareto front, but the behaviour is worse above the reliability index equal approximately to 2.5. The reason is that the local meta-models work nicely only if the proposed individual of the genetic algorithm is not so far from the limit state. In this particular example, the domain is small, and our universal setting of the selection of the number of points for the meta-model assembly to 1/10 of all DoE support points is too narrow. The sampling for larger reliability indices forces the local meta-model to extrapolate, and this is the reason for the worse behaviour at the tail of the Pareto front in higher reliability indices values. This effect is obvious in Figure 7.4. The influence domain is set to 1/10 of all points of DoE that are the closest to the selected individual. The meta-model behaves similarly as the analytical model in the area of the training data. However, the extrapolation around the limit state is evident. The limit state function has four inputs; this figure represents only a projection of the 4D function into 2D with fixed stochastic variables in their mean values. On the contrary, Figure 7.5 shows the space with fixed d_1 and d_2 values to 8.8381 and 12.2061 (the green point from Pareto set and Pareto front in Figure 7.4), where magenta circles depict the same training points for a local meta-model. The original model is hard to imitate by the local meta-model; especially around the limit state (thick contours). All the points in DoE depicted in Figure 7.4, and Figure 7.5 are only projections into 2D, the limit state function is, however, a section of the 4D function (with fixed values on mean values of

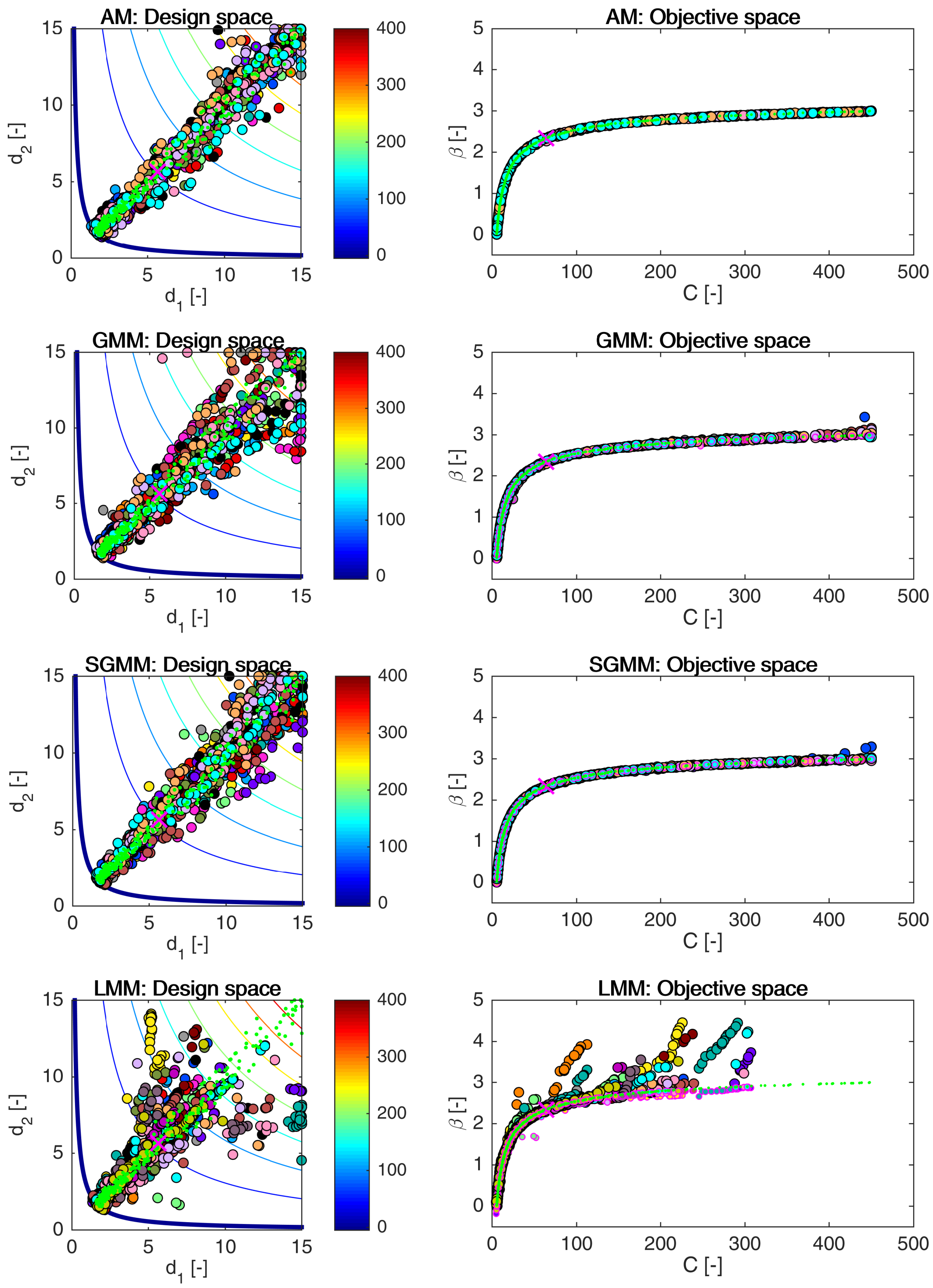


Figure 7.3: Example 1: Pareto sets (circles, left) and Pareto fronts (large circles, right) for all models, namely an analytical model (AM), global meta-model (GMM), sparse global meta-model (SGMM), and local meta-models (LMM). Larger circles with black edges depict original Pareto sets and fronts. Smaller circles with magenta edges in the objective space represent the recalculated fronts with an analytical model. The magenta cross depicts a single-objective optimum published in [3]. Green dots sets show the superior Pareto set and superior Pareto front. Every colour has the meaning of a different run. The contour plot in the design space represents an analytical limit state function; the bold contour is for the limit state $g(\mathbf{X}) = 0$.

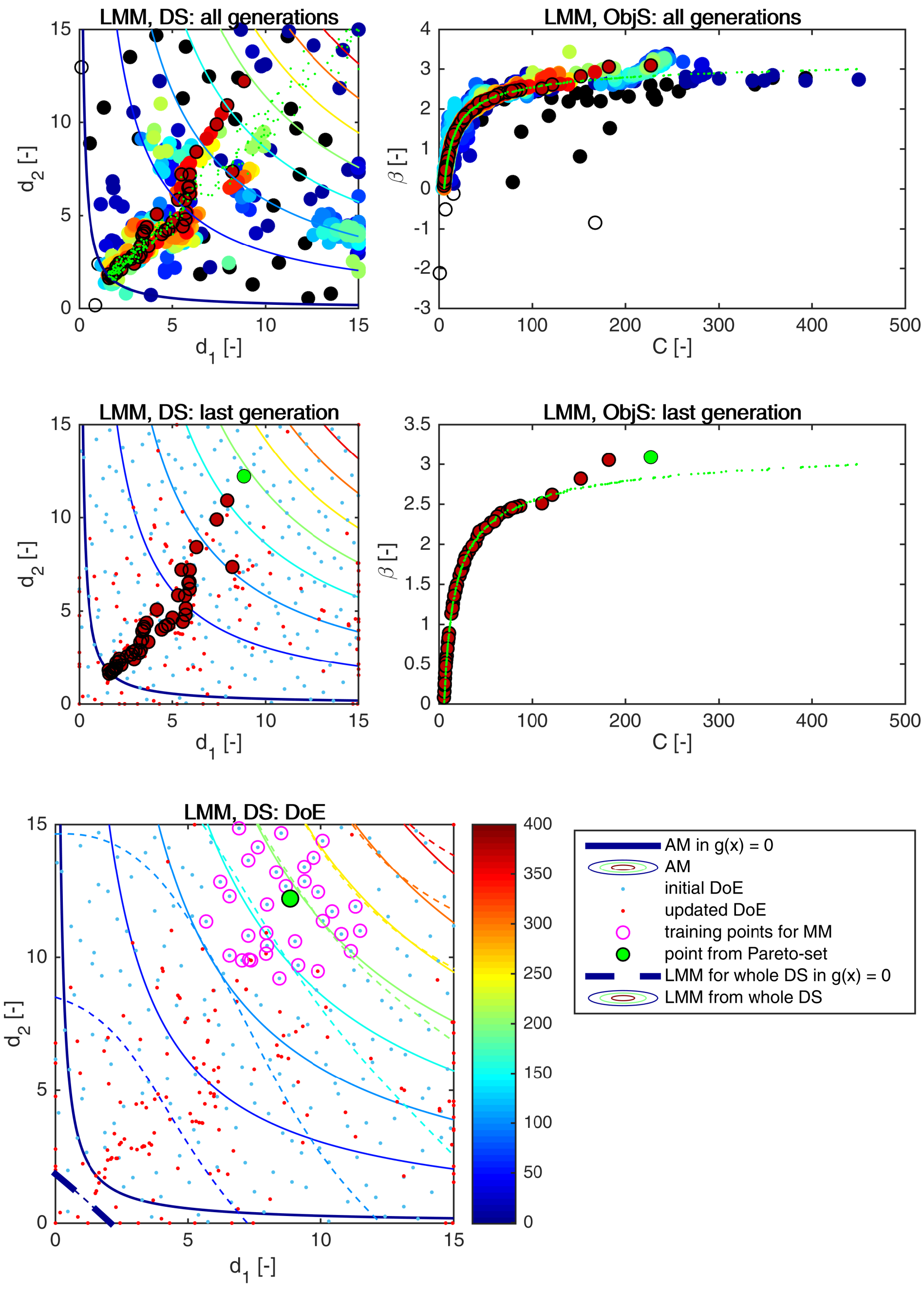


Figure 7.4: Example 1: Pareto sets and Pareto fronts from one MO-RBDO run utilizing local meta-models. (DS - design space, ObjS - objective space, LMM - local meta-models, AM - analytical model, DoE - Design of Experiments.) The first row of pictures represents all generations from the optimization algorithm (circles). The dark blue circles depict the first generation; the next generations continue with cyan, green, yellow, orange, and red colours. Empty circles denote the solutions, not satisfying β constraints. The last generation (Pareto set on the left and Pareto front on the right) is depicted in the second row of pictures together with the DoE (blue dots for initial DoE, red dots for added samples into DoE with an update). Set of green dots represents the superior Pareto front. The last picture represents LMM assembled for one individual (green circle) depicted for the whole design space by dashed contours. Solid contours represent the analytical model in all DS images.

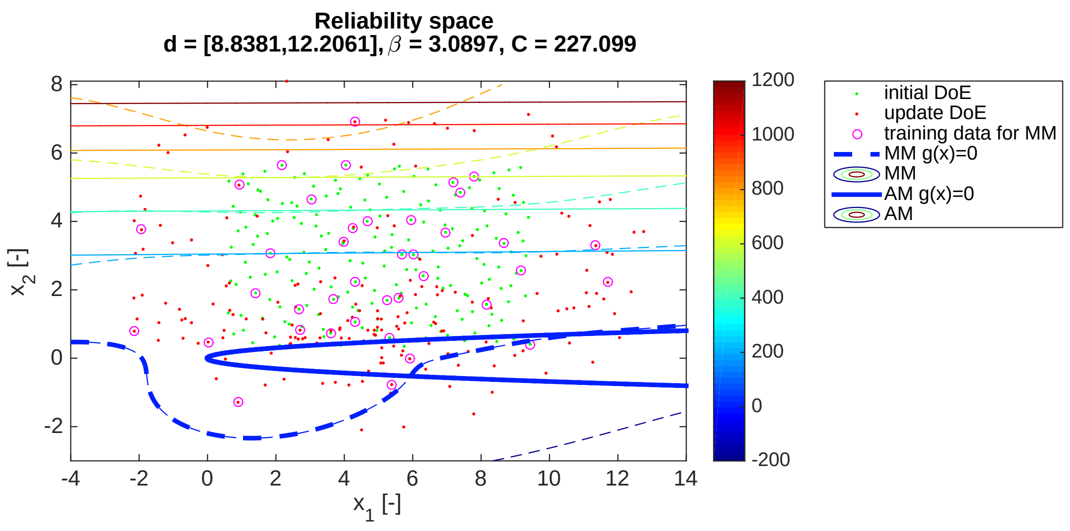


Figure 7.5: Example 1: The reliability space for fixed values of d_1 and d_2 equal to 8.8381 and 12.2061, respectively (the green individual from Figure 7.4). The reliability index is evaluated using the local meta-model.

the stochastic variables in Figure 7.4 and on the green point $\mathbf{d}$ in Figure 7.5). Some samples, therefore, can give the impression that they are closer than already selected points. However, the *k*-NN algorithm is set to choose the closest points to the proposed optimum by the genetic algorithm.

Table 7.3 shows the comparison of the mean values of performance measures described in Section 3.2 and errors; the last two columns represent the minimum and maximum achieved values. The sparse global meta-model behaves the best among all meta-models in unary metrics in total. The second-best meta-model is the global meta-model comparable with the local meta-model only in the Spacing metric. The results of the binary metrics are not as straightforward as the unary metric results. The Generational distance, representing how far is a recalculated Pareto front from a superior Pareto front, shows that the global meta-model was the best, followed by the sparse global meta-model, and the local meta-models behave worst. The Two set coverage metric C(A,B) calculates the proportion of solutions in set B that are weakly dominated by solutions in set A. This operator is non-symmetric. Therefore we calculated both versions of this metric. Metric C(sPF,rPF) indicates a proportionate part of a total of how many solutions of a superior Pareto front weakly dominates the solutions from recalculated Pareto fronts and vice versa, metric C(rPF,sPF) denotes a proportion of the dominating solutions from a recalculated Pareto fronts to a superior Pareto front. Surprisingly, the best performance C(sPF,rPF) was obtained by the local meta-models followed by the sparse global meta-model and subsequently the last global meta-model. The metric C(rPF,sPF) has almost comparable values for the local meta-models and the sparse global meta-model, closely followed by the global meta-model. The Coverage difference of two sets is a similar metric as the Two set coverage metric C(A,B) but operating with the hypervolumes. The Coverage difference of two sets D(A,B) measures the size of the space that is weakly dominated by front A but not weakly dominated by front B. The best performance was again achieved with the sparse global meta-model, followed by the global meta-model and the worst value got the local meta-models. The last part of the table shows the error of the reliability index in the original Pareto fronts. The sparse global meta-model again achieved the best performance, the second-best precise and accurate model was the global meta-model, and the worst model in the accuracy and precision of the obtained reliability index β was the local meta-model. Figure 12.1 in Appendix 12 shows the box-plots for the performance measures and the standard deviation of the Error.

	AM	GMM	LMM	SGMM	min	max
HV(rPF)	1574.5	1569.2	1519.7	1572.2	1456.1	1576.5
S(rPF)	0.0103	0.0125	0.0124	0.0111	0.0048	0.0525
Δ(rPF)	0.41	0.48	0.76	0.45	0.34	0.92
GD(rPF,sPF)	-	$5.5 \cdot 10^{-4}$	$6.5 \cdot 10^{-4}$	$5.7 \cdot 10^{-4}$	$3.1 \cdot 10^{-4}$	$2.3 \cdot 10^{-3}$
C(sPF,rPF)	-	0.643	0.533	0.613	0.298	0.838
C(rPF,sPF)	-	0.019	0.027	0.025	0	0.074
D(sPF,rPF)	-	15.99	65.47	12.93	11.51	129.13
$\min \epsilon(\beta_{\mathrm{oPF}})$	-	0.0002	0.0016	0.0001	0	0.013
$\max \epsilon(\beta_{\mathrm{oPF}})$	-	0.107	0.677	0.046	0.0195	1.77
E $\epsilon(\beta_{\mathrm{oPF}})$	-	0.015	0.120	0.009	0.0063	0.49
std $\epsilon(\beta_{\mathrm{oPF}})$	-	0.020	0.165	0.010	0.0055	0.51

Table 7.3: Example 1: Comparison of performance measures for RBDO utilizing quasi-Monte Carlo simulation with an analytical model (AM), local meta-model (LMM), global meta-model (GMM), and sparse global meta-model (SGMM). Used metrics are a Hypervolume (HV), Spacing (S), Spread (Δ), Generational distance (GD), Two set coverage metric (C(A,B)), Coverage difference of two sets (D(A,B)), and Error of the reliability index β_{oPF}. Datasets: rPF - recalculated Pareto fronts, sPF - superior Pareto front, oPF - original Pareto fronts. The best values are tinged with the green colour, the worst values with the red colour. The scale shows the visual comparison of the results.

7.1.2 Results comparison of different reliability assessment techniques utilizing an analytical model

We run the multi-objective reliability-based design optimization utilizing an analytical model together with different reliability assessment techniques to find the differences between the methods and the subsequent results. All MO-RBDO routines run ten times to get the relevant statistical data. All the simulations were performed on a laptop computer with hardware and software settings defined in Table 11.2. Table 7.4 shows average elapsed times for whole MO-RBDO runs together with the necessary number of limit state function evaluations. We tried to keep the number of limit state functions evaluations approximately the same except for the FORM since FORM is an approximation method, and the larger number of limit state function

	AS	eMC	FORM	IS	MC	SS	SSS
elapsed time [s]	116	57	66	71	20	39	36
g(x)-calls	$5.3 \cdot 10^7$	$4.7 \cdot 10^7$	$1.3 \cdot 10^5$	$4.4 \cdot 10^7$	$4.7 \cdot 10^7$	$3.3 \cdot 10^7$	$4.7 \cdot 10^7$

Table 7.4: Example 1: Comparison of statistics for RBDO utilizing an analytical model and different reliability assessment techniques, namely an Asymptotic sampling (AS), Enhanced Monte Carlo simulation (eMC), First-order reliability method (FORM), Importance sampling (IS), quasi-Monte Carlo simulation (MC), Subset simulation (SS), and Scaled sigma sampling (SSS).

evaluations do not provide the more precise failure probability estimate in comparison with the other simulation methods. Our fastest implementation is unsurprisingly MO-RBDO utilizing a quasi-Monte Carlo simulation; the slowest is MO-RBDO utilizing Asymptotic sampling.

Figure 7.6 and 7.7 show all Pareto sets (on the left) with corresponding Pareto fronts (on the right) obtained from MO-RBDO and different reliability assessment methods. MO-RBDO utilizing an Asymptotic sampling provides Pareto fronts that are noticeably below the recalculated data; this means that an Asymptotic sampling systematically underestimates the reliability index and the design approximation is on the safe side. The superior Pareto front is slightly above the recalculated Pareto fronts. The MO-RBDO utilizing an Enhanced Monte Carlo simulation provides the worst results from the visual point of view. The recalculated Pareto fronts are considerably above the provided data by MO-RBDO using eMC. MO-RBDO utilizing a First-order reliability method surprisingly provides very nice Pareto fronts that are almost identical with the recalculated fronts. MO-RBDO utilizing an Importance sampling provides very good numeric equality of recalculated and original Pareto fronts as well. Since the Pareto fronts do not exceed reliability index equal to 3, the number of quasi-Monte Carlo simulation samples are sufficient for the coefficient of variation of failure probability estimator slightly above the 10% for all individuals. Therefore the Pareto fronts are very precise. The number of samples for the β-index equal to 3 in the quasi-Monte Carlo simulation is equal to $7.41 \cdot 10^4$ for 10% coefficient of variation and we simulate 30,000 samples in each quasi-Monte Carlo simulation for each individual. Since we keep the number of limit state function evaluations approximately the same for all MO-RBDO except for the FORM, we can compare the efficiency of the advanced simulation techniques and the quasi-Monte Carlo simulation. MO-RBDO utilizing a Subset simulation provides Pareto fronts with slightly tremulous tails with the increasing cost function C. The reliability indices mostly overestimate the real value in comparison with the recalculated data from the visual point of view, which is on the hazardous side of the design. MO-RBDO utilizing a Scaled sigma sampling overestimates the reliability indices systematically and therefore provides the most hazardous Pareto fronts from all mentioned MO-RBDO methodologies in this section.

Table 7.5 shows the performance measures and errors for the data depicted in previous figures. The Hypervolume metric values are comparably the same, which means that even though the original Pareto fronts differ especially for MO-RBDO utilizing an Enhanced Monte Carlo simulation, the Pareto sets are located approximately in the same region; therefore, the subsequent recalculation provides almost identical data. All the recalculated Pareto fronts have very low Spacing performance measure, which means that according to this metric, all members are equidistantly spaced. This statement is not valid in comparison with the next performance measure, the Spread, therefore we suppose that the data are distributed in pairs, groups, or both, and these pairs or groups are uniformly distributed. The Spacing metric cannot capture this trend, and the metric is therefore somehow skewed. The Spread differs more considerably; the worst Spread is in the Pareto fronts from MO-RBDO utilizing a Subset simulation, Scaled sigma sampling, and Asymptotic sampling. The rest of the methodologies have approximately the same value of this metric, but none of these has an ideal Spread close to zero. All the recalculated Pareto fronts have a Generational distance metric close to zero. Therefore all of them are close to the superior Pareto front. The result for MO-RBDO utilizing a Scale sigma sampling is the worst of all, the GD metric is higher by one order of magnitude than for other MO-RBDO results in this section, but its GD value is still very close to zero, and therefore the correspondence between the data is very good; see the last set of images in Figure 7.7 - the aqua dots set compared to the small circles with the green edges. The superior Pareto front however weakly dominates more than half of the recalculated Pareto fronts, but almost none of

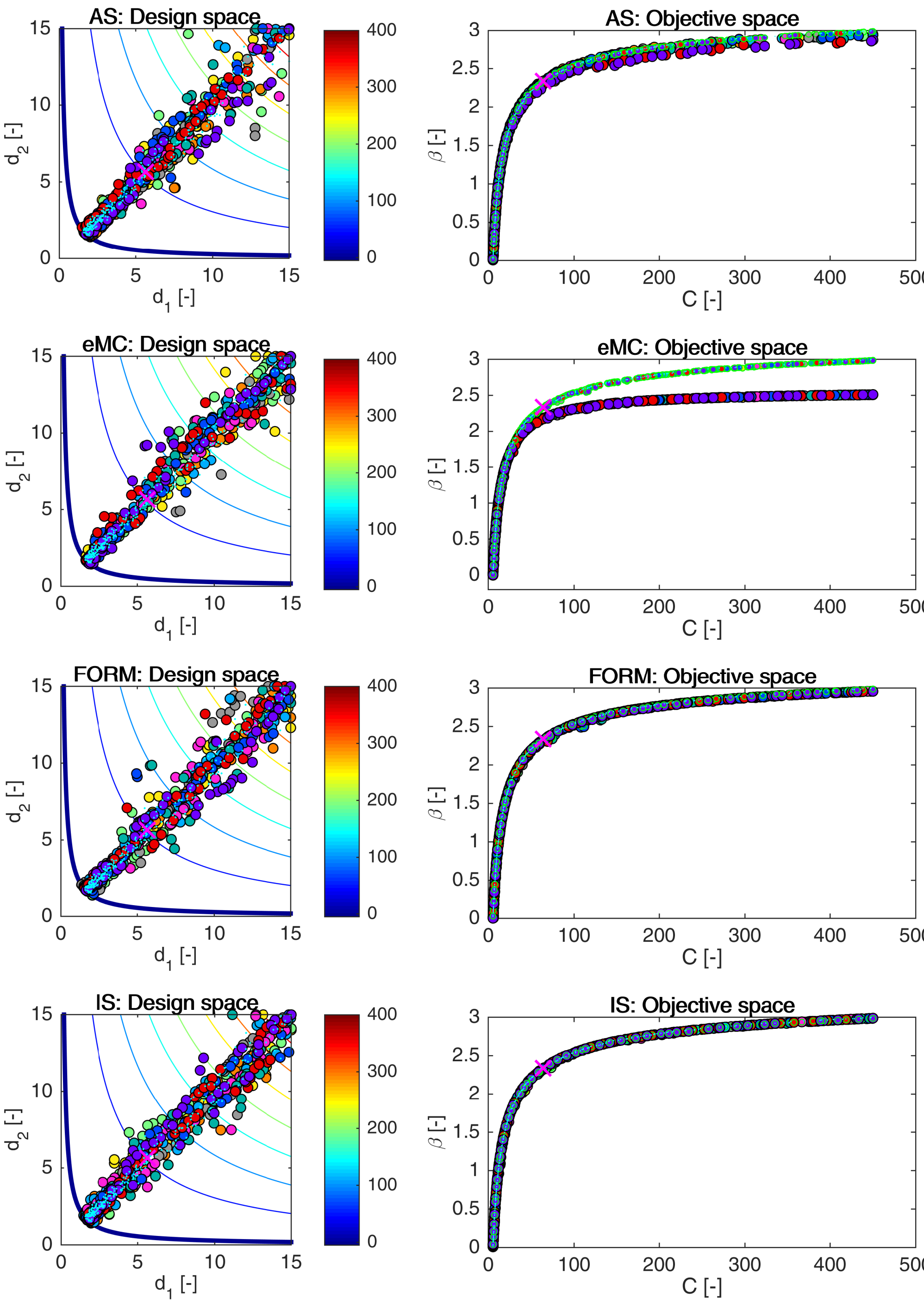


Figure 7.6: Example 1: Pareto sets (circles, left) and Pareto fronts (larger circles with black edges, right) for RBDO utilizing an Asymptotic sampling (AS), Enhanced Monte Carlo simulation (eMC), First-order reliability method (FORM), and Importance sampling (IS) together with an analytical model. The smaller circles with green edges in the objective space represent the recalculated Pareto sets to Pareto fronts with quasi-Monte Carlo simulations with CoV < 5%. The magenta cross represents a result published in [3]. Aqua dots sets show the superior Pareto set and Pareto front obtained with an analytical model together with a quasi-Monte Carlo simulation. Every colour means a different run. The contour plot in the design space represents an analytical limit state function; the bold contour is for the limit state $g(\mathbf{X}) = 0$.

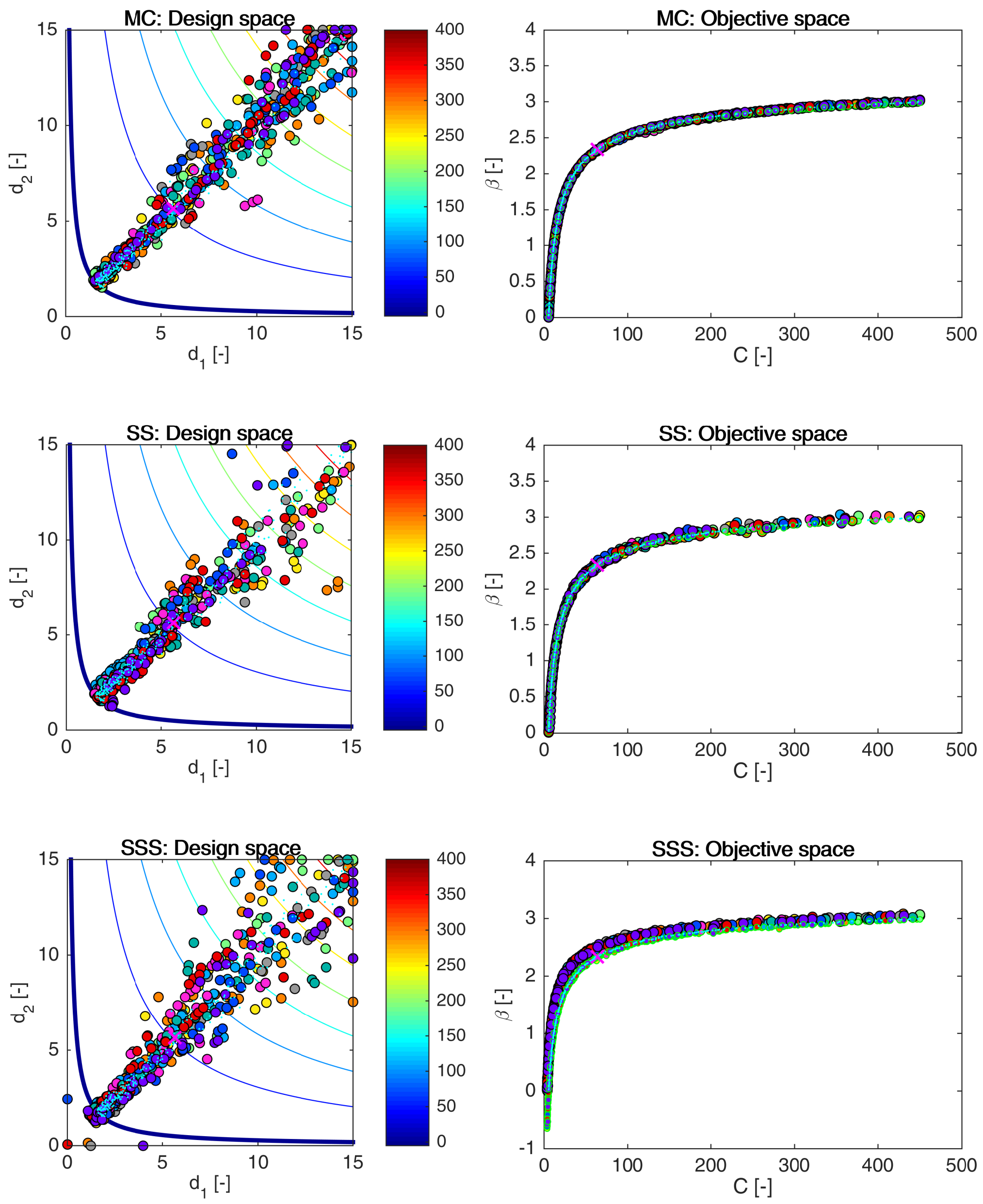


Figure 7.7: Example 1: Pareto sets (circles, left) and Pareto fronts (larger circles with black edges, right) for RBDO utilizing a quasi-Monte Carlo simulation (MC), Subset simulation (SS), and Scaled sigma sampling (SSS) together with an analytical model. The smaller circles with green edges in the objective space represent the recalculated Pareto sets to Pareto fronts with quasi-Monte Carlo simulations with CoV lesser than 5%. The magenta cross represents a result published in [3]. Aqua dots sets show the superior Pareto set and Pareto Pareto front obtained with an analytical model together with quasi-Monte Carlo simulation. Every colour means a different run. The contour plot in the design space represents an analytical limit state function; the bold contour is for the limit state $g(\mathbf{X}) = 0$.

	AS	eMC	FORM	IS	MC	SS	SSS	min	max
HV(rPF)	1667.7	1670.7	1670.7	1670.9	1671.7	1663.1	1667.1	1655.1	1672.7
S(rPF)	0.016	0.011	0.011	0.012	0.011	0.019	0.017	0.008	0.031
Δ(rPF)	0.57	0.39	0.39	0.39	0.39	0.64	0.58	0.26	0.72
GD(sPF,rPF) $[\cdot 10^{-4}]$	5.4	6.4	6.7	6.1	6.3	5.0	49.9	3.4	63.9
C(sPF,rPF)	0.57	0.59	0.62	0.65	0.63	0.51	0.51	0.36	0.82
C(rPF,sPF)	0.026	0.028	0.019	0.024	0.023	0.023	0.018	0	0.041
D(sPF,rPF)	16.32	13.36	13.33	13.09	12.39	20.96	17.57	11.33	29.01
min $\epsilon(\beta_{\text{oPF}})$	0.0003	0.0004	0.007	0	0.00002	0.0002	0.046	0	0.08
max $\epsilon(\beta_{\text{oPF}})$	0.074	0.475	0.036	0.011	0.034	0.077	0.641	0.01	0.69
E $\epsilon(\beta_{\text{oPF}})$	0.022	0.196	0.017	0.003	0.009	0.017	0.286	0.002	0.31
std $\epsilon(\beta_{\text{oPF}})$	0.020	0.179	0.007	0.002	0.009	0.018	0.192	0.002	0.20

Table 7.5: Example 1: Comparison of performance measures for RBDO utilizing an analytical model and different reliability assessment techniques, namely an Asymptotic sampling (AS), Enhanced Monte Carlo simulation (eMC), First-order reliability method (FORM), Importance sampling (IS), quasi-Monte Carlo simulation (MC), Subset simulation (SS), and Scaled sigma sampling (SSS). Used metrics are a Hypervolume (HV), Spacing (S), Spread (Δ), Generational distance (GD), Two set coverage metric (C(A,B)), Coverage difference of two sets (D(A,B)), and Error of the reliability index $\epsilon(\beta)$. The used datasets: rPF - recalculated Pareto front, sPF - superior Pareto front, oPF - original Pareto front, and rF - recalculated front. The best values are tinged with the green colour, the worst values with the red colour. The scale shows the visual comparison of the results.

the recalculated Pareto fronts dominate the superior Pareto front. The size of the space which is weakly dominated by the superior Pareto front but not weakly dominated by recalculated Pareto fronts from MO-RBDO is also very small in comparison with the Hypervolume metric. The last part of the table shows the Error of the reliability assessment techniques. MO-RBDO utilizing an Importance sampling obtained the most precise and accurate Pareto fronts, the worst mean error in the β-index is 0.011, and the standard deviation of the error is 0.003. The next most accurate Pareto fronts were obtained by MO-RBDO utilizing a quasi-Monte Carlo simulation with 30,000 samples for each individual. This result is obtained because the Pareto fronts have a maximum β-index equal to 3 for this example. MO-RBDO utilizing FORM has the second-best precision of the reliability indices. The worst accuracy and precision of the Pareto fronts were obtained by MO-RBDO utilizing a Scaled sigma sampling and second-worst accuracy and precision by MO-RBDO utilizing an Enhanced Monte Carlo simulation. Figure 12.2 in Appendix 12 shows the box-plots for the performance measures and the mean and standard deviation of the error.

7.1.3 Results for approximation using meta-models and reliability assessment techniques

As the final part of the testing of our proposed method, we merged the advanced reliability assessment methods with meta-models. The three best-behaving reliability assessment methods, namely an Asymptotic sampling, Importance sampling, and Subset simulation, are combined with all meta-models. We discuss only two best-behaving meta-models in this section, namely a sparse global meta-model, and global meta-model. The results for the local meta-models are given in Appendix 12 without further comments. The ranks of all methods, together with the best placings for all five examples, are in Table 7.1; the first example is in the first row of the table.

Table 7.6 shows the statistics for MO-RBDO for ten independent runs. All the computations run on a laptop computer with specifications defined in Table 11.2. Our fastest implementation is a combination of a Subset simulation together with a sparse global meta-model closely followed by a global meta-model together with the same simulation technique. Since both models are global, the number of initial samples in DoE is equal to 50. An Asymptotic sampling seems to provide more diverse samples for an update of DoE in comparison with an Importance sampling and Subset simulation. We kept the same setting of the simulation methods for an analytical function as well as for meta-models. The number of meta-model calls is approximately the same for both meta-models for the same simulation technique; from $3.5 \cdot 10^7$ to $8.9 \cdot 10^7$. The number of analytical function evaluations is dependent on the number of added samples to DoE; from 349 to 418 samples in total.

Figure 7.8 and 7.9 show all Pareto sets and Pareto fronts from MO-RBDO utilizing a global meta-model and sparse global meta-model together with all three selected simulation techniques. The difference between the original data and the recalculated counterparts is small from a visual point of view. The superior data nicely imitate the obtained data from MO-RBDO. Our superior Pareto sets and fronts dominate the single-objective optimum published in [3] (see Figure 7.40). MO-RBDO using an Asymptotic sampling and sparse global meta-model provides less stable solutions in both tails as well as MO-RBDO using Subset simulation and both versions of the global meta-model in the upper tail.

Table 7.7 provides performance measures and errors. According to the unary metrics and the errors indicators, MO-RBDO utilizing an Importance sampling performed the best; the differences between the models are inappreciable. The Hypervolumes are comparable for all the methods. The differences among the methods, as well as models, are minimal as well. The Spacing metric is close to zero for all methods, which means that distances between the solutions in the objective space are almost ideal. However, the Spread metric is at least 0.34. The difference between Spacing and the Spread metric is in several points. One difference

	GMM			SGMM		
	AS	IS	SS	AS	IS	SS
elapsed time [sec]	369	618	226	353	584	211
elapsed time [hours]	0.103	0.172	0.063	0.098	0.162	0.059
initial DoE	50	50	50	50	50	50
added samples	368	308	319	354	299	308
analytical g(x)-calls	418	358	369	404	349	358
MM built for opt.	31	31	31	31	31	31
MM-calls	$5.3{\cdot}10^7$	$8.9{\cdot}10^7$	$3.6{\cdot}10^7$	$5.2{\cdot}10^7$	$8.9{\cdot}10^7$	$3.4{\cdot}10^7$

Table 7.6: Example 1: Comparison of statistics for RBDO utilizing an Asymptotic sampling, Importance sampling, and Subset simulation with a global meta-model (GMM), and sparse global meta-model (SGMM).

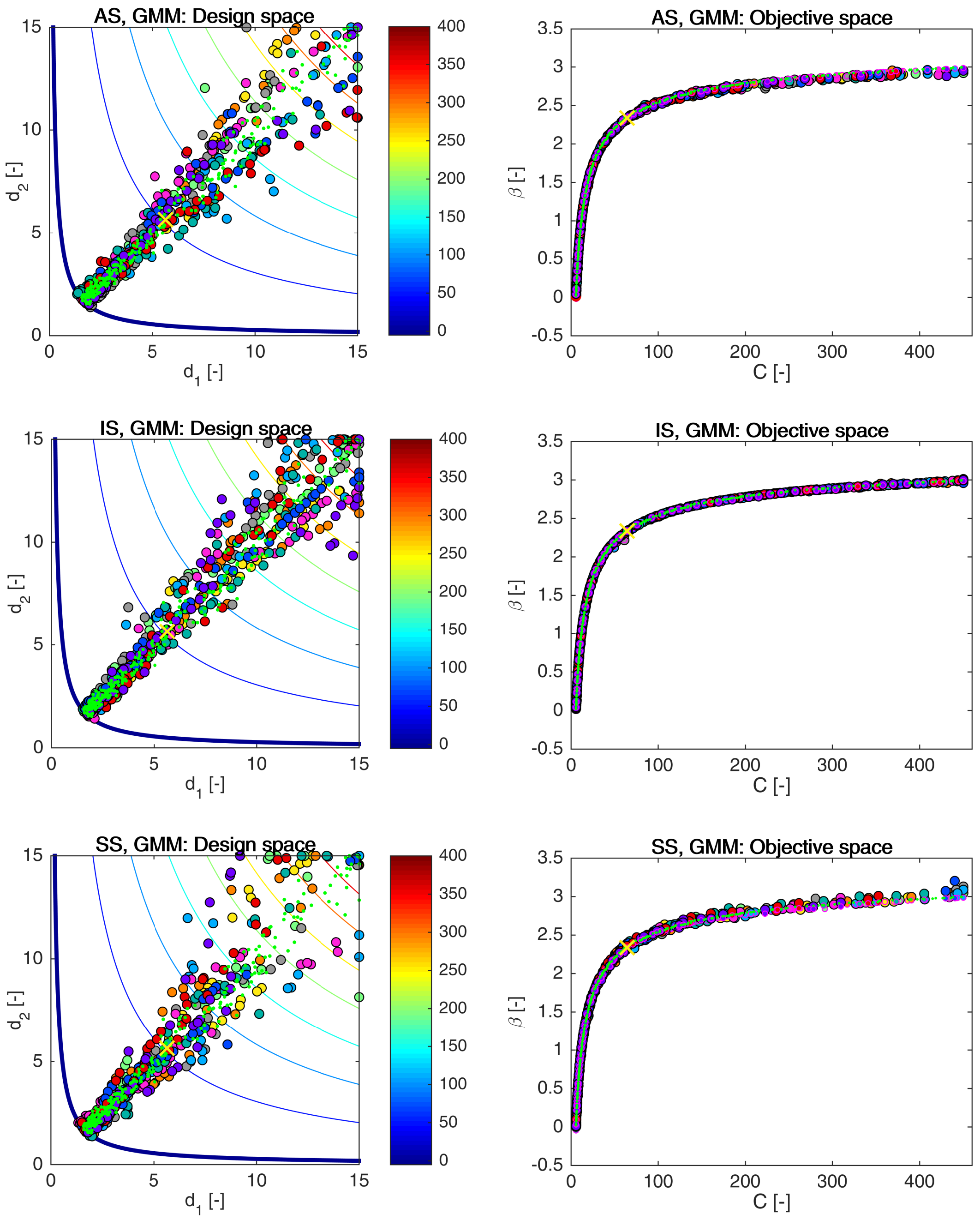


Figure 7.8: Example 1: Pareto sets (circles, left) and Pareto fronts (larger circles with black edges, right) for RBDO utilizing an Asymptotic sampling (AS), Importance sampling (IS), and Subset Simulation (SS) together with a global meta-model (GMM). The smaller circles with magenta edges in the objective space represent the recalculated Pareto sets to Pareto fronts with quasi-Monte Carlo simulations with $\mathrm{CoV} < 5\%$. The yellow cross represents a result published in [3]. Green dots sets show the superior Pareto set and Pareto front obtained with an analytical model together with a quasi-Monte Carlo simulation. Every colour means a different run. The contour plot in the design space represents an analytical limit state function; the bold contour is for the limit state $g(\mathbf{X}) = 0$.

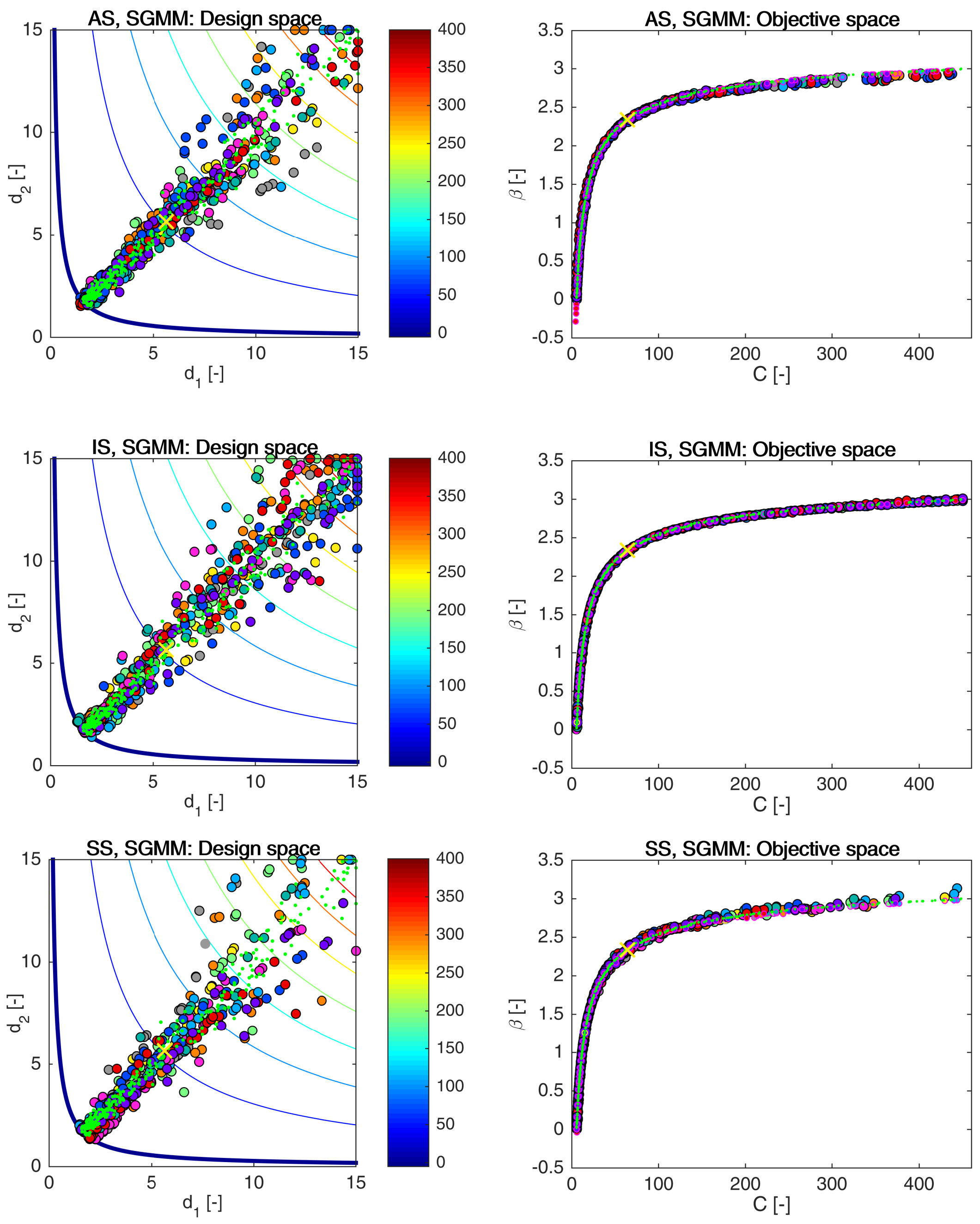


Figure 7.9: Example 1: Pareto sets (circles, left) and Pareto fronts (larger circles with black edges, right) for RBDO utilizing an Asymptotic sampling (AS), Importance sampling (IS), and Subset Simulation (SS) together with a sparse global meta-model (SGMM). The smaller circles with magenta edges in the objective space represent the recalculated Pareto sets to Pareto fronts with quasi-Monte Carlo simulations with $\mathrm{CoV} < 5\%$. The yellow cross represents a result published in [3]. Green dots sets show the superior Pareto set and Pareto front obtained with an analytical model together with a quasi-Monte Carlo simulation. Every colour means a different run. The contour plot in the design space represents an analytical limit state function; the bold contour is for the limit state $g(\mathbf{X}) = 0$.

	GMM			SGMM				
	AS	IS	SS	AS	IS	SS	min	max
HV (rPF)	1395.16	1397.79	1389.42	1395.18	1397.78	1388.87	1367.35	1399.43
S (rPF)	0.020	0.013	0.018	0.021	0.012	0.020	0.009	0.056
Δ (rPF)	0.60	0.42	0.67	0.60	0.39	0.68	0.34	0.83
GD (rPF,sPF)	$5.9{\cdot}10^{-4}$	$7.0{\cdot}10^{-4}$	$6.0{\cdot}10^{-4}$	$7.3{\cdot}10^{-4}$	$7.4{\cdot}10^{-4}$	$5.4{\cdot}10^{-4}$	$3.1{\cdot}10^{-4}$	$2.3{\cdot}10^{-3}$
C(sPF,rPF)	0.56	0.66	0.55	0.50	0.66	0.58	0.37	0.80
C(rPF,sPF)	0.023	0.017	0.027	0.028	0.019	0.022	0.004	0.053
D(sPF,rPF)	15.0	12.3	20.7	15.0	12.4	21.3	10.7	42.8
$\min \epsilon(\beta_{\text{oPF}})$	$3{\cdot}10^{-4}$	$2{\cdot}10^{-6}$	$7{\cdot}10^{-4}$	$6{\cdot}10^{-4}$	$3{\cdot}10^{-5}$	$4{\cdot}10^{-4}$	0	$1.6{\cdot}10^{-3}$
$\max \epsilon(\beta_{\text{oPF}})$	0.06	0.02	0.10	0.09	0.02	0.08	0.01	0.32
E $\epsilon(\beta_{\text{oPF}})$	0.014	0.0039	0.021	0.024	0.0044	0.018	0.003	0.100
std $\epsilon(\beta_{\text{oPF}})$	0.014	0.0046	0.021	0.023	0.0049	0.019	0.002	0.100

Table 7.7: Example 1: Comparison of performance measures for RBDO utilizing different reliability assessment techniques namely an Asymptotic sampling (AS), Importance sampling (IS), and Subset simulation (SS) using a global (GMM) and sparse global meta-model (SGMM). Used metrics are a Hypervolume (HV), Spacing (S), Spread (Δ), Generational distance (GD), Two set coverage metric (C(A,B)), Coverage difference of two sets (D(A,B)), and Error of the reliability index $\epsilon(\beta)$. The used datasets: rPF - recalculated Pareto front, sPF - superior Pareto front, oPF - original Pareto front, and rF - recalculated front. The best values are tinged with the green colour, the worst values with the red colour. The scale shows the visual comparison of the results.

is that the cumulative distances are divided by the number of solutions in the Spacing metric and by the ideal cumulative distances in the Spread metric. Both measures can be zero in the ideal case; however, the Spacing does not consider any other solutions except for the solutions belonging to the Pareto front, while the Spread is zero only if the extreme solutions merge with the boundary solutions in the Pareto front and the distances between the solutions are equal. In case that the extreme solutions that are selected by the user in advance do not merge, the Spread metric is nonzero. This condition is however omitted in case of the Spacing metric. The second difference is in search of the closest point. The Spacing metric looks for the closest solution among all solutions for each one; this means that it can omit the pairwise, groupwise, or both spread since the pairs, groups, or both are always close to each other; on the contrary, the Spacing metric can detect only outlier solutions and not the gaps between groups. The Spread metric employs the sorted distances, and therefore it can embrace the larger gaps between the consecutive solutions. Accordingly, these two metrics provide different data in our work. We select the extreme solutions as the boundary solutions from the union of the superior Pareto front and the actual recalculated Pareto front.

The binary metrics shows that all the recalculated Pareto fronts are close to the superior Pareto front. The superior Pareto front weakly dominates at least half of the solutions from the recalculated Pareto fronts, and the recalculated Pareto fronts weakly dominate only a minimal number of solutions belonging to the superior Pareto front. The worst results for the binary metrics come from the MO-RBDO utilizing an Importance sampling with one exception, the Coverage difference of two sets, which is the best for MO-RBDO utilizing IS. The error is the smallest for the MO-RBDO utilizing IS regardless of the meta-model. The error is even smaller for the combination of the sparse global meta-model and an Importance sampling (E $\epsilon(\beta_{\text{oPF}}) = 0.0044$, std $\epsilon(\beta_{\text{oPF}}) = 0.0049$) than for the preconditioned Monte Carlo simulation and the sparse global meta-model (E $\epsilon(\beta_{\text{oPF}}) = 0.0089$, std $\epsilon(\beta_{\text{oPF}}) = 0.010$). The reason is that the Importance sampling provides better samples for DoE update since the Importance sampling density samples around the design point instead of the mean values. The updating procedure has more points to choose from the Importance sampling than from the preconditioned MC, and the limit state of the meta-model can be very precise after the update from the Importance sampling. Just for comparison, MO-RBDO utilizing an analytical model and an Importance sampling also performed the best if the error indicators are considered (E $\epsilon(\beta_{\text{oPF}}) = 0.0026$, std $\epsilon(\beta_{\text{oPF}}) = 0.0024$). Figure 12.3 in Appendix 12 shows the box-plots for the performance measures and the mean and standard deviation of the error.

7.2 Example 2: Mathematical problem with a highly nonlinear series system reliability

This problem considers two stochastic design variables X_1 and X_2 contributing to the cost function with their mean values and concurrently to the limit state function as the stochastic variables. The problem has two component limit state functions, where the components are connected into the series system. Therefore, we use the minimum value from both limit state functions values, and this representative value determines whether the system fails or not. Mathematically written

$$\min \quad C(\mu_{X_1}, \mu_{X_2}) = (\mu_{X_1} - 3.7)^2 + (\mu_{X_2} - 4)^2 \tag{7.9}$$

$$\max \quad \beta(\mathbf{X}) = -\Phi^{-1}(\text{Prob}[g(\mathbf{X}) \leq 0]) \tag{7.10}$$

$$g(\mathbf{X}) = \min \begin{pmatrix} -x_1 \sin(4x_1) - 1.1x_2 \sin(2x_2) \\ x_1 + x_2 - 3 \end{pmatrix} \tag{7.11}$$

$$0 \leq \mu_{X_1} \leq 3.7, \quad 0 \leq \mu_{X_2} \leq 4. \tag{7.12}$$

Stochastic design variable X_1 and X_2 have a normal distribution, the mean value μ_{X_1} and μ_{X_2} represent the design variables bounded from below with 0 and from above with 3.7 and 4, respectively. The standard deviations are both equal to 0.1. The variables are statistically independent. For practical purposes, we limited the reliability index β to the range $0 \leq \beta \leq 4.5$. The cost function, as well as the limit state function, are depicted in Figure 7.10.

This example was first studied as the single-objective optimization problem utilizing a surrogate model by [109] and subsequently as the decoupling approach by [32]. The optimization problem of the original single-objective optimization problem is according to reference [109]

$$\min \quad C(\mu_{X_1}, \mu_{X_2}) = (\mu_{X_1} - 3.7)^2 + (\mu_{X_2} - 4)^2 \tag{7.13}$$

$$\text{s.t.} \quad g_1(\mathbf{X}) = -X_1 \sin(4X_1) - 1.1X_2 \sin(2X_2) \tag{7.14}$$

$$g_2(\mathbf{X}) = X_1 + X_2 - 3, \tag{7.15}$$

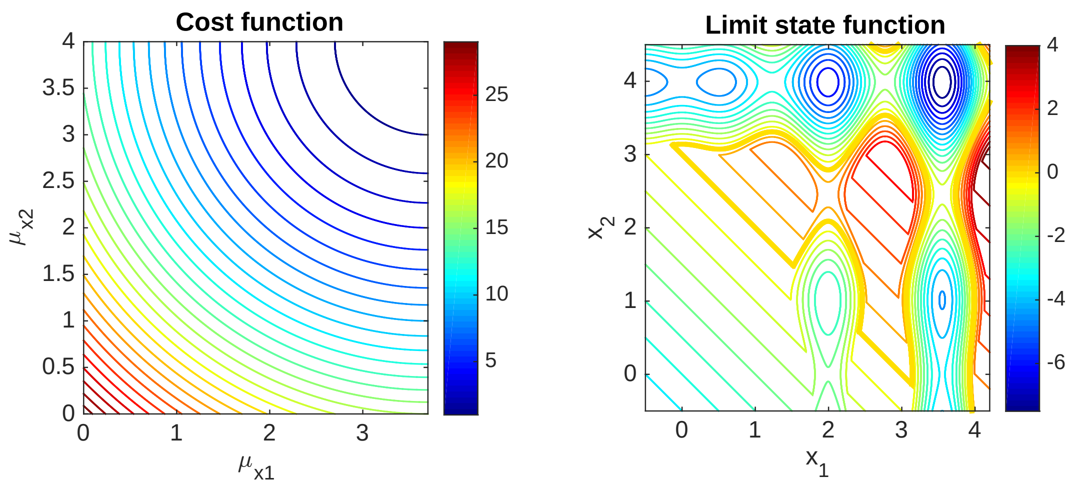


Figure 7.10: Example 2: The cost function (left) and the limit state function (right). The bold orange contour (right) is a limit state $g(\mathbf{x}) = 0$; the failure region is for $g(\mathbf{x}) \leq 0$, i.e. from the yellow to the blue on the scale.

and according to reference [32] in a probabilistic formulation

$$\min \quad C(\mu_{X_1}, \mu_{X_2}) = (\mu_{X_1} - 3.7)^2 + (\mu_{X_2} - 4)^2 \tag{7.16}$$

$$s.t. \quad \text{Prob}\Big[g(\mathbf{X}) \leq 0\Big] \leq \Phi(-\beta^t) \tag{7.17}$$

$$g_1(\mathbf{X}) = -X_1 \sin(4X_1) - 1.1X_2 \sin(2X_2) \tag{7.18}$$

$$g_2(\mathbf{X}) = X_1 + X_2 - 3 \tag{7.19}$$

$$0 \leq \mu_{X_1} \leq 3.7, \quad 0 \leq \mu_{X_2} \leq 4. \tag{7.20}$$

Table 7.8 summarizes the single-objective optima published in the literature.

Reference	μ_{X_1}	μ_{X_2}	$C(\mu_{X_1}, \mu_{X_2})$ [-]		β [-]	
			published	corrected	published	corrected
[59]	2.81	3.25	1.26	1.3546	1.67	2.083
[109]	2.8235	3.2963	-	1.2634	2	1.6764
[109]	2.7971	3.2266	-	1.4134	2.9	2.3118
[32]	2.8163	3.2768	1.3038	1.3039	1.837	1.8575

Table 7.8: Example 2: Published single-objective optima.

7.2.1 Comparison of meta-models and analytical model results together with a preconditioned quasi-Monte Carlo simulation

First of all, the study of the behaviour of the meta-models was carried out as in the previous Example 1. A preconditioned quasi-Monte Carlo simulation was used as a reliability assessment method since we have the best experiences with its stability. To get relevant outputs, we run each optimization procedure with the same setting and the same meta-model several times. Table 7.9 shows the average values for one optimization run. The analytical model is the simplest; the time of the MO-RBDO evaluation with its utilization is the fastest. Local meta-models are the

	AM	LMM	GMM	SGMM
elapsed time [hours] (1 thread)	1.37	5.94	23.61	22.88
elapsed time [hours] (8 threads)	0.80	2.81	5.26	4.85
initial DoE	0	200	50	50
added samples	0	228	323	296
analytical g(x)-calls	$8.21 \cdot 10^9$	428	373	346
MM built for opt.	0	3050	31	31
MM built for update	0	4208	0	0
MM-calls	0	$8.48 \cdot 10^9$	$7.32 \cdot 10^9$	$7.53 \cdot 10^9$

Table 7.9: Example 2: Comparison of statistics for RBDO utilizing a quasi-Monte Carlo simulation with an analytical model (AM), local meta-models (LMM), global meta-model (GMM), and sparse global meta-model (SGMM).

fastest type of a meta-model that we implemented; followed by the sparse global meta-model and the global meta-model. We run the code in serial on one thread as well as parallel at eight threads. The speed-up is evident particularly for the global meta-model evaluations; the speed-up is not linear, but it is possible to save up to 79% of computational time. The total number of analytical function evaluations was the highest for the local meta-models, followed by the global meta-model and the sparse global meta-model if we take into account only MO-RBDO with meta-models. All three versions of meta-models, as well as the analytical model, were simulated comparably in the order of magnitude for the reliability assessment purposes.

Figure 7.11 depicts all Pareto fronts and Pareto sets from several runs. All the models show similar behaviour with very few exceptions in outliers, namely two components of the Pareto front using the global meta-model and similarly the sparse global meta-model.

Table 7.10 shows the comparison of performance measures described in Section 3.2. Since the quality of all meta-models is excellent, there are only minor deviations in the metrics. The analytical model got the best metric value for the Hypervolume; on the other side, it got the worst value for the Spread metric. The global meta-model was insignificantly better in unary metrics among the meta-models. The Spacing measure is close to zero for all models; therefore, almost all members in Pareto fronts are equidistantly spaced. On the other side, the Spread metric Δ is approximately 0.37, which means that the Spread is not ideal, however, if we compare the meta-models metric values to the analytical model metric value, they are almost identical. Therefore, the Spread metric is affected by choice of the multi-objective optimization algorithm and not by the meta-model itself. The difference in these two metrics can be influenced by the pairwise spread, which is omitted by the Spacing metric and not by the Spread metric. The local meta-models were insignificantly better in binary metrics. A Generational distance metric shows that all the Pareto fronts obtained by utilizing meta-models are very close to the superior Pareto front. However, the superior Pareto front weakly dominates almost half of the solutions obtained by using meta-models according to the Two set coverage metric. The error shows that the global meta-model had the smallest errors in the evaluation of the limit state function in general except for the minimum error, which was the smallest for the sparse global meta-model. However, all the errors are very low, and therefore meta-models show the very accurate and precise prediction of the limit state function. Figure 12.4 in Appendix 12 shows the box-plots for the performance measures and the mean and standard deviation of the error.

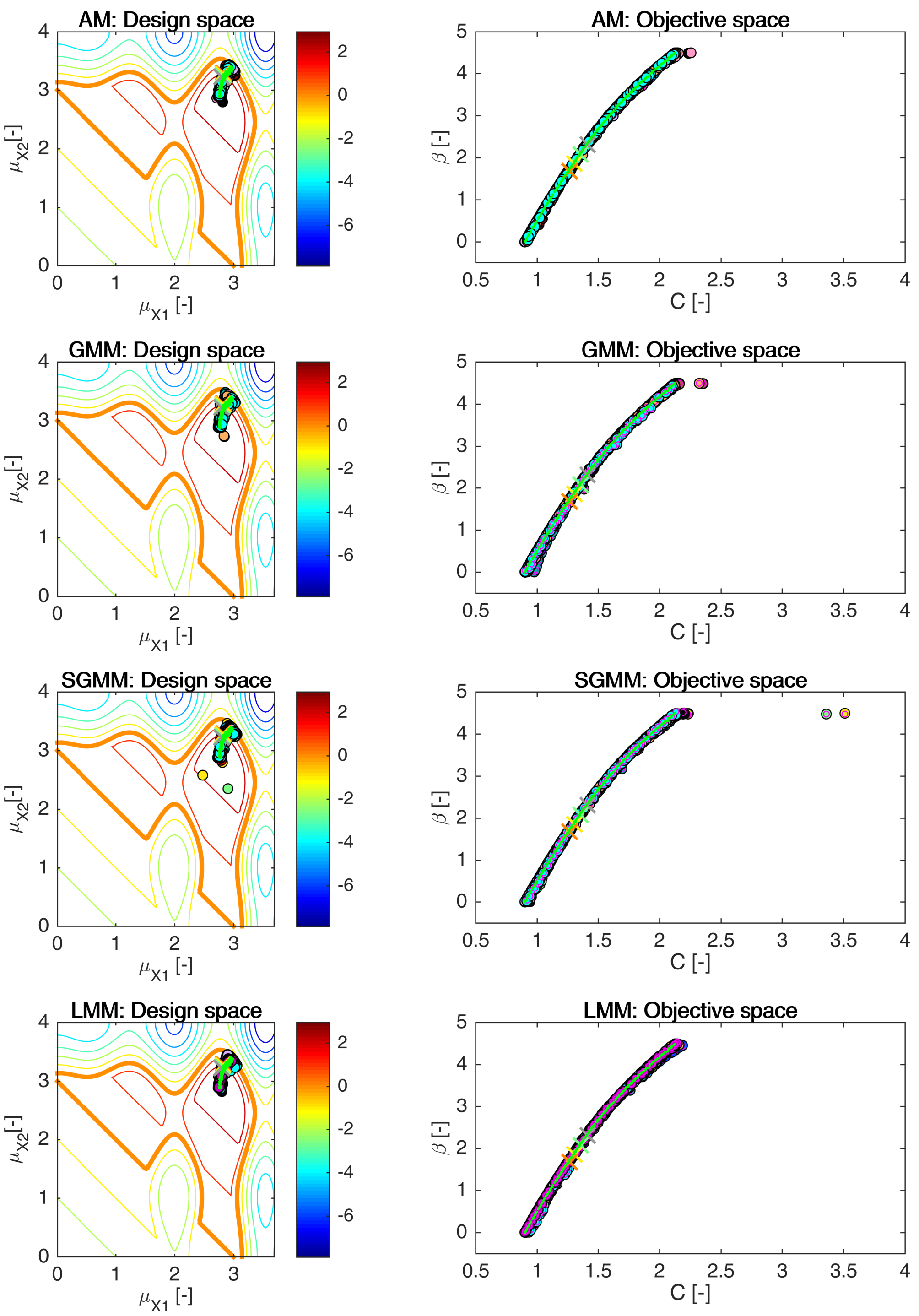


Figure 7.11: Example 2: Pareto sets (circles, left) and Pareto fronts (large circles, right) for all models, namely an analytical model (AM), global meta-model (GMM), sparse global meta-model (SGMM), and local meta-models (LMM). Smaller circles with magenta edges in the objective space represent the recalculated fronts with an analytical model. Crosses represent results published in [59, 109, 32]. Green dots sets show the superior Pareto set and superior Pareto front. Every colour has the meaning of a different run. The contour plot in the design space represents an analytical limit state function; the bold contour is for the limit state $g(\mathbf{X}) = 0$.

	AM	GMM	LMM	SGMM	min	max
HV	12.99	12.91	12.90	12.89	12.70	13.03
S	0.0108	0.0098	0.0102	0.0103	0.0072	0.0156
Δ	0.37	0.37	0.38	0.39	0.29	0.47
GD	-	0.00059	0.00058	0.00055	0.00033	0.00180
C(PF$_{\text{AM}}$,PF$_{\text{MM}}$)	-	0.49	0.47	0.51	0.20	0.75
C(PF$_{\text{MM}}$,PF$_{\text{AM}}$)	-	0.030	0.033	0.030	0.004	0.067
D(PF$_{\text{AM}}$,PF$_{\text{MM}}$)	-	0.19	0.21	0.22	0.07	0.40
$\min \epsilon(\beta_{\text{oPF}})$	-	0.0002	0.0002	0.0001	0	0.002
$\max \epsilon(\beta_{\text{oPF}})$	-	0.032	0.043	0.040	0.016	0.093
E $\epsilon(\beta_{\text{oPF}})$	-	0.0080	0.0090	0.0085	0.0063	0.014
std $\epsilon(\beta_{\text{oPF}})$	-	0.0070	0.0085	0.0079	0.0039	0.019

Table 7.10: Example 2: Comparison of performance measures for RBDO utilizing a quasi-Monte Carlo simulation with an analytical model (AM), local meta-models (LMM), global meta-model (GMM), and sparse global meta-model (SGMM). Used metrics are a Hypervolume (HV), Spacing (S), Spread (Δ), Generational distance (GD), Two set coverage metric (C(A,B)), Coverage difference of two sets (D(A,B)), and Error of the reliability index β_{oPF}. The best values are tinged with the green colour, the worst values with the red colour. The scale shows the visual comparison of the results.

7.2.2 Results comparison of different reliability assessment techniques utilizing an analytical model

This section provides results for the multi-objective reliability-based design optimization utilizing an analytical model and approximation techniques for the reliability assessment minimizing the error in the model part. The differences in the Pareto fronts are caused only by the simulation techniques or the First-order reliability method. All the simulations were run on the laptop computer with the hardware and software settings recorded in Table 11.2. The setting of all the reliability assessment methods is described at the beginning of this chapter.

The number of limit state function evaluations is lower by two orders of magnitude if compared to results in the previous section, see Table 7.11 and 7.9. Our fastest implementation is MO-RBDO utilizing a quasi-Monte Carlo simulation followed by a Scale sigma sampling. The slowest implementation is MO-RBDO utilizing an Asymptotic sampling followed by an Importance sampling and Enhanced Monte Carlo simulation.

	AS	eMC	FORM	IS	MC	SS	SSS
elapsed time [s]	91	86	49	87	15	44	23
g(x)-calls	$6.3 \cdot 10^7$	$4.7 \cdot 10^7$	$6.2 \cdot 10^4$	$5.9 \cdot 10^7$	$4.7 \cdot 10^7$	$4.4 \cdot 10^7$	$4.1 \cdot 10^7$

Table 7.11: Example 2: Comparison of statistics for RBDO utilizing an analytical model and different reliability assessment techniques, namely an Asymptotic sampling (AS), Enhanced Monte Carlo simulation (eMC), First-order reliability method (FORM), Importance sampling (IS), quasi-Monte Carlo simulation (MC), Subset simulation (SS), and Scaled sigma sampling (SSS).

The Importance sampling, as well as the First-order reliability method, have to be extended for the series system reliability since these methods work only for the components. The extension is the same for both methods. The reliability is evaluated for each limit state function together with the unit normal to the hyperplanes. The multi-normal CDF is subsequently evaluated for the final failure probability. The Enhanced Monte Carlo simulation is also used in the different form than for the component reliability; we used the recommendations in the NAESS et al. paper [136] described in Section 4.4.5.

A Scaled sigma sampling can have difficulties if the limit state function contains a finite failure domain or a finite safe domain creating a bounded area. The combination of both cases is shown in Figure 7.12; the limit state function consists of two functions, one is highly nonlinear, and together with the second one, it creates the island of the safe space. With an enlarged area of the design space, the safe space appears behind the failure region with larger values of variable x_1 once more. The vector $p_{F,h,q}^{MC}$ in SSS has to contain only increasing values of the failure probabilities with the increasing q since the growing σ_q performs the growing failure probability in every quasi-Monte Carlo simulation. However, because of the high nonlinearity often changing the sign of the limit state particularly with the increasing values of a variable x_1 in this example, the $p_{F,h,q}^{MC}$ contains decreasing numbers from the beginning and the increasing numbers at the end with the increasing q. The theory behind the Scaled sigma sampling is then not working and the resulting approximation of the failure probability p_F can be even larger than 1 (which is a wrong result). We handled this problem in this example by changing vector σ into smaller values to avoid generating samples outside the failure region to hit the safe region with the large values of variable x_1.

Figure 7.13 and 7.14 show the Pareto sets on the left and the Pareto fronts on the right from different MO-RBDO runs utilizing seven different reliability assessment techniques. The bold

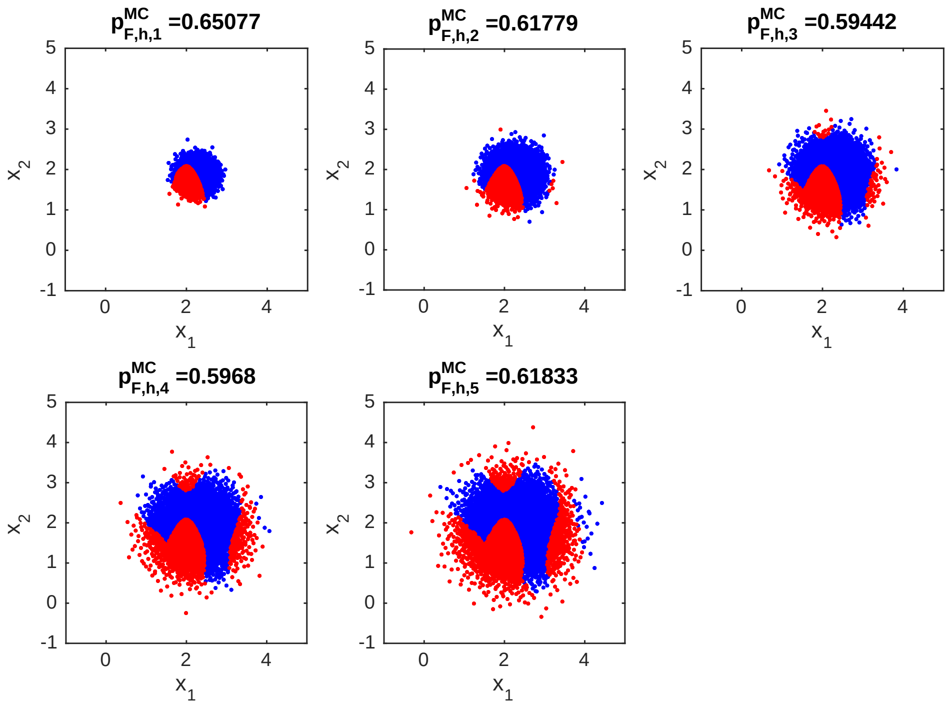


Figure 7.12: Example 2: Scaled sigma sampling on a combination of finite safe and failure domains. Red dots show the samples in the failure domain, while the blue dots represent the samples in the safe domain.

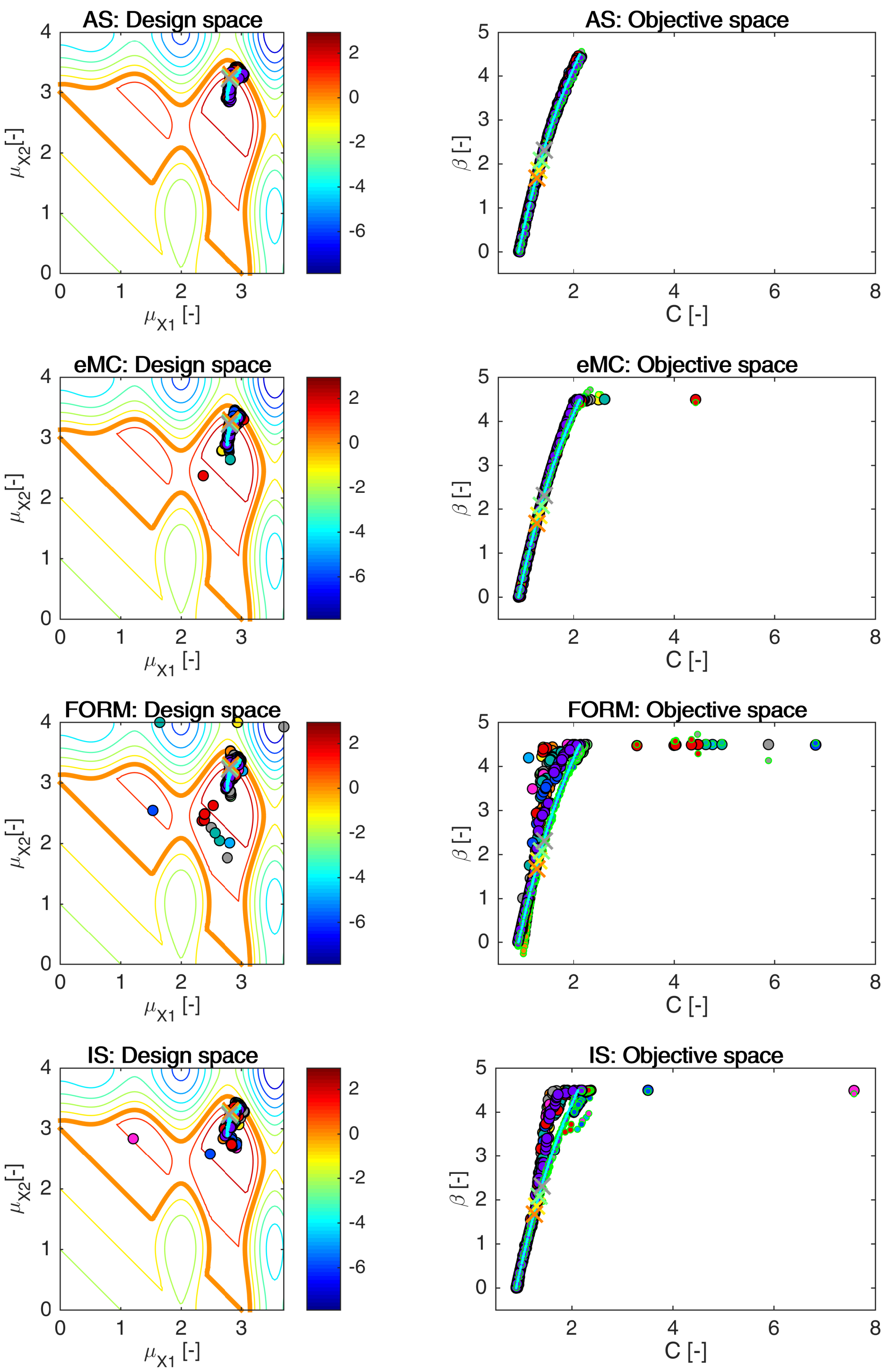


Figure 7.13: Example 2: Pareto sets (circles, left) and Pareto fronts (larger circles with black edges, right) for RBDO utilizing an Asymptotic sampling (AS), Enhanced Monte Carlo simulation (eMC), First-order reliability method (FORM), and Importance sampling (IS) together with an analytical model. The smaller circles with green edges in the objective space represent the recalculated Pareto sets to Pareto fronts with quasi-Monte Carlo simulations with CoV lesser than 5%. Orange, yellow, green and grey crosses represent results published in [109], [32], [59], and [32]. Aqua dots sets show the superior Pareto set and superior Pareto front. Every colour means a different run. The contour plot represents the limit state function.

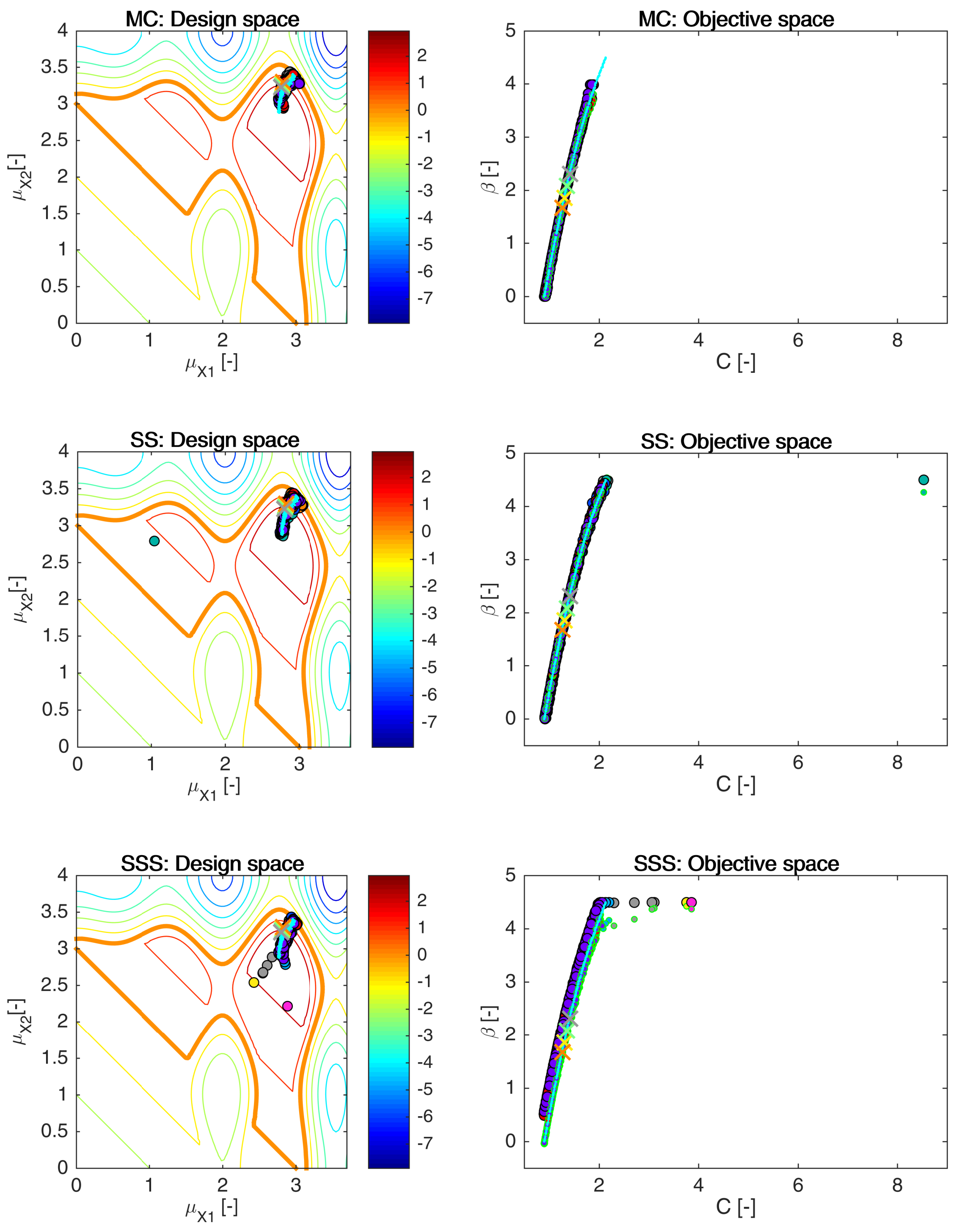


Figure 7.14: Example 2: Pareto sets (circles, left) and Pareto fronts (larger circles with black edges, right) for RBDO utilizing a quasi-Monte Carlo simulation (MC), Subset simulation (SS), and Scaled sigma sampling (SSS) together with an analytical model. The smaller circles with green edges in the objective space represent the recalculated Pareto sets to Pareto fronts with quasi-Monte Carlo simulations with CoV lesser than 5%. Orange, yellow, green and grey crosses represent results published in [109], [32], [59], and [32]. Aqua dots sets show the superior Pareto set and superior Pareto front. Every colour means a different run. The contour plot represents the limit state function.

orange contour represents the limit state dividing the space into the safe and the failure region; the safe region is inside the orange island, moreover, if the bounds of the problem are large, and this can happen for several sampling techniques such as an Asymptotic Sampling, a Scale sigma sampling, the failure region will turn into the valley, and the safe region appear the soonest with the increasing variable X_2. The bounds of the image are nevertheless set to the bounds of the design variables μ_{X_1} and μ_{X_2}.

The datasets in the objective space obtained with MO-RBDO utilizing an Asymptotic sampling have visually very nice correspondence among the obtained Pareto fronts, the recalculated fronts and the superior Pareto front. The Pareto fronts obtained with MO-RBDO utilizing an Enhanced Monte Carlo simulation have very nice similarity between the recalculated fronts and the superior Pareto front. However, this methodology is slightly more sensitive to the bounding of the reliability index from the above to the value 4.5, which explains the red individual with the largest value of the cost function and the β-index close to 4.5. This individual is indifferent with the other individuals from its Pareto front. MO-RBDO utilizing a First-order reliability method provides Pareto fronts that are very inaccurate; one limit state function is highly nonlinear, and the approximations of the reliability indices show large errors. Unfortunately, the Pareto fronts obtained via MO-RBDO utilizing an Importance sampling method suffer from the same problems; the problematic part is the extension of these methodologies into the series system reliability. The Pareto fronts from the MO-RBDO utilizing a quasi-Monte Carlo simulation have nice congruence between the recalculated fronts and the superior Pareto front below the reliability index equal to 4. The reason is that a quasi-Monte Carlo simulation with 30,000 samples cannot capture larger reliability indices than 4; we would increase the number of samples to have more precise reliability indices, but this would appear in the rise of the total number of the limit state function evaluations for each individual in each generation. MO-RBDO utilizing a Subset simulation provides visually very good Pareto fronts, only one individual suffered from the bounding of the Pareto front from above (the teal individual with the cost function value greater than 8). The Pareto fronts from MO-RBDO utilizing a Scale sigma sampling have significantly overestimated the reliability indices, which is unfortunately on the unsafe side of the designing. This method also suffers from the bounding of the Pareto fronts from above.

Table 7.12 provides mean values of the performance measures and errors. The Hypervolume performance measure has an average value around 33; the minimum value was obtained from the Pareto front obtained by MO-RBDO utilizing FORM; this value is an outlier for this methodology. According to this metric, the best-recalculated Pareto fronts were obtained by MO-RBDO utilizing an Enhanced Monte Carlo simulation, closely followed by an Asymptotic Sampling and a Subset simulation. All the recalculated Pareto fronts suffer from the pairwise layout as in the previous example, Section 7.1.2. The Spacing metric provides very low values which refer to the almost ideal spread, while the Spread metric has the average value around 0.55 with the minimum 0.35 (by MO-RBDO utilizing eMC) and the maximum above 1 (by MO-RBDO utilizing FORM). The pairwise layout is partially visible from Figure 7.13 and 7.14 - small circles with green edges in the objective space. Note that the recalculated datasets from mentioned figures are not necessary Pareto fronts since the Pareto efficiency conditions are not used. Recalculated Pareto fronts from MO-RBDO utilizing an Asymptotic sampling, and Enhanced Monte Carlo simulation, and a Subset simulation have the best Spread according to the Δ metric in comparison with other recalculated Pareto fronts. The worst Spread is for the recalculated Pareto fronts obtained via MO-RBDO utilizing FORM followed by MO-RBDO utilizing an Importance sampling.

Low Generational distance values show that the recalculated Pareto fronts are close to the superior Pareto front. The Two set coverage metric reveals that the superior Pareto front weakly

	AS	eMC	FORM	IS	MC	SS	SSS	min	max
HV(rPF)	35.3	35.6	30.0	30.5	31.1	35.1	34.2	10.0	36.1
S(rPF)	0.0063	0.0073	0.0071	0.0078	0.0057	0.0065	0.0058	0.0019	0.014
Δ(rPF)	0.45	0.45	0.83	0.70	0.51	0.46	0.45	0.35	1.10
GD(sPF,rPF) $[\cdot 10^{-4}]$	2.7	2.4	5.7	4.3	2.4	2.5	3.6	1.8	17.1
C(sPF,rPF)	0.54	0.44	0.55	0.50	0.46	0.49	0.66	0.23	0.86
C(rPF,sPF)	0.028	0.032	0.012	0.023	0.026	0.027	0.018	0	0.052
D(sPF,rPF)	1.2	0.9	6.5	5.9	5.4	1.3	2.2	0.4	26.5
min $\epsilon(\beta_{\mathrm{oPF}})$	0.0005	0.0005	0.044	0.0001	$3 \cdot 10^{-5}$	0.0002	0.24	$3 \cdot 10^{-5}$	0.24
max $\epsilon(\beta_{\mathrm{oPF}})$	0.12	0.31	1.92	1.36	0.26	0.138	0.57	0.11	1.9
E $\epsilon(\beta_{\mathrm{oPF}})$	0.024	0.073	0.33	0.25	0.028	0.023	0.43	0.023	0.43
std $\epsilon(\beta_{\mathrm{oPF}})$	0.027	0.075	0.450	0.390	0.053	0.028	0.079	0.027	0.45

Table 7.12: Example 2: Comparison of performance measures for RBDO utilizing different reliability assessment techniques and an analytical model, namely an Asymptotic sampling (AS), Enhanced Monte Carlo simulation (eMC), First-order reliability method (FORM), Importance sampling (IS), quasi-Monte Carlo simulation (MC), Subset simulation (SS), and Scaled sigma sampling (SSS). Used metrics are a Hypervolume (HV), Spacing (S), Spread (Δ), Generational distance (GD), Two set coverage metric (C(A,B)), Coverage difference of two sets (D(A,B)), and Error of the reliability index $\epsilon(\beta)$. The datasets: rPF - recalculated Pareto front, sPF - superior Pareto front, oPF - original Pareto front, and rF - recalculated front. The best values are tinged with the green colour, the worst values with the red colour. The scale shows the visual comparison of the results.

dominates around half of the individuals from recalculated Pareto fronts, while the recalculated Pareto fronts weakly dominate almost none of the individuals from the superior Pareto front. In comparison with the whole hypervolume, the weakly dominated hypervolume is also very small. Almost all of the reliability methods can provide very precise and accurate reliability indices at least on the part of the Pareto front; the exception is a Scaled sigma sampling. On the other hand, the error can be very large for a First-order reliability method and an Importance sampling. A Subset simulation closely followed by an Asymptotic sampling, and a quasi-Monte Carlo simulation provide accurate and precise results. A Scaled sigma sampling produces reliability indices with low accuracy and high precision. A First-order reliability method and Importance sampling are imprecise and inaccurate. Figure 12.5 in Appendix 12 shows the box-plots for the performance measures and the mean and standard deviation of the error.

7.2.3 Results for approximation using meta-models and reliability assessment techniques

As the last part of our experiments, we combined all meta-models and three selected simulation techniques that behaved the best according to the error indicators, namely the mean and the standard deviation of the error. In overall, an Importance sampling, Subset simulation, and Asymptotic sampling were the best three methods. We are aware that the quasi-Monte Carlo simulation and the Enhanced Monte Carlo simulation provided better results for this particular example; however, we wanted to show coherent results through all examples. We discuss only

	GMM			SGMM		
	AS	IS	SS	AS	IS	SS
elapsed time [sec]	393	1183	224	444	1155	237
elapsed time [hours]	0.109	0.329	0.062	0.123	0.321	0.066
initial DoE	50	50	50	50	50	50
added samples	370	733	297	383	799	301
analytical g(x)-calls	420	783	347	433	849	351
MM built for opt.	31	2×31	31	31	2×31	31
MM-calls	$6.23\cdot 10^7$	$1.18\cdot 10^8$	$4.25\cdot 10^7$	$6.19\cdot 10^7$	$1.18\cdot 10^8$	$4.30\cdot 10^7$

Table 7.13: Example 2: Comparison of statistics for RBDO utilizing an Asymptotic sampling, Importance sampling, and Subset simulation with a global meta-model (GMM), and sparse global meta-model (SGMM).

two best-behaving meta-models in this section, namely a global meta-model, and sparse global meta-model. The results for the local meta-model are given in Appendix 12 without further comments. The ranks of all methods, together with the best placings for all five examples, are in Table 7.1; the second example is in the second row of the table.

Each combination of an Asymptotic sampling, Importance sampling, and Subset simulation together with dense and sparse global meta-models in MO-RBDO was run ten times to have valid data for our statistics. We run all the evaluations on the laptop computer with hardware and software specifications defined in Table 11.2. All the simulations were modelled in MATLAB; we use serial MATLAB codes without any inner parallelization. Table 7.13 shows some statistics. Our fastest implementation is MO-RBDO utilizing a Subset simulation and a global meta-model followed closely by a Subset simulation and a sparse global meta-model; one simulation was running approximately 4 minutes. Our slowest implementation is MO-RBDO utilizing an Importance sampling with both types of meta-models; one simulation was running approximately 20 minutes. The number of samples in an initial DoE is equal to 50 for both types of meta-models. The number of samples added into DoE for an update differs among methods, especially for the Importance sampling. The Importance sampling runs for each limit state function separately and the reliability indices together with the correlation matrix for the linearized limit state functions provide the final series system reliability index [139]. Therefore, we assembled two different meta-models since the system limit state function is serial employment of two-component limit state functions.

Each update is two-criterion if the example uses only a component limit state function; the first criterion is space-filling; the second is the shortest distance to the limit state. We had two choices of the updating procedure if several meta-models are used at once. The first choice is a *parallel update* having as many criteria for an update, as many component limit state functions are available plus one criterion for the space-filling needs. The sample from the Importance sampling holds information about only one limit state function; the rest of the limit state function meta-models has to be evaluated together with the space-filling metric. The risk of this method is that the computational effort arises since each of the meta-models has to be evaluated for each sample and not only one meta-model for one sample. If the example uses more than two limit state functions, selecting points to the Pareto front may be more difficult, see the paragraph on many-objective optimization in Chapter 3. The second choice is a *serial* *update*; DoE is updated sequentially as many times as many component limit state functions the system has. Therefore, the update is two-criterion; the first criterion is a space-filling metric; the second criterion is a distance to the limit state only for the actual limit state function. The risk of this method is that we will have more samples for DoE update and therefore, more enumerations of the limit state functions would be necessary. We selected the latter updating methodology, the minimization of computational effort in the meta-model field at the cost of the larger computational efforts in analytical function evaluations. We can afford this reduction since our analytical function is easy and fast to evaluate. In the opposite case, the former method would be more suitable.

We used global meta-models in dense and sparse versions; therefore, 31 meta-models were assembled in case of an Asymptotic sampling and a Subset simulation, and 62 meta-models were assembled for an Importance sampling. The setting of all simulation methods is described at the beginning of this chapter resulting in a different number of meta-model simulations across the simulation methods. The maximum number of meta-model evaluations was performed for MO-RBDO utilizing an Importance sampling regardless of the meta-model type. The lowest number of samples was used for MO-RBDO utilizing a Subset simulation differing slightly in the meta-model type; the minimum number of evaluations was for MO-RBDO utilizing SS and a global meta-model.

Figure 7.15 and 7.16 show the design space and objective space of MO-RBDO utilizing three simulation methods and two meta-models, the former is devoted to results with a global meta-model, the latter to results with a sparse global meta-model. An Asymptotic sampling and a Subset simulation provide very nice results for MO-RBDO regardless of the meta-model type. On the contrary, an Importance sampling works nicely only up to reliability indices equal to 2.3, where LEE and JUNG in [109] found the single-objective optimum (Table 7.8 – optimum with the highest reliability index). Our MO-RBDO utilizing Importance sampling is quite unstable with higher reliability indices. This trend is also visible in Figure 7.13; the instability is evident from the same reliability index value.

Performance measures and error indicators evaluate the quality of the MO-RBDO outputs in Table 7.14. According to unary metrics, the worst simulation method is an Importance sampling. An Asymptotic sampling, in combination with a global meta-model, performs slightly better than other combinations except for an Importance sampling. The results of binary metrics are ambiguous, a Generational distance provides the best results for MO-RBDO utilizing a Subset simulation and a sparse global meta-model, closely followed by MO-RBDO using an Asymptotic sampling and a global meta-model. The other combinations except for an Importance sampling also perform well, therefore recalculated Pareto fronts are close to the superior Pareto front. Even the Importance sampling results are in the same order of magnitude as the other results. The reason is that most of the solutions in the Pareto front are located in the lower part of the front where the error is minimal; we also scaled all the Pareto fronts to intervals $(0, 1)$ to have approximately the same weights on both criteria. The smallest portion of solutions in the superior Pareto front, which weakly dominates solutions in the recalculated Pareto fronts, provides MO-RBDO utilizing an Importance sampling and a sparse global meta-model followed closely by MO-RBDO utilizing SS and SGMM and MO-RBDO using IS and GMM. The largest portion of weakly dominated solutions of superior Pareto front by recalculated Pareto fronts are from the same three methodologies; however, this size of data is insignificant.

The error indicators show that the Importance sampling can provide at least one best result according to the average minimum of the error. As being said, the lower part of the Pareto front is very precise, and the problems occur from the reliability index greater than approximately 3. For that reason, this method has also the largest maximum error. The most accurate and precise

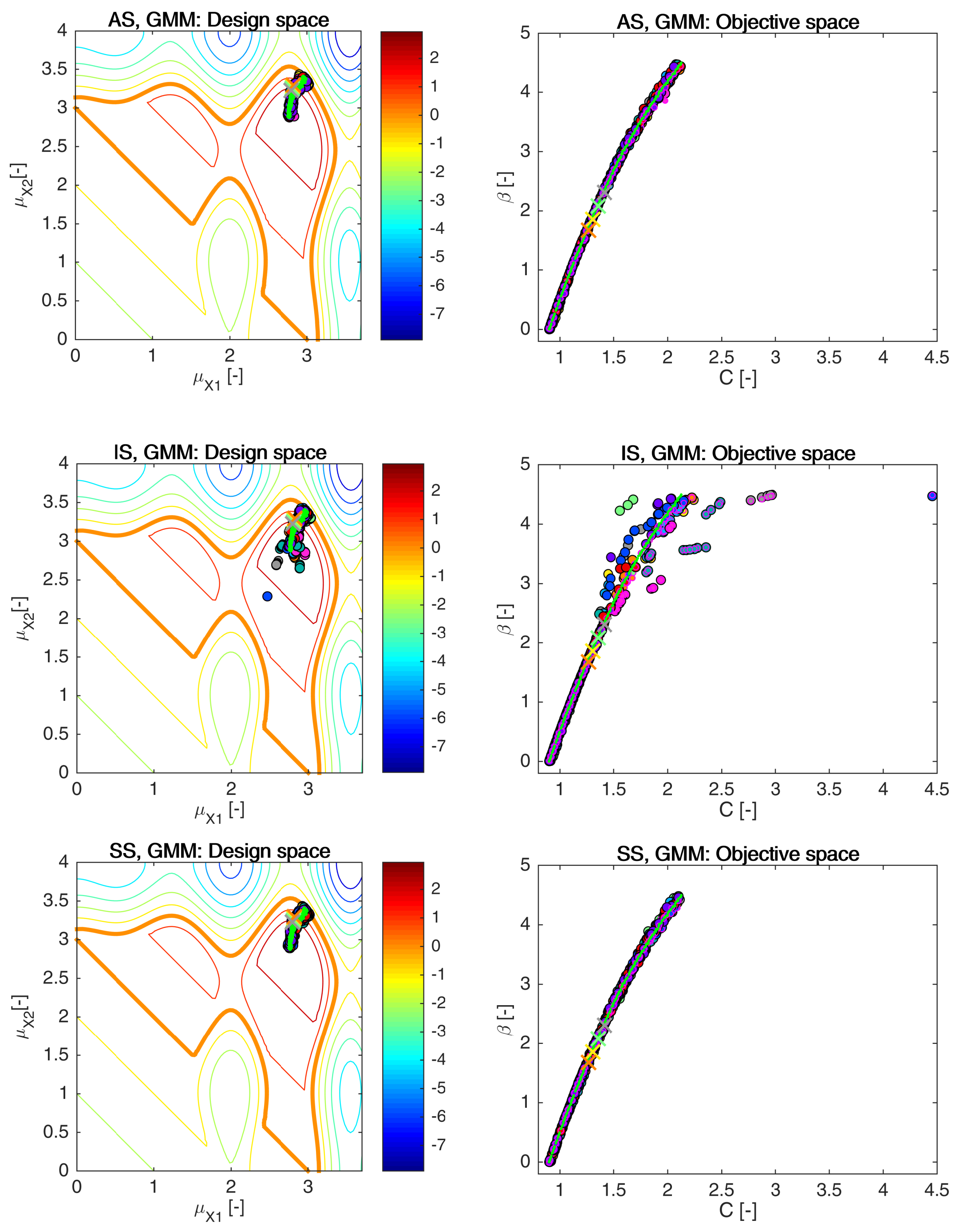


Figure 7.15: Example 2: Pareto sets (circles, left) and Pareto fronts (larger circles with black edges, right) for RBDO utilizing an Asymptotic sampling (AS), Importance sampling (IS), and Subset Simulation (SS) together with a global meta-model (GMM). The smaller circles with magenta edges in the objective space represent the recalculated Pareto sets to Pareto fronts with quasi-Monte Carlo simulations with $\mathrm{CoV} < 5\%$. Orange, yellow, green and grey crosses represent results published in [109], [32], [59], and [32]. Green dots sets show the superior Pareto set and Pareto front obtained with an analytical model together with a quasi-Monte Carlo simulation. Every colour means a different run.

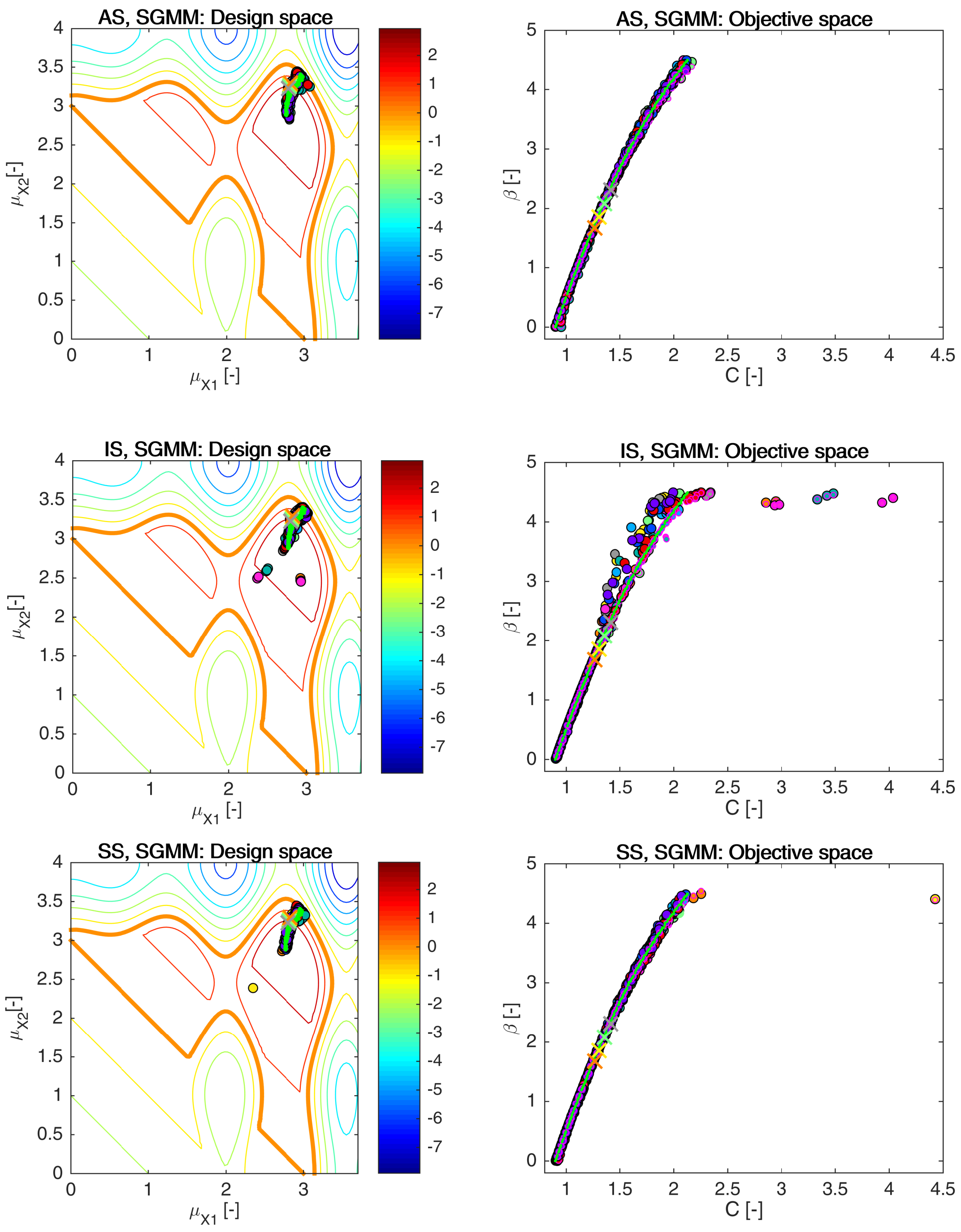


Figure 7.16: Example 2: Pareto sets (circles, left) and Pareto fronts (larger circles with black edges, right) for RBDO utilizing an Asymptotic sampling (AS), Importance sampling (IS), and Subset Simulation (SS) together with a sparse global meta-model (SGMM). The smaller circles with magenta edges in the objective space represent the recalculated Pareto sets to Pareto fronts with quasi-Monte Carlo simulations with $\mathrm{CoV} < 5\%$. Orange, yellow, green and grey crosses represent results published in [109], [32], [59], and [32]. Green dots sets show the superior Pareto set and Pareto front obtained with an analytical model together with a quasi-Monte Carlo simulation. Every colour means a different run.

	GMM			SGMM				
	AS	IS	SS	AS	IS	SS	min	max
HV (rPF)	30.2	26.9	29.8	29.9	27.7	29.5	22.8	30.6
S (rPF)	0.0074	0.0086	0.0082	0.0077	0.0125	0.0077	0.0048	0.0295
Δ (rPF)	0.41	0.72	0.45	0.45	0.73	0.45	0.33	0.85
GD (rPF,sPF)	$2.7{\cdot}10^{-4}$	$5.8{\cdot}10^{-4}$	$2.9{\cdot}10^{-4}$	$3.1{\cdot}10^{-4}$	$3.0{\cdot}10^{-4}$	$2.6{\cdot}10^{-4}$	$2.1{\cdot}10^{-4}$	$1.56{\cdot}10^{-3}$
C(sPF,rPF)	0.50	0.43	0.51	0.58	0.38	0.42	0.24	0.76
C(rPF,sPF)	0.023	0.027	0.025	0.016	0.027	0.028	0	0.056
D(sPF,rPF)	0.78	4.05	1.12	1.05	3.22	1.45	0.38	8.13
$\min \epsilon(\beta_{\text{oPF}})$	$2.{\cdot}10^{-4}$	$5{\cdot}10^{-5}$	$1.1{\cdot}10^{-4}$	$2.6{\cdot}10^{-4}$	$6.5{\cdot}10^{-5}$	$1.4{\cdot}10^{-4}$	0	$8.6{\cdot}10^{-4}$
$\max \epsilon(\beta_{\text{oPF}})$	0.12	0.57	0.13	0.14	0.64	0.12	0.01	1.47
$\text{E}\epsilon(\beta_{\text{oPF}})$	0.023	0.041	0.024	0.026	0.065	0.023	0.004	0.121
$\text{std}\epsilon(\beta_{\text{oPF}})$	0.027	0.116	0.028	0.030	0.148	0.027	0.003	0.320

Table 7.14: Example 2: Comparison of performance measures for RBDO utilizing different reliability assessment techniques namely an Asymptotic sampling (AS), Importance sampling (IS), and Subset simulation (SS) using a global (GMM) and sparse global meta-model (SGMM). Used metrics are a Hypervolume (HV), Spacing (S), Spread (Δ), Generational distance (GD), Two set coverage metric (C(A,B)), Coverage difference of two sets (D(A,B)), and Error of the reliability index $\epsilon(\beta)$. The used datasets: rPF - recalculated Pareto front, sPF - superior Pareto front, oPF - original Pareto front, and rF - recalculated front. The best values are tinged with the green colour, the worst values with the red colour. The scale shows the visual comparison of the results.

results provide MO-RBDO utilizing an Asymptotic sampling and global meta-model and MO-RBDO using a Subset simulation and sparse global meta-model. On the other side, MO-RBDO utilizing an Importance sampling with both models had the least accurate and precise reliability indices; a global meta-model is slightly better. Figure 12.6 in Appendix 12 shows the box-plots for the performance measures and the mean and standard deviation of the error.

7.3 Example 3: Mathematical problem with a series system containing three limit states

This problem considers two stochastic design variables X_1 and X_2 contributing to the cost function with their mean values and simultaneously to the limit state function as the stochastic variables. The problem has three component limit state functions, where the components are connected into the series system. Therefore, we use the minimum value from all limit state functions values, and this representative value determines whether the system fails or not for the majority of the reliability assessment methods. The rest of the methods, namely a First-order reliability method, Importance sampling, and Enhanced Monte Carlo simulation, uses three limit state functions separately as described in the theoretical part, Sections 4.3 and 4.4.

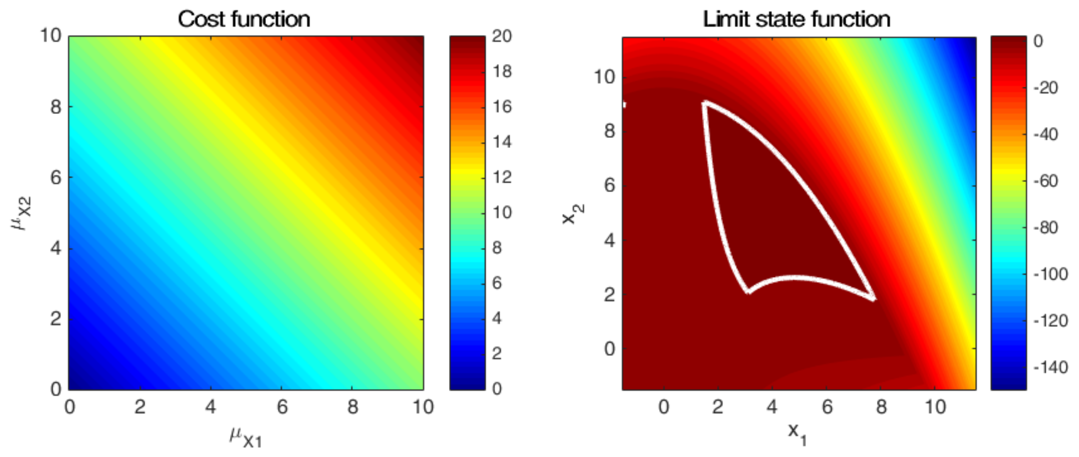


Figure 7.17: Example 3: The cost function (left) and the limit state function (right). The bold white contour (right) is a limit state $g(\mathbf{x}) = 0$; $g(\mathbf{x}) \leq 0$ is valid for the failure region, i.e. the region outside the white island.

Mathematically written

$$\min \quad C(\mu_{X_1}, \mu_{X_2}) = \mu_{X_1} + \mu_{X_2} \tag{7.21}$$

$$\max \quad \beta(\mathbf{X}) = -\Phi^{-1}(\text{Prob}[g(\mathbf{X}) \leq 0]) \tag{7.22}$$

$$g(\mathbf{X}) = \min \begin{pmatrix} \frac{x_1^2 x_2}{20} - 1 \\ \frac{(x_1+x_2-5)^2}{30} + \frac{(x_1-x_2-12)^2}{120} - 1 \\ \frac{80}{x_1^2+8x_2+5} - 1 \end{pmatrix} \tag{7.23}$$

$$0 \leq \mu_{X_{1,2}} \leq 10. \tag{7.24}$$

The third part of the limit state function is reformulated according to [59] to

$$G_3 = \{\mathbf{x} \in \mathbf{X} : g_3(\mathbf{x}) = 80 - (x_1^2 + 8x_2 + 5) \leq 0\} \tag{7.25}$$

due to the possibility of 0 in the denominator. Figure 7.17 depicts cost function as well as the combination of the limit state functions into the series system. Stochastic variable X_1 and X_2 have a normal distribution with the mean value μ_{X_1} and μ_{X_2}, respectively, and both have a standard deviation equal to 0.3. They are statistically independent. Both mean values μ_{X_1} and

Reference	μ_{X_1}	μ_{X_2}	$C(\mu_{X_1}, \mu_{X_2})$ [-]		β [-]	
			published	corrected	published	corrected
[59]	3.46	3.27	6.74	6.73	2.98	2.7762
[174]	3.4406	3.2800	6.7205	6.7206	-	2.7796
[115]	3.441	3.285	6.726	6.726	-	2.7916
[202]: RIA	3.609	3.661	7.270	7.270	2	3.8445
[202]: PMA	3.609	3.660	7.269	7.269	2	3.8449
[55]	3.440	3.287	6.726	6.727	2.9686	2.7951

Table 7.15: Example 3: Published single-objective optima.

μ_{X_2} represent design variables. For practical purposes, we limited the reliability index β to the range $0 \leq \beta \leq 4.5$.

YOUN and CHOI first studied this example as the single-objective problem utilizing a Reliability index approach (RIA) and a Performance measure approach (PMA) in [202]. The single-objective problem was subsequently studied by different authors as well. Table 7.15 summarizes the single-objective optima published in the literature.

7.3.1 Comparison of meta-models and analytical model results together with a preconditioned quasi-Monte Carlo simulation

In this section, we minimize the reliability assessment method error by using a preconditioned quasi-Monte Carlo simulation. Several independent runs with different meta-models, as well as an analytical model, were used to obtain statistical data. Table 7.16 shows average values of some statistics from these runs. The local meta-models were the fastest among the meta-models. We run our code in serial and parallel using eight threads; the best-obtained speed-up was up to 4.72 utilizing sparse global meta-models, followed by speed-up 4.49 utilizing global meta-model, and the slowest speed-up is 2.12 utilizing the local meta-models. The initial DoE was set to 200 samples in local meta-models, and 50 samples to both global meta-models using the sparse as well as the dense Gramm matrix. The total number of evaluated analytical functions in methods utilizing meta-models was from 346 for SGMM to 438 for LMM. 31 global meta-models, or 3050 local meta-models were assembled during the optimization process. Local meta-models need an assembly for each individual in each generation and another assembly for each potential candidate sample to be added into DoE. In total, meta-models were simulated from 6.1 billion times for SGMM to 7.3 billion times for LMM for the reliability assessment evaluation. The analytical model needs a similar number of samples for the reliability assessment, i.e. 7.5 billion samples.

All the independent runs are depicted in Figure 7.18. With only a few exceptions, RBDO utilizing a global meta-model terminated into the Pareto fronts that are almost identical to the superior Pareto front. The problems occur for reliability indices above 4. RBDO utilizing local meta-models was fast, and the results are excellent. Except for one individual from one RBDO

	AM	LMM	GMM	SGMM
elapsed time [hours] (1 thread)	1.25	6.50	24.81	22.82
elapsed time [hours] (8 threads)	0.69	2.58	5.40	4.56
initial DoE	0	200	50	50
added samples	0	238	300	296
analytical g(x)-calls	$7.48 \cdot 10^9$	438	350	346
MM built for opt.	0	3050	31	31
MM built for update	0	2015	0	0
MM-calls	0	$7.29 \cdot 10^9$	$7.00 \cdot 10^9$	$6.07 \cdot 10^9$

Table 7.16: Example 3: Comparison of statistics for RBDO utilizing a quasi-Monte Carlo simulation with an analytical model (AM), local meta-models (LMM), global meta-model (GMM), and sparse global meta-model (SGMM).

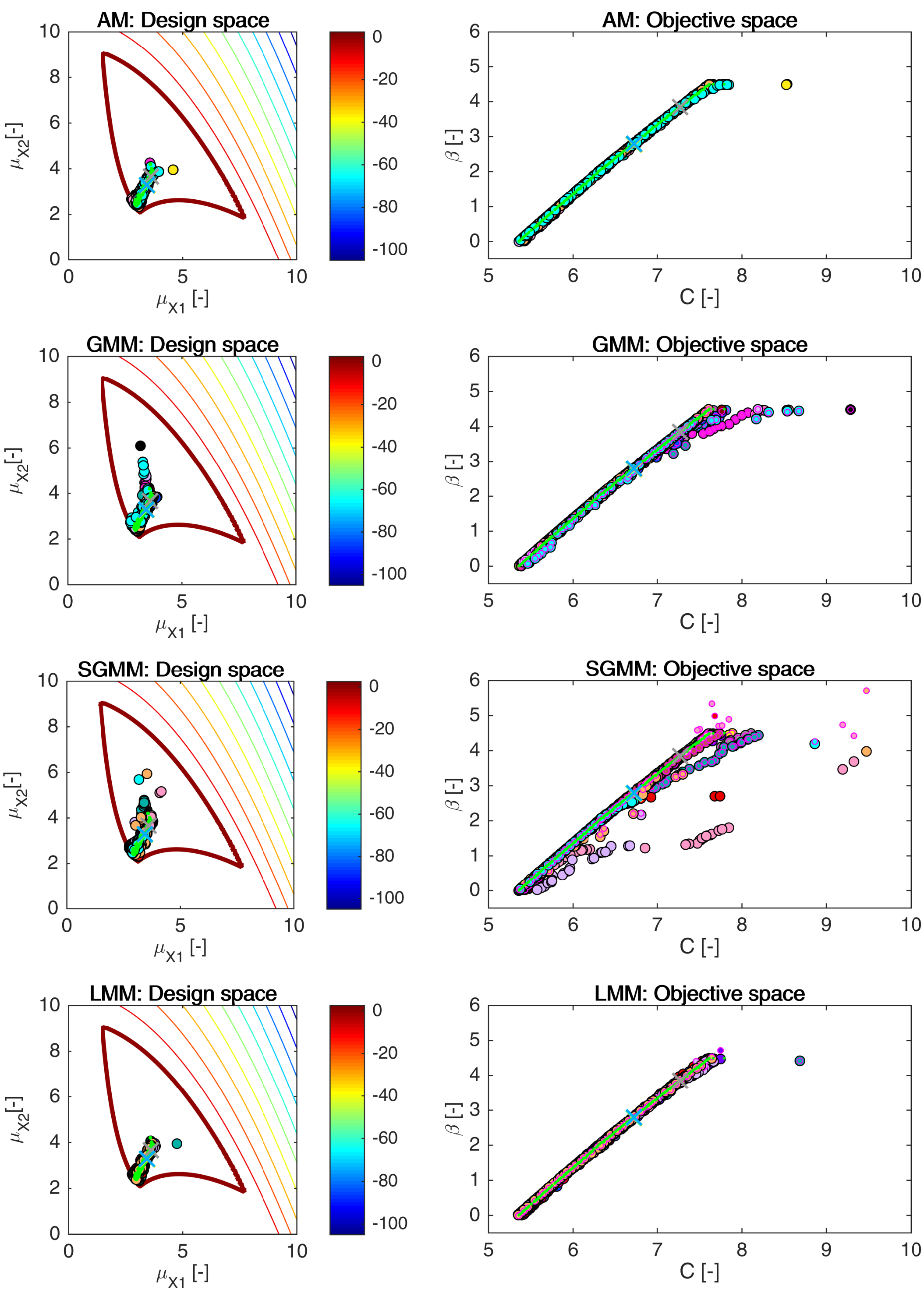


Figure 7.18: Example 3: Pareto sets (circles, left) and Pareto fronts (large circles, right) for all models, namely an analytical model (AM), global meta-model (GMM), sparse global meta-model (SGMM), and local meta-models (LMM). Smaller circles with magenta edges in the objective space represent the recalculated fronts with an analytical model. Crosses represent results published in [59, 174, 115, 202, 55]. Green dots sets show the superior Pareto set and the superior Pareto front. Every colour has the meaning of a different run. The contour plot in the design space represents an analytical limit state function; the bold contour is for the limit state $g(\mathbf{X}) = 0$.

run, the Pareto fronts are almost mutually identical as well as almost identical to the superior Pareto front. RBDO utilizing sparse global meta-models got the worst results visually for some runs. Figure 7.19 shows the whole optimization process of RBDO utilizing SGMM for a run with a poor Pareto front. The whole procedure started very promisingly (first row of pictures), and it is visible that the optimization algorithm reached the results similar to the superior Pareto front in around the 25th generation. Figure 7.21 shows a sparse global meta-model assembled for selected generations in MO-RBDO. The meta-model works fine from the early generations up to 25th generation. After the 25th generation, the meta-model seems to be overfitted. Therefore, this meta-model is too flexible and fitting the noise as well as the actual behaviour of the model. The problematic area is in the plateau (see Figure 7.17) that emerged with the union of three limit state functions, and this plateau is hard to fit. Figure 7.20 shows a good termination of the sparse global meta-model and Figure 7.22 unifies images of the assembled sparse global meta-models for selected generations together with DoE. Even the fifth generation of RBDO in this run uses a good approximation of the meta-model around the limit state. The updating procedure concentrates the new samples (red dots) in the DoE, mainly in the plateau to detect any other potential limit states to distinguish the space into the safe regions and failure regions.

Table 7.17 shows the comparison of performance measures for RBDO utilizing a preconditioned quasi-Monte Carlo simulation together with an analytical model and meta-models. All the data represent mean values for all performed runs. A closer inspection of the column with the sparse global meta-model metric values shows that this model performed worst in total. The Hypervolume measure value is the smallest from all other values for different models on the average. These values are affected by the poor-quality Pareto fronts that terminated below the superior Pareto front; this trend can be seen in Figure 7.18, third row, the second column of images, small circles with magenta edges. Note that not all the individuals from the recalcu-

	AM	GMM	LMM	SGMM	min	max
HV	18.06	17.73	18.03	17.22	12.84	19.48
S	0.0079	0.0084	0.0084	0.0141	0.0051	0.0674
Δ	0.39	0.43	0.39	0.53	0.29	1.06
GD	-	0.00087	0.00049	0.00131	0.00029	0.00546
C(PF_{AM},PF_{MM})	-	0.56	0.47	0.56	0.22	0.82
C(PF_{MM},PF_{AM})	-	0.013	0.027	0.017	0	0.074
D(PF_{AM},PF_{MM})	-	0.52	0.25	1.20	0.13	5.40
$\min \epsilon(\beta_{oPF})$	-	0.0001	0.0004	0.0026	0	0.0414
$\max \epsilon(\beta_{oPF})$	-	0.027	0.094	0.542	0.018	3.720
E $\epsilon(\beta_{oPF})$	-	0.007	0.020	0.124	0.006	1.162
std $\epsilon(\beta_{oPF})$	-	0.006	0.019	0.135	0.005	1.306

Table 7.17: Example 3: Comparison of performance measures for RBDO utilizing a quasi-Monte Carlo simulation with an analytical model (AM), local meta-model (LMM), global meta-model (GMM), and sparse global meta-model (SGMM). Used metrics are a Hypervolume (HV), Spacing (S), Spread (Δ), Generational distance (GD), Two set coverage metric (C(A,B)), Coverage difference of two sets (D(A,B)), and Error of the reliability index ($\epsilon(\beta)$). The best values are tinged with the green colour, the worst values with the red colour. The scale shows the visual comparison of the results.

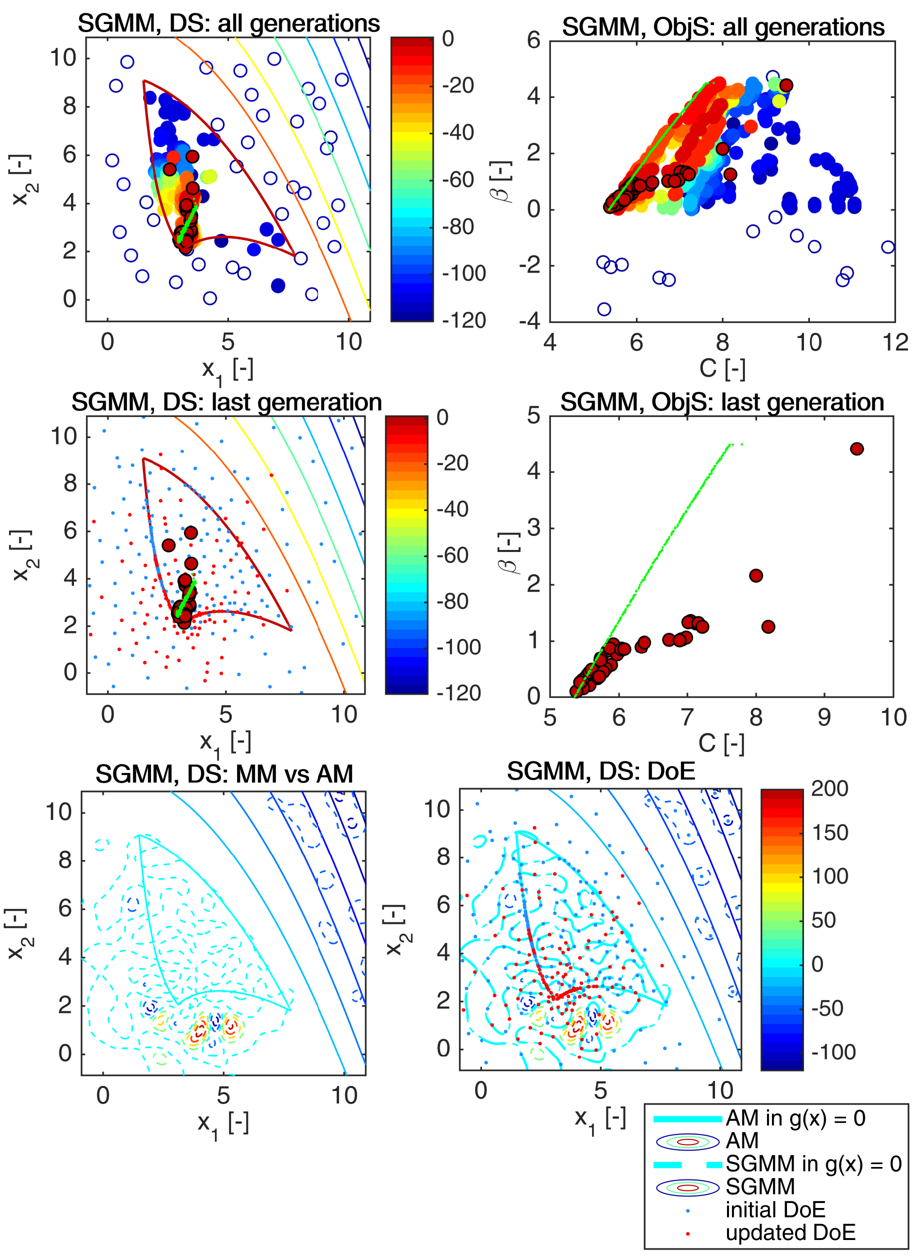


Figure 7.19: Example 3: Pareto sets and Pareto fronts from one MO-RBDO run utilizing a sparse global meta-model with poor results. Abbreviations: DS - design space, ObjS - objective space, SGMM - sparse global meta-model, AM - analytical model, MM - meta-model, DoE - Design of Experiments. The first row of pictures represents all generations from the optimization algorithm (circles). The colour scale of circles show the number of generation in the MO-RBDO procedure; the corresponding colour-bar is in the first-row second-column image. The last generation (a Pareto set on the left and a Pareto front on the right) is depicted in the second row of pictures together with DoE (dots). Set of green dots represents a superior Pareto set and Pareto front obtained by an analytical model. The last row of images represents a sparse global meta-model assembled for the last generation of NSGA-II (dashed contours) and the analytical model (solid contours) together with the initial (blue dots) and updated DoE (red dots). The bold contour on the right image highlights the limit state.

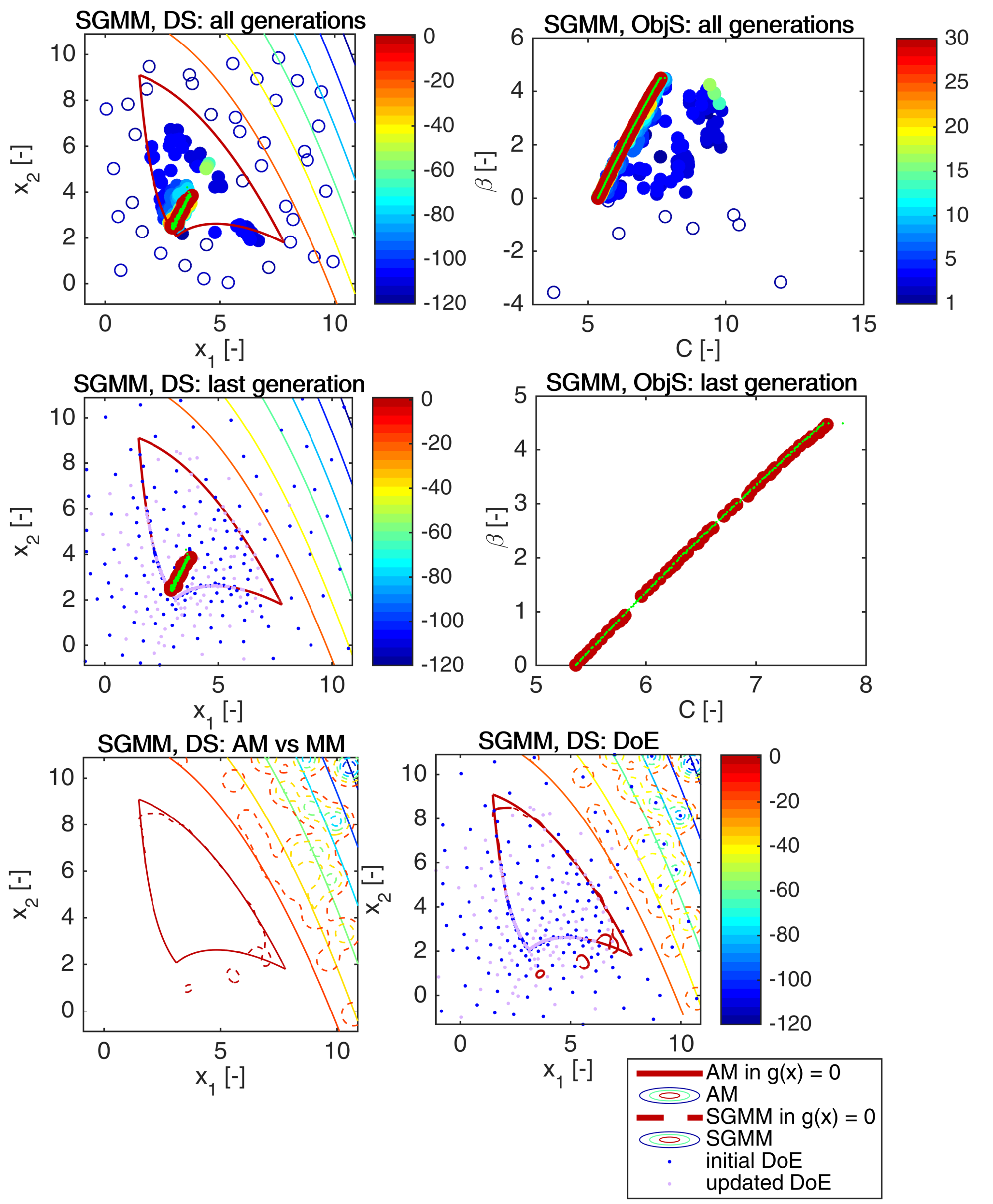


Figure 7.20: Example 3: Pareto sets and Pareto fronts from one MO-RBDO run utilizing a sparse global meta-model with good results. Abbreviations: DS - design space, ObjS - objective space, SGMM - sparse global meta-model, AM - analytical model, MM - meta-model, DoE - Design of Experiments. The first row of pictures represents all generations from the optimization algorithm (circles). The colour scale of circles show the number of generation in the MO-RBDO procedure; the corresponding colour-bar is in the first-row second-column image. The last generation (a Pareto set on the left and a Pareto front on the right) is depicted in the second row of pictures together with DoE (dots). Set of green dots represents a superior Pareto set and Pareto front obtained by an analytical model. The last row of images represents a sparse global meta-model assembled for the last generation of NSGA-II (dashed contours) and the analytical model (solid contours) together with the initial (blue dots) and the updated DoE (violet dots).

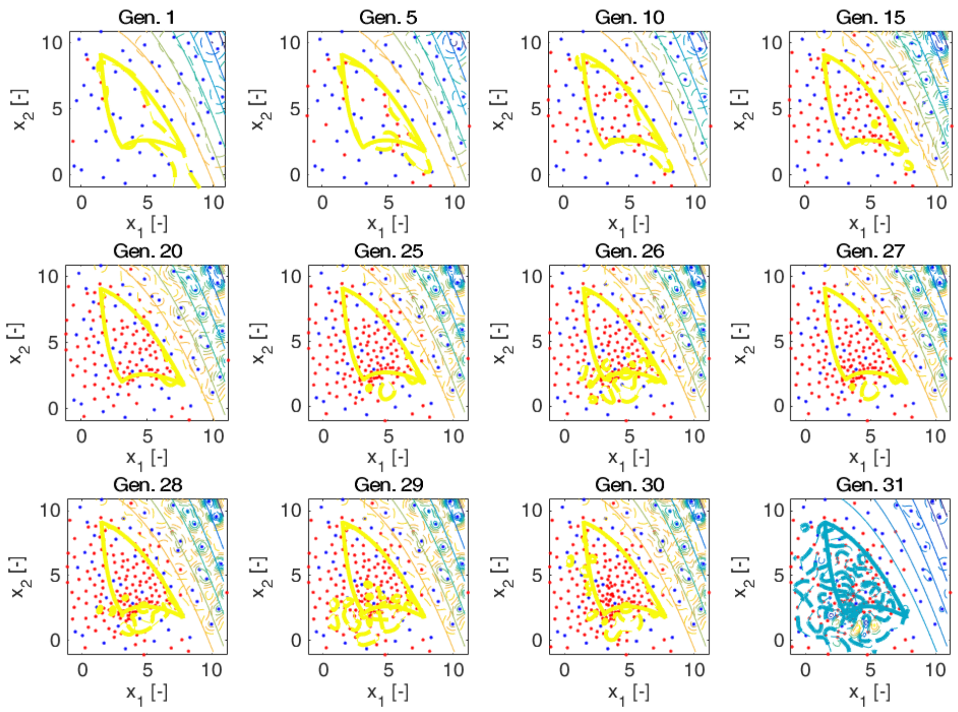


Figure 7.21: Example 3: The sparse global meta-model (dashed lines) compared with the analytical model (solid lines) depicted for selected generations of MO-RBDO run for an optimization run shown in Figure 7.19. Blue dots represent the initial DoE; red dots are for the updated DoE.

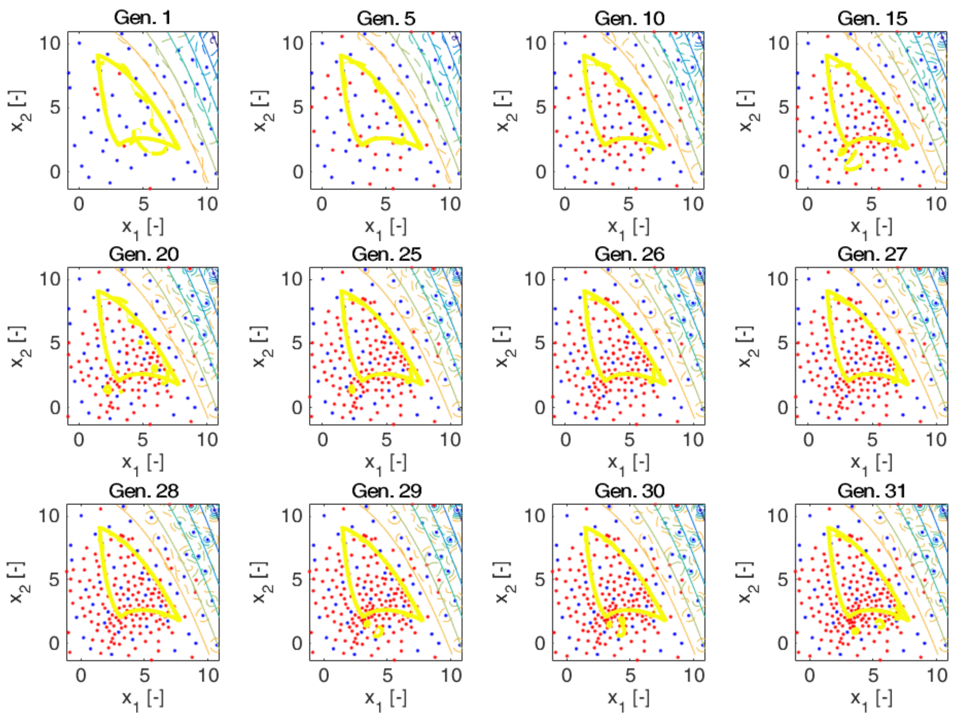


Figure 7.22: Example 3: The sparse global meta-model (dashed lines) compared with the analytical model (solid lines) depicted for selected generations of MO-RBDO run represented in Figure 7.20. Blue dots symbolize an initial DoE; red dots are for an updated DoE.

lated fronts (small circles with magenta edges) emerge in the final recalculated Pareto fronts. The Spread and the Spacing measures also have the worst results for SGMM, which means that the individuals in the Pareto fronts are poorly distributed creating clusters of individuals. The General distance metric shows that the Pareto fronts from RBDO utilizing SGMM are the farthest from the superior Pareto front. The Two set coverage metric has not as bad results for SGMM as other metrics; this value can be comparable with the metric value for the global meta-model. The Two set coverage metric value can be influenced by the large scale of the obtained data and by the outlying data, the median of the Two set coverage metric for RBDO utilizing SGMM is again the highest from all values of this metric, which is evident from Figure 12.7 in Appendix 12. The space that is weakly dominated by the superior Pareto front but not weakly dominated by the recalculated Pareto front from RBDO utilizing meta-models D($\mathrm{PF_{AM}}$,$\mathrm{PF_{MM}}$) has the worst results for SGMM again. MO-RBDO using the sparse global meta-model shows the largest error even in all observed error indicators, it provides the least accurate and precise reliability indices. On the contrary, it is apparent from this table that the local meta-models performed the best among all used meta-models. The Hypervolume metric value is very close to the value for RBDO utilizing an analytical model. The Spacing metric value is comparable with the metric value for RBDO using AM and the same for the RBDO utilizing GMM. The Spread of the Pareto fronts is also comparable with RBDO using AM. The binary metrics provide the best results for RBDO utilizing LMM as well. The Pareto fronts from RBDO using LMM are closest to the superior Pareto front, the weakly dominated individuals, as well as hyperspace, is the smallest among RBDO using any meta-model. Surprisingly, the global meta-model provides the most precise and accurate data for the reliability index prediction; however, the RBDO utilizing LMM still provides very nice results and high-quality Pareto fronts with the best elapsed times of computations. Figure 12.7 in Appendix 12 shows the box-plots for the performance measures and the mean and standard deviation of the error.

7.3.2 Results comparison of different reliability assessment techniques utilizing an analytical model

We run our multi-objective reliability-based design optimization algorithm with the analytical model only but using different reliability assessment methodologies described in Sections 4.3 and 4.4. All MO-RBDO methodologies were run ten times for each reliability assessment method to have enough statistical data; Table 7.18 shows the number of analytical function simulations together with elapsed time of the serial code. The setting of the reliability methods is described at the beginning of this chapter. If we compare the number of limit state function

	AS	eMC	FORM	IS	MC	SS	SSS
elapsed time [s]	85	106	77	115	13	36	26
g(x)-calls	$5.7 \cdot 10^7$	$4.7 \cdot 10^7$	$9.9 \cdot 10^4$	$4.5 \cdot 10^7$	$4.7 \cdot 10^7$	$3.9 \cdot 10^7$	$4.1 \cdot 10^7$

Table 7.18: Example 3: Comparison of statistics for RBDO utilizing an analytical model and different reliability assessment techniques, namely Asymptotic sampling (AS), Enhanced Monte Carlo simulation (eMC), First-order reliability method (FORM), Importance sampling (IS), quasi-Monte Carlo simulation (MC), Subset simulation (SS), and Scaled sigma sampling (SSS).

evaluations in the previous section (RBDO combining a preconditioned quasi-MC with metamodels) with the number of limit state function evaluations obtained by simulation techniques, the difference is lower by two orders of magnitude. MO-RBDO utilizing a First-order reliability method is even lower by five orders of magnitude in comparison with the previous section. Our fastest implementation of MO-RBDO is with quasi-Monte Carlo simulation, followed by a Scaled sigma sampling and a Subset simulation. The slowest implementation is MO-RBDO utilizing an Importance sampling, followed by an Enhanced Monte Carlo simulation.

Figure 7.23 and 7.24 depict all the independent runs of MO-RBDO utilizing different reliability assessment techniques. With few little exceptions, all the Pareto fronts are almost identical to the recalculated front data, and the superior Pareto front is close to these data as well. Small perturbations are at both tails of the Pareto fronts in MO-RBDO utilizing an Asymptotic sampling. An Enhanced Monte Carlo simulation has difficulties in capturing correctly larger reliability indices than approximately four. The Pareto fronts from MO-RBDO utilizing a First-order reliability method suffer from the bounding of the reliability index to the 4.5 value as well as the Pareto fronts from MO-RBDO utilizing an Importance sampling. A quasi-Monte Carlo simulation in MO-RBDO utilizing q-MC would need more samples than 30,000 to evaluate the upper tail of the Pareto fronts correctly, therefore, the method itself bounded the Pareto front from above artificially at value 4. The Pareto fronts obtained via MO-RBDO utilizing a Subset simulation have small perturbations on the upper tails. The Pareto fronts obtained via MO-RBDO utilizing a Scaled sigma sampling have visible perturbation on both tails as well as suffering from the bounding of the reliability index from above.

	AS	eMC	FORM	IS	MC	SS	SSS	min	max
HV(rPF)	12.9	13.4	13.1	13.1	11.9	12.7	12.9	11.7	13.9
S(rPF)	0.0091	0.01	0.01	0.0093	0.0087	0.0105	0.0089	0.0059	0.0147
Δ(rPF)	0.43	0.43	0.45	0.41	0.52	0.48	0.42	0.31	0.61
GD(SPF,rPF) $[\cdot 10^{-4}]$	5.8	12.7	8.3	7.0	5.0	5.0	9.3	3.0	28.1
C(sPF,rPF)	0.49	0.48	0.49	0.46	0.49	0.52	0.81	0.26	0.90
C(rPF,sPF)	0.028	0.057	0.027	0.031	0.020	0.018	0.004	0	0.093
D(sPF,rPF)	0.32	0.16	0.15	0.15	1.31	0.53	0.37	0.12	1.50
min $\epsilon(\beta_{\text{oPF}})$	0.0002	0.014	0.001	0.0001	0.0001	0.0004	0.003	0	0.035
max $\epsilon(\beta_{\text{oPF}})$	0.11	0.48	0.18	0.13	0.27	0.22	0.19	0.03	1.12
E $\epsilon(\beta_{\text{oPF}})$	0.020	0.094	0.036	0.024	0.025	0.029	0.061	0.005	0.12
std $\epsilon(\beta_{\text{oPF}})$	0.023	0.081	0.050	0.038	0.051	0.044	0.042	0.0055	0.20

Table 7.19: Example 3: Comparison of performance measures for RBDO utilizing different reliability assessment techniques and an analytical model, namely an Asymptotic sampling (AS), Enhanced Monte Carlo simulation (eMC), First-order reliability method (FORM), Importance sampling (IS), quasi-Monte Carlo simulation (MC), Subset simulation (SS), and Scaled sigma sampling (SSS). Used metrics are a Hypervolume (HV), Spacing (S), Spread (Δ), Generational distance (GD), Two set coverage metric (C(A,B)), Coverage difference of two sets (D(A,B)), and Error of the reliability index ($\epsilon(\beta)$). The datasets: rPF - recalculated Pareto front, sPF - superior Pareto front, oPF - original Pareto front, and rF - recalculated front. The best values are tinged with the green colour, the worst values with the red colour. The scale shows the visual comparison of the results.

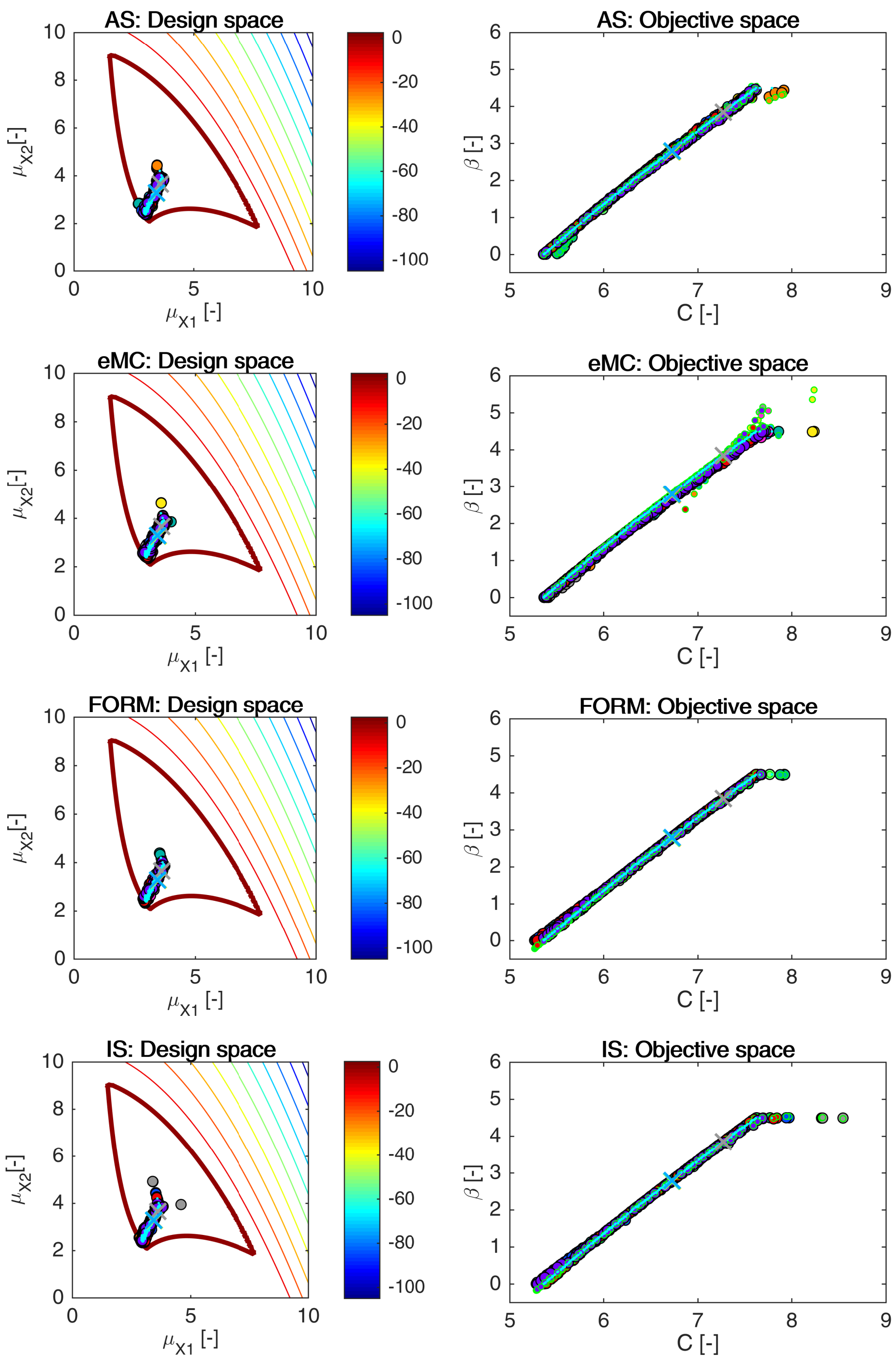


Figure 7.23: Example 3: Pareto sets (circles, left) and Pareto fronts (larger circles with black edges, right) for RBDO utilizing an Asymptotic sampling (AS), Enhanced Monte Carlo simulation (eMC), First-order reliability method (FORM), and Importance sampling (IS) together with an analytical model. The smaller circles with green edges in the objective space represent the recalculated Pareto sets to Pareto fronts with a quasi-Monte Carlo simulation with CoV lesser than 5%. A yellow, green, and orange cross are hidden behind a blue cross, and they represent solutions published in [59], [174], [115], and [55]; the grey cross represents optima published in [202] (RIA as well as PMA methodology). Aqua dots sets show the superior Pareto set and Pareto Pareto front obtained with an analytical model together with a quasi-Monte Carlo simulation. Every colour means a different run. The contours represent the limit state function which corresponds to the colour bar with the limit state highlighted with a thick line.

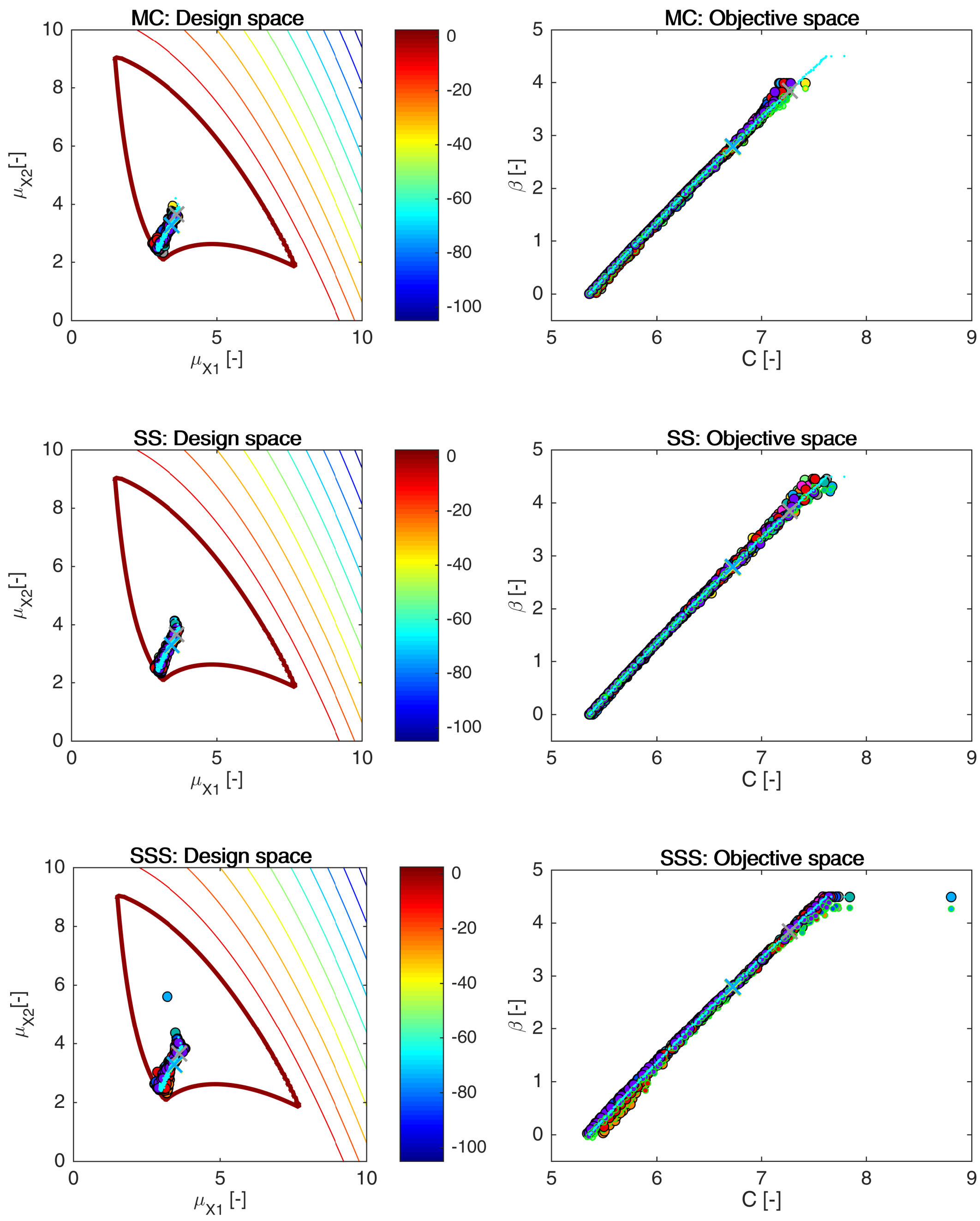


Figure 7.24: Example 3: Pareto sets (circles, left) and Pareto fronts (larger circles with black edges, right) for RBDO utilizing a quasi-Monte Carlo simulation (MC), Subset simulation (SS), and Scaled sigma sampling (SSS) together with an analytical model. The smaller circles with green edges in the objective space represent the recalculated Pareto sets to Pareto fronts with a quasi-Monte Carlo simulation with CoV lesser than 5%. A yellow, green, and orange cross are hidden behind a blue cross, and they represent solutions published in [59], [174], [115], and [55]; the grey cross represents optima published in [202] (RIA as well as PMA methodology). Aqua dots sets show the superior Pareto set and Pareto Pareto front obtained with an analytical model together with a quasi-Monte Carlo simulation. Every colour means a different run. The contours represent the limit state function which corresponds to the colour bar with the limit state highlighted with a thick line.

Table 7.19 shows the performance measures for RBDO utilizing different reliability assessment techniques and an analytical model. The Hypervolume metric has almost identical outputs with only small perturbations. The Spacing performance measure indicates that the individuals in the Pareto fronts are distributed almost ideally, but the Spread performance measure is in modest opposition. The same problem, as in Example 1 and Example 2, is in the pairwise, groupwise, or both, distribution of the individuals, which is unfortunately unrecognizable by this metric. A Generational distance is close to zero, which means that the superior Pareto front is very close to the recalculated Pareto fronts. The superior Pareto front weakly dominates almost half of the individuals in all the recalculated Pareto fronts on average with an exception for the Pareto fronts from MO-RBDO utilizing a Scaled sigma sampling. Oppositely, only a small portion of the solutions in the recalculated Pareto fronts weakly dominates the superior Pareto front. All of the methods can achieve a small value of the minimum of the error at least on the part of the Pareto fronts. The maximum error is also acceptable except for an Enhanced Monte Carlo simulation. The mean value of the error is minimal for the Asymptotic sampling, followed by the Importance sampling and surprisingly a quasi-Monte Carlo simulation. The standard deviation of the error is the smallest again for an Asymptotic sampling, followed by an Importance sampling and a Scaled sigma sampling. The Scaled sigma sampling provides a consistently shifted reliability indices which means that this method provides precise but not accurate responses, while an Enhanced Monte Carlo simulation is the method with the worst mean of the error accompanied by the worst standard deviation of the error. The most precise and accurate method is an Asymptotic sampling; the second-best is an Importance sampling. Figure 12.8 in Appendix 12 shows the box-plots for the performance measures and the mean and standard deviation of the error.

7.3.3 Results for approximation using meta-models and reliability assessment techniques

In the previous sections, we studied the influence of the different meta-models and different reliability assessment methods on the performance and errors separately. The three best-behaving reliability assessment methods, namely an Asymptotic sampling, Importance sampling, and Subset simulation, are combined with all meta-models. We discuss only two best-behaving meta-models in this section, namely local meta-models and global meta-model. The results for the sparse global meta-model are given in Appendix 12 without further comments. The ranks of all methods, together with the best placings for all five examples, are in Table 7.1; the third example is in the third row of the table.

We run all the combinations of our methodology with different meta-models and selected simulation methods ten times to have relevant data for statistics that are summarized in Table 7.20. We used a laptop computer with a software and hardware specification defined in Table 11.2. Our fastest implementation is MO-RBDO utilizing a Subset simulation and local meta-models, followed by the same meta-model and a Subset simulation, and an Asymptotic sampling using local meta-models. In general, the local meta-models have a faster evaluation that a global meta-model if the same simulation method is considered. The number of samples in the initial DoE differs with the type of meta-model used. The number of samples that are added into DoE as an update differs from the used methods. The largest portion of the samples was added if an Importance sampling was used as the reliability method. Since the Importance sampling works with limit state functions separately, we extended the update definition and kept separate meta-models for each function. The update is implemented serial as described in

	GMM			LMM		
	AS	IS	SS	AS	IS	SS
elapsed time [sec]	384	1567	209	215	544	153
elapsed time [hours]	0.107	0.435	0.058	0.060	0.151	0.043
initial DoE	50	50	50	200	200	200
added samples	337	1314	300	218	303	246
analytical g(x)-calls	387	1364	350	418	503	446
MM built for opt.	31	31	31	3050	3050	3050
MM built for update	0	0	0	788	3714	1237
MM-calls	$5.61{\cdot}10^7$	$9.05{\cdot}10^7$	$3.68{\cdot}10^7$	$5.98{\cdot}10^7$	$9.05{\cdot}10^7$	$3.39{\cdot}10^7$

Table 7.20: Example 3: Comparison of statistics for RBDO utilizing an Asymptotic sampling, Importance sampling, and Subset simulation with a global meta-model (GMM), and local meta-models (LMM).

Section 7.2.3. The total number of meta-model simulations is in the same order of magnitude for all used variants of MO-RBDO.

Figure 7.25 and 7.26 show the results of MO-RBDO using different simulation methods with a global meta-model and local meta-models, respectively. From the visual point of view, the best results were obtained with an Importance sampling. The local meta-models seem to be more stable than global meta-model within whole Pareto fronts, especially in the upper tails. Several Pareto fronts obtained using a global meta-model show significant differences from the superior Pareto front. With closer observation, an Importance sampling, however, does not have the correspondence between the recalculated fronts and the original Pareto fronts as the other simulation methods, especially in the lower tails of fronts.

The first, the second, and the third part of Table 7.21 present unary and binary metrics and error indicators. The fourth part shows the average (and total) number of rejected individuals because of the $\pm\infty$ values for reliability indices. The unary metrics prefer the Importance sampling; a Hypervolume, Spacing and Spread have the best results. MO-RBDO using an Asymptotic sampling and global meta-model has great results for a Spacing metric. Binary metrics prefer results obtained with local meta-models to global meta-model. The generational distance is small; the recalculated Pareto fronts are close to the superior Pareto front in general. On average, at least 43% of individuals in the superior Pareto front weakly dominates recalculated Pareto front individuals (MO-RBDO using SS and LMM); the worst result is from MO-RBDO utilizing an Asymptotic sampling and a global meta-model with 61% individuals from recalculated Pareto fronts weakly dominated by individuals from the superior Pareto front. On the contrary, only a minimal portion of solutions from recalculated Pareto fronts weakly dominates individuals from the superior Pareto front; the best result is from MO-RBDO using an Importance sampling and local meta-models with 2.9% of weakly dominated individuals. Errors indicators clearly show that a global meta-model provides more precise results for a reliability index evaluation than local meta-models. In general, an Importance sampling can get the minimum error value, which means that the meta-model together with this sampling method, can credibly imitate the true behaviour at some sub-domains. The Importance sampling with a global meta-model also has the largest maximum error on average. However, 0.35 is an acceptable error. This value is unfortunately influenced by one outlier with error value equal to

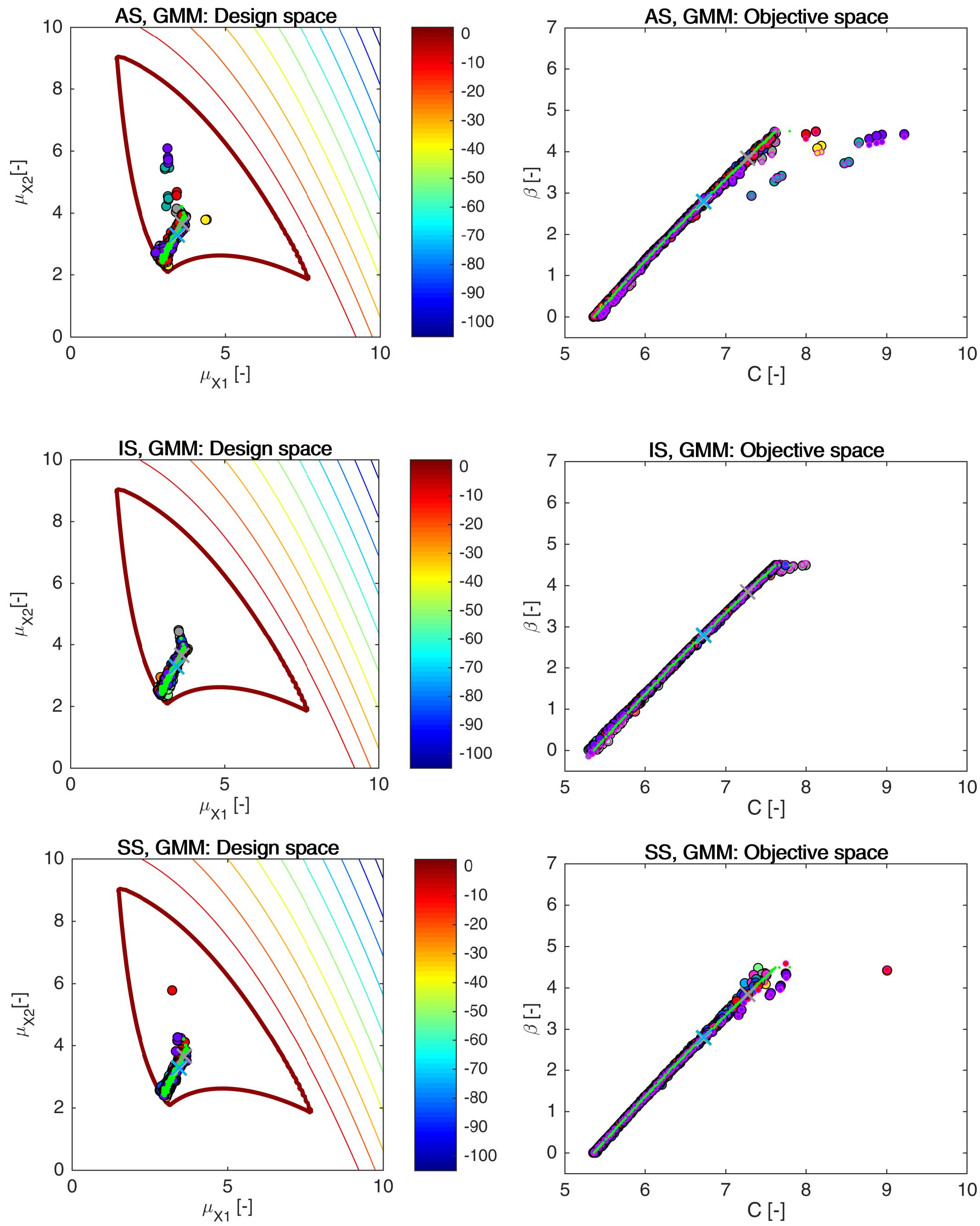


Figure 7.25: Example 3: Pareto sets (circles, left) and Pareto fronts (larger circles with black edges, right) for RBDO utilizing an Asymptotic sampling (AS), Importance sampling (IS), and Subset Simulation (SS) together with a global meta-model (GMM). The smaller circles with magenta edges in the objective space represent the recalculated Pareto sets to Pareto fronts with quasi-Monte Carlo simulations with $\mathrm{CoV} < 5\%$. A yellow, green, and orange cross are hidden behind a blue cross, and they represent solutions published in [59], [174], [115], and [55]; the grey cross represents optima published in [202] (RIA as well as PMA methodology). Green dots sets show the superior Pareto set and Pareto front obtained with an analytical model together with a quasi-Monte Carlo simulation. Every colour means a different run. The contour plot in the design space represents an analytical limit state function; the bold contour is for the limit state $g(\mathbf{X}) = 0$.

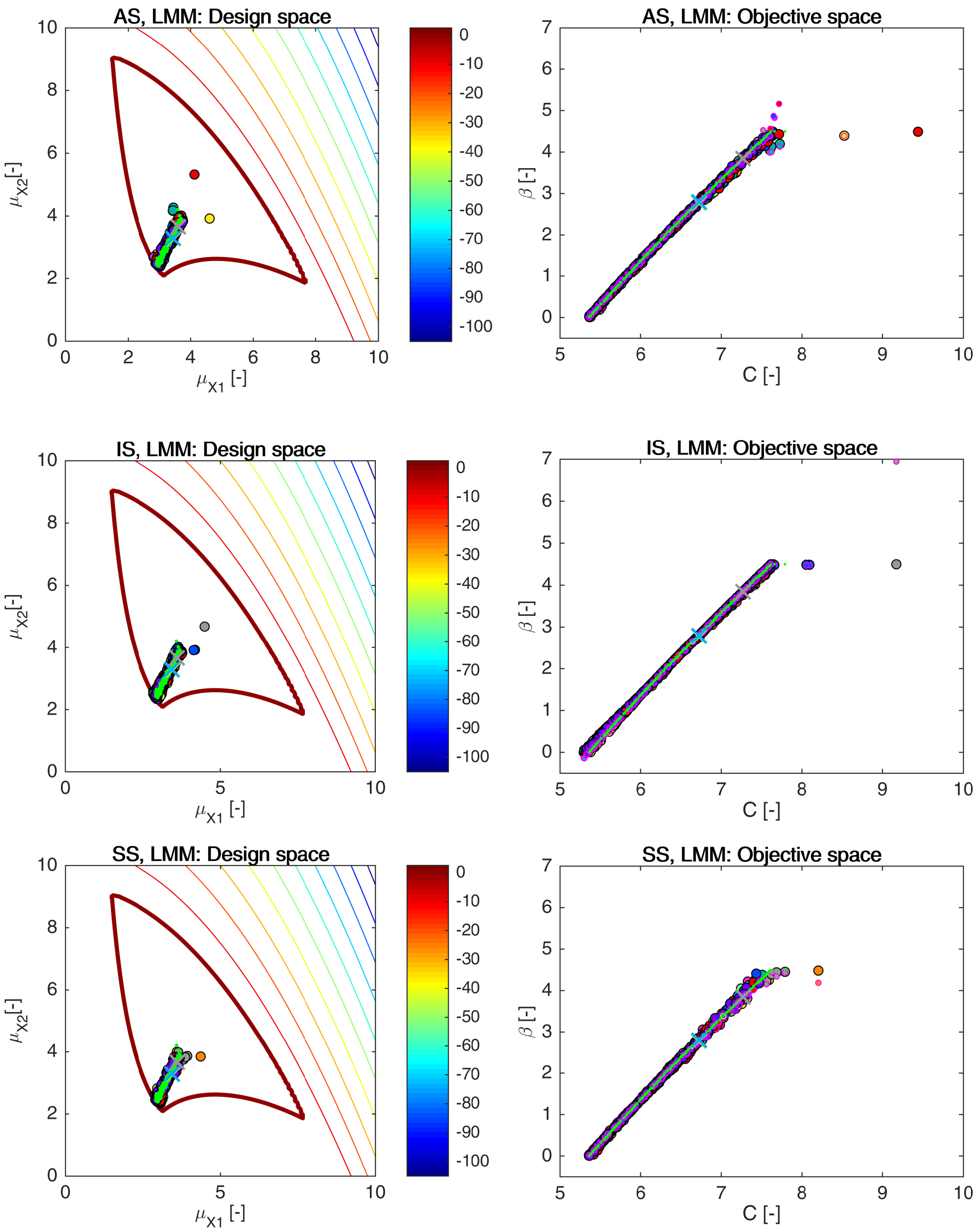


Figure 7.26: Example 3: Pareto sets (circles, left) and Pareto fronts (larger circles with black edges, right) for RBDO utilizing an Asymptotic sampling (AS), Importance sampling (IS), and Subset Simulation (SS) together with local meta-models (LMM). The smaller circles with magenta edges in the objective space represent the recalculated Pareto sets to Pareto fronts with quasi-Monte Carlo simulations with $\mathrm{CoV} < 5\%$. A yellow, green, and orange cross are hidden behind a blue cross, and they represent solutions published in [59], [174], [115], and [55]; the grey cross represents optima published in [202] (RIA as well as PMA methodology). Green dots sets show the superior Pareto set and Pareto front obtained with an analytical model together with a quasi-Monte Carlo simulation. Every colour means a different run. The contour plot in the design space represents an analytical limit state function; the bold contour is for the limit state $g(\mathbf{X}) = 0$.

	GMM			LMM				
	AS	IS	SS	AS	IS	SS	min	max
HV (rPF)	17.18	18.11	17.28	18.24	18.13	17.21	14.86	19.61
S (rPF)	0.0062	0.0061	0.0091	0.0082	0.0065	0.0079	0.0043	0.0180
Δ (rPF)	0.44	0.41	0.52	0.45	0.41	0.50	0.34	0.68
GD (rPF,sPF)	$7.1{\cdot}10^{-4}$	$4.2{\cdot}10^{-4}$	$4.3{\cdot}10^{-4}$	$6.3{\cdot}10^{-4}$	$3.8{\cdot}10^{-4}$	$2.9{\cdot}10^{-4}$	$2.2{\cdot}10^{-4}$	$2.94{\cdot}10^{-3}$
C(sPF,rPF)	0.61	0.54	0.52	0.50	0.47	0.43	0.26	0.90
C(rPF,sPF)	0.020	0.020	0.016	0.026	0.029	0.024	0.004	0.062
D(sPF,rPF)	1.06	0.19	0.99	0.34	0.15	1.03	0.11	3.38
$\min \epsilon(\beta_{\mathrm{oPF}})$	$1.6{\cdot}10^{-4}$	$7{\cdot}10^{-5}$	$2.4{\cdot}10^{-4}$	$2.7{\cdot}10^{-4}$	$8{\cdot}10^{-5}$	$4.8{\cdot}10^{-4}$	0	$1.42{\cdot}10^{-3}$
$\max \epsilon(\beta_{\mathrm{oPF}})$	0.11	0.11	0.24	0.25	0.35	0.22	0.06	2.45
$\mathrm{E}\epsilon(\beta_{\mathrm{oPF}})$	0.020	0.022	0.026	0.032	0.029	0.033	0.006	0.077
$\mathrm{std}\epsilon(\beta_{\mathrm{oPF}})$	0.023	0.033	0.042	0.043	0.066	0.042	0.010	0.346
rejected ind.	0	0	0.1 (1)	0.1 (1)	0	0	0	0.1 (1)

Table 7.21: Example 3: Comparison of performance measures for RBDO utilizing different reliability assessment techniques namely an Asymptotic sampling (AS), Importance sampling (IS), and Subset simulation (SS) using a global (GMM) and local meta-models (LMM). Used metrics are a Hypervolume (HV), Spacing (S), Spread (Δ), Generational distance (GD), Two set coverage metric (C(A,B)), Coverage difference of two sets (D(A,B)), and Error of the reliability index $\epsilon(\beta)$. The used datasets: rPF - recalculated Pareto front, sPF - superior Pareto front, oPF - original Pareto front, and rF - recalculated front. The best values are tinged with the green colour, the worst values with the red colour. The scale shows the visual comparison of the results.

2.45 (Figure 7.26, grey generation, the individual with C = 9.172, $\beta_{\mathrm{oPF}} = 4.496$, $\beta_{\mathrm{rF}} = 6.944$); if we omit this outlier, the average maximum error for MO-RBDO using SS and GMM is 0.112. The average mean and the standard deviation of the error has the lowest values for an Asymptotic sampling in combination with a global meta-model, which means that this method is the most accurate and precise, followed by an Importance sampling and a global meta-model, the latter combination of methods has almost the same error means but the slightly larger standard deviation of the error. The combination of simulation methods with local meta-models are indifferent if we consider both means and standard deviations at the same time since an Importance sampling simulating LMM has the smallest mean and the larger standard deviation of the error among all LMM combinations and the rest of the methods vice versa. Figure 12.9 in Appendix 12 shows the box-plots for the performance measures and the mean and standard deviation of the error.

7.4 Example 4: A short column under oblique bending

A short column under axial loading F and bi-axial bending M_1 and M_2 as depicted in Figure 7.27 is considered in this example. The column has a rectangular cross-section with width b and height h. Geometrically and physically linear behaviour is assumed. This problem considers four stochastic variables and two stochastic design variables. The stochastic variables are applied bending moments M_1 and M_2, yield stress σ_0, and applied normal force F. The stochastic design variables are width b and height h contributing to the cost function by their mean values such that the cross-sectional area of the column is minimized. The problem has one component limit state function employing all stochastic variables as well as stochastic design variables sorted in a vector $\mathbf{X} = [b, h, M_1, M_2, F, \sigma_0]$. The collapse of the structure occurs with the ultimate plastic state. Therefore, we consider this state as the limit state.

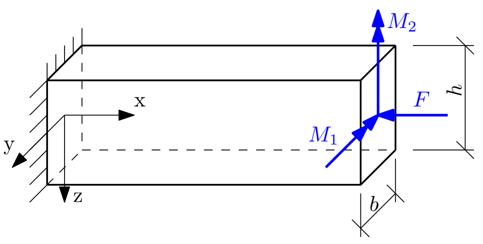


Figure 7.27: Example 4: A short column under oblique bending.

Problem definition is

$$\begin{aligned}
\min \quad & C(\mu_b, \mu_h) = \mu_b \cdot \mu_h && (7.26)\\
\max \quad & \beta(\mathbf{X}) = -\Phi^{-1}(\text{Prob}[g(\mathbf{X}) \leq 0]) && (7.27)\\
& g(\mathbf{X}) = 1 - \frac{4M_1}{bh^2\sigma_0} - \frac{4M_2}{b^2h\sigma_0} - \left(\frac{F}{bh\sigma_0}\right)^2 && (7.28)\\
& 0.5 \leq \mu_b/\mu_h \leq 2. && (7.29)
\end{aligned}$$

For more details, see Appendix 10. The axial force F, bending moments M_1 and M_2 and yield stress σ_0 are statistically independent random variables with Gumbel and Weibull distribution. Cross-section dimensions b and h are also statistically independent variables with a log-normal distribution. The statistical description of variables is in Table 7.22.

This example is used in the literature quite often in different modifications. We overtook and modified a version with a cost function equal to the cross-sectional area from [3] into the multi-objective version. A different version can be seen, e.g. in [59] that was inspired by [168], where a cost function considers a life-time function containing the built costs together with the

Variable		Distribution	Mean	C.o.V
M_1	N.mm	GUM	$2.5 \cdot 10^8$	30%
M_2	N.mm	GUM	$1.25 \cdot 10^8$	30%
F	N	GUM	$2.5 \cdot 10^6$	20%
σ_0	MPa	WEI	40	10%
b	mm	LN	μ_b	5%
h	mm	LN	μ_h	5%

Table 7.22: Example 4: A probabilistic description of the short column under oblique bending example.

Reference	$C(\mu_b^{opt}, \mu_h^{opt})$	β
[3]: RIA	0.2087	2.999
[3]: KKT	0.2268	2.999
[3]: SLA	0.2077	2.978

Table 7.23: Example 4: Published single-objective optima

failure costs. The original version from [3] is

$$\min \quad C(\mu_b, \mu_h) = \mu_b \cdot \mu_h \tag{7.30}$$

$$s.t. \quad \text{Prob}\left[1 - \frac{4M_1}{bh^2\sigma_0} - \frac{4M_2}{b^2h\sigma_0} - \left(\frac{F}{bh\sigma_0}\right)^2 \leq 0\right] \leq p_F^T \tag{7.31}$$

$$0.5 \leq \mu_b/\mu_h \leq 2. \tag{7.32}$$

The target reliability index β^T is equal to 3 which corresponds with the target probability of failure p_F^T equal to $1.35 \cdot 10^{-3}$. Unfortunately, AOUES et al. [3] do not show the optimum of design variables, only the value of the cost function together with the corresponding reliability index. Therefore, we could not verify the accuracy of their results. AOUES et al. compare several single-objective reliability methods. We selected results from three methods, namely Reliability index approach (RIA), classical RBDO replaced by Karush-Kuhn-Tucker optimality conditions of the First-order reliability method (KKT) and Single loop approach (SLA) for our comparison purposes. Table 7.23 summarizes the selected optima.

7.4.1 Comparison of meta-models and analytical model results together with a preconditioned quasi-Monte Carlo simulation

Each MO-RBDO procedure using a preconditioned quasi-Monte Carlo simulation and a different model was run several times to get the required statistical data. Table 7.24 shows some interesting statistics. Average time spent on the MO-RBDO evaluation utilizing an analytical model with eight threads was 1.62 hour, MO-RBDO utilizing local meta-models evaluation

	AM	LMM	GMM	SGMM
elapsed time [hours] (8 threads)	1.62	2.78	0.06	0.06
initial DoE	0	200	50	50
added samples	0	108	374	378
analytical g(x)-calls	$1.10 \cdot 10^{10}$	308	424	428
MM built for opt.	0	3050	31	31
MM built for update	0	441	0	0
MM-calls	0	$7.00 \cdot 10^9$	$2.52 \cdot 10^7$	$2.49 \cdot 10^7$

Table 7.24: Example 4: Comparison of statistics for RBDO utilizing a quasi-Monte Carlo simulation with an analytical model (AM), local meta-models (LMM), global meta-model (GMM), and sparse global meta-model (SGMM).

was approximately one hour long. On the other side, RBDO utilizing global meta-models in the sparse, as well as the dense version of the Gramm matrix, was suspiciously fast. This fast evaluation time together with the lower number of meta-models calls than for GMM and SGMM indicates problems during the optimization, which is evident from Figure 7.28. The global versions of meta-models failed, and the Pareto fronts are completely distorted. Since the meta-models predicted lower reliability indices than is the true value, the time, as well as the number of the total limit state function evaluations, was different from MO-RBDO utilizing an analytical model evaluation. The number of samples in the initial DoE was identical to the setting in other examples, 200 samples for LMM and 50 samples for GMM and SGMM. The updating procedure added 108 samples into DoE for MO-RBDO utilizing LMM, and almost four hundred samples for MO-RBDO utilizing GMM and SGMM. Even this number of added samples into DoE was not able to improve the behaviour of the meta-model to provide satisfactory Pareto fronts.

All the Pareto sets and Pareto fronts obtained from MO-RBDO utilizing all types of used models are depicted in Figure 7.28. From the visual evaluation, the best results matching the superior Pareto front are obtained by MO-RBDO utilizing local meta-models. The global meta-models, as well as the sparse global meta-models, failed in many cases to find the correct solutions even for small reliability indices cases. Figure 7.29 shows one MO-RBDO simulation utilizing a global meta-model to analyze the problem more closely. It is evident that the optimization found the false Pareto front in the very early generation, and this remains unchanged for the rest of the optimization run. The update is mainly around the genetic algorithm population. This DoE data are however only a 2D projection from 6D space omitting stochastical behaviour of M_1, M_2, F, and σ_0 variables. The contours of the analytical model (solid lines) unfortunately do not correspond to the global meta-model (dashed lines) where thick lines for both models highlight the limit states. The problem is in the definition of the model itself, as evident in the last row of images, where the left image represents the analytical model and the right image is for the global meta-model after the last updating procedure of DoE. This analytical model is very flat on the majority of the design space, and the steep descent is around the zero values of the width and the height of the analysed column. This behaviour is very hard to imitate for the global meta-models if the DoE update concentrates in the different area of the design space than in this steepest descent of the function. The local meta-models omit this problem since the procedure does not assemble the model for the whole domain but only for the vicinity around the optimum candidate solution.

Table 7.25 shows the performance measures for MO-RBDO utilizing all models. The analytical model behaves the best in all unary metrics. Recalculated Pareto fronts from MO-RBDO using local meta-models obtained the best Hypervolume among meta-models. On the other side, a Spacing was the best for a sparse global meta-model, and a Spread is comparable for both global meta-models. It follows that MO-RBDO utilizing local meta-models has the best shape of the Pareto front approximation but the distribution among the individuals is better in MO-RBDO utilizing GMM and SGMM. However, the latter two metrics are not so important as the Hypervolume performance measure. The second part of the table takes into account the recalculated Pareto fronts compared to the superior Pareto fronts. The differences between reliability indices values were significant; the recalculated fronts are depicted in Figure 7.28 in the objective space with smaller circles with magenta edges. Note that the Pareto fronts are not restricted to values $[0, 4.5]$ but are stretched to larger magnitudes because the constraint is considered only in MO-RBDO. Since the limit state function is flat in the most of the design space which the meta-models cannot imitate very well, the recalculated Pareto fronts turned out to be better than original Pareto fronts from meta-models if they are compared to the superior

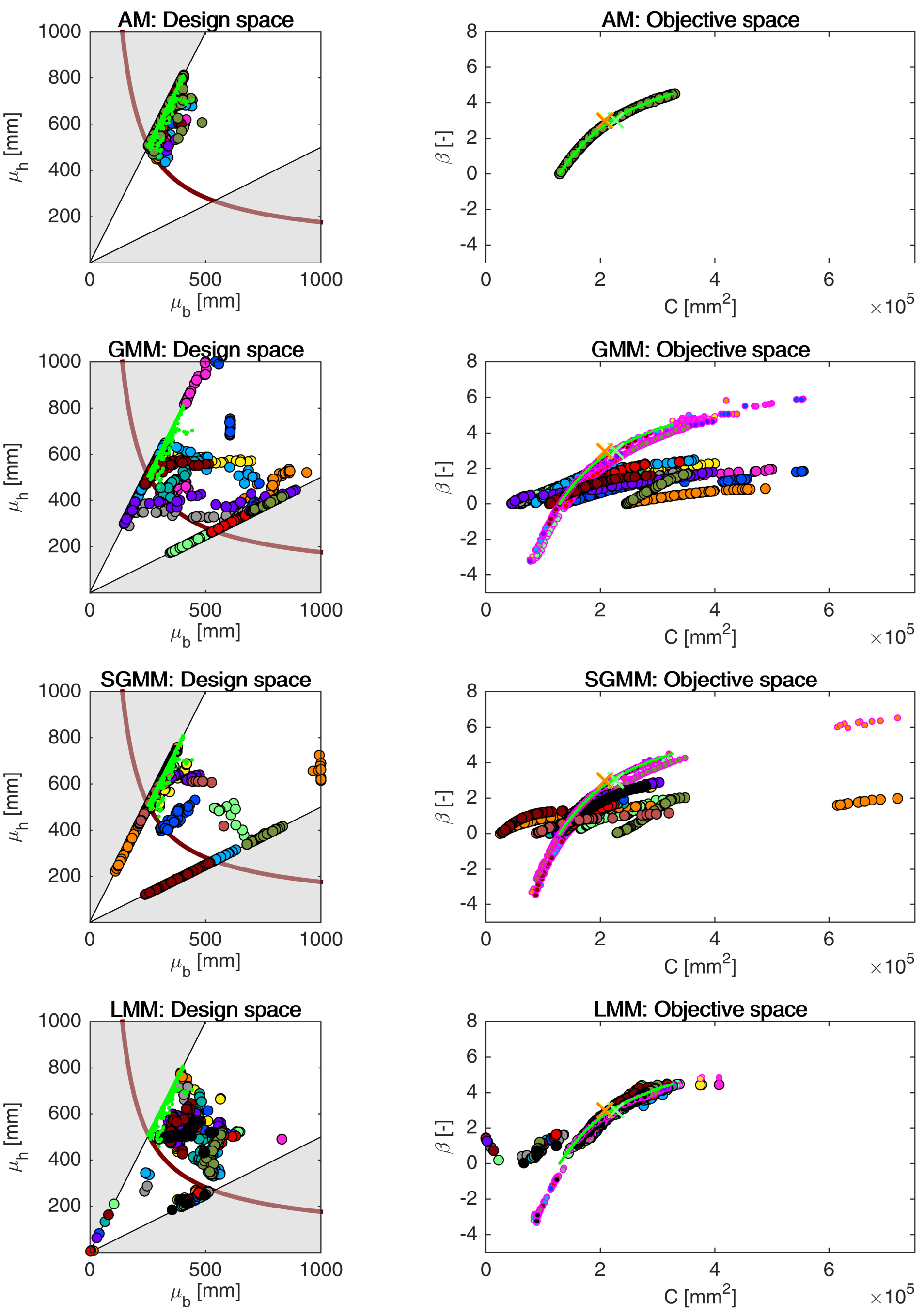


Figure 7.28: Example 4: Pareto sets (circles, left) and Pareto fronts (large circles, right) for all models, namely an analytical model (AM), global meta-model (GMM), sparse global meta-model (SGMM), and local meta-models (LMM). Smaller circles with magenta edges in the objective space represent the recalculated fronts with an analytical model. Crosses represent results published in [3]. Green dots sets show the superior Pareto set and superior Pareto front. Every colour has the meaning of a different run. The bold contour represents the limit state $g(\mathbf{X}) = 0$. The grey shaded areas represent inadmissible subspace by the constraint in Equation 7.29.

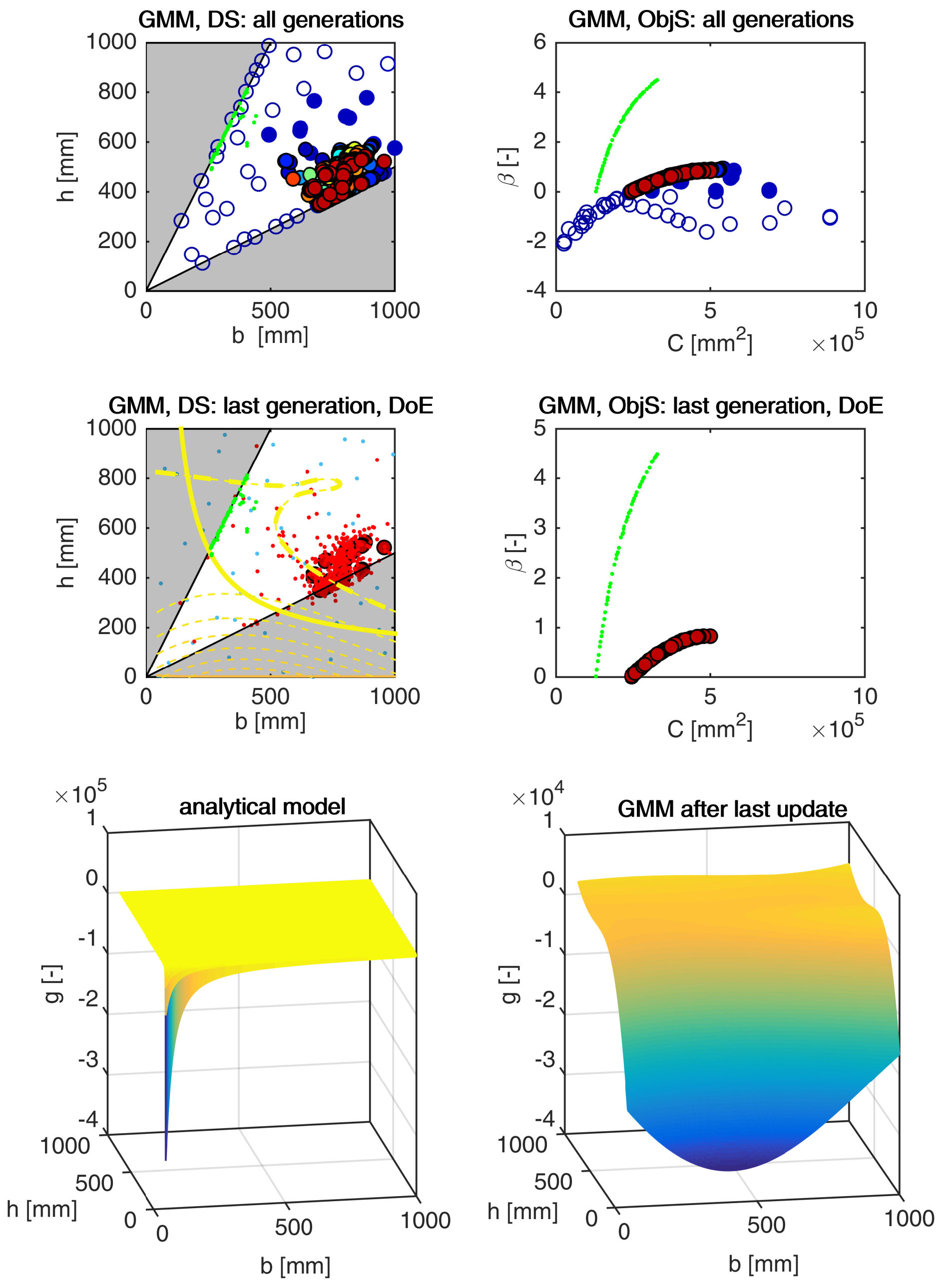


Figure 7.29: Example 4: Pareto sets and Pareto fronts from one MO-RBDO run utilizing a global meta-model. Abbreviations: DS - design space, ObjS - objective space, GMM - global meta-model, DoE - Design of Experiments. The first row of pictures represents all generations from the optimization algorithm (circles). Empty circles denote the solutions not satisfying β constraints. The dark blue circles depict the first generation; the next generations continue with cyan, green, yellow, orange, and red colours. The last generation (Pareto set on the left and Pareto front on the right) is depicted in the second row of pictures together with DoE. Set of green dots depicts the superior Pareto set and the superior Pareto front. The dashed contours serve for the representation of GMM and the solid contours for the analytical model. The bold contours are for the limit state $g(\mathbf{X}) = 0$. The last row of images represents an analytical model (left) for fixed values of M_1, M_2, F, and σ_0 on their mean values; and a global meta-model assembled for the last generation of NSGA-II. The grey shaded areas represent inadmissible subspace by the constraint in Equation 7.29.

Pareto front (green dots set). However, several individuals had to be rejected since the recalculated reliability indices β were equal to $\pm\infty$. We had to reject at least one solution per Pareto front obtained by MO-RBDO utilizing LMM. On the other hand, MO-RBDO utilizing SGMM or GMM were able to get a Pareto front without $\pm\infty$ values after recalculation at least once. MO-RBDO utilizing SGMM provided the worst Pareto front, where the recalculation rejected 38 solutions out of 50 solutions.

The Generational distance is comparable for all-meta-models on an average; the recalculated fronts are very close to the superior Pareto front. However, mostly all solutions from the superior Pareto front weakly dominate the solutions in the Pareto fronts obtained by MO-RBDO utilizing GMM or LMM; the Pareto fronts obtained by MO-RBDO utilizing SGMM are weakly dominated only from a half. The recalculated Pareto fronts dominate almost none of the solutions from the superior Pareto front. According to the Two set coverage metric, MO-RBDO utilizing SGMM provided the best-obtained fronts. By contrast, the Coverage difference of two sets metric prefers the Pareto fronts obtained by MO-RBDO utilizing LMM.

As can be seen from Figure 7.28 in the objective space, if larger circles are compared to the smaller ones, the best minimum error of the reliability index is obtained using local meta-models. On the other hand, if we observe the maximum error, the values are poor for all meta-models. Local meta-models are accurate but not precise; on the contrary, a sparse global meta-model is the most precise but nor accurate. However, the standard deviation of the error does not differ much for all three models as the mean value of the error. These values of error

	AM	GMM	LMM	SGMM	min	max
HV(rPF)	$4.9 \cdot 10^6$	$4.5 \cdot 10^6$	$4.7 \cdot 10^6$	$4.3 \cdot 10^6$	$2.4 \cdot 10^6$	$5.1 \cdot 10^6$
S(rPF)	0.0046	0.0094	0.0072	0.0065	0.0022	0.0370
Δ(rPF)	0.38	0.69	0.98	0.69	0.35	1.13
GD(rPF,sPF)	-	0.0085	0.0080	0.0090	0.0002	0.0612
C(sPF,rPF)	-	0.81	0.90	0.55	0.17	1.00
C(rPF,sPF)	-	0.005	0.001	0.031	0	0.097
D(sPF,rPF)	-	$5.4 \cdot 10^5$	$2.9 \cdot 10^5$	$7.0 \cdot 10^5$	$4.1 \cdot 10^4$	$2.7 \cdot 10^6$
min $\epsilon(\beta_{\text{oPF}})$	-	0.470	0.003	0.406	$6.1 \cdot 10^{-5}$	2.836
max $\epsilon(\beta_{\text{oPF}})$	-	3.244	2.969	2.195	0.191	5.061
E $\epsilon(\beta_{\text{oPF}})$	-	1.659	0.398	1.178	0.074	3.622
std $\epsilon(\beta_{\text{oPF}})$	-	0.786	0.787	0.530	0.053	1.567
rejected ind.	-	3.75 (45)	1.23 (16)	3.79 (53)	0	38

Table 7.25: Example 4: Comparison of performance measures for RBDO utilizing a quasi-Monte Carlo simulation with an analytical model (AM), local meta-models (LMM), global meta-model (GMM), and sparse global meta-model (SGMM). Used metrics are a Hypervolume (HV), Spacing (S), Spread (Δ), Generational distance (GD), Two set coverage metric (C(A,B)), Coverage difference of two sets (D(A,B)), Error of the reliability index ($\epsilon(\beta)$), and a number of the mean value of rejected individuals with a total number of rejected individuals in parentheses from all 12 runs with 50 individuals. datasets: rPF - recalculated Pareto fronts, sPF - superior Pareto front, oPF - original Pareto front. The best values are tinged with the green colour, the worst values with the red colour. The scale shows the visual comparison of the results.

are slightly skewed since we had to reject plenty of solutions with the recalculated β-indices equal to $\pm\infty$; the number of rejected individuals is in the last row of the table. The mean number of rejected solutions per one front is tinged, the values in parentheses represent the total number of rejected solutions per 12 fronts, i.e. 600 solutions in total. The minimum and maximum rejected solutions are per one recalculated front. Figure 12.10 in Appendix 12 shows the box-plots for the performance measures and the mean and standard deviation of the error.

7.4.2 Results comparison of different reliability assessment techniques utilizing an analytical model

We run our multi-objective reliability-based design optimization with an analytical model to test the behaviour of different reliability assessment techniques. We run each MO-RBDO with a different reliability assessment method ten times to check the behaviour and to have enough statistical data. Table 7.26 shows some statistics. Our implementation of MO-RBDO utilizing a quasi-Monte Carlo simulation was the fastest, followed by an Enhanced Monte Carlo simulation and Scaled sigma sampling. On the other hand, our slowest implementation was MO-RBDO utilizing an Importance sampling followed by MO-RBDO utilizing a First-order reliability method. An Enhanced Monte Carlo simulation is extremely unstable in this benchmark. We had to switch the recommended Levenberg-Marquart least-squares optimization algorithm for searching for the parameters into a Trust region reflective algorithm since the Levenberg-Marquart least-squares optimization does not use the bounds for the inputs. However, if the b parameter is lesser than any λ, the x_j part of the simplification becomes a complex number since the mantissa of the exponential function is a negative number, and this negative number is then raised to the power of c, which is a real number. In the same way, we had to restrict the parameters from below because if the algorithm found the optimum with the parameters in the order of magnitude larger than 3, the probability of failure becomes Not a Number (NaN). It seems that the minimized function has several local optima or the problem is multi-modal.

Figure 7.30 and 7.31 show all the resulting data from all MO-RBDO runs.The Pareto fronts obtained from MO-RBDO utilizing an Asymptotic sampling are visually almost identical as their recalculated counterpart with a small exception on the right tail of the Pareto front, the recalculated fronts almost imitate the superior Pareto front. The Pareto fronts from MO-RBDO utilizing an Enhanced Monte Carlo simulation have good performance below reliability index equal to approximately 3; the original Pareto fronts above this value differ from their recalculated counterparts and the superior Pareto front. The orange Pareto front also suffers from bounding of the reliability index to 4.5 as is the definition of the optimization task. The Pareto

	AS	eMC	FORM	IS	MC	SS	SSS
elapsed time [s]	143	68	278	335	31	121	84
g(x)-calls	$7.1 \cdot 10^7$	$4.7 \cdot 10^7$	$1.8 \cdot 10^5$	$4.5 \cdot 10^7$	$4.7 \cdot 10^7$	$3.9 \cdot 10^7$	$4.4 \cdot 10^7$

Table 7.26: Example 4: Comparison of statistics for RBDO utilizing an analytical model and different reliability assessment techniques, namely an Asymptotic sampling (AS), Enhanced Monte Carlo simulation (eMC), First-order reliability method (FORM), Importance sampling (IS), quasi-Monte Carlo simulation (MC), Subset simulation (SS), and Scaled sigma sampling (SSS).

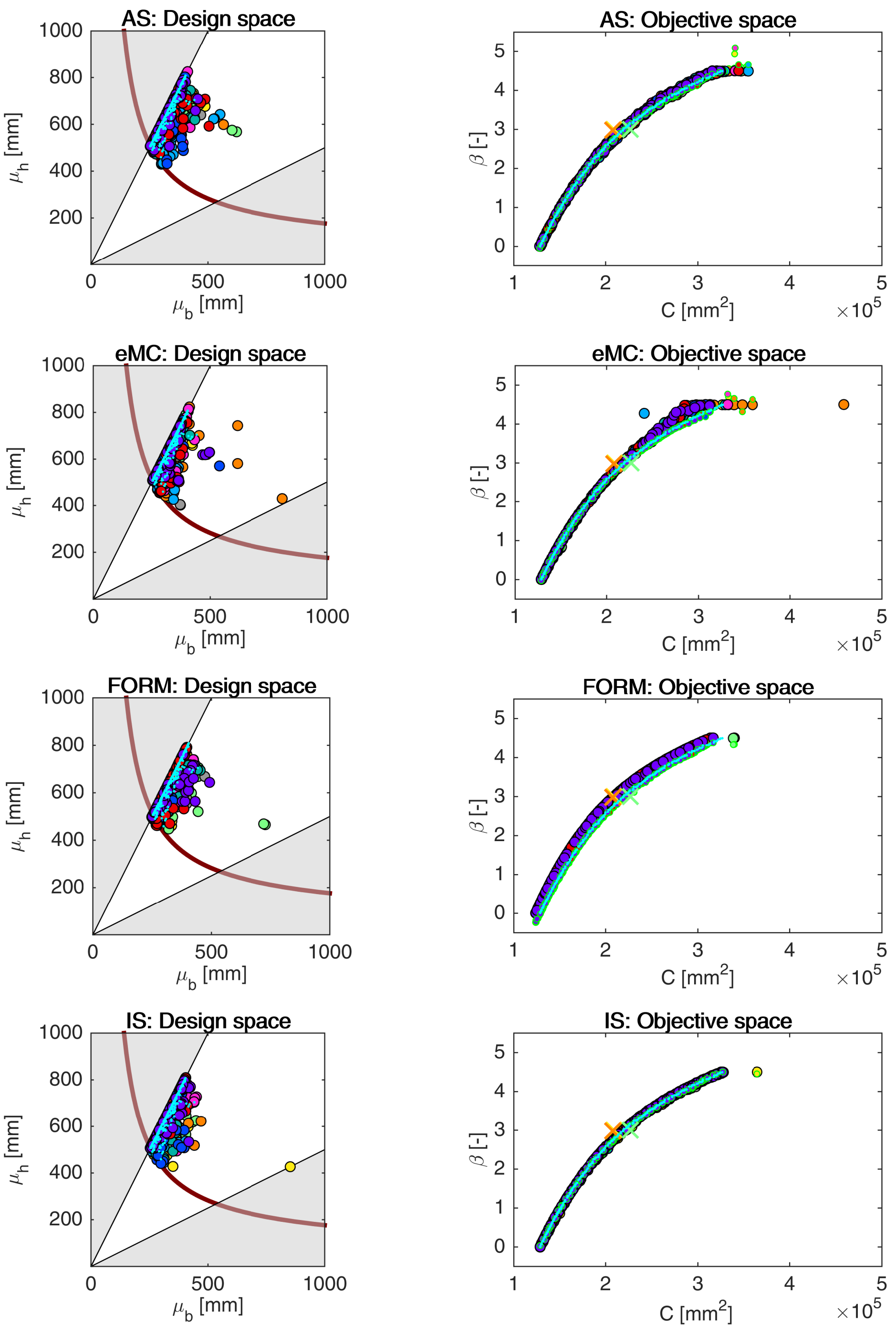


Figure 7.30: Example 4: Pareto sets (circles, left) and Pareto fronts (larger circles with black edges, right) for RBDO utilizing an Asymptotic sampling (AS), Enhanced Monte Carlo simulation (eMC), First-order reliability method (FORM), and Importance sampling (IS) together with an analytical model. The smaller circles with green edges in the objective space represent the recalculated Pareto sets to Pareto fronts with a quasi-Monte Carlo simulation with CoV lesser than 5%. An orange cross represents RIA and SLA optimum published in [3], a green cross represents KKT optimum published ibid. Aqua dots sets show the superior Pareto set and superior Pareto Pareto front. Every colour means a different run. The contour plot in the design space represents an analytical limit state function; the bold contour is for the limit state $g(\mathbf{X}) = 0$.

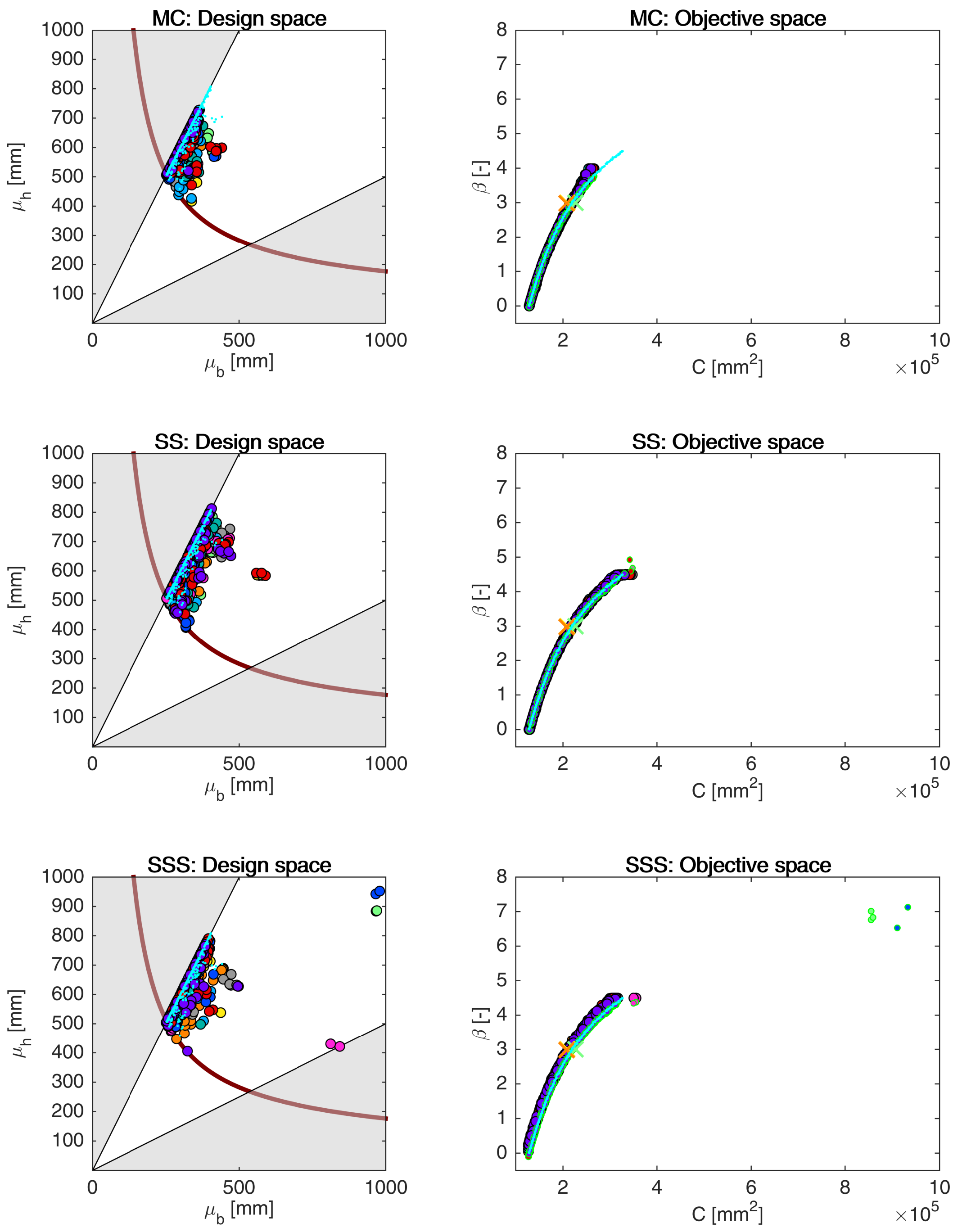


Figure 7.31: Example 4: Pareto sets (circles, left) and Pareto fronts (larger circles with black edges, right) for RBDO utilizing a quasi-Monte Carlo simulation (MC), Subset simulation (SS), and Scaled sigma sampling (SSS) together with an analytical model. The smaller circles with green edges in the objective space represent the recalculated Pareto sets to Pareto fronts with a quasi-Monte Carlo simulation with CoV lesser than 5%. An orange cross represents RIA and SLA optimum published in [3], a green cross represents KKT optimum published ibid. Aqua dots sets show the superior Pareto set and superior Pareto front. Every colour means a different run. The contour plot in the design space represents an analytical limit state function; the bold contour is for the limit state $g(\mathbf{X}) = 0$.

fronts obtained via a First-order reliability method are consistently underestimated in the reliability index objective function, the recalculated fronts, however, imitate the superior Pareto front. MO-RBDO utilizing an Importance sampling provides one of the best solutions among the rest of the reliability assessment methods, the Pareto fronts, recalculated fronts, and superior Pareto fronts have the same shape with only one exception suffering from the bounding of the reliability index from above to value 4.5. MO-RBDO utilizing a quasi-Monte Carlo simulation cannot provide reliable data with larger reliability index than 4 since a quasi-Monte Carlo simulation would need more samples than 30,000. Up to reliability index 3.5, the shape of the original Pareto fronts traces the recalculated fronts as well as the superior Pareto front. The Pareto fronts obtained via a Subset simulation provide visually similar shape as the recalculated fronts as well as the superior Pareto front, again with a small exception at the upper tail of the Pareto fronts. The Pareto fronts obtained via MO-RBDO utilizing a Scaled sigma sampling have consistently underestimated reliability indices. However, their recalculated counterparts imitate the shape of the superior Pareto front with a few exceptions - the outlier solutions depicted with the pink, the light green and the royal blue colours; these individuals differ in the design space as well as in the objective space.

Table 7.27 shows the performance measures and the error of the reliability indices. The Hypervolume values differ from 3.3 million to 3.9 million, the smallest Hypervolume value was approximately obtained by MO-RBDO utilizing MC, while the largest by MO-RBDO utilizing IS closely followed by AS and SS. The Spacing performance measure indicates that the individuals in the Pareto fronts are almost ideally distributed, but the Spread performance measure

	AS	eMC	FORM	IS	MC	SS	SSS	min	max
HV(rPF) $[\cdot 10^6]$	3.90	3.80	3.85	3.92	3.34	3.89	3.82	3.28	3.93
S(rPF) $[\cdot 10^{-3}]$	6.6	6.6	5.4	5.6	5.3	5.9	6.3	3.6	9.2
Δ(rPF)	0.45	0.47	0.45	0.42	0.56	0.43	0.50	0.33	0.62
GD(rPF,sPF) $[\cdot 10^{-4}]$	2.8	2.6	7.2	2.7	2.5	2.6	3.0	2.1	7.9
C(sPF,rPF)	0.63	0.73	0.57	0.63	0.60	0.67	0.63	0.43	0.82
C(rPF,sPF)	0.044	0.028	0.037	0.048	0.023	0.041	0.031	0	0.068
D(sPF,rPF) $[\cdot 10^4]$	4.4	14	8.9	1.9	60	4.8	12	1.2	66
min $\epsilon(\beta_{\text{oPF}})$	0.002	0.001	0.12	0.001	0.0003	0.001	0.105	0.0003	0.14
max $\epsilon(\beta_{\text{oPF}})$	0.22	0.49	0.30	0.062	0.34	0.19	0.39	0.062	1.02
E $\epsilon(\beta_{\text{oPF}})$	0.045	0.093	0.24	0.023	0.046	0.046	0.21	0.023	0.24
std $\epsilon(\beta_{\text{oPF}})$	0.045	0.115	0.043	0.016	0.069	0.043	0.053	0.016	0.17

Table 7.27: Example 4: Comparison of performance measures for RBDO utilizing different reliability assessment techniques and an analytical model, namely an Asymptotic sampling (AS), Enhanced Monte Carlo simulation (eMC), First-order reliability method (FORM), Importance sampling (IS), quasi-Monte Carlo simulation (MC), Subset simulation (SS), and Scaled sigma sampling (SSS). Used metrics are a Hypervolume (HV), Spacing (S), Spread (Δ), Generational distance (GD), Two set coverage metric (C(A,B)), Coverage difference of two sets (D(A,B)), and Error of the reliability index ($\epsilon(\beta)$). The datasets: rPF - recalculated Pareto front, sPF - superior Pareto front, oPF - original Pareto front, and rF - recalculated front. The best values are tinged with the green colour, the worst values with the red colour. The scale shows the visual comparison of the results.

is in modest opposition. The same problem occurred most likely in Examples 1 - 3 in the pair-wise distribution of the individuals, which is unfortunately unrecognizable by this metric. The Spread metric has similar values for all methods on average, the best Spread is between individuals from MO-RBDO utilizing IS closely followed by MO-RBDO utilizing SS, the worst Spread is between solutions in Pareto fronts from MO-RBDO utilizing MC.

The Generational distance is small for all methods, the distance between the recalculated Pareto fronts and the superior Pareto front is therefore really small as visible from Figure 7.30 and 7.31. The smallest proportion of weakly dominated solutions by the superior Pareto front is in the recalculated Pareto fronts obtained by MO-RBDO utilizing FORM on average, followed by MO-RBDO utilizing MC. On the other hand, the largest proportion of weakly dominated solutions is in the recalculated Pareto fronts from an Enhanced Monte Carlo simulation, followed by a Subset simulation. 4.8% of solutions from the recalculated Pareto fronts obtained by MO-RBDO utilizing an Importance sampling dominated solutions in the superior Pareto front, which is the best result. On the contrary, the worst result is from MO-RBDO utilizing a quasi-Monte Carlo simulation. The smallest size of the space that is weakly dominated by the superior Pareto front but not weakly dominated by the recalculated Pareto fronts is obtained by MO-RBDO utilizing an Importance sampling, followed by MO-RBDO utilizing an Asymptotic sampling, and MO-RBDO utilizing a Subset simulation. Compared to the Hypervolume, this space is twice the order of the Coverage difference of two sets.

The smallest minimum error was obtained via a plain quasi-Monte Carlo simulation followed by an Importance sampling and a Subset Simulation. The quasi-Monte Carlo simulation can provide very precise reliability indices for low values, even with a low number of samples. The best result of the maximum error is for an Importance sampling; the maximum error was 0.062 on average. An Enhanced Monte Carlo simulation provided results with the worst maximum error on average as well as in total, which is the maximum of the maximum error 1.02.

The most accurate and precise method is an Importance sampling. The second- and the third-best method is an Asymptotic sampling and a Subset simulation. If we compare these two methods, the Asymptotic sampling is slightly more accurate, and the Subset simulation is slightly more precise; however, both methods are relatively comparable. A FORM has the same precision as a Subset simulation, but the accuracy is the worst of all methods. This trend is evident from Figure 7.30, where the original Pareto fronts are translated vertically from the recalculated fronts. An Enhanced Monte Carlo simulation provides Pareto fronts that have the worst precision. This behaviour is also evident from Figure 7.30, where the upper tails of the Pareto fronts indeed differ from the recalculated solutions, but the bottom tails are relatively precise and accurate. Figure 12.11 in Appendix 12 shows the box-plots for the performance measures and the mean and standard deviation of the error.

7.4.3 Results for approximation using meta-models and reliability assessment techniques

The last part of the results is the combination of the meta-models and different reliability assessment techniques. The three best-behaving reliability assessment methods, namely an Asymptotic sampling, Importance sampling, and Subset simulation, are combined with all meta-models. We discuss only two best-behaving meta-models in this section, namely local meta-models and global meta-model. The results for the sparse global meta-model are given in Appendix 12 without further comments. The ranks of all methods, together with the best

	SGMM			LMM		
	AS	IS	SS	AS	IS	SS
elapsed time [sec]	116	1364	158	442	706	404
elapsed time [hours]	0.032	0.379	0.044	0.123	0.196	0.112
initial DoE	50	50	50	200	200	200
added samples	355	398	369	80	111	100
analytical g(x)-calls	405	448	419	280	311	300
MM built for opt.	31	31	31	3050	3050	3050
MM built for update	0	0	0	322	544	430
MM-calls	$1.18{\cdot}10^7$	$8.41{\cdot}10^7$	$1.93{\cdot}10^7$	$6.11{\cdot}10^7$	$8.65{\cdot}10^7$	$4.59{\cdot}10^7$

Table 7.28: Example 4: Comparison of statistics for RBDO utilizing an Asymptotic sampling, Importance sampling, and Subset simulation with a sparse global meta-model (SGMM), and local meta-models (LMM).

placings for all five examples, are in Table 7.1; the fourth example is in the fourth row of the table. We run all MO-RBDO with all combinations of selected simulation techniques and meta-models ten times to have relevant data for our statistics. We used a laptop computer with hardware and software specifications defined in Table 11.2. Our fastest implementation is MO-RBDO utilizing an Asymptotic sampling and sparse global meta-model, followed by MO-RBDO using a Subset simulation and sparse global meta-model, see Table 7.28. Nonetheless, with a closer observation of the results with sparse global meta-model employment in Figure 7.33, the original Pareto fronts are different from the recalculated counterparts; the reliability indices are significantly underestimated in most cases, and therefore fewer samples are necessary to evaluate the reliability indices. Even though an updating procedure terminated with more samples in DoE than for local meta-models, local meta-models show the better prediction of the limit state function, which has an impact on the reliability index prediction.

Figure 7.32 and 7.33 show the design and the objective space of the optimization results together with their recalculated counterparts. Unfortunately, the sparse global meta-model provides an unsatisfying response of the model as evident from Figure 7.32. We got similar results with a preconditioned quasi-Monte Carlo simulation, see Figure 7.28. The bottom image on the left in Figure 7.29 shows the analytical model, the image on the right the global meta-model assembled after the last update of DoE. The sparse global meta-model has a similar trend. It is tough to imitate the steep trend of the model with low values of width and height of the column. The global meta-model, in general, is therefore not suitable for this type of problems. On the contrary, local meta-models are assembled only in the vicinity of the individuals of the genetic algorithm. The steepest trend of the function is in the failure region, and therefore the individuals are not concentrated around the hard-imitating subspace. The local meta-models are assembled outside this subspace, and the limit state function prediction is more accurate than the sparse global meta-model response. Unfortunately, the simulation methods using local meta-models still fail in the prediction of the response in the failure region.

Table 7.29 is devoted to the qualitative appraisal of the Pareto front results. The Hypervolume metric provides better results for the recalculated Pareto fronts from MO-RBDO utilizing local meta-models; values of this metric are almost comparable for different reliability methods and meta-models. The Spacing metric has results close to zero for all presented combinations

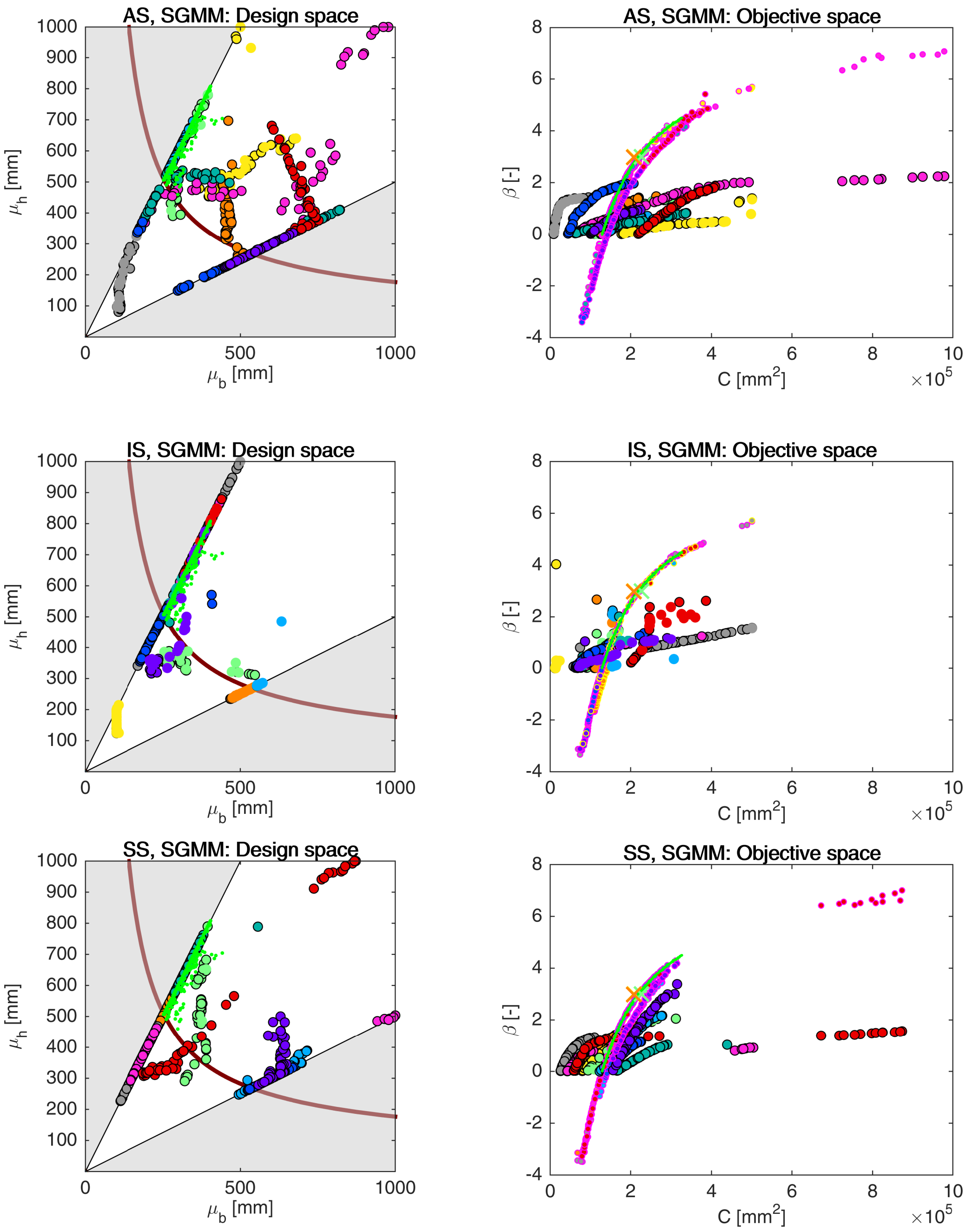


Figure 7.32: Example 4: Pareto sets (circles, left) and Pareto fronts (larger circles with black edges, right) for RBDO utilizing an Asymptotic sampling (AS), Importance sampling (IS), and Subset Simulation (SS) together with a sparse global meta-model (SGMM). The smaller circles with magenta edges in the objective space represent the recalculated Pareto sets to Pareto fronts with quasi-Monte Carlo simulations with $\mathrm{CoV} < 5\%$. An orange cross represents RIA and SLA optimum published in [3], a green cross represents KKT optimum published ibid. Green dots sets show the superior Pareto set and Pareto front obtained with an analytical model together with a quasi-Monte Carlo simulation. Every colour means a different run. The contour plot in the design space represents an analytical limit state function; the bold contour is for the limit state $g(\mathbf{X}) = 0$.

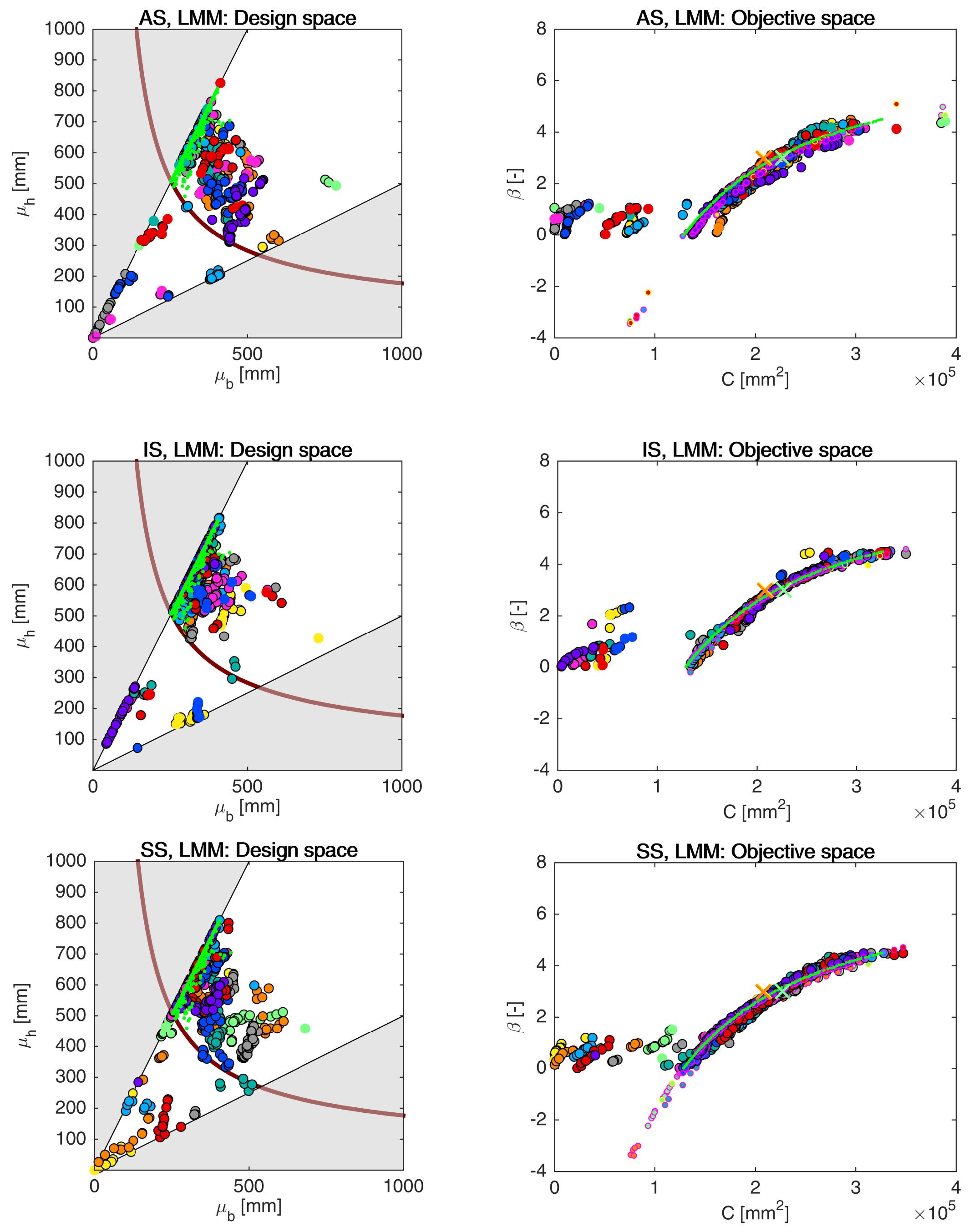


Figure 7.33: Example 4: Pareto sets (circles, left) and Pareto fronts (larger circles with black edges, right) for RBDO utilizing an Asymptotic sampling (AS), Importance sampling (IS), and Subset Simulation (SS) together with local meta-models (LMM). The smaller circles with magenta edges in the objective space represent the recalculated Pareto sets to Pareto fronts with quasi-Monte Carlo simulations with $\mathrm{CoV} < 5\%$. An orange cross represents RIA and SLA optimum published in [3], a green cross represents KKT optimum published ibid. Green dots sets show the superior Pareto set and Pareto front obtained with an analytical model together with a quasi-Monte Carlo simulation. Every colour means a different run. The contour plot in the design space represents an analytical limit state function; the bold contour is for the limit state $g(\mathbf{X}) = 0$.

	SGMM			LMM				
	AS	IS	SS	AS	IS	SS	min	max
HV (rPF)	$5.9\cdot10^6$	$5.3\cdot10^6$	$5.6\cdot10^6$	$6.62\cdot10^6$	$6.7\cdot10^6$	$6.8\cdot10^6$	0	$7.0\cdot10^6$
S (rPF)	0.009	0.039	0.010	0.0097	0.0092	0.0063	0	0.18
Δ (rPF)	0.77	1.05	0.73	0.92	0.79	0.77	0.52	1.54
GD (rPF,sPF)	0.024	0.014	0.025	0.0040	0.00035	0.0033	0.00015	0.16
C(sPF,rPF)	0.65	0.44	0.54	0.88	0.79	0.78	0	1
C(rPF,sPF)	0.010	0.011	0.006	0.005	0.011	0.011	0	0.081
D(sPF,rPF)	$1.2\cdot10^6$	$1.9\cdot10^6$	$1.5\cdot10^6$	$4.7\cdot10^5$	$3.37\cdot10^5$	$3.38\cdot10^5$	$1.0\cdot10^5$	$7.1\cdot10^6$
$\min\epsilon(\beta_{\text{oPF}})$	0.83	0.51	0.65	0.015	0.0043	0.0046	0	3.5
$\max\epsilon(\beta_{\text{oPF}})$	3.63	3.46	3.02	1.62	0.59	1.20	0.14	5.4
$\text{E}\epsilon(\beta_{\text{oPF}})$	2.13	1.73	1.84	0.32	0.12	0.27	0.04	4.00
$\text{std}\epsilon(\beta_{\text{oPF}})$	0.82	0.69	0.70	0.39	0.13	0.35	0.04	1.63
rejected ind.	8.9	10.7	10.9	7.8	6.8	4.6	0	50
Σ rejected ind.	89	107	109	78	68	46	46	109

Table 7.29: Example 4: Comparison of performance measures for RBDO utilizing different reliability assessment techniques namely an Asymptotic sampling (AS), Importance sampling (IS), and Subset simulation (SS) using a sparse global (SGMM) and local meta-models (LMM). Used metrics are a Hypervolume (HV), Spacing (S), Spread (Δ), Generational distance (GD), Two set coverage metric (C(A,B)), Coverage difference of two sets (D(A,B)), and Error of the reliability index $\epsilon(\beta)$. The used datasets: rPF - recalculated Pareto front, sPF - superior Pareto front, oPF - original Pareto front, and rF - recalculated front. The best values are tinged with the green colour, the worst values with the red colour. The scale shows the visual comparison of the results.

of methodologies, which means that not so much outliers or separate individuals exist in the recalculated Pareto fronts. On the contrary, the Spread metric provides larger values which means that the spread of the data is not ideal. The recalculated Pareto fronts are surprisingly not as far from the superior Pareto front as their original counterparts. The recalculated Pareto fronts from MO-RBDO utilizing local meta-models are closer to the superior Pareto front than recalculated Pareto fronts obtained from MO-RBDO utilizing sparse global meta-models. The number of weakly dominated solutions is higher for the recalculated Pareto fronts using local meta-models than sparse global meta-models; at least 44% of members are weakly dominated on average. For the global meta-models, at least 78% of members from recalculated Pareto fronts are weakly dominated by the superior Pareto front on average. An amount of weakly dominated solutions in the superior Pareto front by the recalculated fronts is low. Nevertheless, the recalculated Pareto fronts differ from the original Pareto fronts. Therefore, the comparison of the dominated solutions is only for the record to keep the consistency of the results.

The bottom half of Table 7.29 presents the most critical data since the original Pareto fronts differ from their recalculated counterparts. The local meta-models can be very precise at some subspaces according to the minimum error; on the contrary, global meta-models have the best performance with the 0.51 difference in the reliability index on average. The difference between the true reliability index and its approximation with the simulation technique and a meta-model is enormous for the maximum error, especially for a global meta-model. The largest maximum error for local meta-models is 1.62 for an Asymptotic sampling; however, the maximum error using local meta-models with a preconditioned quasi-Monte Carlo simulation in Section 7.4.1 is 2.969 (see Table 7.25), and the maximum error using an analytical model and Asymptotic sampling is 0.22 (see Table 7.27). These results mean that Asymptotic sampling can provide better potential candidates for the DoE update than a preconditioned quasi-Monte Carlo simulation. Therefore, we can assembly better local meta-models with DoE update candidates from an Asymptotic sampling compared to a preconditioned quasi-Monte Carlo simulation.

The mean error for local meta-models is no more than 0.32 on average for an Asymptotic sampling, while for the sparse global meta-model, the best average value is 1.73. The difference in the standard deviation between the models is also significant. MO-RBDO using an Importance sampling and local meta-models provides the most accurate and precise results in this Section. However, if we compare results from local meta-model utilizing a preconditioned quasi-Monte Carlo simulation (E $\epsilon(\beta_{\text{oPF}})$ = 0.398, std $\epsilon(\beta_{\text{oPF}})$ = 0.787, see Table 7.25), the advanced simulation techniques perform better (E $\epsilon(\beta_{\text{oPF}})$ = 0.12, std $\epsilon(\beta_{\text{oPF}})$ = 0.13 for an Importance sampling, see Table 7.29). If we compare the results for an Importance sampling using an analytical model (E $\epsilon(\beta_{\text{oPF}})$ = 0.023, std $\epsilon(\beta_{\text{oPF}})$ = 0.016, see Table 7.27) and local meta-models (E $\epsilon(\beta_{\text{oPF}})$ = 0.12, std $\epsilon(\beta_{\text{oPF}})$ = 0.13, see Table 7.29), the error is lower-order value in case of an analytical model. However, the result is excellent, considering that only 311 simulations of an analytical model were performed.

Unfortunately, not all individuals from the recalculated fronts are depicted in Figure 7.32 and 7.33. Last part of Table 7.29 shows quantities of individuals that have reliability indices equal to $\pm\infty$ after recalculation by a quasi-Monte Carlo simulation with CoV lesser than 5%. Since we run ten independent MO-RBDO simulations for all methodology combinations, the average number of rejected individuals due to $\pm\infty$ is ten times lower than their sum. Each original Pareto set/front contains 50 individuals. MO-RBDO using a sparse global meta-model and Importance sampling provided once such inaccurate reliability index values, that all individuals in recalculated fronts were equal to $\pm\infty$ and the whole front had to be discarded after recalculation. Figure 12.12 in Appendix 12 shows the box-plots for the performance measures and the mean and standard deviation of the error.

7.5 Example 5: A 23-bar truss

The 23-bar planar truss bridge is a simply supported structure, see Figure 7.34. Bars are divided into two groups; upper and lower chords (11 members) are included in one group, and 12 diagonals create the second group. The vertically oriented static loading is located in all six upper chord nodes. All truss bars are made from the same material. LEE and KWAK first published a probabilistic description of the problem in paper [108], where they calculated only reliability of this benchmark. Subsequently, DUBOURG defined a single-objective optimization problem in his PhD thesis [59].

Table 7.30 gives the probabilistic description of ten statistically independent random variables. Young's modulus E_1 and E_2, as well as cross-sectional areas assigned to two groups A_1 and A_2, are lognormally distributed; six gravity loads P_i have a Gumbel distribution. Table 7.30 does not define the means of the cross-sectional areas since they represent design variables to be optimized. This problem, therefore, considers eight stochastic variables with given distributions described exactly by two statistical moments and two stochastic design variables with given distributions and standard deviations, where the mean values are the optimized targets.

The design rule in this task is that the mid-span displacement should not exceed maximum

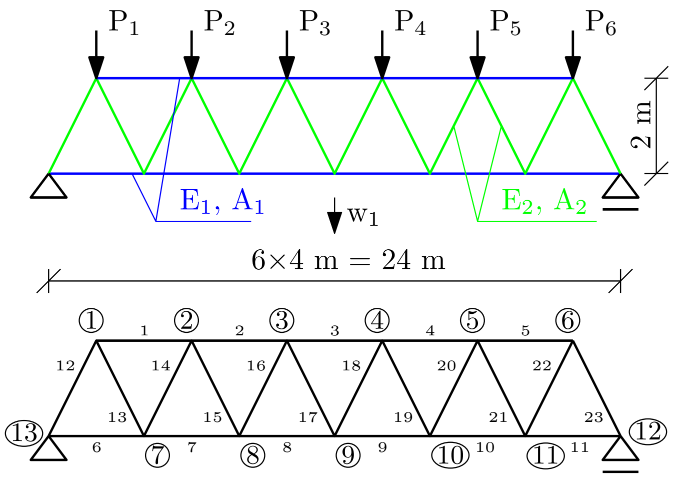


Figure 7.34: A 23-bar plane truss bridge.

displacement $w_{\max} = 10$ cm mathematically expressed as

$$g(\mathbf{x}) = w_{\max} - |w_1(\mathbf{x})|; \tag{7.33}$$

positive or zero $g(\mathbf{x})$ denotes safety of the system, negative $g(\mathbf{x})$ indicates failure. All random variables are arranged in the vector $\mathbf{x}$. Note that the mid-span displacement need not be the maximal one due to the randomness of variables; however, the mid-span displacement is presumed to be the maximum one for the limit state function purposes. Since the structure is statically determinate, the analytical formulation of the mid-span displacement is simple to obtain:

$$\begin{aligned} w_{\max} \quad = \quad & -\frac{2\sqrt{2}\,A_1\,P_1 + 6\sqrt{2}\,A_1\,P_2 + 10\sqrt{2}\,A_1\,P_3 + 10\sqrt{2}\,A_1\,P_4}{A_1\,A_2\,E_2} + \\ \ldots \quad & +\frac{6\sqrt{2}\,A_1\,P_5 + 2\sqrt{2}\,A_1\,P_6}{A_1\,A_2\,E_2} - \frac{36\,A_2\,E_2\,P_1 + 100\,A_2\,E_2\,P_2}{A_1\,A_2\,E_1\,E_2} + \\ \ldots \quad & +\frac{140\,A_2\,E_2\,P_3 + 140\,A_2\,E_2\,P_4 + 100\,A_2\,E_2\,P_5 + 36\,A_2\,E_2\,P_6}{A_1\,A_2\,E_1\,E_2}. \end{aligned} \tag{7.34}$$

Considering the mean values of all the variables (SUDRET in [183] used $\mu_{\mathrm{A1}} = 2 \cdot 10^{-3}$ m^2 and $\mu_{\mathrm{A2}} = 1 \cdot 10^{-3}$ m^2), the vertical displacement in the middle of the truss structure is 7.78 cm.

The optimal design of the structure should be on the one hand the most light-weighted as possible; on the other hand, the safest as possible. These two criteria are antagonistic. DUBOURG in [59] formulated the single-objective optimization problem as

$$\mathbf{d}^* = \arg\min_{\mathbf{d}\in\mathbb{D}} L_1\mu_{A1} + L_2\mu_{A2} : \text{Prob}[g(\mathbf{x}(\mathbf{d})) \le 0] \le \Phi(-\beta_0), \tag{7.35}$$

Variable		Distribution	Mean	Standard deviation
E_1, E_2	Pa	Lognormal	$2.1 \cdot 10^{11}$	$2.1 \cdot 10^{10}$
A_1	m^2	Lognormal	μ_{A1}	$2 \cdot 10^{-4}$
A_2	m^2	Lognormal	μ_{A2}	$1 \cdot 10^{-4}$
$P_1 \ldots P_6$	N	Gumbel	$5 \cdot 10^4$	$7.5 \cdot 10^3$

Table 7.30: A probabilistic description of the 23-bar planar truss bridge.

in which L_1 and L_2 are cumulative lengths of all bars in each group, β_0 is a prescribed reliability index equal to 3 and $\Phi(\cdot)$ is a Laplace function that represents a cumulative distribution function of the standard normal distribution. $\mathbb{D}$ space is bounded to $[6\cdot 10^{-4}, 6\cdot 10^{-3}]\times[3\cdot 10^{-4}, 3\cdot 10^{-3}]$. DUBOURG in [59] provides an optimum obtained by metamodel-based RBDO utilizing Kriging as $\mathbf{d}_{opt}$ = $[2.53\cdot 10^{-3}, 8.13\cdot 10^{-4}]$ with a cost function equal to 0.1388 m^3 and a probability index $\beta = 3.05$.

We reformulated the single-objective problem to the multi-objective formulation into

$$\min \quad C(\mu_{A_1}, \mu_{A_2}) = \quad L_1\mu_{A1} + L_2\mu_{A2} \tag{7.36}$$
$$\max \quad \beta(\mathbf{X}) = -\Phi^{-1}(\text{Prob}[g(\mathbf{X}) \le 0]) \tag{7.37}$$
$$g(\mathbf{X}) = w_{\max} - |w_1(\mathbf{x})|. \tag{7.38}$$

For practical purposes, the reliability index β was limited to the range $0 \le \beta \le 4.5$.

7.5.1 Comparison of meta-models and analytical model results together with a preconditioned quasi-Monte Carlo simulation

First of all, we analyzed the behaviour of the meta-models and the analytical model with a preconditioned quasi-Monte Carlo simulation. We run the problem on one thread and eight threads. Table 7.31 shows some interesting statistical data from several optimization runs. MO-RBDO utilizing an analytical model (AM) was the fastest among all models. The speed-up of the parallelization is from 2.08 for MO-RBDO using AM to 4.60 for MO-RBDO utilizing sparse global meta-model (SGMM). The number of samples in the initial DoE was set to the same values as in the previous examples regardless of the dimensionality of the problem, therefore 50 samples for both versions of global meta-models and 200 samples for local meta-models (LMM). MO-RBDO utilizing LMM added 154 points on an average, MO-RBDO using both versions of GMM added around 430 samples on an average. Quantity of meta-models assembled during the optimization was 3050 for MO-RBDO utilizing LMM since each individual in each generation needs unique meta-model. MO-RBDO utilizing GMM or SGMM identically constructed a unique meta-model for the first generation and 30 offspring generation, 31 generations in total. Local meta-models version of MO-RBDO needs a new meta-model for each candidate solution

	AM	LMM	GMM	SGMM
elapsed time (1 thread) [hours]	4.61	14.65	21.56	26.76
elapsed time (8 threads) [hours]	2.22	4.24	5	5.82
initial DoE	0	200	50	50
added samples	0	154	422	433
analytical g(x)-calls	$9.45\cdot 10^9$	354	472	483
MM built for opt.	0	3050	31	31
MM built for update	0	689	0	0
MM-calls	0	$8.32\cdot 10^9$	$5.16\cdot 10^9$	$5.26\cdot 10^9$

Table 7.31: Example 5: Comparison of statistics for RBDO utilizing a quasi-Monte Carlo simulation with an analytical model (AM), local meta-models (LMM), global meta-model (GMM), and sparse global meta-model (SGMM).

to be updated with the DoE; therefore 689 meta-models were assembled for this purposes on an average. The total number of meta-models evaluations is almost identical for all used models at least in the order of magnitude.

Figure 7.35 depicts all the independent MO-RBDO runs.MO-RBDO utilizing a global meta-model, as well as a sparse global meta-model, have problems to approximate the upper tail of Pareto fronts. These methodologies underestimate the β-indices greater than or equal to 3; the approximation is therefore on the safe side. However, the problem with meta-models for β greater than three is evident. The recalculated Pareto fronts with both types of global meta-models have tail ending around β-index equal to 8. On the other hand, MO-RBDO utilizing LMM provides excellent Pareto fronts that are close to the superior Pareto front from a visual point of view. The Pareto set obtained by MO-RBDO using AM is the most resembling the Pareto set from MO-RBDO using LMM.

Table 7.32 shows the comparison of performance measures described in Section 3.2. The unary metrics evince the best results for the recalculated Pareto fronts obtained from MO-RBDO utilizing the analytical model. The Hypervolume, however, does not have a large difference in the mean values of the metrics for any model. The recalculated Pareto fronts obtained by MO-RBDO using both versions of the global meta-models have similar Spacing metric and Spread metric values. The recalculated Pareto fronts obtained by MO-RBDO utilizing LMM have a nice behaviour according to a Spacing metric but not according to a Spread performance measure. The pairwise or the groupwise distribution do not affect the Spacing metric but the Spread metric, which happened in this case as well. A Generational distance is very close to the zero in all cases, which means that the recalculated Pareto fronts are close to the superior Pareto front. The solutions in the superior Pareto front dominate the solutions in the recalculated Pareto

	AM	GMM	LMM	SGMM	min	max
HV	0.79	0.77	0.76	0.78	0.65	0.84
S	0.0053	0.0083	0.0073	0.0080	0.0038	0.0180
Δ	0.37	0.43	0.59	0.43	0.31	0.81
GD	-	$8.3 \cdot 10^{-4}$	$7.9 \cdot 10^{-4}$	$8.1 \cdot 10^{-4}$	$3.3 \cdot 10^{-4}$	$3.9 \cdot 10^{-3}$
C(PF$_{\mathrm{AM}}$,PF$_{\mathrm{MM}}$)	-	0.75	0.81	0.72	0.53	1.00
C(PF$_{\mathrm{MM}}$,PF$_{\mathrm{AM}}$)	-	0.019	0.030	0.025	0	0.107
D(PF$_{\mathrm{AM}}$,PF$_{\mathrm{MM}}$)	-	0.03	0.04	0.02	0.01	0.15
$\min \epsilon(\beta_{\mathrm{oPF}})$	-	0.021	0.004	0.008	0	0.164
$\max \epsilon(\beta_{\mathrm{oPF}})$	-	3.012	0.449	1.872	0.02	7.89
$\mathrm{E}\epsilon(\beta_{\mathrm{oPF}})$	-	0.812	0.126	0.473	0.01	2.61
$\mathrm{std}\epsilon(\beta_{\mathrm{oPF}})$	-	0.792	0.109	0.519	0.00	2.02
rejected ind.	-	1.11 (21)	0 (0)	0.90 (17)	0	2

Table 7.32: Example 5: Comparison of performance measures for RBDO utilizing a quasi-Monte Carlo simulation with an analytical model (AM), local meta-models (LMM), global meta-model (GMM), and sparse global meta-model (SGMM). Used metrics are a Hypervolume (HV), Spacing (S), Spread (Δ), Generational distance (GD), Two set coverage metric (C(A,B)), Coverage difference of two sets (D(A,B)), and Error of the reliability index ($\epsilon(\beta)$). The best values are tinged with the green colour, the worst values with the red colour. The scale shows the visual comparison of the results.

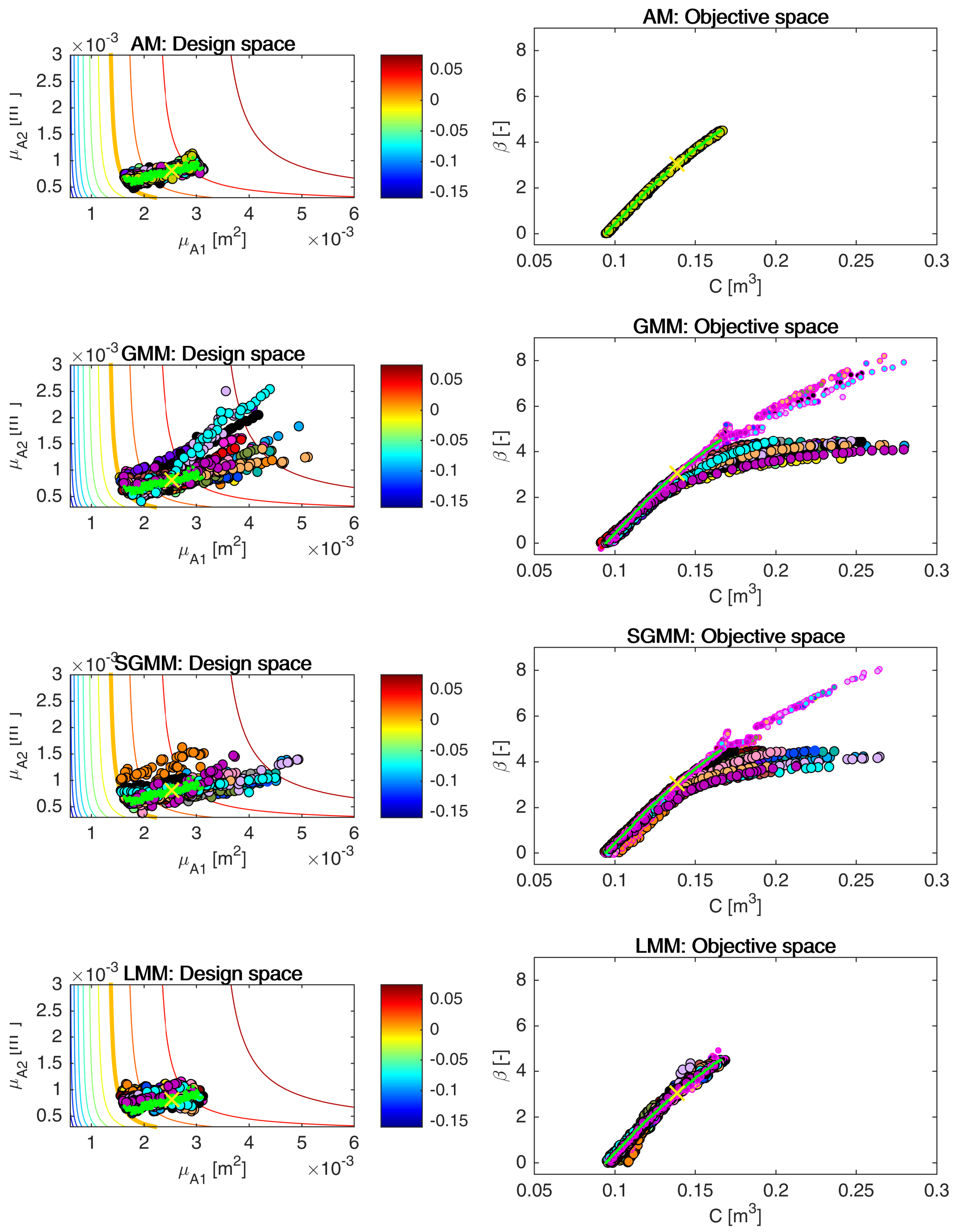


Figure 7.35: Example 5: Pareto sets (circles, left) and Pareto fronts (large circles, right) for all models, namely an analytical model (AM), global meta-model (GMM), sparse global meta-model (SGMM), and local meta-models (LMM). Smaller circles with magenta edges in the objective space represent the recalculated fronts with an analytical model. A magenta cross is for a result published in [59]. Green dots sets show the superior Pareto set and superior Pareto front. Every colour has the meaning of a different run. The contour plot in the design space represents an analytical limit state function; the bold contour is for the limit state $g(\mathbf{X}) = 0$.

fronts obtained by MO-RBDO utilizing any meta-model in at least 72% on average; recalculated Pareto fronts obtained by MO-RBDO utilizing SGMM have the best behaviour, MO-RBDO utilizing LMM provided the worst data. On the other hand, 3% solutions of recalculated Pareto front obtained by MO-RBDO using LMM dominated the solutions of the superior Pareto front, which is approximately the best result for the Two set coverage metric $C(rPF, sPF)$. MO-RBDO utilizing SGMM provided the smallest size of the space that is weakly dominated by the superior Pareto front but not weakly dominated by the recalculated Pareto front.

Overall, the most precise and accurate data provided MO-RBDO utilizing local meta-models, which is evident both from the comparison of the original Pareto fronts with their recalculated counterparts in Figure 7.35 and from all error indicators in Table 7.32. On the contrary, the worst error is on the original Pareto fronts from MO-RBDO utilizing global meta-models. We had to discard approximately 1.11 solutions from each Pareto front obtained by MO-RBDO utilizing GMM, the total number of discarded solutions was 21 out of 950 (19 independent runs, 50 individuals in the last generation). We did not discard any solution from the Pareto fronts obtained by MO-RBDO utilizing LMM. The recalculation by a quasi-Monte Carlo simulation with CoV lesser than 5% discarded approximately 0.9 solutions from each Pareto front obtained by MO-RBDO utilizing SGMM; the total number of discarded solutions was 17 out of 950.

7.5.2 Results comparison of different reliability assessment techniques utilizing an analytical model

We run the multi-objective reliability-based design optimization with an analytical model and several different reliability assessment techniques to minimize the error and examine the behaviour of simulation methods and an approximation technique. Table 7.33 shows some statistics. We run all the methods ten times on a computer with hardware and software specifications defined in Table 11.1 and work with these data in the following section. The number of limit state function evaluations is two orders of magnitude smaller for simulation methods in comparison with MO-RBDO utilizing a preconditioned quasi-MC. Our fastest implementation is MO-RBDO utilizing a quasi-Monte Carlo simulation; the same primacy is obtained in the previous four examples as well. The second fastest implementation is MO-RBDO utilizing an Enhanced Monte Carlo simulation followed by MO-RBDO utilizing a Scaled sigma sampling. Our slowest implementation is MO-RBDO utilizing an Asymptotic sampling, the problem is the setting most likely, but we prefer to keep the setting the same for all five testing benchmarks. The second slowest is the implementation of MO-RBDO utilizing an Importance sampling followed closely by MO-RBDO utilizing a Subset simulation.

	AS	eMC	FORM	IS	MC	SS	SSS
elapsed time [s]	409	82	218	287	45	272	111
g(x)-calls	$6.3 \cdot 10^7$	$4.7 \cdot 10^7$	$3.2 \cdot 10^5$	$4.4 \cdot 10^7$	$4.7 \cdot 10^7$	$4.8 \cdot 10^7$	$4.4 \cdot 10^7$

Table 7.33: Example 5: Comparison of statistics for RBDO utilizing an analytical model and different reliability assessment techniques, namely an Asymptotic sampling (AS), Enhanced Monte Carlo simulation (eMC), First-order reliability method (FORM), Importance sampling (IS), quasi-Monte Carlo simulation (MC), Subset simulation (SS), and Scaled sigma sampling (SSS).

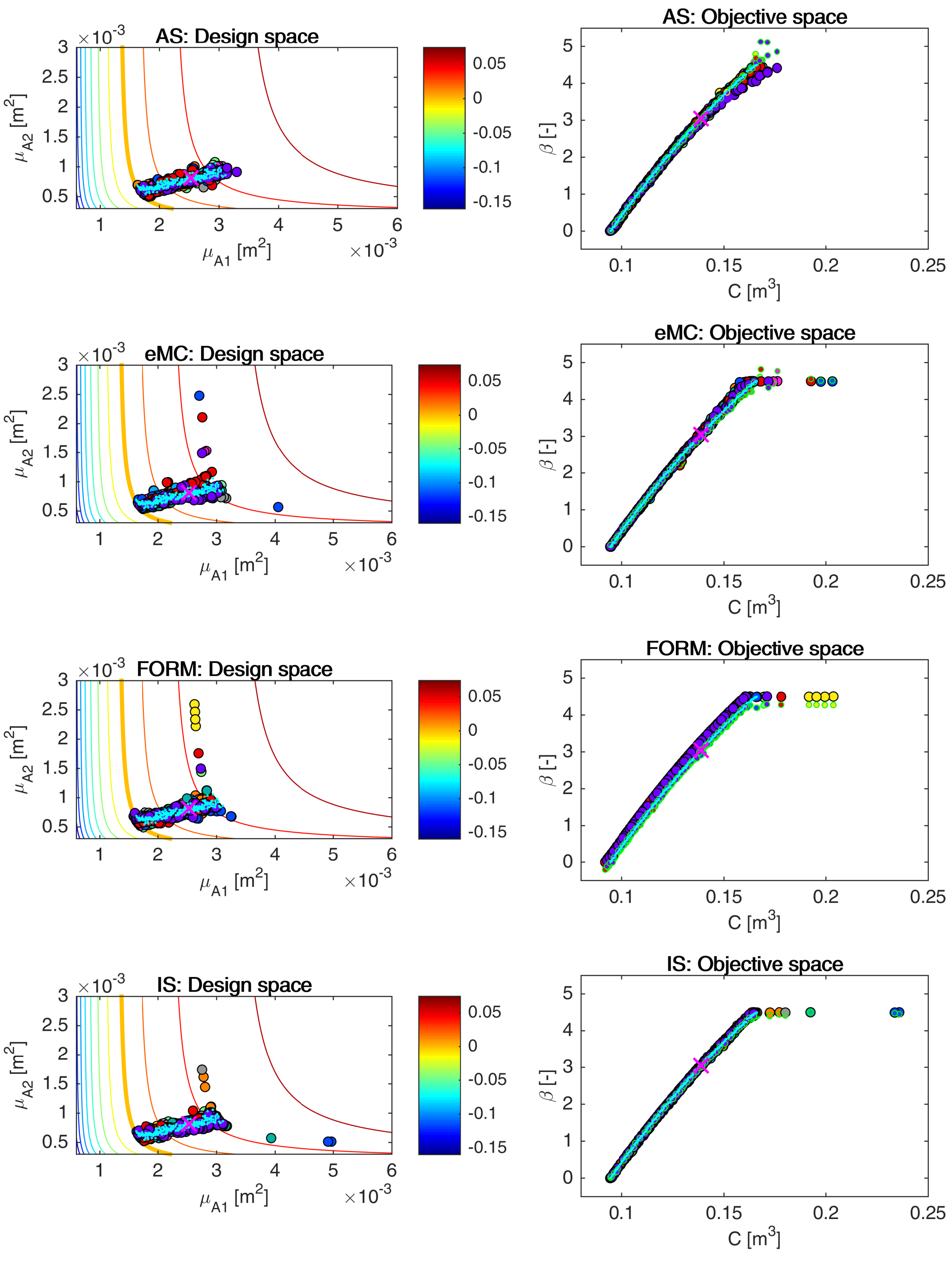


Figure 7.36: Example 5: Pareto sets (circles, left) and Pareto fronts (larger circles with black edges, right) for RBDO utilizing an Asymptotic sampling (AS), Enhanced Monte Carlo simulation (eMC), First-order reliability method (FORM), and Importance sampling (IS) together with an analytical model. The smaller circles with green edges in the objective space represent the recalculated Pareto sets to Pareto fronts with a quasi-Monte Carlo simulation with CoV lesser than 5%. A magenta cross represents an optimum published in [59]. Aqua dots sets show the superior Pareto set and superior Pareto front. Every colour means a different run. The contour plot in the design space represents an analytical limit state function; the bold contour is for the limit state $g(\mathbf{X}) = 0$.

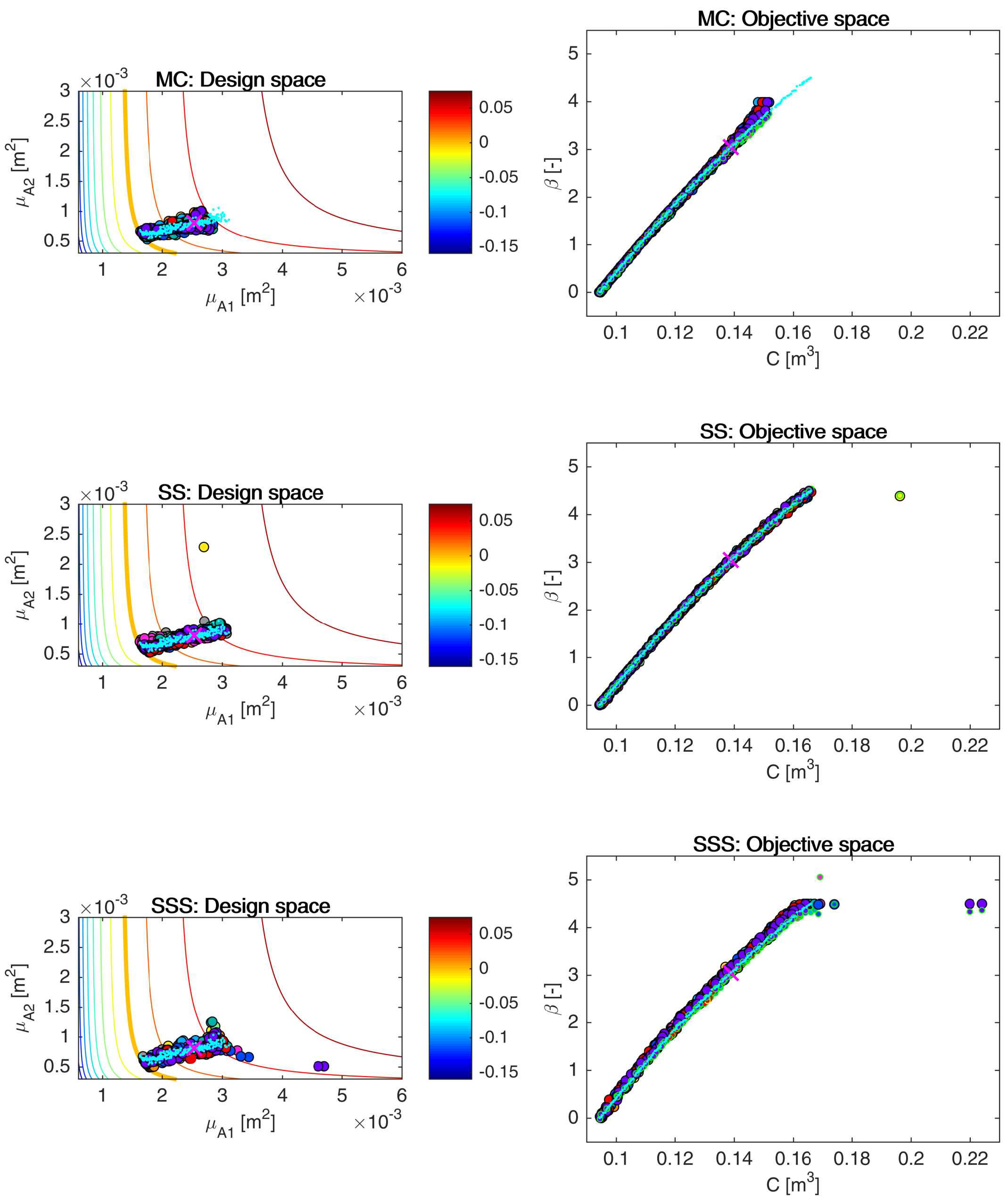


Figure 7.37: Example 5: Pareto sets (circles, left) and Pareto fronts (larger circles with black edges, right) for RBDO utilizing a quasi-Monte Carlo simulation (MC), Subset simulation (SS), and Scaled sigma sampling (SSS) together with an analytical model. The smaller circles with green edges in the objective space represent the recalculated Pareto sets to Pareto fronts with quasi-Monte Carlo simulations with CoV lesser than 5%. A magenta cross represents an optimum published in [59]. Aqua dots sets show the superior Pareto set and Pareto Pareto front obtained with an analytical model together with quasi-Monte Carlo simulation. Every colour means a different run. The contour plot in the design space represents an analytical limit state function; the bold contour is for the limit state $g(\mathbf{X}) = 0$.

Figure 7.36 and 7.37 show all the data from the multi-objective reliability-based design optimization together with the recalculation of the Pareto sets by the quasi-Monte Carlo simulation with a coefficient of variation lesser than 5%. The Pareto fronts obtained from MO-RBDO utilizing an Asymptotic sampling differ visually from their recalculated counterparts for the reliability index greater than 3; however, an Asymptotic sampling underestimated the β-index which is on the safe side of the design. The Pareto fronts obtained via MO-RBDO utilizing an Enhanced Monte Carlo simulation differ visually as well. Unfortunately, this method overestimated the reliability indices, and the design is on the unsafe side of the design. The Pareto fronts also suffer from the bounding the β-index to the value 4.5 from above. A First-order reliability method systematically underestimated the reliability indices. The original Pareto fronts are vertically translated in comparison with their recalculated counterparts. This method also suffers from the bounding of the β-index to the value 4.5 from above. The Pareto fronts from MO-RBDO utilizing an Importance sampling visually imitate the recalculated fronts as well as the superior Pareto front. Still, the upper tails of the original Pareto fronts also suffer from the bounding of the reliability index from above. A quasi-Monte Carlo simulation in MO-RBDO would need more samples than 30,000 for the upper tails of the Pareto fronts, the shape of the original Pareto fronts with a β-index from the interval $[0, 3]$ imitates nicely the recalculated counterparts as well as the superior Pareto front. The Pareto fronts obtained from MO-RBDO utilizing a Subset simulation have similar shapes like their recalculated counterparts and the superior Pareto front with only one outlier from one simulation out of 10 runs. MO-RBDO using a Scaled sigma sampling overestimates the reliability indices with a growing cost function, and therefore the Pareto fronts are slightly rotated around the lower tail of the Pareto front. This method also suffers from the bounding of the β-index from above in one run out of 10 runs.

Table 7.34 shows the performance measures and the error for the Pareto fronts obtained from MO-RBDO. The Hypervolume is between values 0.76 and 0.79 with one exception - the Hypervolume calculated on the Pareto fronts obtained from MO-RBDO utilizing a quasi-Monte Carlo simulation. The best value is however obtained on the Pareto fronts from MO-RBDO using an Asymptotic sampling. The Spacing metric is close to zero for all the methodologies as in all previous examples; however, the Spread metric is more extensive; therefore, the spread of individuals in Pareto fronts is not ideal. The Generational distance is small for all the methodologies; the recalculated Pareto fronts are therefore close to the superior Pareto front. The lowest portion of weakly dominated individuals in recalculated Pareto fronts by a superior Pareto front is from the MO-RBDO utilizing a Subset simulation, followed by a Scaled sigma sampling and by a First-order reliability method. On the contrary, the most significant percentage of weakly dominated solutions in recalculated Pareto fronts by a superior Pareto front is from MO-RBDO utilizing an Enhanced Monte Carlo simulation. In general, only a small portion of the recalculated data weakly dominated the superior Pareto front; the best result is from MO-RBDO utilizing an Asymptotic sampling where 3.8% of Pareto front weakly dominated the superior Pareto front.

A quasi-Monte Carlo simulation can obtain minimal error; however, it can also get the maximum error on average. The problem is that any Monte Carlo simulation needs a higher number of samples for higher reliability indices. Therefore the lower tail of the Pareto front contains individuals with the best beta-index values, and the upper-tail individuals have the most significant errors. The maximum of the maximum error surprisingly came from an Asymptotic sampling ($\max \epsilon(\beta_{\text{oPF}} = 0.91)$, the last column). On average, the maximum error is smallest for a Subset simulation, which also provides the most accurate and precise reliability indices; this is the absolute winner in the field of the minimization of error indicators. The second best is an Importance sampling with a small minimum and maximum error, mean value as

	AS	eMC	FORM	IS	MC	SS	SSS	min	max
HV(rPF)	0.79	0.77	0.76	0.78	0.67	0.78	0.78	0.66	0.83
S(rPF)	0.0086	0.0068	0.0062	0.0061	0.0055	0.0062	0.0068	0.0046	0.013
Δ(rPF)	0.47	0.45	0.45	0.42	0.54	0.40	0.45	0.28	0.61
GD(rPF,sPF) $[\cdot 10^{-4}]$	4.1	2.9	7.6	2.8	2.8	2.8	2.8	2.2	10.5
C(sPF,rPF)	0.61	0.64	0.57	0.61	0.61	0.54	0.57	0.44	0.79
C(rPF,sPF)	0.038	0.037	0.027	0.034	0.031	0.037	0.034	0	0.064
D(sPF,rPF)	0.016	0.028	0.039	0.018	0.124	0.016	0.020	0.004	0.137
min $\epsilon(\beta_{\text{oPF}})$	0.001	0.001	0.151	0.0005	0.0003	0.0002	0.007	0	0.17
max $\epsilon(\beta_{\text{oPF}})$	0.27	0.28	0.27	0.10	0.35	0.085	0.25	0.057	0.91
E $\epsilon(\beta_{\text{oPF}})$	0.043	0.055	0.228	0.027	0.043	0.021	0.102	0.018	0.24
std $\epsilon(\beta_{\text{oPF}})$	0.057	0.067	0.027	0.025	0.068	0.018	0.053	0.015	0.203

Table 7.34: Example 5: Comparison of performance measures for RBDO utilizing different reliability assessment techniques and an analytical model, namely an Asymptotic sampling (AS), Enhanced Monte Carlo simulation (eMC), First-order reliability method (FORM), Importance sampling (IS), quasi-Monte Carlo simulation (MC), Subset simulation (SS), and Scaled sigma sampling (SSS). Used metrics are a Hypervolume (HV), Spacing (S), Spread (Δ), Generational distance (GD), Two set coverage metric (C(A,B)), Coverage difference of two sets (D(A,B)), and Error of the reliability index ($\epsilon(\beta)$). The datasets: rPF - recalculated Pareto front, sPF - superior Pareto front, oPF - original Pareto front, and rF - recalculated front. The best values are tinged with the green colour, the worst values with the red colour. The scale shows the visual comparison of the results.

well as standard deviation. A First-order reliability method provides precise but not accurate reliability indices for this example; this trend is visible from Figure 7.36, where the original Pareto fronts are translated from the recalculated fronts. The last precise results came from a quasi-Monte Carlo simulation; the problem is in a small number of samples that influences the upper tail of the Pareto fronts. Pareto fronts from MO-RBDO utilizing an Enhanced Monte Carlo simulation show the same trend; the results are even less accurate than from a quasi-Monte Carlo simulation. Figure 12.14 in Appendix 12 shows the box-plots for the performance measures and the mean and standard deviation of the error.

7.5.3 Results for approximation using meta-models and reliability assessment techniques

We selected three best-behaving reliability assessment methods, and we combined them with all meta-models in our MO-RBDO methodology. We present only two best-behaving meta-models according to Table 7.1, namely local meta-models and sparse global meta-model. Appendix 12 contains the data from MO-RBDO using global meta-model. The ranks of all methods, together with the best placings for all five examples, are in Table 7.1; the fifth example is in the fifth row of the table.

We run MO-RBDO with each combination of selected methods ten times to have solid data for our statistics. Our fastest implementation is MO-RBDO using a Subset simulation and sparse global meta-model, the second-best is MO-RBDO utilizing an Asymptotic sampling and local meta-models, and the third-best is MO-RBDO using a Subset simulation and local meta-models. Table 7.35 presents the statistical data from all runs. The number of meta-model simulations is ten to the power of seven.

Figure 7.38 presents all obtained data from optimization runs with some additional information. Except for one Pareto front, the sparse global meta-model works well up to the reliability index equal to 3. This reliability index approximately corresponds with the single objective optimum; a yellow cross depicts this solution. The tails of original Pareto fronts with reliability index larger than three considerably differ from their recalculated counterparts as well as the superior Pareto front. Moreover, the superior Pareto front is bounded to reliability indices [0,4.5], but we keep the recalculated Pareto fronts without any bounding to see their real values. The cost function values are identical for the original and the recalculated individuals, Therefore, the corresponding values are only vertically translated. If the original individual does not have any recalculated counterpart depicted in the objective space, the recalculated reliability index is equal to $\pm\infty$. The best reliability assessment method seems to be the Importance sampling in combination with the sparse global meta-model because the upper tails differ the least in comparison with the other methods. Figure 7.39 shows results from MO-RBDO using local meta-models. The original Pareto fronts from all selected simulation methods have nice congruence with the superior Pareto front. The best simulation method seems to be the Importance sampling because the original data is most closely related to their recalculated values. Except for two individuals from the Pareto fronts calculated using Subset simulation, this method also gave nice results.

Table 7.36 presents the results of performance measures, errors, and rejected individuals due to $\pm\infty$ in the reliability indices after recalculation. The average values for Hypervolume for all combinations of methodologies are almost the same. The minimum value of the Hypervolume is for the Pareto fronts from MO-RBDO using AS and SGMM – the yellow recalculated Pareto front in Figure 7.38, this value is an outlier also visible in Figure 12.15 in Appendix 12 and it is profoundly influencing the mean value. If omitted, the mean value is 0.91, a value comparable to the best placed MO-RBDO using SS and SGMM. A Spacing metric has low values. Therefore, there is the ideal spacing between individuals or at least ideal spacing between individuals in groups. MO-RBDO utilizing a Subset simulation and sparse global meta-model provides

	SGMM			LMM		
	AS	IS	SS	AS	IS	SS
elapsed time [sec]	718	2198	496	554	1299	546
elapsed time [hours]	0.199	0.610	0.138	0.154	0.361	0.152
initial DoE	50	50	50	200	200	200
added samples	400	387	362	138	134	130
analytical g(x)-calls	450	437	412	338	334	330
MM built for opt.	31	31	31	3050	3050	3050
MM built for update	0	0	0	545	547	489
MM-calls	$6.01\cdot10^7$	$8.95\cdot10^7$	$4.31\cdot10^7$	$6.25\cdot10^7$	$8.92\cdot10^7$	$4.89\cdot10^7$

Table 7.35: Example 5: Comparison of statistics for RBDO utilizing an Asymptotic sampling, Importance sampling, and Subset simulation with a sparse global meta-model (SGMM), and local meta-models (LMM).

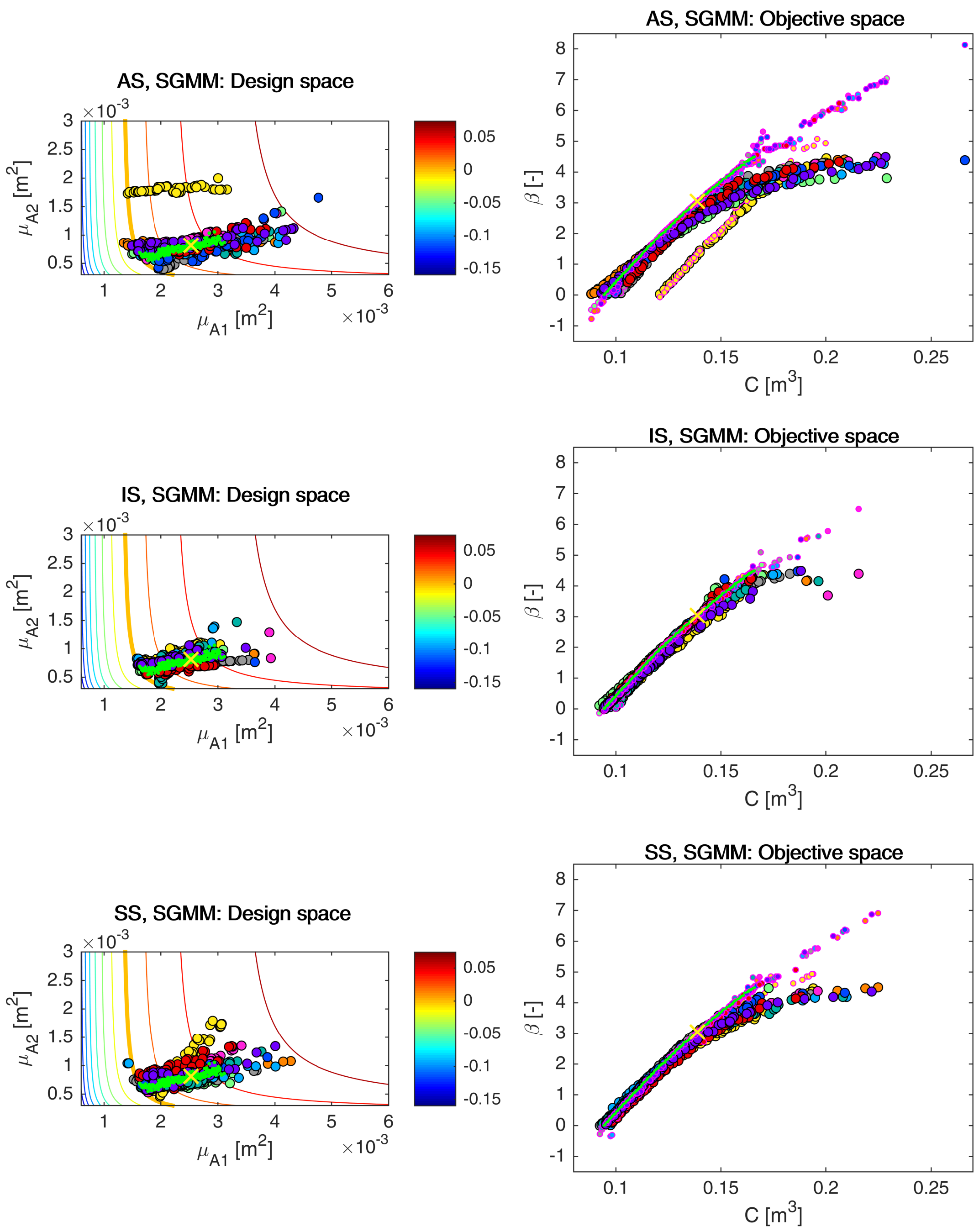


Figure 7.38: Example 5: Pareto sets (circles, left) and Pareto fronts (larger circles with black edges, right) for RBDO utilizing an Asymptotic sampling (AS), Importance sampling (IS), and Subset Simulation (SS) together with a sparse global meta-model (SGMM). The smaller circles with magenta edges in the objective space represent the recalculated Pareto sets to Pareto fronts with quasi-Monte Carlo simulations with $\mathrm{CoV} < 5\%$. The yellow cross is for the result published in [59]. Green dots sets show the superior Pareto set and Pareto front obtained with an analytical model together with a quasi-Monte Carlo simulation. Every colour means a different run. The contour plot in the design space represents an analytical limit state function; the bold contour is for the limit state $g(\mathbf{X}) = 0$.

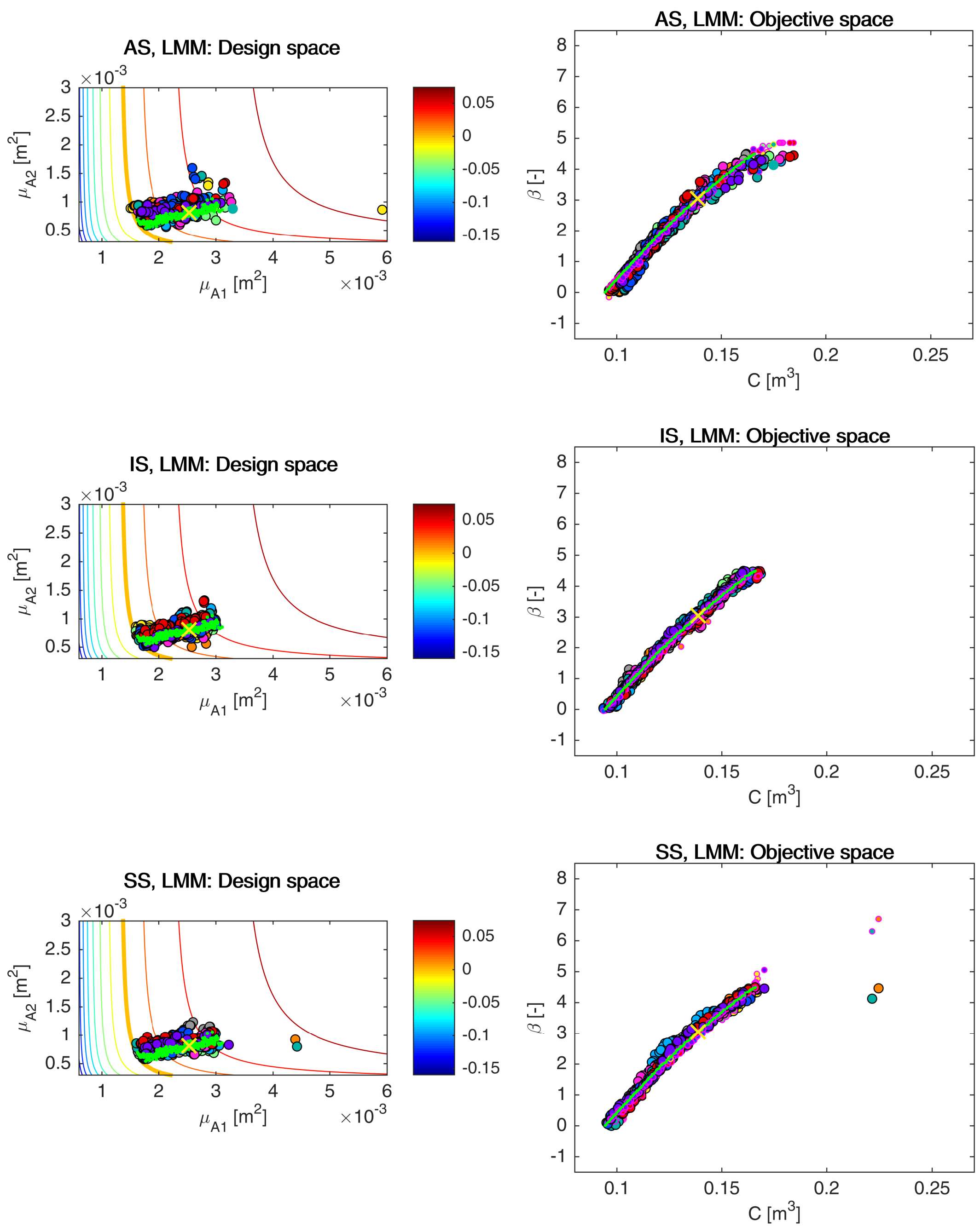


Figure 7.39: Example 5: Pareto sets (circles, left) and Pareto fronts (larger circles with black edges, right) for RBDO utilizing an Asymptotic sampling (AS), Importance sampling (IS), and Subset Simulation (SS) together with local meta-models (LMM). The smaller circles with magenta edges in the objective space represent the recalculated Pareto sets to Pareto fronts with quasi-Monte Carlo simulations with $\mathrm{CoV} < 5\%$. The yellow cross is for the result published in [59]. Green dots sets show the superior Pareto set and Pareto front obtained with an analytical model together with a quasi-Monte Carlo simulation. Every colour means a different run. The contour plot in the design space represents an analytical limit state function; the bold contour is for the limit state $g(\mathbf{X}) = 0$.

	SGMM			LMM				
	AS	IS	SS	AS	IS	SS	min	max
HV (rPF)	0.88	0.89	0.91	0.87	0.89	0.90	0.65	0.96
S (rPF)	$8.6{\cdot}10^{-3}$	$7.8{\cdot}10^{-3}$	$7.6{\cdot}10^{-3}$	$1.0{\cdot}10^{-2}$	$6.1{\cdot}10^{-3}$	$6.9{\cdot}10^{-3}$	$3.6{\cdot}10^{-3}$	$2.3{\cdot}10^{-2}$
Δ (rPF)	0.54	0.57	0.47	0.70	0.58	0.55	0.39	0.88
GD (rPF,sPF)	$2.7{\cdot}10^{-3}$	$5.3{\cdot}10^{-4}$	$8.3{\cdot}10^{-4}$	$8.2{\cdot}10^{-4}$	$5.3{\cdot}10^{-4}$	$5.6{\cdot}10^{-4}$	$3.1{\cdot}10^{-4}$	$1.6{\cdot}10^{-2}$
C(sPF,rPF)	0.76	0.75	0.79	0.86	0.75	0.79	0.52	1.00
C(rPF,sPF)	0.022	0.019	0.023	0.012	0.017	0.014	0	0.068
D(sPF,rPF)	0.050	0.039	0.026	0.063	0.039	0.031	0.006	0.28
$\min \epsilon(\beta_{\mathrm{oPF}})$	0.040	0.017	0.005	0.005	0.003	0.003	$3.5{\cdot}10^{-5}$	0.31
$\max \epsilon(\beta_{\mathrm{oPF}})$	2.3	0.9	1.6	0.8	0.4	0.8	0.2	3.7
$\mathrm{E}\epsilon(\beta_{\mathrm{oPF}})$	0.50	0.15	0.24	0.16	0.10	0.15	0.06	0.78
$\mathrm{std}\epsilon(\beta_{\mathrm{oPF}})$	0.49	0.17	0.32	0.15	0.09	0.15	0.05	0.78
rejected ind.	0.8	0	0.4	0	0	0	0	1.0
Σ rejected ind.	8	0	4	0	0	0	0	8

Table 7.36: Example 5: Comparison of performance measures for RBDO utilizing different reliability assessment techniques namely an Asymptotic sampling (AS), Importance sampling (IS), and Subset simulation (SS) using a sparse global meta-model (SGMM) and local meta-models (LMM). Used metrics are a Hypervolume (HV), Spacing (S), Spread (Δ), Generational distance (GD), Two set coverage metric (C(A,B)), Coverage difference of two sets (D(A,B)), and Error of the reliability index $\epsilon(\beta)$. The used datasets: rPF - recalculated Pareto front, sPF - superior Pareto front, oPF - original Pareto front, and rF - recalculated front. The best values are tinged with the green colour, the worst values with the red colour. The scale shows the visual comparison of the results.

the best distribution among the individuals according to the Spread metric. On the contrary, the worst distribution of individuals has MO-RBDO using an Asymptotic sampling and local meta-models on average. Overall, unary metrics tend to prefer a Subset simulation over an Asymptotic sampling, but the results are not unambiguous. Similar results are also found for binary metrics, which lack a clear preference for the methodology. A Generational distance provides better results for an Importance sampling with any meta-model and a Subset simulation with a local meta-model; their recalculated Pareto fronts are closest to the superior Pareto front. A superior Pareto front weakly dominates at least 75% of individuals from recalculated Pareto fronts on average. These results were obtained with MO-RBDO using an Importance sampling with both meta-models. All solutions were dominated by the superior Pareto front twice in the case of MO-RBDO using an Asymptotic sampling, once with a sparse global meta-model (yellow Pareto front from Figure 7.38) and once with local meta-models (purple Pareto front from Figure 7.39). Similarly, only a small portion of data dominates a superior Pareto front, maximally 2.3% on average for a Subset simulation and sparse global meta-model. In overall, binary metrics tend to prefer an Importance sampling regardless of the meta-model.

If we consider the accuracy and precision of the Pareto fronts, MO-RBDO achieved the best results while using an Importance sampling and local meta-models. If only mean and standard deviation of the error is considered, the second-best results are obtained by the rest of the simulation methods and local meta-models and an Importance sampling and sparse global meta-model. A Subset simulation using local meta-models and an Importance sampling using a sparse global meta-model are slightly more accurate than an Asymptotic sampling using local meta-models. On the contrary, an Asymptotic sampling and a Subset simulation both using local meta-models were slightly more precise than an Importance sampling using a sparse global meta-model. For local meta-models, none of the individuals was rejected after recalculation by a quasi-Monte Carlo simulation with CoV lesser than 5% due to $\pm\infty$ in reliability indices, and only a few individuals were rejected for sparse global meta-model with an Asymptotic sampling and a Subset simulation. Figure 12.15 in Appendix 12 shows the box-plots for the performance measures and the mean and standard deviation of the error.

7.6 Summary results for all examples

We tested our proposed methodology with three different types of meta-models and seven different reliability assessment methods. To keep the results well arranged, we divided the results into three different sub-sections. The first part contains results with a preconditioned quasi-Monte Carlo simulation and different types of meta-models as well as an analytical model. The quasi-Monte Carlo simulation is computationally expensive, especially for low failure probabilities and low coefficient of variation of the probability failure predictors. Therefore, for saving the computational demands, we estimated the needed number of samples for a constant coefficient of variation of the failure probability prediction by a Subset simulation. This part tends to minimize the error of the reliability assessment and concentrate on the error of the meta-models. Since we do not have any benchmark for validation, we constructed the superior Pareto set with the corresponding Pareto front from solutions resulting from the preconditioned quasi-Monte Carlo simulation and the analytical model. We validated the quality of the superior Pareto front with the single-objective optima found in the literature by using the Pareto-efficiency conditions. The solutions in the superior Pareto fronts dominate the single-objective optima with only one exception, Example 4. Unfortunately, AOUES et al. in [3] published only the values of the cost function values with the corresponding reliability indices and therefore, we could not verify the published values with our methodology as we did for the other examples. Figure 7.40 shows superior Pareto sets and superior Pareto fronts compared to all single-objective optima. The first column of images represents the design space, the second column is dedicated to the objective space, and the third column zooms in the single objective optima compared to our superior Pareto fronts. The second part embraces the results with an analytical model and different types of reliability assessment techniques. This part concentrates on the minimization of the model error and investigates the error of the reliability index prediction. Finally, the third part interconnects the best meta-models with the best reliability assessment methods. We selected three best-performing reliability methods the same for all tested examples and two best-performing meta-models differing from example to example. The most important criteria for the selection of methods are mean of the reliability index error and its standard deviation from our point of view. Therefore, we evaluated the Pareto-efficiency of these two criteria, and the result is shown in Table 7.1. Best-performing reliability assessment methods were an Importance sampling, Subset simulation, and Asymptotic sampling in overall.

We presented five examples of testing our methodology. Each example differs among others. The first example contains only one limit state function using two deterministic design variables and two stochastic variables from normal space. The second example is much more complicated than the first one for its nonlinearity of one limit state function; overall, the system reliability comprises two components connected into the series system, and therefore two limit

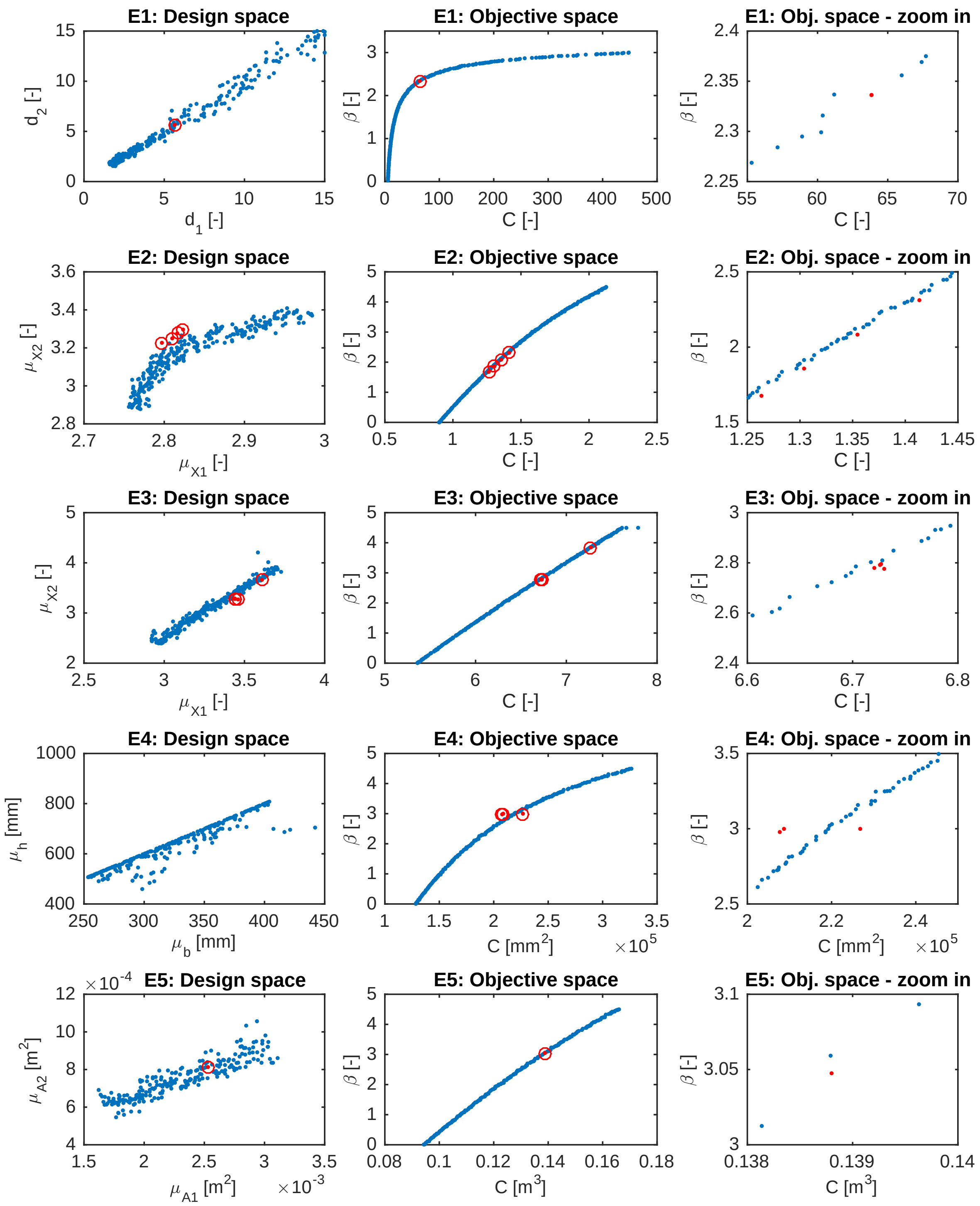


Figure 7.40: Superior Pareto sets and superior Pareto fronts (blue sets) are compared to the single-objective optima borrowed from literature (red sets).

state functions are defined. The two input variables are stochastic design variables from normal space. The third example works with three limit state functions that are connected into the series system; the two input variables are stochastic design variables from normal space. The first three examples are mathematical problems with different degrees of nonlinearity, which is concentrated only into the limit state functions. The fourth example is a short column under oblique bending, where the ultimate plastic state as the limit state reveals high nonlinearity. The two stochastic design variables are from lognormal space, and four stochastic variables are from Gumbel and Weibull spaces – 6 non-normal stochastic variables in total. The fifth example is a 23-bar truss; the mid-span displacement represents the limit state function. Two stochastic design variables are from lognormal space, and eight stochastic variables are from lognormal and Gumbel space – 10 non-normal stochastic variables in total. A summary of the presented examples is in Table 7.37.

Example	1	2	3	4	5
Type of problem	math.	math.	math.	short column	23-bar truss
A number of det. des. var.	2	0	0	0	0
A number of stoch. des. var.	0	2	2	2	2
A number of stoch. var.	2	0	0	4	8
The total number of var.	4	2	2	6	10
A number of LSFs	1	2	3	1	1
Type of a reliability system	component	series	series	component	component
Type of a stochastic space	normal	normal	normal	LN-GUM-WEI	LN-GUM

Table 7.37: Description of presented examples. Abbreviations: math. - a mathematical problem, det. - deterministic, stoch. - stochastic, des. - design, var. - variables, LSFs - limit state functions, LN - lognormal, GUM - Gumbel - WEI - Weibull.

Each example reveals a different type of problem that we had to solve. **Example 1** is complicated because of its small design space and the nonlinearity of the limit state function. The local meta-models suffered from the restriction of the design space, and we found that this type of meta-model is not suitable if the general setting, which we used for all examples, is used. We could improve the behaviour of the meta-model with the extension of the influence domain. However, if the design space is so small that the Pareto set is spread over its entire space or at least its entire band including both extreme intervals of variables, the influence domain would merge with the entire design space and the local meta-model would not differ from the global meta-model. Both global meta-models work well with minimal errors, a sparse global meta-model even better than a dense meta-model. Different reliability assessment methods using an analytical model perform well except for an Enhanced Monte Carlo simulation and Scaled sigma sampling. The best performing method for Example 1 is an Importance sampling if we take into account only the reliability index. Because of the poor performance of the local meta-models, we combined only global meta-models with the Asymptotic sampling, Importance sampling, and Subset simulation for results discussion. The Importance sampling using both versions of global meta-models performed even better than the rest of the reliability assessment techniques utilizing an analytical model which is evident from Figure 7.41. If we compare the mean value of the error with the number of analytical model evaluations for whole RBDO, the most economical method is the Importance sampling using a global meta-model with an acceptable error compared to the evaluation with the analytical model as evident from Figure 7.42.

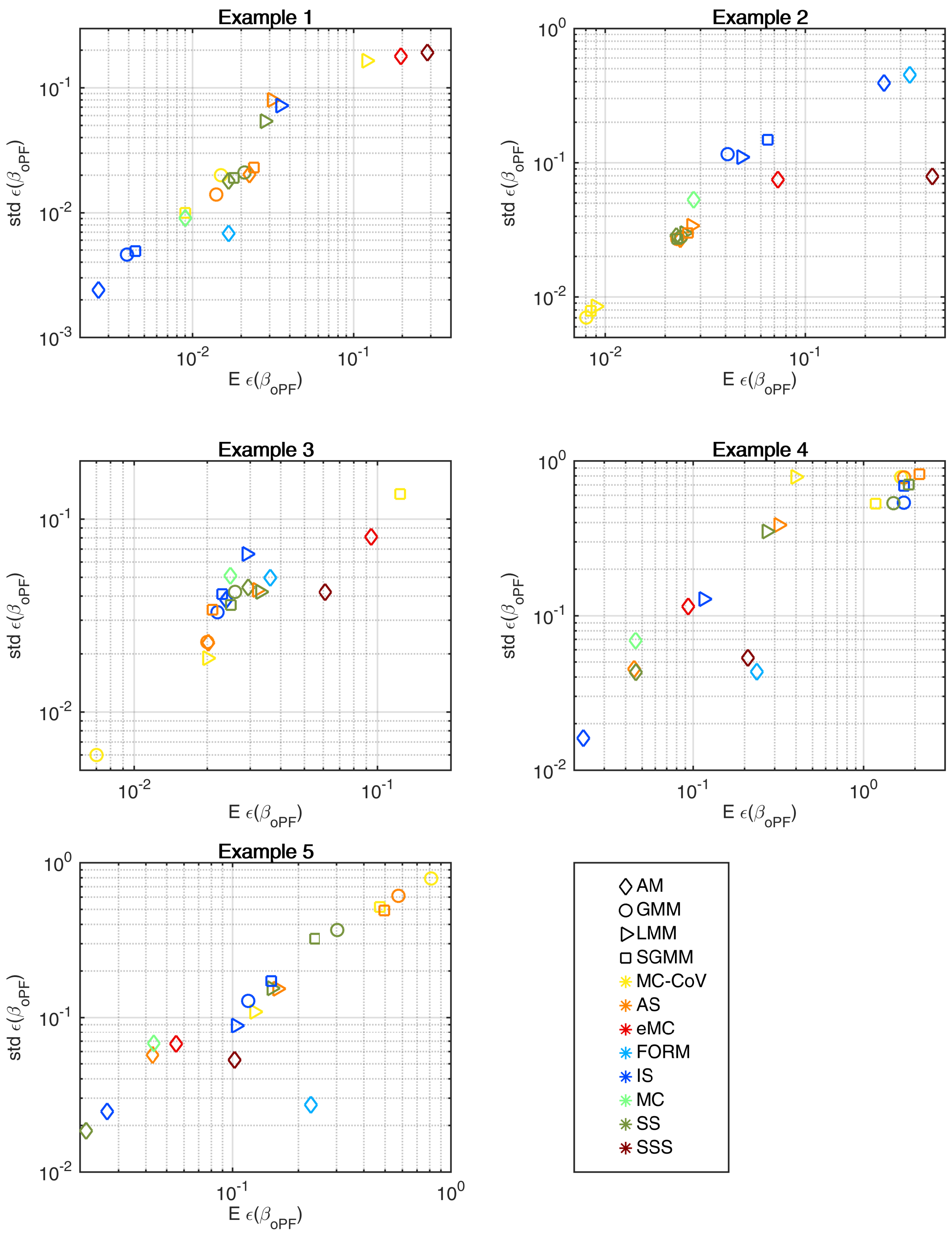


Figure 7.41: Scatter plot of the mean value and standard deviation of the error for all examples.

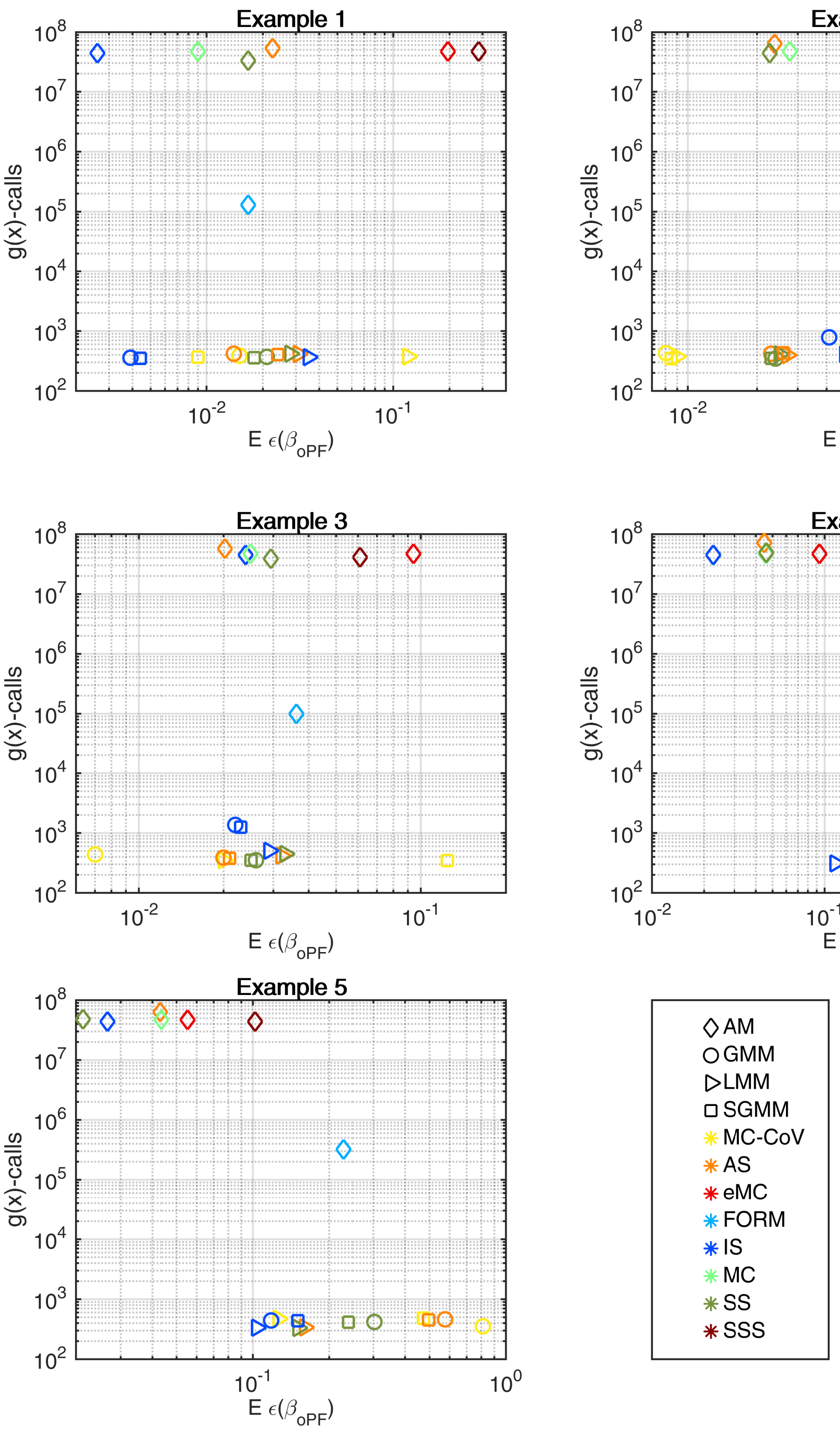


Figure 7.42: Scatter plot of the mean value of the error and number of analytical model simulations for all examples.

Overall, both global meta-models performed very well in this example with a minimum mean error. Table 7.38 compares both error indicators, and the number of analytical function simulations by Pareto-efficiency conditions; an Importance sampling with an analytical model and both global meta-models are the most economical low error methods for Example 1.

Example 2 is complicated for its high nonlinearity and the series system reliability. A connection of two limit state functions creates a bounded safe space, but solutions are only bound to one limit state. Our RBDO using a preconditioned quasi-Monte Carlo simulation simulating all meta-models performed great with small error indicators. If we compare reliability assessment methods using an analytical model, an Importance sampling and First-order reliability method had difficulties with the reliability index precision. Both methods use the same system reliability evaluation principle, differing only in the evaluation of partial component reliability. Besides, Importance sampling uses the translation of the probability density to a design point, which is searched in the same way as First-order reliability method. Therefore, the problem of these methods is in the high degree of nonlinearity of one component limit state function. A Scaled sigma sampling using an analytical model had an even larger mean error of the reliability index but a smaller standard deviation of this error in comparison with an Importance sampling and FORM. An Asymptotic sampling and Subset simulation had comparable error results for the analytical model and both global meta-models. The results from an Importance sampling are interesting because the mean and the standard deviation of the reliability index error for the results obtained by the analytical model were greater than for results obtained from both meta-models; the error for the meta-models and the Importance sampling were subtracted from each other. Table 7.38 shows the ranks if Pareto-efficiency conditions are used for a mean and standard deviation of the reliability index error and the number of simulations of the analytical model. The best-behaving methods, according to these three criteria, are RBDO using a preconditioned quasi-Monte Carlo simulation with local meta-models and sparse global meta-model.

The complexity of **Example 3** is in a serial system connecting three limit state functions that create a bounded safe space and the plateau of the failure region around the limit state border. The solutions in the design space converge into the intersection of two limit states. This intersection and the plateau is hard to imitate by the sparse global meta-model; the sparse global meta-model is easily overfitted in this case. The mean and the standard deviation of the reliability index error for RBDO using a preconditioned quasi-Monte Carlo simulation and the sparse

	MC-CoV			AS	eMC	FORM	IS	MC	SS	SSS	AS	IS	SS	AS	IS	SS	AS	IS	SS
No.	LMM	GMM	SGMM	AM							GMM			LMM			SGMM		
1	3	4	2	5	6	2	1	2	4	7	3	1	3	5	3	5	4	1	2
2	1	2	1	3	6	7	7	5	3	5	3	5	2	3	4	3	4	6	2
3	1	1	1	2	6	4	4	5	4	4	2	3	2	3	3	3	2	3	2
4	2	2	2	2	3	1	1	2	2	1	3	3	2	1	1	1	3	4	2
5	3	2	5	2	2	1	1	2	1	2	6	2	4	2	1	1	5	2	3

Table 7.38: Sorting to fronts according to Pareto efficiency conditions with three different criteria, namely the mean and the standard deviation of the reliability index error, and a number of analytical model simulations. The best solution has rank equal to one and tinted with the green colour. The yellow colour represents an average solution, and the red colour indicates the worst solution.

global meta-model were the largest from all tested methods. In contrast, the global meta-model with a dense Gram matrix used by the preconditioned quasi-Monte Carlo simulation had the lowest error rate when considering the mean value and the standard deviation of the reliability index error as evident from Figure 7.41. The preconditioned quasi-Monte Carlo simulation using the local meta-models has a similar error as the Asymptotic sampling using the analytical model and the global meta-model. A very nice result, however, is that the accuracy and precision of the reliability index from an Asymptotic sampling is the same from the global meta-model as from the analytical model. As shown in Figure 7.42, RBDO using an Asymptotic sampling and an analytical model needs several orders of magnitude more simulation of this analytical meta-model than the global meta-model. However, both Importance sampling and Subset simulation using the global meta-model performed well with a low number of analytical model simulations. If we compare the mean and standard deviation of the reliability index error and the number of analytical model simulations via Pareto-efficiency conditions, the best-behaving methods are RBDO with preconditioned quasi-Monte Carlo simulations using all three types of meta-models.

Example 4 is the first application of a simple construction having one limit state function and six random variables that are from a non-normal space. Thus, the complexity of this example lies also in the non-linearity of the transformation. The limit state function has a very steep trend with low design variables inputs that is hard to imitate by the global meta-models. The last complexity is in the constraining of the design space by the inequality constraints. According to Figure 7.41, the reliability indices obtained by all the reliability assessment methods using an analytical model were more precise than the results from obtained by any meta-model. If we consider the accuracy of the results, an Importance sampling using local meta-models got better results than a Scaled sigma sampling and FORM both using the analytical model. From all meta-models, local meta-models got more precise and accurate results if we combined them with advanced simulation techniques, namely an Importance sampling, Subset simulation, and Asymptotic sampling. If we combined local meta-models with a preconditioned quasi-Monte Carlo simulation, the results were more accurate than any global meta-model, but the precision is worse than most of the global meta-models and simulation techniques. Using advanced simulation techniques, the behaviour of meta-models has mostly improved. The exception is the sparse global meta-model, where results with preconditioned quasi-Monte Carlo simulation are more precise and accurate than with advanced simulation techniques. In the global meta-model, an Asymptotic sampling slightly impaired accuracy in comparison with preconditioned quasi-Monte Carlo simulation. A Subset simulation turned out best for the global meta-model. If we compare the accuracy concerning the number of simulations of the analytical model as depicted in Figure 7.42, then the clear winner is an Importance sampling with the analytical model and with the local meta-models. The former results are more accurate than the latter, but the latter is computationally less demanding by five orders of magnitude. If we compare the accuracy, precision and amount of evaluation of the analytical function as recorded in Table 7.38, then the best methods are FORM, Importance sampling and Subset simulation while using an analytical model and Importance sampling and Subset simulation while using local meta-models. FORM is the least accurate of all reliability methods while using an analytical model, but its results are the third best in terms of precision. FORM as an approximation method also requires orders of magnitude fewer simulations of the analytical function than simulation techniques.

The complexity of **Example 5** is in the number of stochastic variables; this example has ten non-normal variables from lognormal and Gumbel space. Therefore, the nonlinearity is in the component limit state function as well as in the transformation. If a preconditioned quasi-Monte Carlo simulation was used, RBDO with both global meta-models had difficulties with reliability indices greater than approximately 3. The reason is that the meta-limit state is shifted closer to the mean values for the extreme values of variables in comparison with the analytical limit state. These extreme regions are described by sampling only for higher values of the reliability indices. The global meta-model may be less accurate at these extreme values due to two reasons; first, because of insufficient description by support points of DoE, and second, the trend of the function in extreme values is worse predicted due to missing information of the function beyond the meta-model bounds. Local meta-models assembled for design variables with subsequent higher values of reliability indices avoid the latter problem because they are assembled only on small subspace around the actual values of design variables. This behaviour is evident in Table 7.32 and Figure 7.41. Figure 7.41 shows that all simulation methods achieve better results using the analytical function. Local meta-models used by an Importance sampling provide a better prediction of the reliability indices than if a preconditioned quasi-Monte Carlo simulation uses them. An Importance sampling proposes better candidate support points for a DoE update than other simulation methods. The enhancement of the global meta-model by the samples from the Importance sampling is also evident from Figure 7.41, where the mean and the standard deviation of the reliability index error is less than for other simulation methods using both versions of global meta-models. For most of the data in Figure 7.41, the mean value of the error increases with the standard deviation. The biggest exception is FORM, which proposes a systematically wrong reliability index, i.e. large mean value of the error with small standard deviations; therefore, the reliability indices are precise but not accurate. Considering the number of simulations of the analytical function concerning the mean of the reliability index error, as shown in Figure 7.42, then the best-behaving method is RBDO using local meta-models simulated by an Importance sampling and RBDO using a Subset simulation with an analytical model. The latter variant is more accurate than the former one, but the latter needs five orders of magnitude more analytical evaluations than the former. If we consider three criteria into the Pareto-efficiency conditions as mean of the reliability index error and its standard deviation and the number of analytical function simulations, the best approximation methods are FORM, Importance sampling and Subset simulation while using an analytical model, and an Importance sampling and Subset simulation while using local meta-models, as evident from Table 7.38.

Chapter 8

Conclusions

This thesis aimed to develop a methodology that provides fast and computationally simple solution of multi-objective reliability-based design optimization. However, our formulation of MO-RBDO is different from the formulation in the literature. Several papers [44, 118, 121, 122, 178, 184] define multi-objective RBDO essentially as an extension of single-objective RBDO so that there are several objective functions, but the reliability stays in the constraints inequalities. We formulate the reliability as an objective function because we want to answer the question: "How much does the reliability cost?" with the Pareto fronts as an answer. That is why we formulated the assignment of the reliability problem as one of the objective functions. Since this formulation is, to the author's knowledge, unique in the structural optimization field, we looked for the least computationally demanding implementation, optimized its implementation and validated the methodology on selected examples. In terms of computational complexity, the greatest demands appeared in the area of selection of (i) a multi-objective algorithm, (ii) fast and reliable evaluation of reliability, (iii) and effective replacement of the original model with a meta-model. We solved the (i) objective problem of computational complexity by studying various literature and found the most used and most recommended algorithm, which does not use the archive or the external population to preserve elitist solutions. Because our methodology consists of updated DoE in each generation, recalculating the external archive for each new generation would dramatically increase the computational demands. We solved the (ii) objective problem by finding seven different methods for evaluating the reliability in the available literature and using these methods together with the original model in multi-objective RBDO to compare their qualities for our problem. We then mapped the resulting Pareto sets to recalculated "Pareto" fronts using a quasi-Monte Carlo simulation with a guaranteed coefficient of variation below 5%. We used quotation marks because these "Pareto" fronts may not be Pareto-optimal, as the Pareto-efficiency conditions are not used in the mapping. We used these recalculated fronts only to evaluate the errors of the reliability methods. However, for the computation of other metrics evaluating the quality of the Pareto fronts in general, we selected only the Pareto-optimal solutions from these fronts. We solved the (iii) objective problem by formulating two special forms of the radial basis functions model, namely local and sparse global meta-models, and we compare these meta-models with the classical global meta-model. We also compared their error first separately with the preconditioned quasi-Monte Carlo simulation, and then we applied them with selected best reliability methods. On all these results, we again evaluated the errors of the reliability index.

In the introductory chapter, we outlined the background of our research, together with significance. In Chapter 2, we focused on various methods of reliability-based design optimization.

The third chapter describes optimization in general, possibilities of solving multi-objective optimization via single-objective optimization solvers including disadvantages of this solution, the transition from single-objective optimization to multi-objective version, the main principles of multi-objective optimization including principles of domination, and the main differences between multi- and many-objective optimization. This chapter also details the multi-objective optimization solver Non-dominated sorting genetic algorithm II as we use it. It is an evolutionary elitist-preserving algorithm that employs two main principles, the non-dominated sorting and a crowding distance as a diversity preserving mechanism. To create a new generation, we use a tournament selection with two criteria – a non-dominated sorting ranks and a crowding distance to select parents into the mating pool, simulated binary crossover operator creates two offspring members from two parental members, and Gaussian mutation operator mutates some individuals with a prescribed probability. The non-dominated sorting approach also works with the constraints. Therefore we do not use any penalty functions for them. This chapter terminates with performance measures evaluating the quality of the Pareto fronts; they are divided into three categories: metrics evaluating the closeness to the Pareto front, diversity among non-dominated solutions, and metrics evaluating closeness and diversity at the same time. We use these metrics for rating the resulting Pareto fronts in Chapter 7.

The fourth chapter serves as a brief introduction into the statistics describing the distribution of a single variable, including the central tendency and dispersion. It continues with the introduction of the failure probability definition for components as well as systems. A First-order reliability method is the main representative of the approximation techniques in this work; however, we focus more on simulation techniques, namely a Monte Carlo simulation, Importance sampling, Asymptotic sampling, Subset simulation, Enhanced Monte Carlo simulation, and Scaled sigma sampling. Since we work with the series systems in Chapter 7, we described the evaluation of the system reliability for each method. Each approximation and simulation method shows the failure probability evaluation, including graphical illustrations.

The fifth chapter describes the meta-models. Two main representatives of interpolation models are described in more details, namely a Radial basis functions model and Kriging. In our methodology, we work with three types of Radial basis functions model definition—-a global meta-model using a dense Gram matrix, a sparse global meta-model applying only an influence domain for a construction sample resulting in a sparse Gram matrix on the whole domain, and local meta-models constructing a dense Gram matrix only on the influence domain. The two latter formulations are our author's work. In this chapter, we also discuss the implementation details and the optimization of the meta-model performance, since the meta-model assembly and its simulation takes a significant portion of the overall evaluation time. A compute-intensive segment is the evaluation of the interpoint distances; these distances are used both for meta-model assembly and simulation of the meta-model. We propose five different methods for the interpoint distances evaluation, and we compare them for different numbers of support points of Designs of Experiments in terms of computational time. For a dense and sparse Gram matrix, we also compare computational demands in terms of time and memory usage, and a sparse meta-model error compared to a dense meta-model. The initial meta-models need updates to gain better precision in the domain of importance. Therefore, we divided these updates into several categories based on the meta-model usage. Different updates are used for reliability assessment and reliability-based design optimization. In RBDO, updates also differ if the meta-model replaces an objective function and constraints.

The sixth chapter is the main part of the thesis. We propose two updating procedures of the meta-models, the first updates the meta-model independently of optimization, and the second updates the meta-model during the optimization. From our point of view, the latter update is more suitable for RBDO because the simulation techniques' samples also serve as potential candidates for updating the Design of Experiments. This update is described in detail for the global, sparse, and local meta-models. The main idea of this update is the mentioned use of samples from simulation methods. In each generation of the genetic algorithm, selected samples update the Design of Experiments. Depending on the meta-model type, either one global meta-model is generated for each generation or a local meta-model for each individual in each generation. The algorithm selects samples from the simulation method for the DoE update using a special multi-objective algorithm taking into account the distance to the limit state function and the space-filling property.

The seventh chapter presents the verification of the proposed methodology. We showed that our proposed methodology utilizing updated meta-models is better than MO-RBDO utilizing FORM and an original model since FORM needs at least two orders of magnitude more evaluations of the original model than our proposed method as evident from Figure 7.42. We have always been able to achieve better accuracy in the reliability index with at least one meta-model. This discovery is an interesting entry step for further research into how to fill the Pareto front individuals if we add or remove the number of support points in the updated DoE. There are three solutions to this problem. First, it is possible that adding more support points into DoE will greatly improve the accuracy of the reliability indices, as more information about the real problem gets to DoE. Second, DoE may be adequately filled, and the addition of support points do not improve the accuracy of the reliability index. Third, with the addition of data to DoE, the meta-model will be subsequently overfitted and be prone to numerical noise. However, our acquired solutions using meta-models always dominate solutions obtained by using FORM calling the analytical model both in the accuracy of the solution and in the number of simulations of the analytical function.

The global meta-model is the least sensitive to the choice of a simulation method. Conversely, the local meta-models are most sensitive to its choice. If we evaluate the ranks for all used methods concerning the mean and standard deviation of the error, and compare these ranks for a preconditioned quasi-Monte Carlo simulation and averaged values via advanced simulation techniques (AST) as shown in Table 8.1 or Figure 7.41, then the sum of the differences across all examples is the smallest just for the global meta-model and largest for local meta-models. If we evaluate the influence of a preconditioned quasi-MC and AST for each example, as shown in Table 8.1, then the behaviour improved ones, twice, and twice by using advanced simulation techniques for the GMM, LMM, and SGMM. For real applications, it will be necessary to test the meta-model accuracy and precision as an intermediate step for selecting a meta-model type. In case of inappropriate behaviour of the meta-model for the given example, the update procedure cannot improve it to be a credible replacement for the original model.

If we examine the influence of the meta-model on the change of the result of the simulation method using the analytical model, Table 8.2a) shows the trends of accuracy and precision for Example 1 – 3. These examples use mathematical functions as limit state functions that are relatively smooth if we omit the intersections of functions. The global meta-models for these three examples dominate, as both accuracy and precision have improved or only slightly deteriorated regardless of the reliability assessment method compared to the analytical model and the simulation method. The table already contains subtracted values, where a negative number means improvement and a positive number means deterioration. Conversely, local meta-models dominate for application on simple structures as evident from Table 8.2b) for Examples 4 – 5. Limit state functions are no longer as smooth as in Examples 1 – 3, but they have more failure modes and therefore more design points, the design space is larger, and the limit state functions are more nonlinear as in Examples 1 – 3. Global meta-models then fail to capture this

	GMM			SGMM			LMM		
Example	pqMC	$\bar{\mathrm{SM}}$	diff.	pqMC	$\bar{\mathrm{SM}}$	diff.	pqMC	$\bar{\mathrm{SM}}$	diff.
1	13	5.67	7.33	5	7	2.00	7	11.67	4.67
2	3	6.33	3.33	2	7.33	5.33	1	8	7.00
3	2	4.33	2.33	10	4.67	5.33	1	7	6.00
4	8	10	2.00	8	12	4.00	10	6	4.00
5	6	8.67	2.67	10	8.33	1.67	12	6.33	5.67
	32	35	**3.00**	35	39.33	**4.33**	31	39	**8.00**

Table 8.1: Change in the behaviour of the meta-model depending on the simulation method. Ranks evaluate the Pareto-efficiency conditions for a mean and standard deviation of the error. For individual ranks see Table 12.2. Legend: pqMC - ranks for a preconditioned quasi-Monte Carlo simulation, $\bar{\mathrm{SM}}$ - averaged ranks for an Asymptotic sampling, Importance sampling, and Subset simulation, diff. - difference between ranks for pqMC and $\bar{\mathrm{SM}}$, GMM - a global meta-model, LMM - local meta-models, SGMM - a sparse global meta-model. The green tinge represents an average improvement of the meta-model behaviour with the advanced simulation methods in comparison with a preconditioned quasi-Monte Carlo simulation, the red tinge worsening of the same behaviour.

complicated trend, and conversely, local meta-models work very well here due to their smaller influence domain. Also, local meta-models have better accuracy and precision in comparison with both types of meta-models, although the updating procedure ended up with fewer points in DoE in total; they are therefore better copying the original model, and at the same time they need less evaluation of the original model for accuracy and precision.

The resulting computational times published in the tables in Chapter 7 are dependent on our implementation, and the hardware we used that mentioned in Tables 11.1 and 11.2. We know that we have reserves in optimizing the code used, an experienced programmed could certainly achieve even better computing times. We implemented our methods in the MATLAB, which is a computational environment that uses the interpreted language. Certainly, computing demands would also decrease if a programmer reprogrammed our methodology into a compiled language, such as C++. We used MATLAB as it is a leading integrated environment for scientific and technical computing, modelling, simulation, presentation and data analysis. It is a tool for both comfortable, interactive work and development of a wide range of applications. It provides its users powerful graphical and computing tools, as well as extensive function libraries along with a powerful fourth-generation programming language. At the same time, it is possible to work with multi-dimensional arrays, with which MATLAB works vectorized and thus significantly speeds up calculations. MATLAB also provides a library for parallel computations that can further reduce the computational time of a problem. We have also used this library for longer computations and at the same time have verified the speed-up of our implementation against a purely serial solution. Although it could change the computational complexity of a problem in another programming language and on different hardware, the number of simulation of the original model always remains approximately the same. It cannot be said that it will be absolutely the same as it is a stochastic calculation and therefore there will always be some deviation in all the achieved reliability index and the number of support points added to the updated DoE as evident from Table 12.5 in Appendix 12.

Further study needs to be carried out in the field of system reliability. In this thesis, we

	GMM			SGMM			LMM		
Example	AS	IS	SS	AS	IS	SS	AS	IS	SS
accuracy change									
1	-0.008	0.001	0.004	0.001	0.002	0.001	0.008	**0.032**	0.011
2	$-4{\cdot}10^{-4}$	**-0.207**	0.001	0.002	-0.183	$\mathbf{2{\cdot}10^{-4}}$	0.003	-0.200	0.002
3	$-3{\cdot}10^{-5}$	-0.002	-0.003	0.001	-0.001	-0.004	0.012	0.005	0.004
precision change									
1	-0.006	0.002	0.003	0.003	0.003	0.001	0.059	**0.070**	0.036
2	$\mathbf{1{\cdot}10^{-4}}$	-0.275	-0.001	0.003	-0.242	-0.001	0.007	**-0.280**	0.002
3	$5{\cdot}10^{-4}$	-0.006	-0.002	0.011	0.002	-0.009	0.020	0.028	-0.002

(a) Example 1 – 3

	GMM			SGMM			LMM		
Example	AS	IS	SS	AS	IS	SS	AS	IS	SS
accuracy change									
4	1.679	1.699	1.451	**2.080**	1.706	1.794	0.272	0.092	0.225
5	0.532	0.091	0.281	0.453	0.124	0.217	0.117	**0.077**	0.130
precision change									
4	0.737	0.522	0.491	**0.777**	0.674	0.662	0.341	0.112	0.308
5	0.553	0.104	0.349	0.435	0.148	0.305	0.097	**0.064**	0.136

(b) Example 4 – 5

Table 8.2: Accuracy and precision change in the behaviour of the simulation method depending on the meta-model in comparison with the analytical model for a) Example 1 – 3 and b) Example 4 – 5. Legend: GMM - a global meta-model, LMM - local meta-models, SGMM - a sparse global meta-model, IS - an Importance sampling, AS - an Asymptotic sampling, SS - a Subset simulation. The green-yellow-red colour scale represents a deterioration of the method using a meta-model, where green means the least deterioration, while a red shows the worst. Blue colour represents an improvement.

did not solve any parallel systems, and a presented Importance sampling and FORM methods have worked to a limited extent for serial systems. These two methods solve the reliability of systems by an extension above themselves, and because they use this extension in the form of the approximation of the surface by the tangent hyperplane, the result may be inaccurate for limit state functions with high nonlinearity. An Importance sampling is otherwise a very accurate method for reliability estimation. Therefore, its use in the system reliability needs to be further explored. This work could thus be deepened in terms of application to general systems. This thesis also worked only with uncorrelated input variables. However, many applications also include correlated variables, especially when it comes to, for example, loads, or products of the same material made in the same series. Therefore, the study could be repeated for applications with correlated variables. One application of our methodology (Example 4) was carried out on the constrained design space. However, constrained design domains can be significantly more complex, hence the effect of restriction on adding points to DoE and assembling meta-models needs to be investigated. In order to evaluate the reliability of the problem, it is necessary to have a model assembled beyond the domain bounds if the design individual is on the border of this restricted domain. In this work, two objective functions are formulated for each example; the cost function and the reliability evaluated by the β-index. Constraints are defined in the sense of bounding the Pareto front from the bottom and top. Extending the number of objective functions to more than three would mean that multi-objective optimization will become many-objective, and the methodology we propose may not be effective in finding Pareto-optimal solutions. For more than three objective functions, it would be necessary to choose an algorithm other than NSGA-II while maintaining the idea of an updated DoE. Another study would be suitable for determining the influence of more complex constraints on the behaviour of our methodology. In the literature, we have also found examples of RBDO, in which the cost function includes reliability; it is a so-called total cost function [59, 76]. The price of the structure does not only depend on the initial costs but also on the expected failure costs. The initial costs are associated with erection and operation of the structure during its design service life without any failure. Whereas the expected failure costs depend on all the costs associated with the failure and its probability, such as the estimated repair cost, loss of life or the cost of disruption of normal use [76]. If the expected failure costs did not differ from the reliability of the structure, then the same error, that we get for the reliability with the application of the meta-model and advanced simulation techniques, would be reflected in the cost function. The computational complexity of the problem would remain the same. However, if more reliability were to be determined for one example, then the computational complexity would increase with each added reliability objective.

Bibliography

[1] H. Afshari, W. Hare, and S. Tesfamariam. Constrained multi-objective optimization algorithms: Review and comparison with application in reinforced concrete structures. *Applied Soft Computing*, 83:105631, 2019.

[2] G. L. Ang, A. H.-S. Ang, and W. H. Tang. Optimal importance-sampling density estimator. *Journal of engineering mechanics*, 118(6):1146–1163, 1992.

[3] Y. Aoues and A. Chateauneuf. Benchmark study of numerical methods for reliability-based design optimization. *Structural and multidisciplinary optimization*, 41(2):277–294, 2010.

[4] J. S. Arora. *Optimization of structural and mechanical systems*. World Scientific, 2007.

[5] C. Ash. *The probability tutoring book: an intuitive course for engineers and scientists (and everyone else!)*. IEEE Press Piscataway, NJ, USA, 1993.

[6] S. Asmussen and P. W. Glynn. *Stochastic simulation: algorithms and analysis*, volume 57. Springer Science & Business Media, 2007.

[7] S. Au, J. Ching, and J. Beck. Application of subset simulation methods to reliability benchmark problems. *Structural Safety*, 29(3):183–193, 2007. A Benchmark Study on Reliability in High Dimensions.

[8] S.-K. Au and J. L. Beck. Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics*, 16(4):263–277, 2001.

[9] S.-K. Au and Y. Wang. *Engineering risk assessment with subset simulation*. John Wiley & Sons, 2014.

[10] J. Bader and E. Zitzler. Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19(1):45–76, 2011. PMID: 20649424.

[11] A. Basudhar and S. Missoum. Adaptive explicit decision functions for probabilistic design and optimization using support vector machines. *Computers & Structures*, 86(19):1904–1917, 2008.

[12] A. Basudhar, S. Missoum, and A. Harrison Sanchez. Limit state function identification using support vector machines for discontinuous responses and disjoint failure domains. *Probabilistic Engineering Mechanics*, 23(1):1–11, 2008.

[13] P. Beaurepaire, H. Jensen, G. Schuëller, and M. Valdebenito. Reliability-based optimization using bridge importance sampling. *Probabilistic Engineering Mechanics*, 34:48–57, 2013.

[14] M. P. Bendsøe and O. Sigmund. *Optimization of structural topology, shape, and material*, volume 414. Springer, 1995.

[15] B. Betrò. Bayesian methods in global optimization. *Journal of Global Optimization*, 1(1):1–14, 1991.

[16] H.-G. Beyer and K. Deb. On self-adaptive features in real-parameter evolutionary algorithms. *IEEE Transactions on evolutionary computation*, 5(3):250–270, 2001.

[17] B. Bichon, M. Eldred, L. Swiler, S. Mahadevan, and J. McFarland. Efficient global reliability analysis for nonlinear implicit performance functions. *AIAA journal*, 46(10):2459–2468, 2008.

[18] B. J. Bichon. *Efficient Surrogate Modeling for Reliability Analysis and Design*. PhD thesis, Faculty of the Graduate School of Vanderbilt University, Nashville, Tennessee, May 2010.

[19] J.-F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal. *Numerical optimization: theoretical and practical aspects*. Springer Science & Business Media, 2006.

[20] A. Bourchy, L. Barnes, L. Bessette, F. Chalencon, A. Joron, and J. M. Torrenti. Optimization of concrete mix design to account for strength and hydration heat in massive concrete structures. *Cement and Concrete Composites*, 103:233–241, 2019.

[21] U. Bourgund and C. Bucher. Importance sampling procedure using design points (ispud)-a user's manual. *Institute of Engineering Mechanics, University of Innsbruck*, 1986.

[22] K. Breitung. Asymptotic approximations for multinormal integrals. *Journal of Engineering Mechanics*, 110(3):357–366, 1984.

[23] K. Breitung. 40 years FORM: Some new aspects? *Probabilistic Engineering Mechanics*, 42:71–77, 2015.

[24] C. Bucher. Asymptotic sampling for high-dimensional reliability analysis. *Probabilistic Engineering Mechanics*, 24(4):504–510, 2009.

[25] C. Bucher. *Computational Analysis of Randomness in Structural Mechanics: Structures and Infrastructures Book Series*, volume 3. CRC Press, 2009.

[26] C. Bucher and U. Bourgund. A fast and efficient response surface approach for structural reliability problems. *Structural safety*, 7(1):57–66, 1990.

[27] C. G. Bucher. Adaptive sampling — an iterative fast Monte Carlo procedure. *Structural Safety*, 5(2):119–126, 1988.

[28] M. D. Buhmann. *Radial basis functions: theory and implementations*, volume 12. Cambridge university press, 2003.

[29] S. A. Burns. *Recent advances in optimal structural design*. ASCE Publications, 2002.

[30] G. Cai and I. Elishakoff. Refined second-order reliability analysis. *Structural Safety*, 14(4):267–276, 1994.

[31] V. Černý. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of optimization theory and applications*, 45(1):41–51, 1985.

[32] Z. Chen, H. Qiu, L. Gao, and P. Li. An optimal shifting vector approach for efficient probabilistic design. *Structural and Multidisciplinary Optimization*, 47(6):905–920, 2013.

[33] M.-Y. Cheng, D. Prayogo, Y.-W. Wu, and M. M. Lukito. A hybrid harmony search algorithm for discrete sizing optimization of truss structure. *Automation in Construction*, 69:21–33, 2016.

[34] A. Chiralaksanakul and S. Mahadevan. First-order approximation methods in reliability-based design optimization. *Journal of Mechanical Design*, 127(5):851–857, 2005.

[35] P. W. Christensen and A. Klarbring. *An introduction to structural optimization*, volume 153. Springer Science & Business Media, 2008.

[36] R. F. Coelho and P. Bouillard. Multi-objective reliability-based optimization with stochastic metamodels. *Evolutionary computation*, 19(4):525–560, 2011.

[37] C. A. C. Coello. Evolutionary multi-objective optimization: some current research trends and topics that remain to be explored. *Frontiers of Computer Science in China*, 3(1):18–30, 2009.

[38] C. A. C. Coello and G. B. Lamont. *Applications of multi-objective evolutionary algorithms*, volume 1. World Scientific, 2004.

[39] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates. PESA-II: Region-based selection in evolutionary multiobjective optimization. In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, pages 283–290. Morgan Kaufmann Publishers Inc., 2001.

[40] D. W. Corne, J. D. Knowles, and M. J. Oates. The Pareto envelope-based selection algorithm for multiobjective optimization. In *International conference on parallel problem solving from nature*, pages 839–848. Springer, 2000.

[41] N. Cressie. The origins of kriging. *Mathematical Geology*, 22(3):239–252, 1990.

[42] Czech Space Office. ANED-M | CZECH SPACE OFFICE. `https://www.czechspace.cz/en/aned-m`. Accessed: 2020-04-23.

[43] O. Dahmani, S. Bourguet, M. Machmoum, P. Guerin, P. Rhein, and L. Josse. Optimization and reliability evaluation of an offshore wind farm architecture. *IEEE Transactions on Sustainable Energy*, 8(2):542–550, 2016.

[44] K. Dammak and A. El Hami. Multi-objective reliability based design optimization of coupled acoustic-structural system. *Engineering Structures*, 197:109389, 2019.

[45] K. Deb. *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons, 2001.

[46] K. Deb and R. B. Agrawal. Simulated binary crossover for continuous search space. *Complex systems*, 9(2):115–148, 1995.

[47] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *International conference on parallel problem solving from nature*, pages 849–858. Springer, 2000.

[48] K. Deb and H. Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, 2014.

[49] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[50] K. Deb, A. Pratap, S. Agrawal, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. KanGAL Report 200001, Kanpur Genetic Algorithms Laboratory (KanGAL), Indian Institute of Technology Kanpur, Kanpur, PIN 208 016, India, 2000.

[51] K. Deb and S. Tiwari. Omni-optimizer: A procedure for single and multi-objective optimization. In C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, editors, *Evolutionary Multi-Criterion Optimization*, pages 47–61, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[52] S. Degertekin, L. Lamberti, and M. Hayalioglu. Heat transfer search algorithm for sizing optimization of truss structures. *Latin American Journal of Solids and Structures*, 14(3):373–397, 2017.

[53] J. C. V. der Corput. Verteilungsfunktionen. *Proc. Ned. Akad. v. Wet.*, 38:813–821, 1935.

[54] A. Der Kiureghian, H.-Z. Lin, and S.-J. Hwang. Second-order reliability approximations. *Journal of Engineering mechanics*, 113(8):1208–1225, 1987.

[55] T. Dersjö and M. Olsson. Reliability based design optimization with experiments on demand. In *Proceedings of the 10th World Congress on Structural and Multidisciplinary Optimization*, Orlando, Florida, USA, May 2012.

[56] L. Dlouhý, M. Lepš, and M. Novák. Optimization of Reinforced Concrete Frames: A Review. In *Soft Computing Methods for Civil and Structural Engineering*, pages 205–227, Stirling, 2011. Saxe-Coburg Publications.

[57] M. Dorigo and T. Stutzle. *Ant colony optimization*. Cambridge:MIT Press, 2004.

[58] X. Du and W. Chen. Sequential optimization and reliability assessment method for efficient probabilistic design. *Journal of Mechanical Design*, 126(2):225–233, 2004.

[59] V. Dubourg. *Adaptive surrogate models for reliability analysis and reliability-based design optimization*. PhD thesis, Blaise Pascal University – Clermont II, 2011.

[60] V. Dubourg, B. Sudret, and J.-M. Bourinet. Reliability-based design optimization using Kriging surrogates and subset simulation. *Structural and Multidisciplinary Optimization*, 44(5):673–690, 2011.

[61] R. C. Eberhart, Y. Shi, and J. Kennedy. *Swarm intelligence*. Elsevier, 2001.

[62] B. Echard, N. Gayton, and M. Lemaire. AK-MCS: An active learning reliability method combining Kriging and Monte Carlo simulation. *Structural Safety*, 33(2):145–154, 2011.

[63] A. E. Eiben, J. E. Smith, et al. *Introduction to evolutionary computing*, volume 53. Springer, 2003.

[64] I. Enevoldsen and J. Sørensen. Reliability-based optimization in structural engineering. *Structural Safety*, 15(3):169 – 196, 1994.

[65] S. Engelund and R. Rackwitz. A benchmark study on importance sampling techniques in structural reliability. *Structural Safety*, 12(4):255–276, 1993.

[66] European Committee for Standardization. EN 1990 Eurocode – Basis of structural design, 2005.

[67] European Space Agency. FLPP preparing for Europe's next-generation launcher. `http://www.esa.int/Enabling_Support/Space_Transportation/New_Technologies/FLPP_preparing_for_Europe_s_next-generation_launcher`. Accessed: 2020-04-23.

[68] X. Fang, W. Wang, L. He, Z. Huang, Y. Liu, and L. Zhang. Research on improved nsga-ii algorithm and its application in emergency management. *Mathematical Problems in Engineering*, 2018, 2018.

[69] H. Faure. Discrépance de suites associées à un système de numération (en dimension s). *Acta Arithmetica*, 41(4):337–351, 1982.

[70] P. C. Fishburn. Independence in utility theory with whole product sets. *Operations research*, 13(1):28–45, 1965.

[71] C. M. Fonseca and P. J. Fleming. On the performance assessment and comparison of stochastic multiobjective optimizers. In *International Conference on Parallel Problem Solving from Nature*, pages 584–593. Springer, 1996.

[72] C. Forbes, M. Evans, N. Hastings, and B. Peacock. *Statistical distributions*. John Wiley & Sons, 2011.

[73] B. Fornberg and G. Wright. Stable computation of multiquadric interpolants for all values of the shape parameter. *Computers & Mathematics with Applications*, 48(5):853–867, 2004.

[74] A. Forrester, A. Sobester, and A. Keane. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.

[75] R. Foschi, H. Li, and J. Zhang. Reliability and performance-based design: a computational approach and applications. *Structural safety*, 24(2):205–218, 2002.

[76] D. M. Frangopol. Structural optimization using reliability concepts. *Journal of Structural Engineering*, 111(11):2288–2301, 1985.

[77] J. H. Friedman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic time. *ACM Trans. Math. Software*, 3(SLAC-PUB-1549-REV. 2):209–226, 1976.

[78] F. Glover. Tabu search—part I. *ORSA Journal on computing*, 1(3):190–206, 1989.

[79] F. Glover. Tabu search—part II. *ORSA Journal on computing*, 2(1):4–32, 1990.

[80] F. W. Glover and G. A. Kochenberger. *Handbook of metaheuristics*, volume 57. Springer Science & Business Media, 2006.

[81] I. Griva, S. G. Nash, and A. Sofer. *Linear and nonlinear optimization*, volume 108. Siam, 2009.

[82] H.-M. Gutmann. A radial basis function method for global optimization. *Journal of Global Optimization*, 19(3):201–227, 2001.

[83] R. T. Haftka and Z. Gürdal. *Elements of structural optimization*, volume 11. Springer Science & Business Media, 2012.

[84] J. H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2(1):84–90, 1960.

[85] J. M. Hammersley and D. C. Handscomb. *Monte Carlo methods*. Halsted Press, a Division of John Wiley & Sons, Inc., 1964.

[86] A. M. Hasofer and N. C. Lind. An exact and invariant first order reliability format. *Journal of the Engineering Mechanics Division, ASCE*, 100(1):111–121, 1974.

[87] A. Hlobilová and M. Lepš. Multi-objective reliability-based design optimization using subset simulation enhanced by meta-models. In *REC 2016*, Bochum, DE, 2016. Ruhr Universitat Bochum.

[88] A. Hlobilová and M. Lepš. Reliability-based design optimization using local radial basis function interpolations. In *ESCO 2016*, Plzeň, CZ, 2016. Západočeská univerzita.

[89] A. Hlobilová and M. Lepš. Parallelization of reliability-based design optimization using surrogates. In *Proceedings of the Fifth International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering*, Stirling, GB, 2017. Civil-Comp Press Ltd.

[90] A. Hlobilová and M. Lepš. Parameter study on subset simulation for reliability assessment. In *Modern Methods of Experimental and Computational Investigations in Area of Construction II*, Advanced Materials Research, Pfaffikon, CH, 2017. Trans Tech Publications Inc.

[91] M. Hohenbichler and R. Rackwitz. First-order concepts in system reliability. *Structural safety*, 1(3):177–188, 1982.

[92] M. Holický et al. *Příručka pro hodnocení existujících konstrukcí*. Česká technika–nakladatelství ČVUT v Praze, 2007.

[93] J. H. Holland. *Adaptation In Natural And Artificial Systems: An Introductory Analysis With Applications To Biology, Control, And Artificial Intelligence*. A Bradford Book, 1992.

[94] L. Huyse. Solving problems of optimization under uncertainty as statistical decision problems. *AIAA paper*, 1519:1–10, 2001.

[95] H. Ishibuchi, N. Tsukamoto, and Y. Nojima. Evolutionary many-objective optimization: A short review. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 2419–2426, 2008.

[96] D. R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21(4):345–383, 2001.

[97] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

[98] H. Kahn and T. E. Harris. Estimation of particle transmission by random sampling. *National Bureau of Standards applied mathematics series*, 12:27–30, 1951.

[99] F. Kai-Tai. The uniform design: Application of number-theoretic methods in experimental design. *Acta Mathematicae Applicatae Sinica*, 3(4):9, 1980.

[100] H. Kawamura, H. Ohmori, and N. Kito. Truss topology optimization by a modified genetic algorithm. *Structural and Multidisciplinary Optimization*, 23(6):467–473, 2002.

[101] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, et al. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[102] A. D. Kiureghian and O. Ditlevsen. Aleatory or epistemic? Does it matter? *Structural Safety*, 31(2):105–112, 2009.

[103] J. Knowles and D. Corne. The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimisation. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 1, pages 98–105. IEEE, 1999.

[104] J. D. Knowles, L. Thiele, and E. Zitzler. A tutorial on the performance assessment of stochastic multiobjective optimizers -fonseca. TIK-Report 214, Computer Engineering and Networks Laboratory, ETH Zurich, Gloriastrasse 35, ETH-Zentrum, 8092 Zurich, Switzerland, February 2006.

[105] L. Kocis and W. J. Whiten. Computational investigations of low-discrepancy sequences. *ACM Transactions on Mathematical Software (TOMS)*, 23(2):266–294, 1997.

[106] A. Konak, D. W. Coit, and A. E. Smith. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, 91(9):992–1007, 2006.

[107] H. J. Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86:97, 1964.

[108] S. H. Lee and B. M. Kwak. Reliability based design optimization using response surface augmented moment method. `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.73.9180&rep=rep1&type=pdf`, 2005.

[109] T. H. Lee and J. J. Jung. A sampling technique enhancing accuracy and efficiency of metamodel-based RBDO: constraint boundary sampling. *Computers & Structures*, 86(13):1463–1476, 2008.

[110] D. H. Lehmer. Mathematical methods in large-scale computing units. In *Proceedings of a Second Symposium on Large Scale Digital Calculating Machinery*, pages 141–146. Harvard University Press, 1949.

[111] M. Lemaire, A. Chateauneuf, and J.-C. Mitteau. *Structural Reliability*. ISTE, 2010.

[112] M. Lepš. *Single and Multi-Objective Optimization in Civil Engineering with Applications*. PhD thesis, Czech Technical University in Prague, 2005.

[113] M. Lepš. Soft computing methods in concrete mix performance approximation and optimization. `http://klobouk.fsv.cvut.cz/~leps/publications/pdf/habilitation.pdf`, 2009.

[114] M. Lepš and M. Šejnoha. New approach to optimization of reinforced concrete beams. *Computers & Structures*, 81(18–19):1957–1966, August 2003. ISSN: 0045-7949.

[115] F. Li, T. Wu, A. Badiru, M. Hu, and S. Soni. A single-loop deterministic method for reliability-based design optimization. *Engineering Optimization*, 45(4):435–458, 2013.

[116] H.-S. Li. Reliability-based design optimization via high order response surface method. *Journal of Mechanical Science and Technology*, 27(4):1021–1029, 2013.

[117] R. Li, D. K. Lin, and Y. Chen. Uniform design: design, analysis and applications. *International Journal of Materials and Product Technology*, 20(1-3):101–114, 2004.

[118] Z. Li, G. Tian, G. Cheng, H. Liu, and Z. Cheng. An integrated cultural particle swarm algorithm for multi-objective reliability-based design optimization. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 228(7):1185–1196, 2014.

[119] H. Lian, P. Kerfriden, and S. Bordas. Shape optimization directly from CAD: An isogeometric boundary element approach using t-splines. *Computer Methods in Applied Mechanics and Engineering*, 317:1–41, 2017.

[120] J. Liang, C. Yue, and B. Qu. Multimodal multi-objective optimization: A preliminary study. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 2454–2461. IEEE, 2016.

[121] J. Lim, Y. S. Jang, H. S. Chang, J. C. Park, and J. Lee. Multi-objective genetic algorithm in reliability-based design optimization with sequential statistical modeling: an application to design of engine mounting. *Structural and Multidisciplinary Optimization*, 61(3):1253–1271, 2020.

[122] X. Liu, Q. Fu, N. Ye, and L. Yin. The multi-objective reliability-based design optimization for structure based on probability and ellipsoidal convex hybrid model. *Structural Safety*, 77:48–56, 2019.

[123] Y. Liu, H. Ishibuchi, Y. Nojima, N. Masuyama, and K. Shang. A double-niched evolutionary algorithm and its behavior on polygon-based problems. In A. Auger, C. M. Fonseca, N. Lourenço, P. Machado, L. Paquete, and D. Whitley, editors, *Parallel Problem Solving from Nature – PPSN XV*, pages 262–273, Cham, 2018. Springer International Publishing.

[124] M. Mangal and J. C. Cheng. Automated optimization of steel reinforcement in RC building frames using building information modeling and hybrid genetic algorithm. *Automation in Construction*, 90:39–57, 2018.

[125] Matheburg. Efficiently compute pairwise squared euclidean distance in MATLAB. `http://stackoverflow.com/questions/23911670/efficiently-compute-pairwise-squared-euclidean-distance-in-matlab`. Accessed: 2016-09-04.

[126] G. Matheron. Principles of geostatistics. *Economic geology*, 58(8):1246–1266, 1963.

[127] MathWorks. Box plot – MATLAB boxplot. `https://www.mathworks.com/help/stats/boxplot.html`. Accessed: 2019-12-26.

[128] MathWorks. Classification using nearest neighbors – MATLAB & simulink. `https://www.mathworks.com/help/stats/classification-using-nearest-neighbors.html`. Accessed: 2017-07-23.

[129] MathWorks. Pairwise distance between two sets of observations – MATLAB pdist2. `https://www.mathworks.com/help/stats/pdist2.html`. Accessed: 2017-03-23.

[130] M. Matsumoto and T. Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 8(1):3–30, 1998.

[131] N. Metropolis and S. Ulam. The Monte Carlo method. *Journal of the American statistical association*, 44(247):335–341, 1949.

[132] J. Moe. Fundamentals of optimization. *Computers & Structures*, 4(1):95–113, 1974.

[133] D. C. Montgomery, D. C. Montgomery, and D. C. Montgomery. *Design and analysis of experiments*, volume 7. Wiley New York, 1997.

[134] E. Myšáková, A. Pospíšilová, and M. Lepš. Optimized design of computer experiments: A review. In *Computational Methods for Engineering Technology*, Stirling, GB, 2014. Saxe-Coburg Publications.

[135] E. Myšáková, A. Pospíšilová, and M. Lepš. Multiobjective adaptive updating of surrogate models. In I. Zolotarev, editor, *Proceedings of the 19th International Conference Engineering Mechanics 2013*. Institute of Thermomechanics, Academy of Sciences of the Czech Republic, v.v.i., Prague, 2013.

[136] A. Naess, B. Leira, and O. Batsevych. System reliability analysis by enhanced Monte Carlo simulation. *Structural safety*, 31(5):349–355, 2009.

[137] H. Nakayama, K. Inoue, and Y. Yoshimori. Approximate optimization using computational intelligence and its application to reinforcement of cable-stayed bridges. In P. Neittaanmäki, T. Rossi, S. Korotov, E. Onate, J. Périaux, and D. Knörzer, editors, *European Congress on Computational Methods in Applied Sciences and Engineering*, 2004.

[138] H. Niederreiter. Point sets and sequences with small discrepancy. *Monatshefte für Mathematik*, 104(4):273–337, Dec 1987.

[139] E. Nikolaidis, D. M. Ghiocel, and S. Singhal. *Engineering design reliability handbook*. CRC Press, 2005.

[140] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[141] H. Pandey, G. Dhiman, M. Soni, A. Slowik, and H. Kaur. A novel hybrid evolutionary algorithm based on hypervolume indicator and reference vector adaptation strategies for many-objective optimization. *Engineering with Computers*, feb 2020.

[142] M. Papadrakakis, N. Lagaros, and V. Plevris. Design optimization of steel structures considering uncertainties. *Engineering Structures*, 27(9):1408–1418, 2005.

[143] I. Papaioannou, W. Betz, K. Zwirglmaier, and D. Straub. MCMC algorithms for subset simulation. *Probabilistic Engineering Mechanics*, 41:89–103, 2015.

[144] R. Picelli, S. Townsend, C. Brampton, J. Norato, and H. Kim. Stress-based shape and topology optimization with the level set method. *Computer Methods in Applied Mechanics and Engineering*, 329:1–23, 2018.

[145] C. Piret. *Analytical and numerical advances in radial basis functions*. PhD thesis, University of Colorado, 2007.

[146] A. Pospíšilová and M. Lepš. Global optima for size optimization benchmarks by branch and bound principles. *Acta Polytechnica*, 52(6), 2012.

[147] A. Pospíšilová and M. Lepš. Multi-objective optimization with Asymptotic sampling for RBDO. In Y. Tsompanakis, editor, *Proceedings of the Third International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering*. Civil-Comp Press, Stirlingshire, UK, 2013. Paper 2.

[148] A. Pospíšilová and M. Lepš. Parameter study of Asymptotic sampling for truss structures reliability. In *Proceedings of Structural and Physical Aspects of Civil Engineering*, Košice, Slovakia, 2013. Technical University of Kosice.

[149] A. Pospíšilová and M. Lepš. Adaptive update of surrogate models for reliability-based design optimization: A review. In *20 th International Conference Engineering Mechanics 2014*, Brno, CZ, 2014. Brno University of Technology.

[150] A. Pospíšilová and M. Lepš. Comparison of advanced simulation techniques for reliability assessment. In *Proceedings of the 5th Conference Nano & Macro Mechanics*, Prague, CZ, 2014. Czech Technical University.

[151] A. Pospíšilová and M. Lepš. Multi-objective reliability-based design optimization utilizing an adaptively updated surrogate model. In J. K. Y. Tsompanakis and B. Topping, editors, *Proceedings of the Fourth International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering*. Civil-Comp Press, Stirlingshire, UK, 2015. Paper 8.

[152] A. Pospíšilová and M. Lepš. Comparison of different simulation techniques for reliability-based design optimization. In *Engineering Mechanics 2016 – Book of full texts*, Prague, CZ, 2016. Institute of Thermomechanics, AS CR, v.v.i.

[153] A. Pospíšilová, M. Lepš, D. Rypl, and B. Patzák. Shape optimization using isogeometric analysis and particle swarm optimization. In B. Topping, editor, *Proceedings of the Eleventh International Conference on Computational Structures Technology*. Civil-Comp Press, Stirlingshire, UK, 2012. Paper 220.

[154] A. Pospíšilová, E. Myšáková, and M. Lepš. Multi-objective adaptive DoE for RBDO. In D. Novák and M. Voěechovský, editors, *Proceedings of the 11th International Probabilistic Workshop*. Ing. Vladislav Pokorný – LITERA, Brno, Czech Republic, 2013.

[155] Praha: Úřad pro technickou normalizaci, metrologii a státní zkušebnictví. ČSN EN 1990, eurokód: Zásady navrhování konstrukcí, 2004. Třídící znak 73 0002.

[156] R. Rackwitz. Reliability analysis—A review and some perspectives. *Structural safety*, 23(4):365–395, 2001.

[157] R. Rackwitz and B. Flessler. Structural reliability under combined random load sequences. *Computers & Structures*, 9(5):489–494, 1978.

[158] Randomness and Integrity Services Ltd. RANDOM.ORG – true random number service. `https://www.random.org`. Accessed: 2018-07-24.

[159] S. S. Rao. *Reliability-based design*. McGraw-Hill Companies, 1992.

[160] S. S. Rao. *Engineering optimization: theory and practice*. John Wiley & Sons, 2019.

[161] T. Ray, K. Tai, and kin Chye Seow. Multiobjective design optimization by an evolutionary algorithm. *Engineering Optimization*, 33(4):399–424, 2001.

[162] S. Rice. Distribution of quadratic forms in normal random variables—evaluation by numerical integration. *SIAM Journal on Scientific and Statistical Computing*, 1(4):438–448, 1980.

[163] K. Riley, M. Hobson, and S. Bence. Mathematical methods for physics and engineering, 1999.

[164] S. Rippa. An algorithm for selecting a good value for the parameter c in radial basis function interpolation. *Advances in Computational Mathematics*, 11(2-3):193–210, 1999.

[165] N. Riquelme, C. Von Lücken, and B. Baran. Performance metrics in multi-objective optimization. In *Computing Conference (CLEI), 2015 Latin American*, pages 1–11. IEEE, 2015.

[166] M. Rosenblatt. Remarks on a multivariate transformation. *The annals of mathematical statistics*, 23(3):470–472, 1952.

[167] S. M. Ross. *Introduction to probability and statistics for engineers and scientists*. Academic Press, fourth edition, 2009.

[168] J. Royset, A. Der Kiureghian, and E. Polak. Reliability-based optimal structural design by the decoupling approach. *Reliability Engineering & System Safety*, 73(3):213–221, Sept. 2001.

[169] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–435, 1989.

[170] K. Sastry and D. Goldberg. *Search Methodologies*, chapter Genetic Algorithms, pages 97–125. Springer, 2005.

[171] J. R. Schott. Fault tolerant design using single and multicriteria genetic algorithm optimization. Technical report, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 1995. Master Thesis.

[172] G. I. Schuëller, H. J. Pradlwarter, and P. Koutsourelakis. A comparative study of reliability estimation procedures for high dimensions. In *16th ASCE Engineering mechanics conference*, pages 16–18. Citeseer, 2003.

[173] G. Schuëller and H. Jensen. Computational methods in optimization considering uncertainties – An overview. *Computational Methods in Optimization Considering Uncertainties*, 198(1):2–13, Nov. 2008.

[174] S. Shan and G. G. Wang. Reliable design space and complete single-loop reliability-based design optimization. *Reliability Engineering & System Safety*, 93(8):1218–1230, 2008.

[175] M. Sichani, S. Nielsen, and C. Bucher. Applications of Asymptotic sampling on high dimensional structural dynamic problems. *Structural Safety*, 33(4–5):305–316, 2011.

[176] T. W. Simpson, J. Poplinski, P. N. Koch, and J. K. Allen. Metamodels for computer-based engineering design: Survey and recommendations. *Engineering with computers*, 17(2):129–150, 2001.

[177] I. M. Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 7(4):784–802, 1967.

[178] L.-K. Song, C.-W. Fei, J. Wen, and G.-C. Bai. Multi-objective reliability-based design optimization approach of complex structure with multi-failure modes. *Aerospace Science and Technology*, 64:52–62, 2017.

[179] W. R. Spillers and K. M. MacBain. *Structural optimization*. Springer Science & Business Media, 2009.

[180] N. Srinivas and K. Deb. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, Fall 1994.

[181] R. Srinivasan. *Importance sampling: Applications in communications and detection*. Springer Science & Business Media, 2013.

[182] Statinfer. Efficient Matlab (I): pairwise distances. `https://statinfer.wordpress.com/2011/11/14/efficient-matlab-i-pairwise-distances`. Accessed: 2016-09-15.

[183] B. Sudret. *Uncertainty propagation and sensitivity analysis in mechanical models – Contributions to structural reliability and stochastic spectral methods*, 2007. Habilitation à diriger des recherches, Universitè Blaise Pascal, Clermont-Ferrand, France.

[184] G. Sun, H. Zhang, J. Fang, G. Li, and Q. Li. Multi-objective and multi-case reliability-based design optimization for tailor rolled blank (trb) structures. *Structural and Multidisciplinary Optimization*, 55(5):1899–1916, 2017.

[185] S. Sun, X. Li, H. Liu, K. Luo, and B. Gu. Fast statistical analysis of rare circuit failure events via scaled-sigma sampling for high-dimensional variation space. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34:1096–1109, 2013.

[186] B. Teplý and D. Novák. *Spolehlivost stavebních konstrukcí*. University textbook. Akademické nakladatelství CERM, Brno, 1998. Vysoké učení technické v Brně, Fakulta stavební.

[187] J. Tu and K. Choi. A new study on reliability based design optimization. *ASME Journal of Mechanical Design*, 121(4):557–564, 1999.

[188] T. Tušar and B. Filipič. Visualization of Pareto front approximations in evolutionary multiobjective optimization: A critical review and the prosection method. *IEEE Transactions on Evolutionary Computation*, 19(2):225–245, 2014.

[189] M. Valdebenito and G. Schuëller. Reliability-based optimization considering design variables of discrete size. *Engineering Structures*, 32(9):2919–2930, 2010.

[190] M. Valdebenito and G. Schuëller. A survey on approaches for reliability-based optimization. *Structural and Multidisciplinary Optimization*, 42(5):645–663, Nov. 2010.

[191] V. Vapnik. *Statistical learning theory Wiley*. John Wiley & Sons, 1998.

[192] D. A. V. Veldhuizen. *Multiobjective evolutionary algorithms: classifications, analyses and new innovations*. PhD thesis, Dayton, OH: Air Force Institute of Technology, 1999. Technical Report No. AFIT/DS/ENG/99-01.

[193] A. K. Verma, S. Ajit, and D. R. Karanki. *Reliability and safety engineering*, volume 43. Springer, 2010.

[194] Z. Vitingerová. *Evolutionary algorithms for multi-objective parameter estimation*. PhD thesis, Czech Technical University in Prague, 2010.

[195] M. Vořechovský. Hierarchical subset Latin hypercube sampling for correlated random vectors. In Y. T. B.H.V. Topping, editor, *Proceedings of the First International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering*. Civil-Comp Press, Stirlingshire, UK, 2009.

[196] R. Weber, H.-J. Schek, and S. Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *VLDB*, volume 98, pages 194–205, 1998.

[197] F. Xiong, Y. Xiong, W. Chen, and S. Yang. Optimizing Latin hypercube design for sequential sampling of computer experiments. *Engineering Optimization*, 41(8):793–810, 2009.

[198] I.-T. Yang and Y.-H. Hsieh. Reliability-based design optimization with cooperation between support vector machine and particle swarm optimization. *Engineering with Computers*, 29(2):151–163, 2013.

[199] R. Yang and L. Gu. Experience with approximate reliability-based optimization methods. *Structural and Multidisciplinary Optimization*, 26(1-2):152–159, 2004.

[200] 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(6):450–479, 2011.

[201] G. G. Yen and Haiming Lu. Dynamic multiobjective evolutionary algorithm: adaptive cell-based rank and density estimation. *IEEE Transactions on Evolutionary Computation*, 7(3):253–274, 2003.

[202] B. D. Youn and K. K. Choi. A new response surface methodology for reliability-based design optimization. *Computers & structures*, 82(2):241–256, 2004.

[203] X. Yu and A. M. Khambadkone. Reliability analysis and cost optimization of parallel-inverter system. *IEEE Transactions on Industrial Electronics*, 59(10):3881–3889, 2011.

[204] E. P. Zafiropoulos and E. N. Dialynas. Reliability and cost optimization of electronic devices considering the component failure rate uncertainty. *Reliability Engineering & System Safety*, 84(3):271–284, 2004.

[205] T. A. Zang, M. J. Hemsch, M. W. Hilburger, S. P. Kenny, J. M. Luckring, P. Maghami, S. L. Padula, and W. J. Stroud. *Needs and opportunities for uncertainty-based multidisciplinary design methods for aerospace vehicles.* National Aeronautics and Space Administration, Langley Research Center, 2002.

[206] G. R. Zavala, A. J. Nebro, F. Luna, and C. A. Coello Coello. A survey of multi-objective metaheuristics applied to structural optimization. *Structural and Multidisciplinary Optimization*, 49(4):537–558, Apr 2014.

[207] Q. Zhang and H. Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.

[208] W. Zhao and Z. Qiu. An efficient response surface method and its application to structural reliability and reliability-based optimization. *Finite Elements in Analysis and Design*, 67(0):34–42, 2013.

[209] J.-H. Zhu, W.-H. Zhang, and L. Xia. Topology optimization in aircraft and aerospace structures design. *Archives of Computational Methods in Engineering*, 23(4):595–622, 2016.

[210] A. Žilinskas. A review of statistical models for global optimization. *Journal of Global Optimization*, 2(2):145–153, 1992.

[211] E. Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications.* PhD thesis, Swiss Federal Institute of Technology Zurich, November 1999.

[212] E. Zitzler, M. Laumanns, and L. Thiele. Spea2: Improving the strength Pareto evolutionary algorithm. Tech. rep. 103, Computer engineering and networks laboratory (TIK), Swiss federal institute of technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich,Switzerland, 2001.

[213] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4):257–271, 1999.

[214] K. M. Zuev. *Subset Simulation Method for Rare Event Estimation: An Introduction*, pages 1–25. Springer Berlin Heidelberg, Berlin, Heidelberg, 2021.

[215] K. M. Zuev, J. L. Beck, S.-K. Au, and L. S. Katafygiotis. Bayesian post-processor and other enhancements of subset simulation for estimating failure probabilities in high dimensions. *Computers & Structures*, 92—93:283–296, 2012.

Chapter 9

Distributions for continuous variables

9.1 Uniform distribution

A uniform distribution (also called a rectangular distribution) is from a family of symmetric statistical distributions. If an event x from all possible events X is in the range $[a, b]$, the probability of the occurrence of each possible events x is the same. The bounds a and b are the parameters of the distribution. The probability of the occurrence of the event x outside these bounds is zero.

The probability density function of a continuous uniform random variable over the interval $[a, b]$ is given by

$$f_X(x) = \begin{cases} \frac{1}{b-a}, & a \leq x \leq b \\ 0, & x < a \vee x > b. \end{cases} \tag{9.1}$$

It is a symmetrical and monotonic constant function on the interval $[a, b]$.

The cumulative distribution function is given by

$$F_X(x) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \leq x < b \\ 1, & x \geq b. \end{cases} \tag{9.2}$$

The first two statistical moments, i.e. the mean and the variance, are given by

$$\mathrm{E}X = \frac{b+a}{2}, \tag{9.3}$$

$$\mathrm{Var}X = \frac{(b-a)^2}{12}. \tag{9.4}$$

If there is a necessity to evaluate the parameters of the distribution, i.e. for the knowledge of the bounds of the uniform space, the parameters a and b are given by

$$b = \sqrt{3\mathrm{Var}X} + \mathrm{E}X, \tag{9.5}$$

$$a = \mathrm{E}X - \sqrt{3\mathrm{Var}X}. \tag{9.6}$$

The example of the probability density function and the corresponding cumulative density function is depicted in Figure 9.1. The parameters of this distribution are $a = 2$ and $b = 7$ and analogous statistical moments are $\mathrm{E}X = 4.5$ and $\mathrm{Var}X = \frac{25}{12} = 2.0833$.

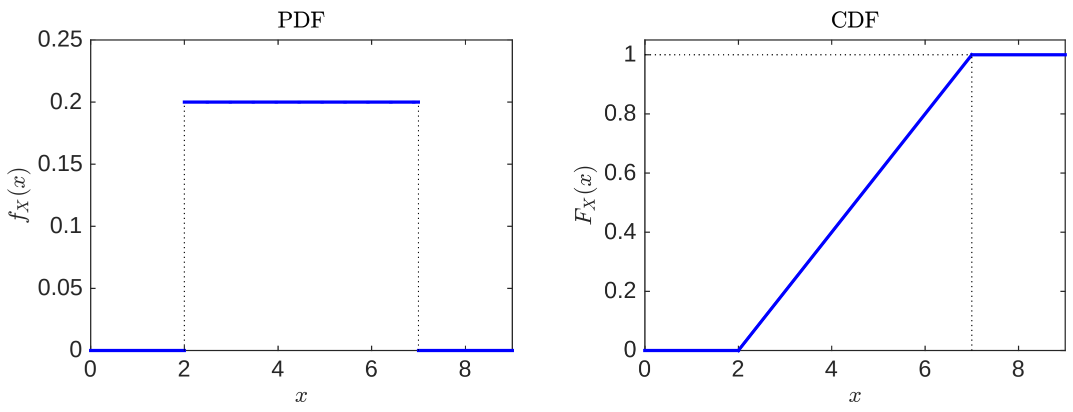


Figure 9.1: Probability density function (PDF, left) and cumulative density function (CDF, right) for the uniform variable.

A **standard uniform distribution** is a special case of the uniform distribution; the bounds are restricted into the interval $[a, b] = [0, 1]$. This distribution is often used in random variable generators that are available in many programming languages as well as in the statistical and the computational programs.

9.2 Normal distribution

A normal distribution (also called Gaussian distribution after its discoverer CARL FRIEDRICH GAUSS) is from a family of symmetric statistical distributions. It is often denoted by $\mathrm{N}(\mu, \sigma^2)$.

The probability density function of a continuous normal random variable is given by

$$f_X(x|\mu, \sigma) = \frac{1}{\sigma_X\sqrt{2\pi}}\mathrm{e}^{-\frac{(x-\mu_X)^2}{2\sigma_X^2}}, \tag{9.7}$$

where x is from the interval $(-\infty, \infty)$, the parameter μ is from the interval $(-\infty, \infty)$ and the parameter σ is greater than 0. The parameters μ and σ are closely related to the first two statistical moments, i.e.

$$\mathrm{E}X = \mu, \tag{9.8}$$
$$\mathrm{Var}X = \sigma^2. \tag{9.9}$$

The cumulative distribution function is given by

$$F_X(x|\mu, \sigma) = \int\limits_{-\infty}^{x} f_T(t)\mathrm{d}t. \tag{9.10}$$

Moreover, it is feasible to enumerate numerically.

The **standard normal distribution** is a standardized form of a normal distribution with $\mu = 0$ and $\sigma = 1$. Corresponding probability distribution function is

$$f_X(x) = \frac{1}{\sqrt{2\pi}}\mathrm{e}^{-\frac{x^2}{2}}. \tag{9.11}$$

The cumulative distribution function of the standard normal distribution is often denoted by $\Phi(z)$. To convert normally distributed variable into the standard normal, the formula

$$U = \frac{X - \mu}{\sigma} \tag{9.12}$$

is used, in which U is standard normally distributed variable and X is normal distributed variable. Figure 9.2 shows the probability density function (PDF) and corresponding cumulative distribution function (CDF) for a variable x that is from the standard normal space.

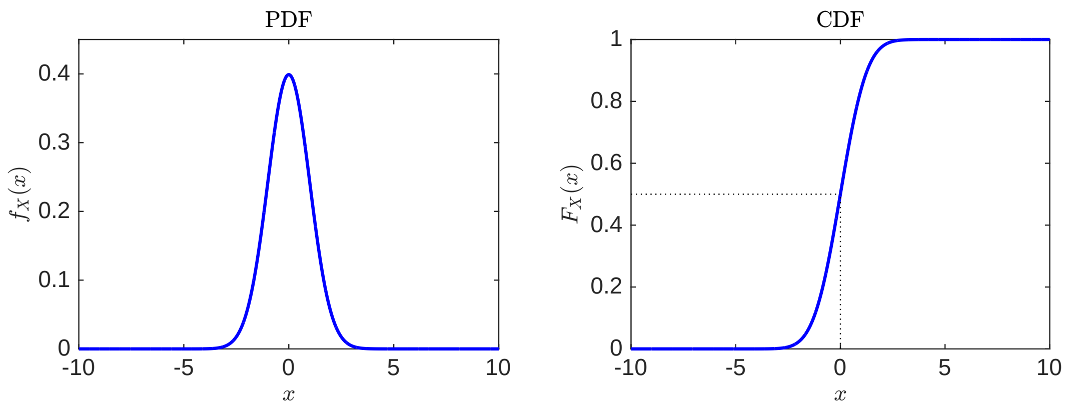


Figure 9.2: Probability density function (PDF, left) and cumulative density function (CDF, right) for the standard normal variable.

In the structural engineering, a normal distribution is used for modelling some kinds of material properties such as a strength of a material, modelling of dead loading and geometrical properties of the structure such as outer sizes of elements [92]. It is also favourable for stochastic variables with a low coefficient of variation (e.g. CoV less than 0.3).

9.3 Log-normal distribution

The variable X has a log-normal distribution if its logarithm is normally distributed, i.e. the variable X is log-normally distributed if a variable $Y = \ln(X)$ is normally distributed. The probability density function of log-normal distribution is non-symmetrical, and all events x are positive real values. In this thesis, we consider only two parameters in this distribution, namely μ and σ.

The probability density function of a continuous log-normal random variable is given by

$$f_X(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi}x\sigma} \exp\left(-\frac{(\ln x - \mu)^2}{2\sigma^2}\right), \tag{9.13}$$

the event x is from the interval $(0, \infty)$, the parameter μ is from the interval $(-\infty, \infty)$ and the parameter σ is greater than 0.

The cumulative distribution function and its corresponding inverse function are given by

$$\begin{aligned} F_X(x|\mu,\sigma) &= \Phi\left(\frac{\ln x - \mu}{\sigma}\right), & (9.14)\\ F_X^{-1}(y) &= \exp\left(\sigma\Phi^{-1}(x) + \mu\right), & (9.15) \end{aligned}$$

The first two statistical moments, the mean and the variance are given by

$$\mathrm{E}X = \exp\left(\mu + \frac{\sigma^2}{2}\right) \tag{9.16}$$

$$\mathrm{Var}X = \exp\left((2\mu + \sigma^2)(\exp(\sigma^2) - 1)\right). \tag{9.17}$$

The two parameters μ and σ of the log-normal distribution are calculable from the statistical moments as

$$\mu = \ln\left(\frac{(\mathrm{E}X)^2}{\sqrt{\mathrm{Var}X + (\mathrm{E}X)^2}}\right), \tag{9.18}$$

$$\sigma = \sqrt{\ln\left(\frac{\mathrm{Var}X}{(\mathrm{E}X)^2 + 1}\right)}. \tag{9.19}$$

Figure 9.3 shows the probability density function (PDF) and corresponding cumulative distribution function (CDF) for three variables x that are from the log-normal space.

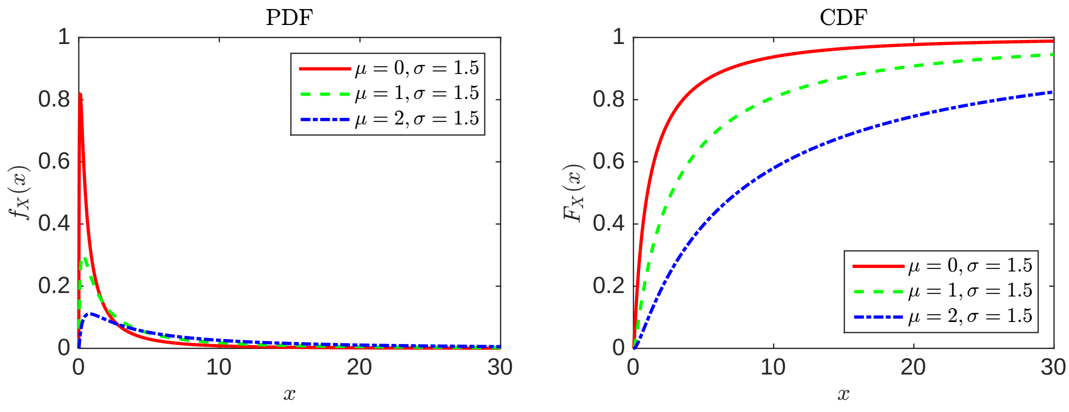


Figure 9.3: Probability density function (PDF, left) and cumulative density function (CDF, right) for three log-normal variables; parameter μ differs values 0, 1 and 2, but the parameter σ remains the same with the value 1.5.

In the structural engineering, a log-normal distribution is usually used for modelling of the material properties such as the strength of a material and Young modulus, some kinds of loading, and the geometrical properties of the structure. The positive skewness is for the low strength materials, and the negative skewness is for the high strength materials [92].

9.4 Exponential distribution

An exponential distribution is an asymmetrical statistical distribution. It has only one parameter called scale parameter, which is greater than 0. This parameter can be alternatively defined as λ, which is connected to the scale parameter via the hazard function $\lambda = 1/b$ [72].

The probability density function of a continuous exponential random variable is given by

$$f_X(x|b) = \frac{1}{b}\exp\left(\frac{-x}{b}\right), \tag{9.20}$$

the variable x as well as the parameter b are both greater than zero.

The cumulative distribution function and its corresponding inverse function are given by

$$F_X(x|b) = 1 - \exp\left(\frac{-x}{b}\right), \tag{9.21}$$

$$F_X^{-1}(y|b) = -b\ln(1-y). \tag{9.22}$$

The first two statistical moments, the mean and the variance are given by

$$\mathrm{E}X = b, \tag{9.23}$$

$$\mathrm{Var}X = b^2. \tag{9.24}$$

The examples of the probability density functions and cumulative distribution functions of exponential variables with three different scale parameters are depicted in Figure 9.4.

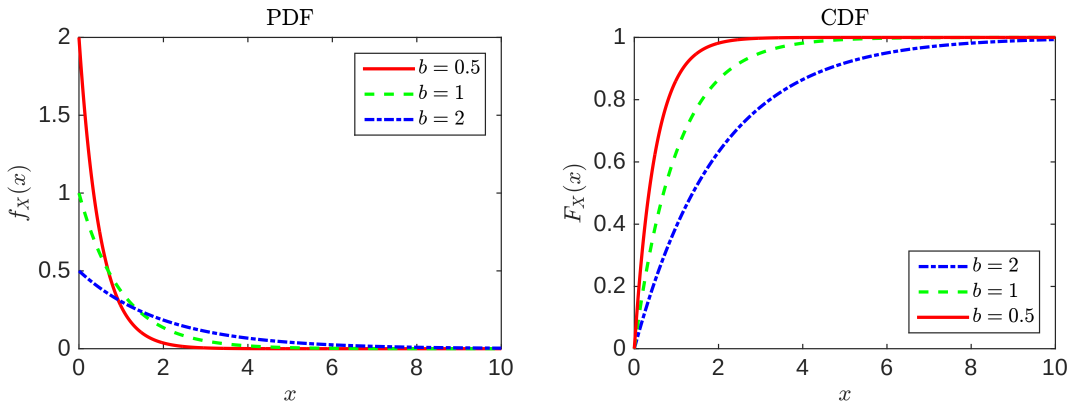


Figure 9.4: Probability density function (PDF, left) and cumulative density function (CDF, right) for three exponential variables; the scale parameter b differs in values 0.5, 1 and 2.

9.5 Gumbel distribution

A Gumbel distribution is also called an *extreme value distribution* because it can model a distribution of the largest quantity of values. Originally, it was applied in the estimation of the flood levels [72]. It has two forms; the largest extreme is used for modelling of structural loading, and the smallest extreme is used for modelling of load capacity. Since we use it for modelling of the loading in this thesis, we will work with the largest extreme. The difference in these two forms is in the reversal of the sign of x. This distribution utilizes two parameters, the location parameter a and the scale parameter b.

The probability density function of a continuous Gumbel random variable is given by

$$f_X(x|a,b) = \frac{1}{b}\exp\left(-\frac{x-a}{b}\right)\exp\left(-\exp\left(-\frac{x-a}{b}\right)\right), b > 0. \tag{9.25}$$

The cumulative distribution function and its corresponding inverse function are given by

$$F_X(x|a,b) = \exp\left(-\exp\left(-\frac{x-a}{b}\right)\right), b > 0, \tag{9.26}$$

$$F_X^{-1}(y|a,b) = -b\ln\left(-\ln(y)\right) + a, 0 < y < 1. \tag{9.27}$$

$$\tag{9.28}$$

The first two statistical moments, the mean and the variance are given by

$$\mathrm{E}X = a + \gamma b, \tag{9.29}$$

$$\mathrm{Var}X = \frac{\pi^2 b^2}{6}, \tag{9.30}$$

where γ is Euler-Mascheroni constant which is approximately equal to 0.5772 15664 9.

The parameters a and b are given by

$$a = \mathrm{E}X - \gamma b, \tag{9.31}$$

$$b = \frac{\sqrt{\mathrm{Var}X \cdot 6}}{\pi}. \tag{9.32}$$

Figure 9.5 shows examples of three Gumbel variables (the largest extreme form) with their probability density functions as well as cumulative distribution functions. The variables differ only in scale parameter b, which is from the set $[1, 2, 3]$; the location parameter remains $a = 0$. By contrast, Figure 9.6 shows the alternation of the location parameter to the set of values $[0, 1, 2]$ and the scale parameter remains equal $b = 2$ for all three Gumbel variables.

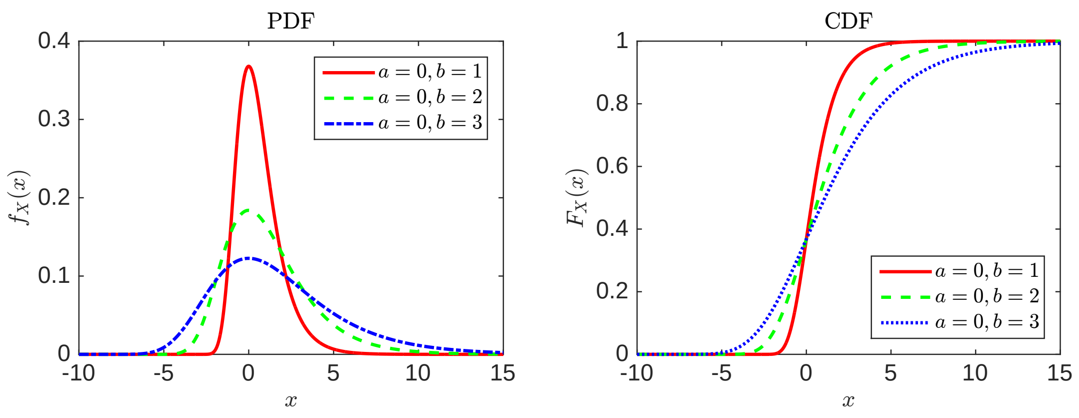


Figure 9.5: Probability density function (PDF, left) and cumulative density function (CDF, right) for three Gumbel (the largest extreme form) variables; the scale parameter b differs from the set of values $[1, 2, 3]$, but the location parameter a remains the same with the value 0.

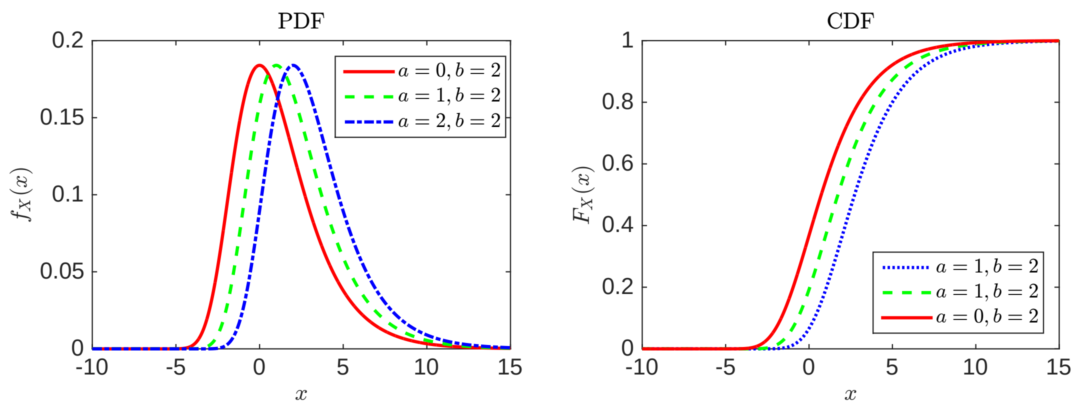


Figure 9.6: Probability density function (PDF, left) and cumulative density function (CDF, right) for three Gumbel variables; the location parameter a differs from the set $[0, 1, 2]$, but the scale parameter b remains the same with the value 2.

9.6 Weibull distribution

The Weibull distribution is a continuous probability distribution, which was named after WALODDI WEIBULL. The Weibull distribution exists in two versions, a two-parameter distribution with a scale parameter η, and a shape parameter β; this distribution is also called a bi-Weibull distribution [72], and a three-parameter distribution with a scale parameter, a shape parameter, and a location parameter γ. We work with the bi-Weibull distribution, which is defined in the following passage. This distribution is usually used for material properties [92] or as a lifetime distribution in reliability applications [72]. In the case of lifetime distribution utilization, the bi-Weibull distribution can represent decreasing, constant, or increasing failure rates, which correspond to the three sections of the “bathtub curve”. The bi-Weibull distribution then represents combinations of two such phases of life [72].

The probability density function of a continuous Weibull random variable is given by

$$f_X(x|\eta,\beta) = \frac{\beta x^{\beta-1}}{\eta^\beta}\exp\left(-\left(\frac{x}{\eta}\right)^\beta\right), \tag{9.33}$$

the random variable x and the parameters η and β are from range $(0,\infty)$.

The cumulative distribution function and its corresponding inverse function are given by

$$F_X(x|\eta,\beta) = 1-\exp\left(-\left(\frac{x}{\eta}\right)^\beta\right), \tag{9.34}$$

$$F_X^{-1}(y|\eta,\beta) = \eta(-\log(1-y))^{\frac{1}{\beta}}, 0 \le y \le 1. \tag{9.35}$$

(9.36)

The first two statistical moments, the mean and the variance are given by

$$\mathrm{E}X = \eta\Gamma\left(1+\frac{1}{\beta}\right), \tag{9.37}$$

$$\mathrm{Var}X = \eta^2\left(\Gamma\left(1+\frac{2}{\beta}\right)-\left(\Gamma\left(1+\frac{1}{\beta}\right)\right)^2\right), \tag{9.38}$$

where $\Gamma(\cdot)$ is a gamma function defined as a definite integral for positive real numbers as

$$\Gamma(n) = \int_0^\infty e^{-x}x^{n-1}\mathrm{d}x. \tag{9.39}$$

The parameters η and β are not able to obtain directly from the previous equations because of the gamma function. Therefore, a crude approximation of parameters can be estimated via

$$\beta_0 = \left(\frac{\sqrt{\mathrm{Var}X}}{\mathrm{E}X}\right)^{-1.086}, \tag{9.40}$$

$$\eta_0 = \frac{\mathrm{E}X}{\Gamma\left(1+\frac{1}{\beta}\right)}. \tag{9.41}$$

If the accuracy is not sufficient, the more precise evaluation is possible. The parameter η is expressed from Equation 9.37, substituted into Equation 9.38 and the equation is simplified to get

$$\frac{(\mathrm{Var}X)^2}{(\mathrm{E}X)^2} - \frac{\Gamma\left(1+\frac{2}{\beta}\right)}{\left(\Gamma\left(1+\frac{1}{\beta}\right)\right)^2} + 1 = 0. \tag{9.42}$$

The root-finding procedure can find the root of this function equal to the parameter β. The initial guess of the β parameter in Equation 9.40 is advantageous to use to shorten the iterative procedure. The parameter η is subsequently easy to get via

$$\eta = \frac{\mathrm{E}X}{\Gamma\left(1+\frac{1}{\beta}\right)}. \tag{9.43}$$

Figure 9.7 shows examples of four bi-Weibull variables with their probability density functions as well as cumulative distribution functions. The variables differ only in the shape parameter β, which is from the set $[1, 2, 3, 4]$; the scale parameter η remains 1. By contrast, Figure 9.8 shows the alternation of the scale parameter to the set of values $[1, 2, 3, 4]$, and the shape parameter remains equal to 2 for all four bi-Weibull variables.

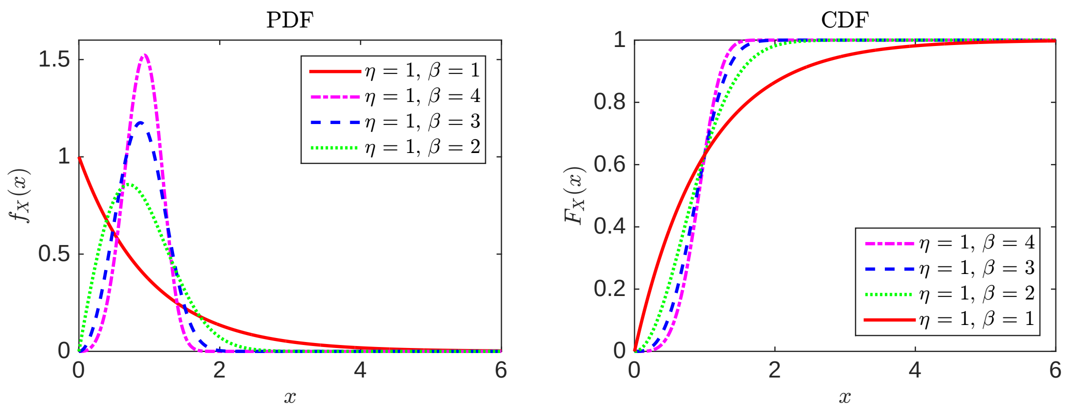


Figure 9.7: Probability density function (PDF, left) and cumulative density function (CDF, right) for four bi-Weibull distribution variables; the shape parameter β differs from the set of values $[1, 2, 3, 4]$, but the scale parameter η remains the same with the value 1.

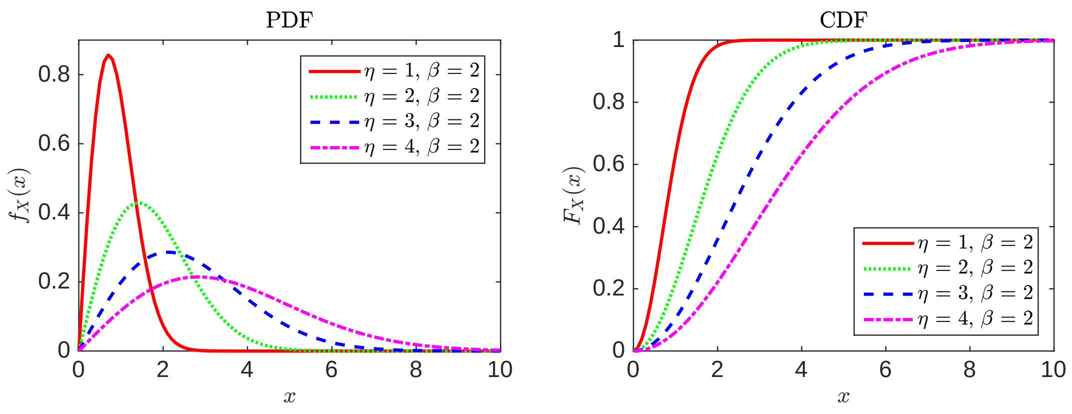


Figure 9.8: Probability density function (PDF, left) and cumulative density function (CDF, right) for four bi-Weibull distribution variables; the scale parameter η differs from the set of values $[1, 2, 3, 4]$, but the shape parameter β remains the same with the value 2.

We provide a simple code in MATLAB to obtain the parameters β and η for a bi-Weibull distribution if a mean and variance are known.

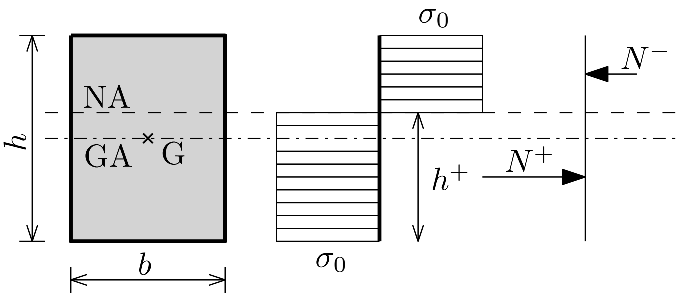
Parameter estimation of bi-Weibull distribution

```
EX = 40;            % mean value
Var = 16;           % variance
std = sqrt(Var);    % standard deviation

% A function handle with the left side of Equation 9.42:
f = @(beta)std^2/EX^2-gamma(1+2./beta)./gamma(1+1./beta).^2+1;

beta0 = (std/EX).^-1.086;  % initial guess of beta (Equation 9.40)
beta = fzero(f,beta0);     % solve for beta
eta = EX/gamma(1+1/beta);  % substitute to find eta
```

Chapter 10

Ultimate plastic state under bending and tension

The collapse of the structure occurs with the ultimate plastic state. Fig. 10.1 shows the stress diagram and force decomposition for the rectangular cross-section under bending and tension in 2D. The special cases of the neutral axis position are depicted in Fig. 10.2.



$$
\begin{aligned}
&N^- = \sigma_0 b(h-h^+)\\
&N^+ = \sigma_0 b h^+\\
&N = N^+ + N^- = \sigma_0 b(2h^+ - h)\\
&M^- = \sigma_0 b(h-h^+)(\tfrac{h}{2} + \tfrac{h+h^+}{2})\\
&M^+ = \sigma_0 b h^+ \tfrac{h-h^+}{2}\\
&M = M^+ + M^- = \sigma_0 b h^+(h-h^+)
\end{aligned}
$$

Figure 10.1: Ultimate plastic state of the rectangular cross-section under bending and tension. NA indicates neutral axis and GA indicates the centre of gravity axis. σ_0 is yield stress.

To put together the bending moment M and the normal force N into one equation, the height of the tension area h^+ and the height of the compressed area $(h - h^+)$ can be expressed from the normal force N as

$$h^+ = \frac{h}{2} + \frac{N}{2\sigma_0 b}, \tag{10.1}$$

$$h - h^+ = \frac{h}{2} - \frac{N}{2\sigma_0 b}. \tag{10.2}$$

Replacing h^+ and $(h-h^+)$ in the bending moment M and considering M_{pl} equals to $\frac{bh^2\sigma_0}{4}$ leads to

$$M_{pl} - M - \left(\frac{N^2}{4\sigma_0 b}\right) = 0. \tag{10.3}$$

Dividing the whole equation with the M_{pl} and considering N_{pl} equals to $bh\sigma_0$, the final resulting equation is

$$1 - \frac{M}{M_{pl}} - \left(\frac{N}{N_{pl}}\right)^2 = 0. \tag{10.4}$$

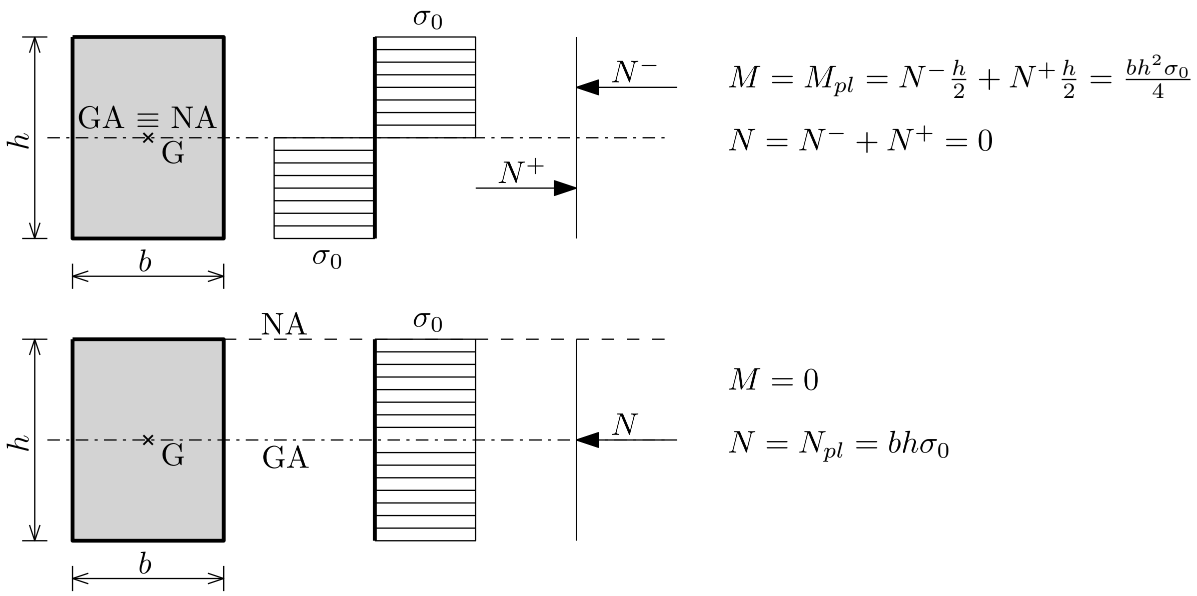


Figure 10.2: Special cases of the neutral axis position in the ultimate plastic state of the rectangular cross-section under bending and tension. NA indicates neutral axis and GA indicates the centre of gravity axis. σ_0 is yield stress.

For the rectangular shape of cross-section holds that

$$1 - \frac{4M}{bh^2\sigma_0} - \frac{N^2}{b^2h^2\sigma_0} = 0. \tag{10.5}$$

Extending it into 3D, we get

$$1 - \frac{4M_y}{bh^2\sigma_0} - \frac{4M_z}{b^2h\sigma_0} - \frac{N^2}{b^2h^2\sigma_0} = 0. \tag{10.6}$$

Chapter 11

Hardware and software parameters

8 CPUs	Intel(R) Xeon(R) CPU E5520 @ 2.27 GHz
Memory [MB]	12025
MATLAB	R2015a, 64-bit, glnxa64
Operating system	Debian 3.16.7

Table 11.1: Hardware and software settings of Computer 1.

8 CPUs	Intel(R) Core(TM) i7-4702MQ CPU @ 2.20 GHz
Memory [MB]	8192
MATLAB	R2015a, 64-bit
Operating system	Windows 7 Home Premium 64-bit

Table 11.2: Hardware and software settings of Computer 2.

Chapter 12

Unpublished results for Examples 1 - 5

12.1 Box-plots for performance measures

In each box, the 25th and 75th percentiles are represented by the bottom and the top edges, and the median value is symbolized by the red line. The the most extreme data points are depicted with the whiskers and the outliers are plotted individually using the plus symbol [127]. Used metrics are a Hypervolume (HV), Spacing (S), Spread (Δ), Generational distance (GD), Two set coverage metric (C(A,B)), Coverage difference of two sets (D(A,B)), and Error of the reliability index ($\epsilon(\beta_{\mathrm{oPF}})$).

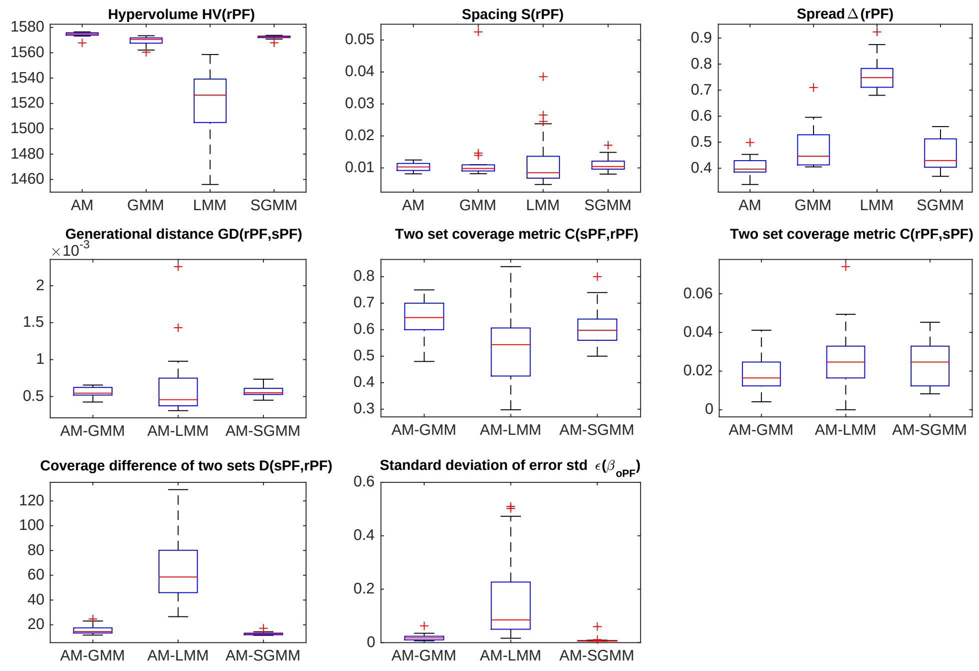


Figure 12.1: Example 1: Performance measures for RBDO utilizing a preconditioned quasi-Monte Carlo simulation with an analytical model (AM), local meta-models (LMM), global meta-model (GMM), and sparse global meta-model (SGMM).

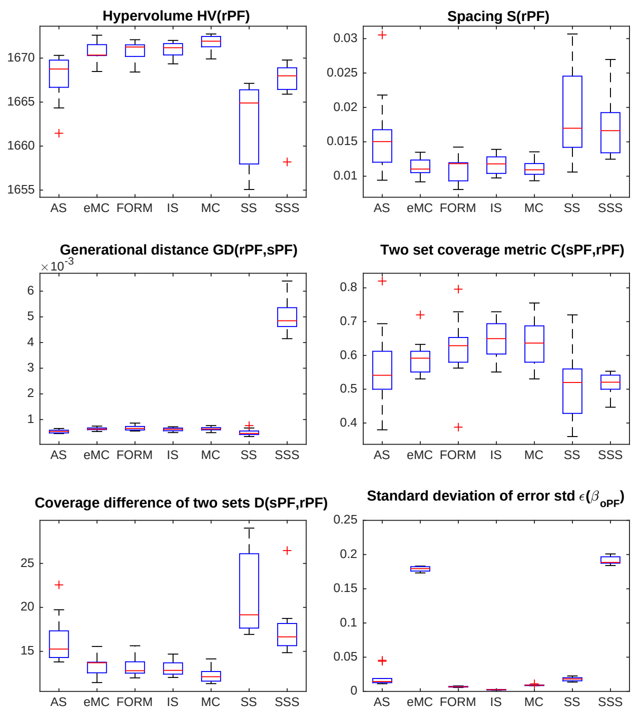


Figure 12.2: Example 1: Performance measures for RBDO utilizing an analytical model and different reliability assessment techniques, namely an Asymptotic sampling (AS), Enhanced Monte Carlo simulation (eMC), First-order reliability method (FORM), Importance sampling (IS), quasi-Monte Carlo simulation (MC), Subset simulation (SS), and Scaled sigma sampling (SSS).

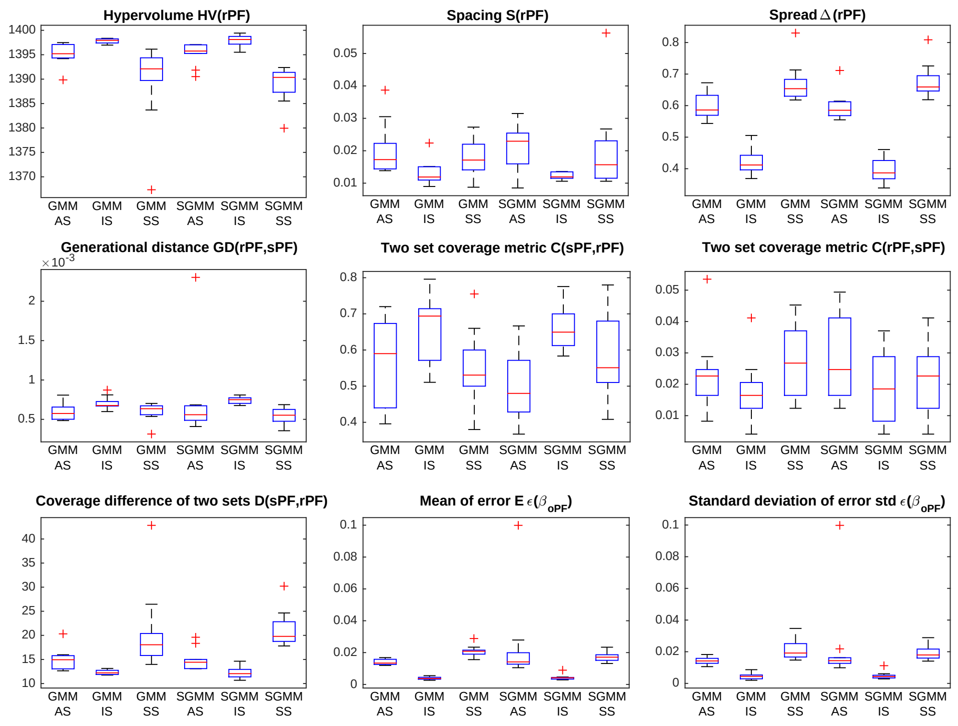


Figure 12.3: Example 1: Performance measures for RBDO utilizing an Asymptotic sampling, Importance sampling, and Subset simulation with a global meta-model (GMM), and sparse global meta-model (SGMM).

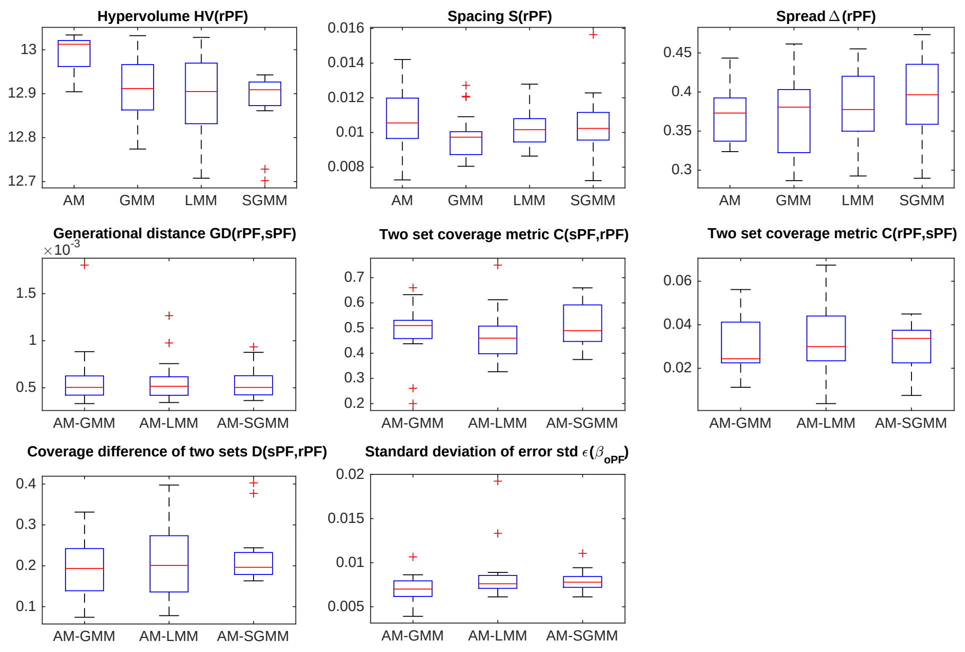


Figure 12.4: Example 2: Performance measures for RBDO utilizing a preconditioned quasi-Monte Carlo simulation with an analytical model (AM), local meta-models (LMM), global meta-model (GMM), and sparse global meta-model (SGMM).

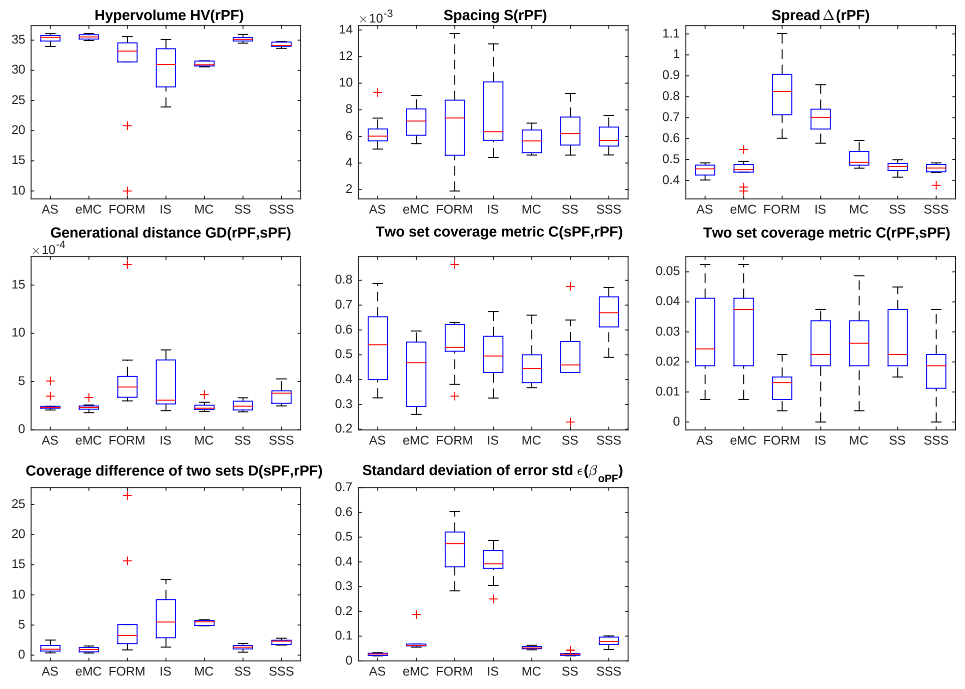


Figure 12.5: Example 2: Performance measures for RBDO utilizing an analytical model and different reliability assessment techniques, namely an Asymptotic sampling (AS), Enhanced Monte Carlo simulation (eMC), First-order reliability method (FORM), Importance sampling (IS), quasi-Monte Carlo simulation (MC), Subset simulation (SS), and Scaled sigma sampling (SSS).

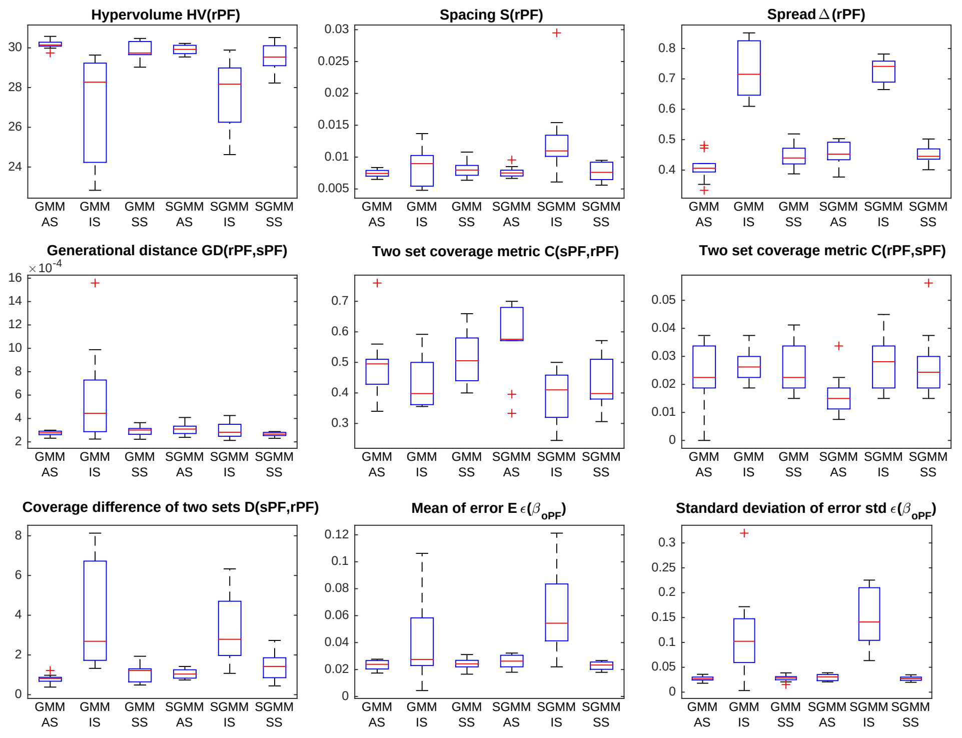


Figure 12.6: Example 2: Performance measures for RBDO utilizing an Asymptotic sampling, Importance sampling, and Subset simulation with a global meta-model (GMM), and sparse global meta-model (SGMM).

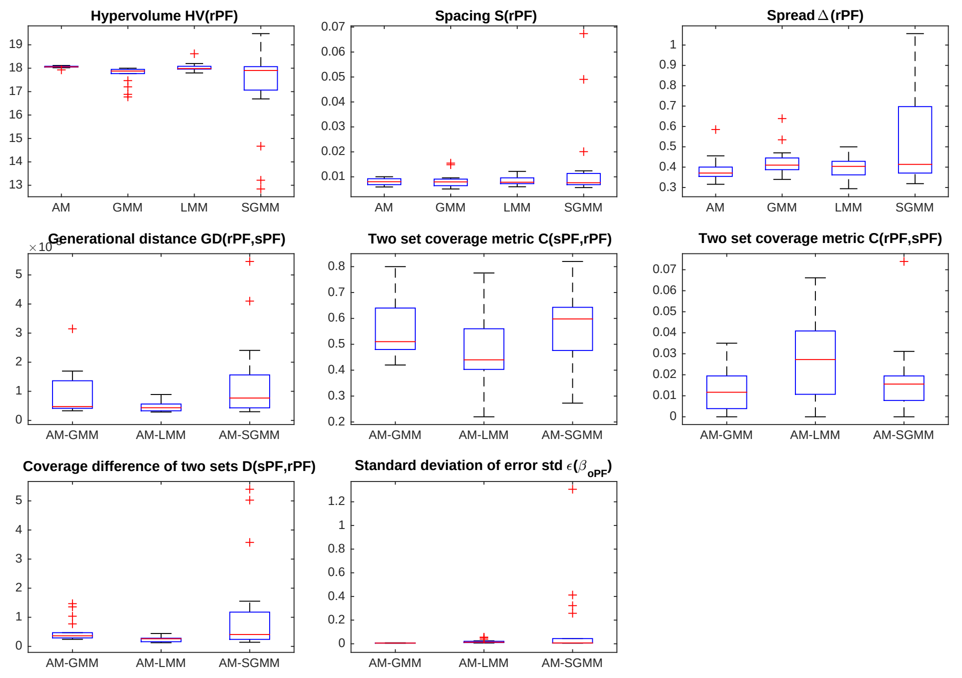


Figure 12.7: Example 3: Performance measures for RBDO utilizing a preconditioned quasi-Monte Carlo simulation with an analytical model (AM), local meta-models (LMM), global meta-model (GMM), and sparse global meta-model (SGMM).

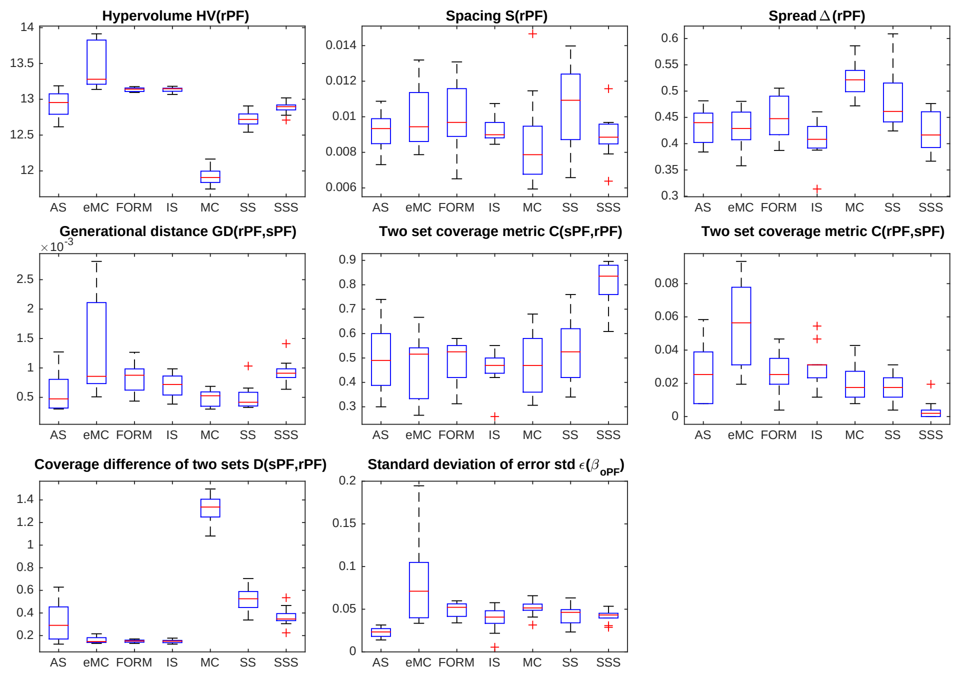


Figure 12.8: Example 3: Performance measures for RBDO utilizing an analytical model and different reliability assessment techniques, namely an Asymptotic sampling (AS), Enhanced Monte Carlo simulation (eMC), First-order reliability method (FORM), Importance sampling (IS), quasi-Monte Carlo simulation (MC), Subset simulation (SS), and Scaled sigma sampling (SSS).

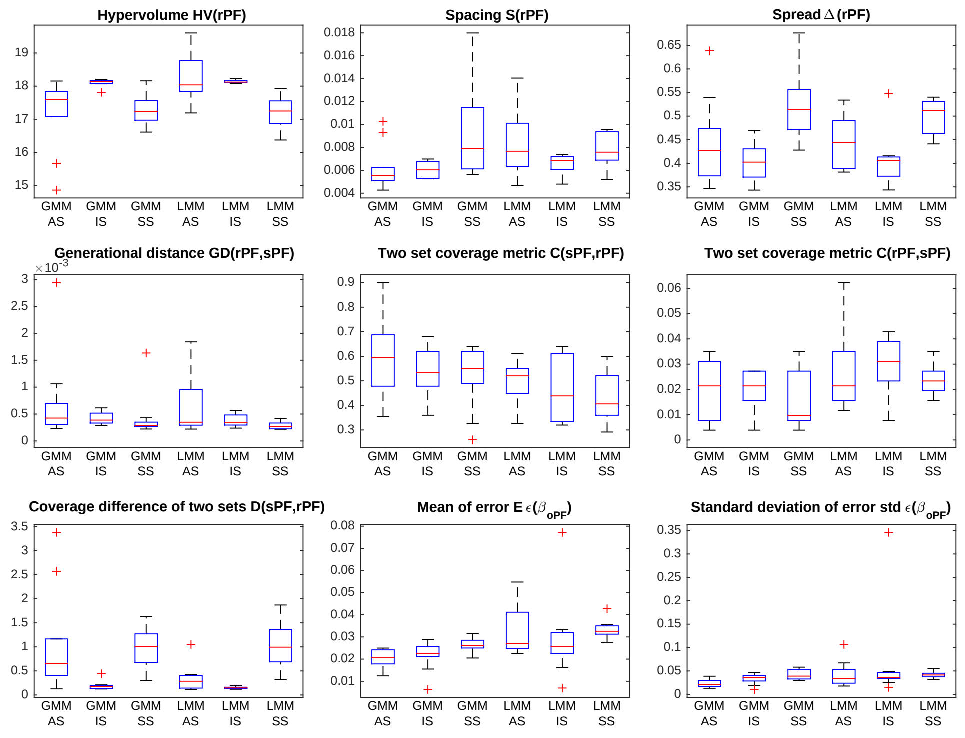


Figure 12.9: Example 3: Performance measures for RBDO utilizing an Asymptotic sampling, Importance sampling, and Subset simulation with a sparse global meta-model (SGMM), and local meta-models (LMM).

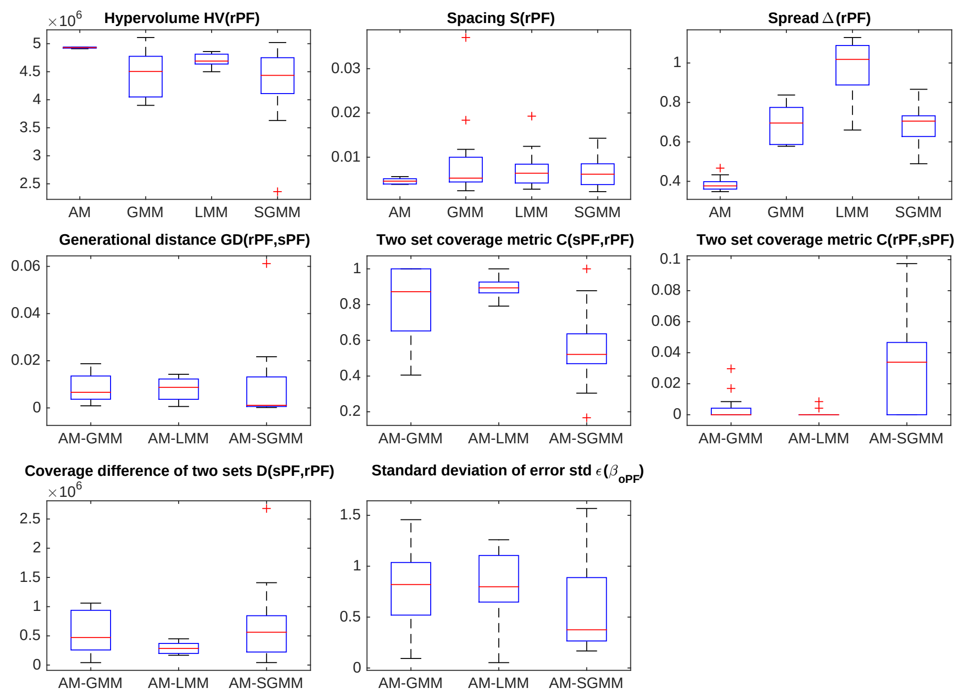


Figure 12.10: Example 4: Performance measures for RBDO utilizing a preconditioned quasi-Monte Carlo simulation with an analytical model (AM), local meta-models (LMM), global meta-model (GMM), and sparse global meta-model (SGMM).

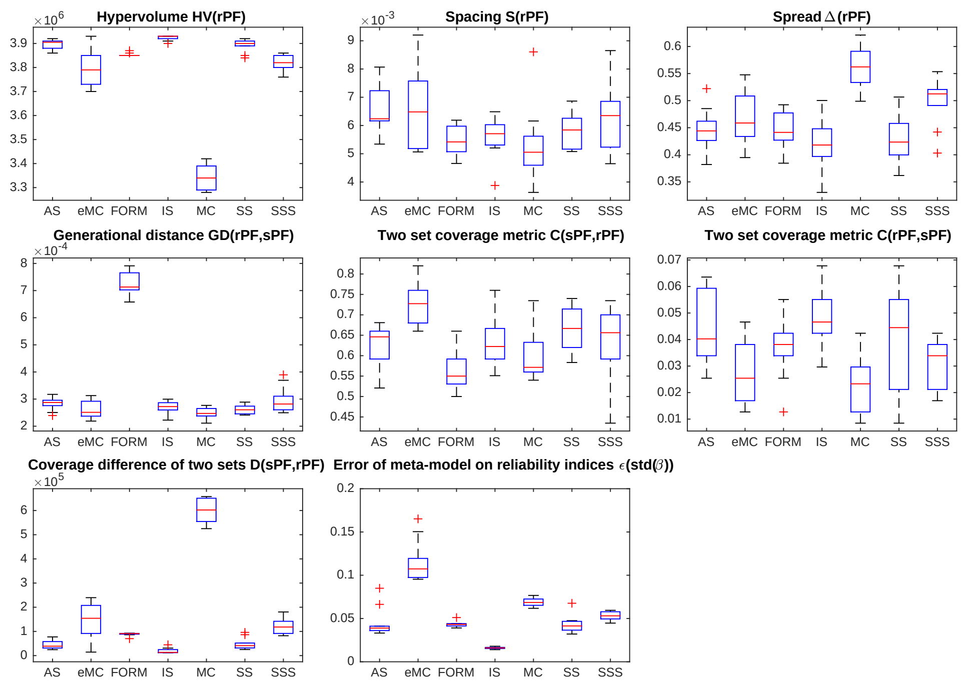


Figure 12.11: Example 4: Performance measures for RBDO utilizing an analytical model and different reliability assessment techniques, namely an Asymptotic sampling (AS), Enhanced Monte Carlo simulation (eMC), First-order reliability method (FORM), Importance sampling (IS), quasi-Monte Carlo simulation (MC), Subset simulation (SS), and Scaled sigma sampling (SSS).

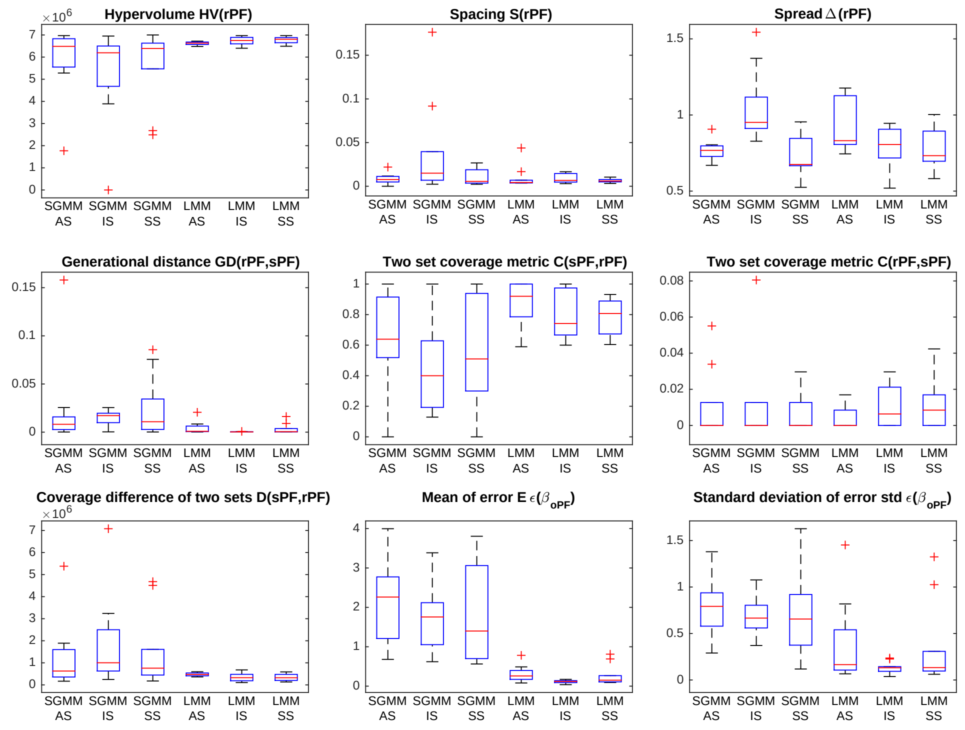


Figure 12.12: Example 4: Performance measures for RBDO utilizing an Asymptotic sampling, Importance sampling, and Subset simulation with a sparse global meta-model (SGMM), and local meta-models (LMM).

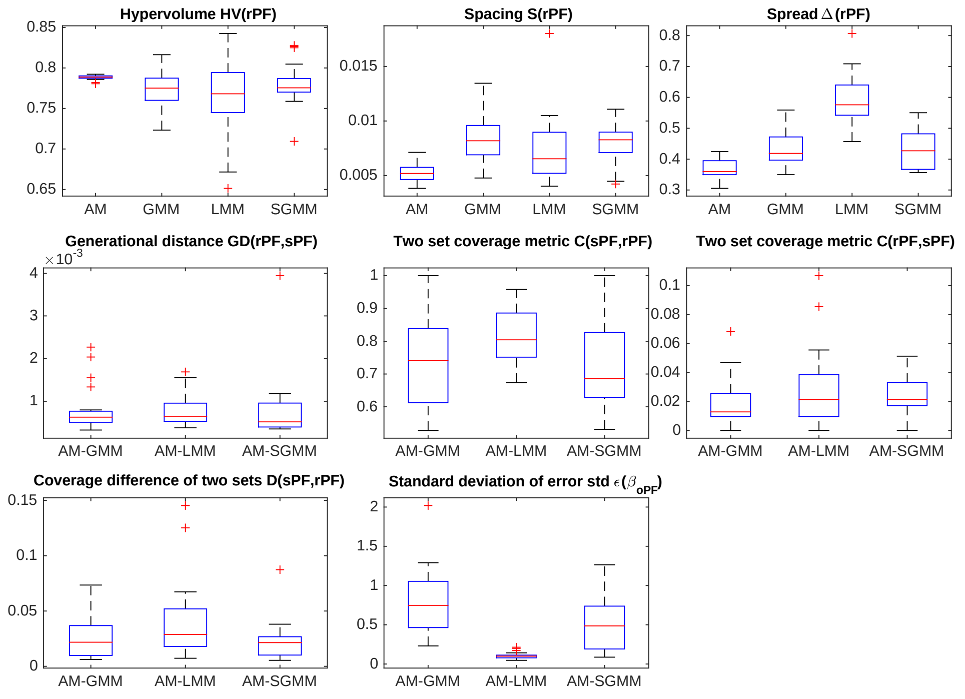


Figure 12.13: Example 5: Performance measures for RBDO utilizing a preconditioned quasi-Monte Carlo simulation with an analytical model (AM), local meta-models (LMM), global meta-model (GMM), and sparse global meta-model (SGMM).

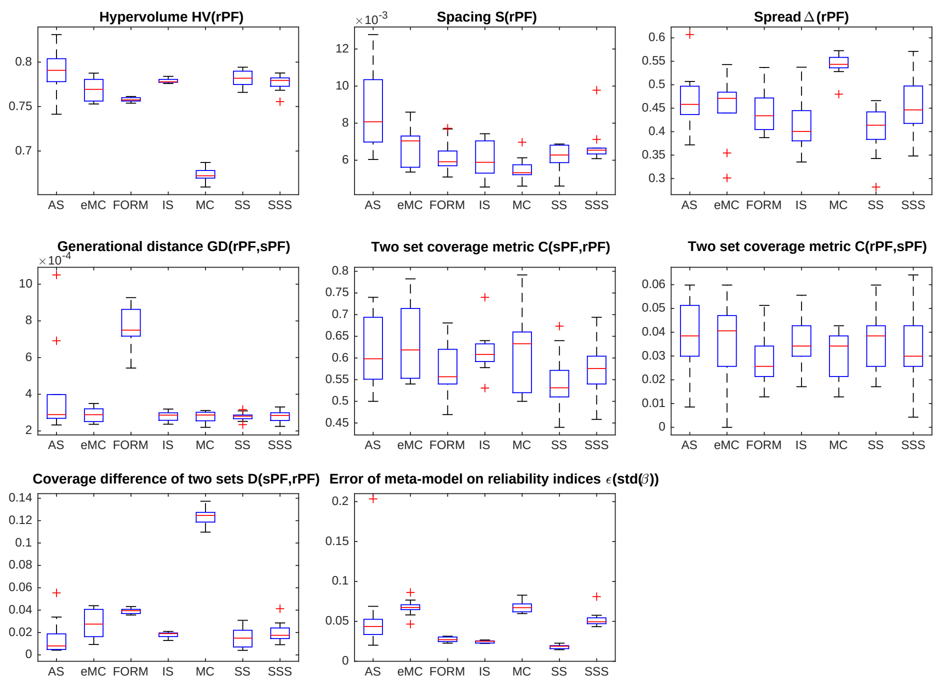


Figure 12.14: Example 5: Performance measures for RBDO utilizing an analytical model and different reliability assessment techniques, namely an Asymptotic sampling (AS), Enhanced Monte Carlo simulation (eMC), First-order reliability method (FORM), Importance sampling (IS), quasi-Monte Carlo simulation (MC), Subset simulation (SS), and Scaled sigma sampling (SSS).

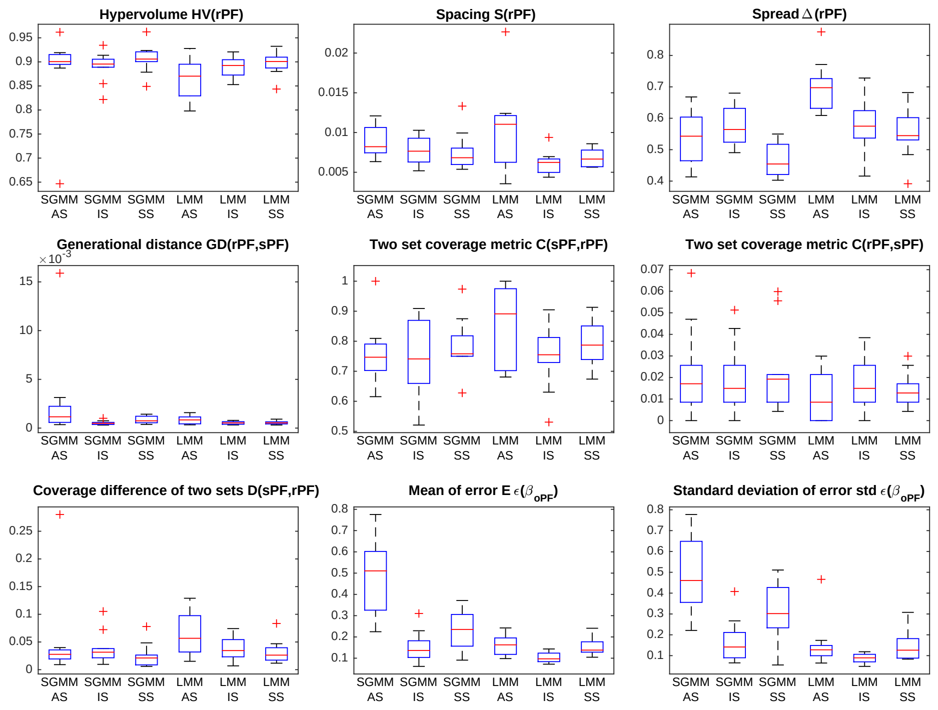


Figure 12.15: Example 5: Performance measures for RBDO utilizing an Asymptotic sampling, Importance sampling, and Subset simulation with a sparse global meta-model (SGMM), and local meta-models (LMM). Used metrics are a Hypervolume (HV), Spacing (S), Spread (Δ), Generational distance (GD), Two set coverage metric (C(A,B)), Coverage difference of two sets (D(A,B)), and mean and standard deviation of Error of the reliability index β ($\epsilon(\beta)$).

12.2 Unpublished combinations of simulation methods and meta-models

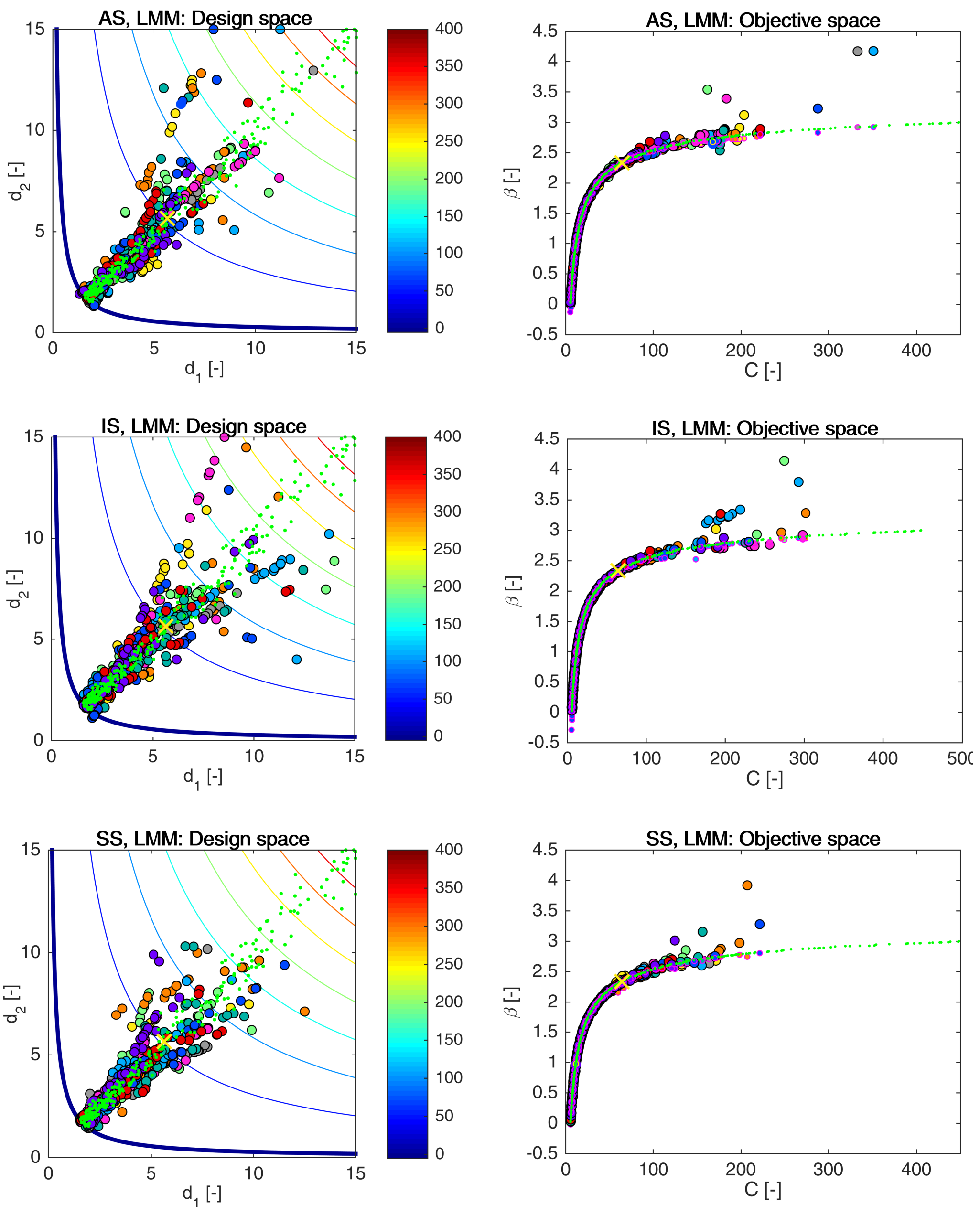


Figure 12.16: Example 1: Pareto sets (circles, left) and Pareto fronts (larger circles with black edges, right) for RBDO utilizing an Asymptotic sampling (AS), Importance sampling (IS), and Subset Simulation (SS) together with local meta-models (LMM). The smaller circles with magenta edges in the objective space represent the recalculated Pareto sets to Pareto fronts with quasi-Monte Carlo simulations with $\mathrm{CoV} < 5\%$. The yellow cross represents a result published by [3]. Green dots sets show the superior Pareto set and Pareto front obtained with an analytical model together with a preconditioned quasi-Monte Carlo simulation. Every colour means a different run.

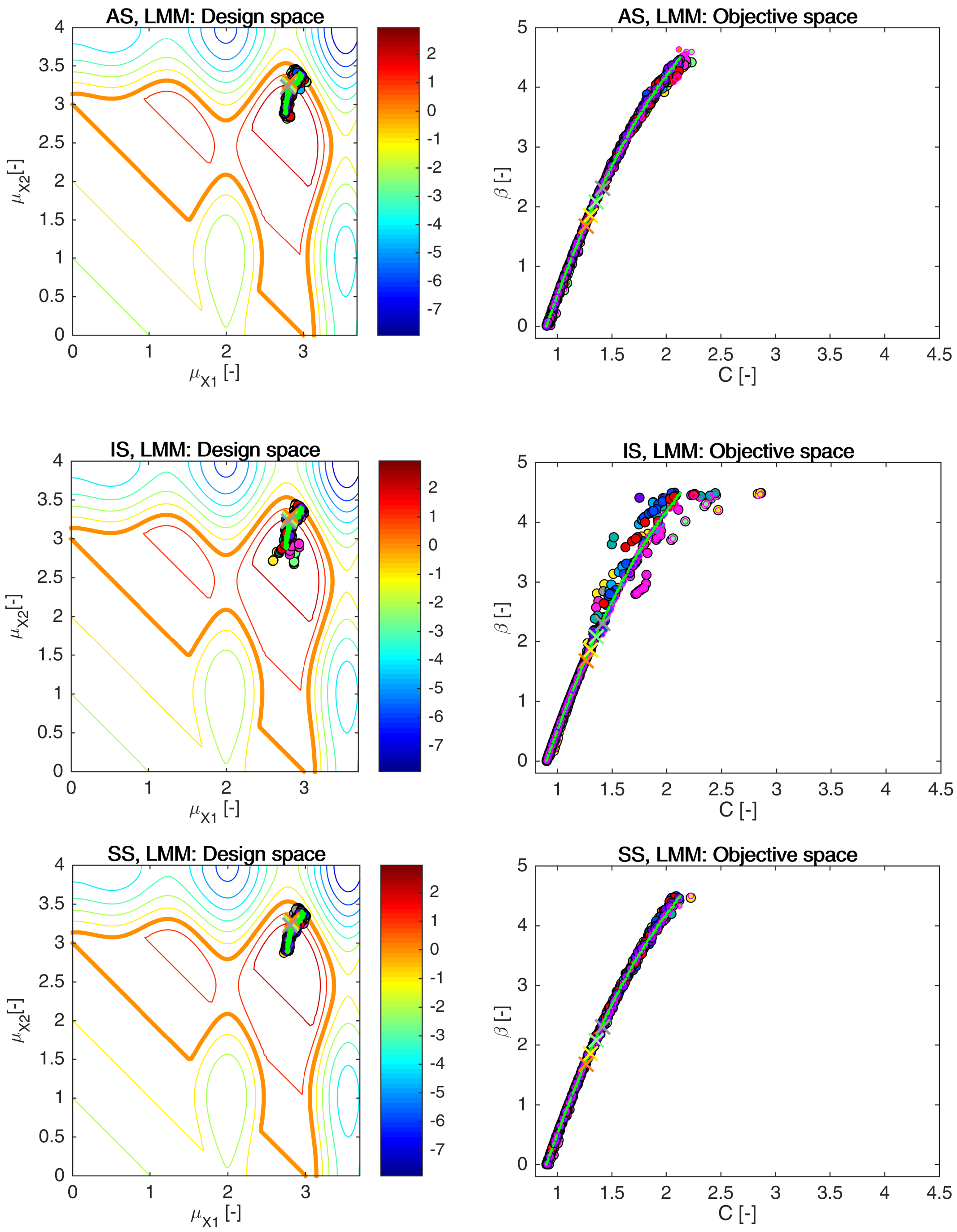


Figure 12.17: Example 2: Pareto sets (circles, left) and Pareto fronts (larger circles with black edges, right) for RBDO utilizing an Asymptotic sampling (AS), Importance sampling (IS), and Subset Simulation (SS) together with local meta-models (LMM). The smaller circles with magenta edges in the objective space represent the recalculated Pareto sets to Pareto fronts with quasi-Monte Carlo simulations with $\mathrm{CoV} < 5\%$. The yellow cross represents a result published by [3]. Green dots sets show the superior Pareto set and Pareto front obtained with an analytical model together with a preconditioned quasi-Monte Carlo simulation. Every colour means a different run.

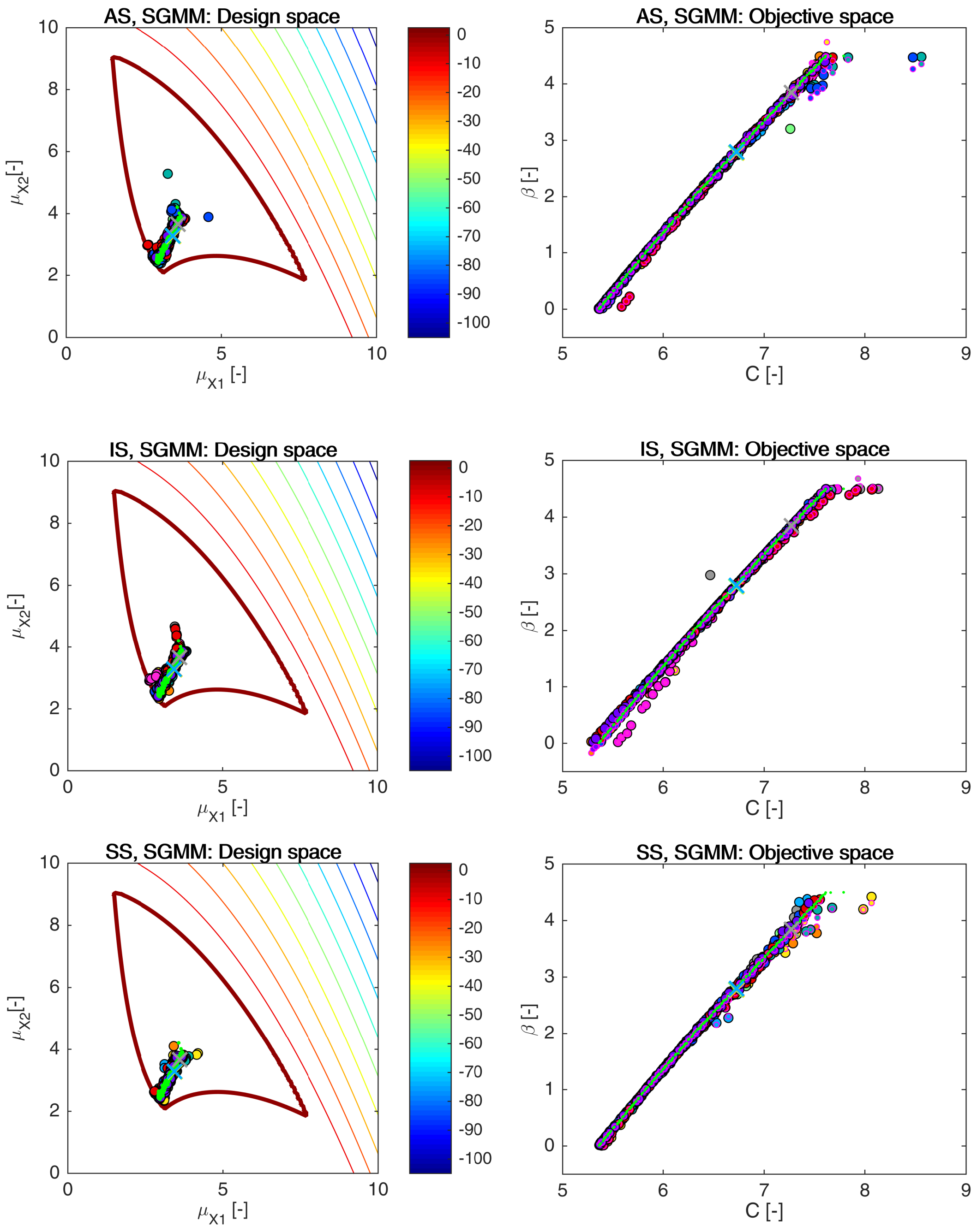


Figure 12.18: Example 3: Pareto sets (circles, left) and Pareto fronts (larger circles with black edges, right) for RBDO utilizing an Asymptotic sampling (AS), Importance sampling (IS), and Subset Simulation (SS) together with a sparse global meta-model (SGMM). The smaller circles with magenta edges in the objective space represent the recalculated Pareto sets to Pareto fronts with quasi-Monte Carlo simulations with $\mathrm{CoV} < 5\%$. The yellow cross represents a result published by [3]. Green dots sets show the superior Pareto set and Pareto front obtained with an analytical model together with a preconditioned quasi-Monte Carlo simulation. Every colour means a different run.

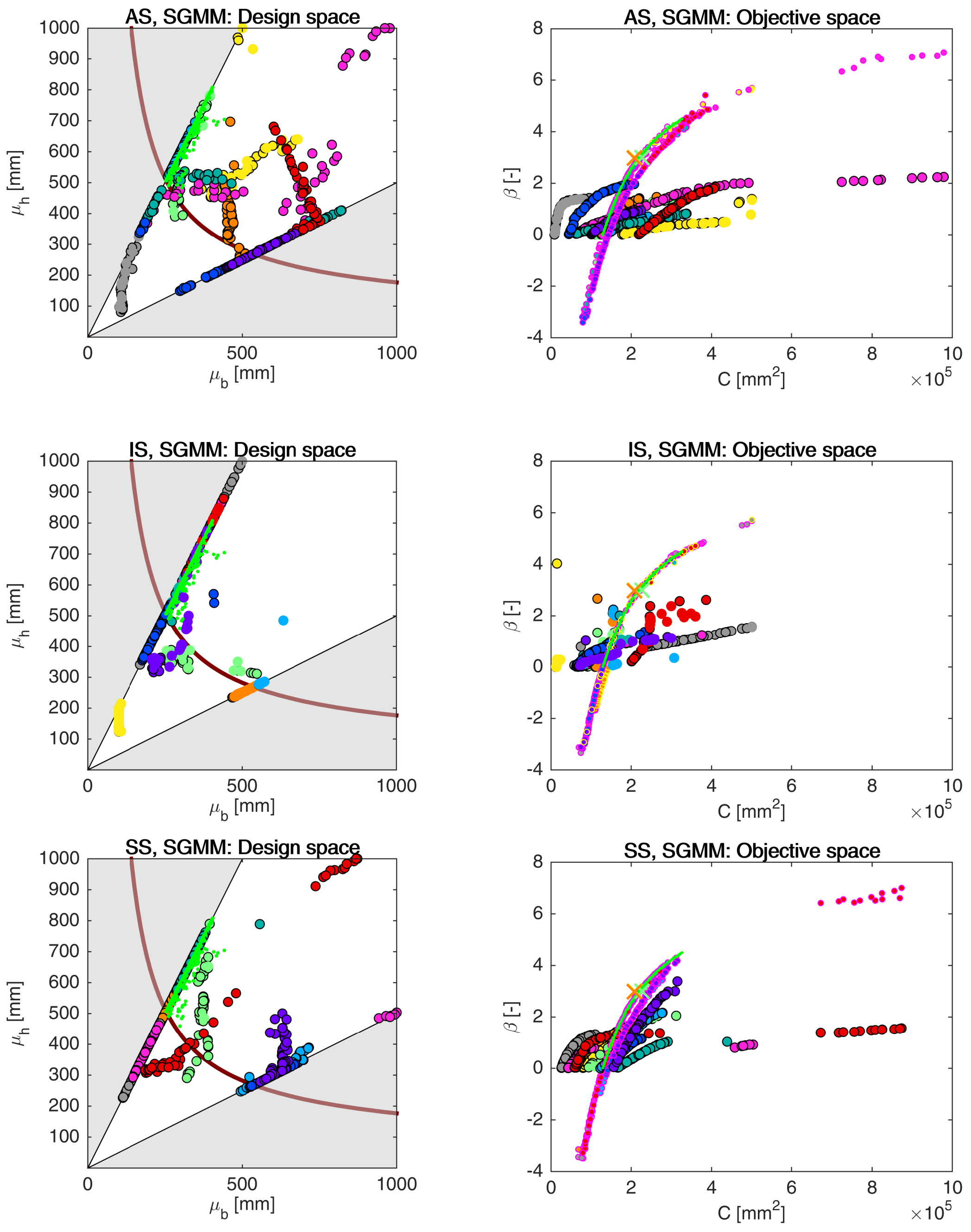


Figure 12.19: Example 4: Pareto sets (circles, left) and Pareto fronts (larger circles with black edges, right) for RBDO utilizing an Asymptotic sampling (AS), Importance sampling (IS), and Subset Simulation (SS) together with a sparse global meta-model (SGMM). The smaller circles with magenta edges in the objective space represent the recalculated Pareto sets to Pareto fronts with quasi-Monte Carlo simulations with $\mathrm{CoV} < 5\%$. The yellow cross represents a result published by [3]. Green dots sets show the superior Pareto set and Pareto front obtained with an analytical model together with a preconditioned quasi-Monte Carlo simulation. Every colour means a different run.

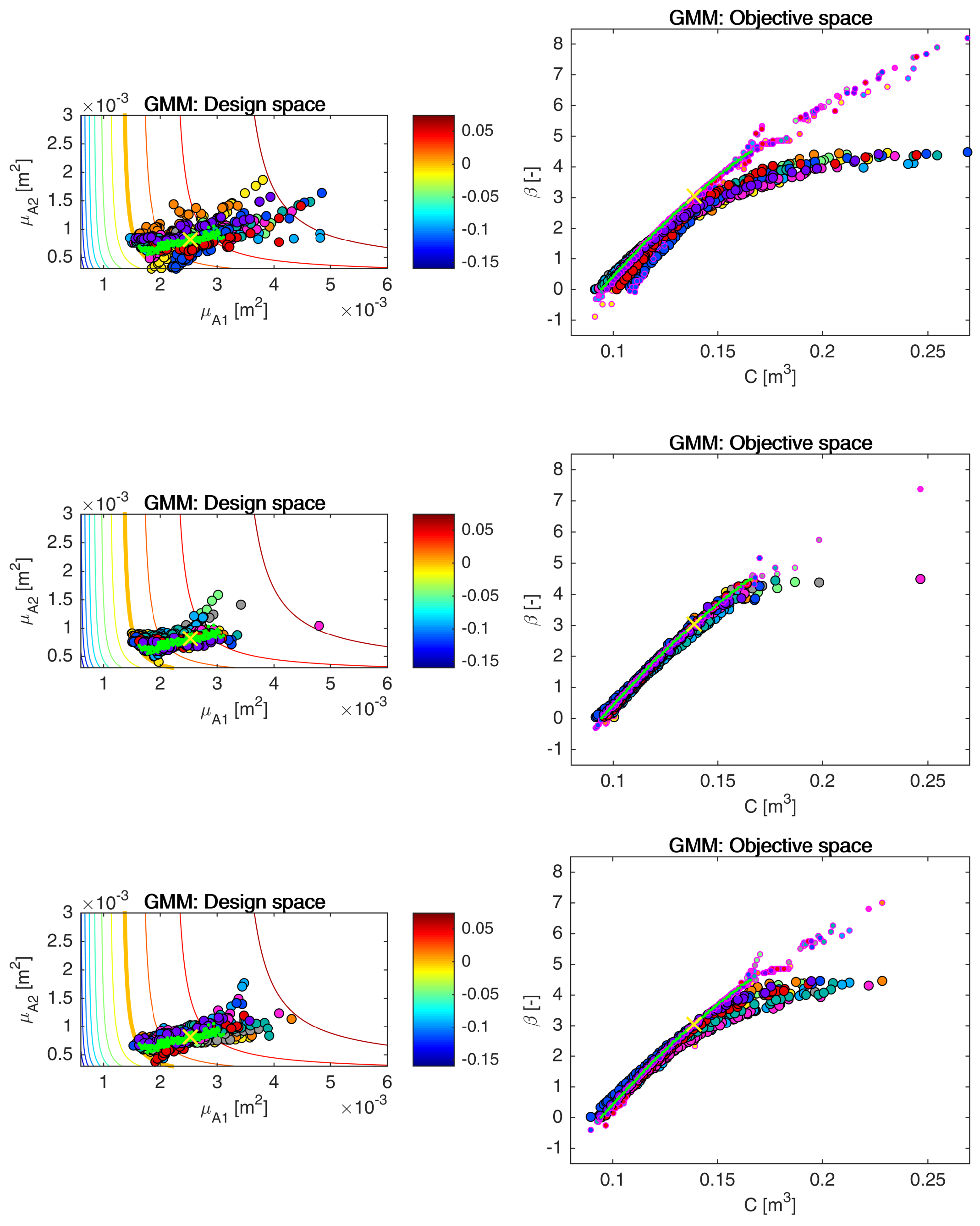


Figure 12.20: Example 5: Pareto sets (circles, left) and Pareto fronts (larger circles with black edges, right) for RBDO utilizing an Asymptotic sampling (AS), Importance sampling (IS), and Subset Simulation (SS) together with a global meta-model (GMM). The smaller circles with magenta edges in the objective space represent the recalculated Pareto sets to Pareto fronts with quasi-Monte Carlo simulations with $\mathrm{CoV} < 5\%$. The yellow cross represents a result published by [3]. Green dots sets show the superior Pareto set and Pareto front obtained with an analytical model together with a preconditioned quasi-Monte Carlo simulation. Every colour means a different run.

12.3 Summary data

	AS	IS	SS	AS	IS	SS	AS	IS	SS
Example	GMM			LMM			SGMM		
1	6	2	9	12	12	11	10	3	8
2	4	10	5	8	10	6	7	11	4
3	3	4	6	7	7	7	4	5	5
4	11	10	9	7	5	6	13	11	12
5	11	6	9	7	5	7	10	7	8

Table 12.1: Ranks from Pareto-efficiency conditions for two criteria, namely the mean and the standard deviation of the error, for approximation in the reliability assessment as well as model.

	MC-CoV			AS	eMC	FORM	IS	MC	SS	SSS	AS	IS	SS	AS	IS	SS	AS	IS	SS
Ex.	LMM	GMM	SGMM	AM							GMM			LMM			SGMM		
1	7	13	5	9	14	4	1	4	7	15	6	2	9	12	12	11	10	3	8
2	1	3	2	4	10	13	12	9	4	11	4	10	5	8	10	6	7	11	4
3	1	2	10	3	9	8	5	6	7	6	3	4	6	7	7	7	4	5	5
4	10	8	8	2	4	3	1	3	2	3	11	10	9	7	5	6	13	11	12
5	12	6	10	3	4	3	2	4	1	3	11	6	9	7	5	7	10	7	8

Table 12.2: Ranks from Pareto-efficiency conditions for two criteria, namely the mean and the standard deviation of the error. This table corresponds with Figure 7.41.

	MC-CoV			AS	eMC	FORM	IS	MC	SS	SSS	AS	IS	SS	AS	IS	SS	AS	IS	SS
Example	LMM	GMM	SGMM	AM							GMM			LMM			SGMM		
1	0.015	0.120	0.009	0.022	0.196	0.017	0.003	0.009	0.017	0.286	0.014	0.004	0.021	0.031	0.035	0.028	0.024	0.004	0.018
2	0.008	0.009	0.008	0.024	0.073	0.334	0.248	0.028	0.023	0.432	0.023	0.041	0.024	0.027	0.048	0.025	0.026	0.065	0.023
3	0.007	0.020	0.124	0.020	0.094	0.036	0.024	0.025	0.029	0.061	0.020	0.022	0.026	0.032	0.029	0.033	0.021	0.023	0.025
4	1.659	0.398	1.178	0.045	0.093	0.236	0.023	0.046	0.046	0.209	1.724	1.721	1.497	0.317	0.115	0.270	2.125	1.728	1.840
5	0.812	0.126	0.473	0.043	0.055	0.228	0.027	0.043	0.021	0.102	0.575	0.118	0.302	0.160	0.104	0.151	0.496	0.150	0.239

Table 12.3: Average values of the mean reliability index error for all examples. The green colour represents the best results; the red colour is for the worst results.

	MC-CoV			AS	eMC	FORM	IS	MC	SS	SSS	AS	IS	SS	AS	IS	SS	AS	IS	SS
Example	LMM	GMM	SGMM	AM							GMM			LMMl			SGMM		
1	0.020	0.165	0.010	0.020	0.179	0.007	0.002	0.009	0.018	0.192	0.014	0.005	0.021	0.080	0.072	0.054	0.023	0.005	0.019
2	0.007	0.009	0.008	0.027	0.075	0.450	0.390	0.053	0.028	0.079	0.027	0.116	0.028	0.034	0.110	0.030	0.030	0.148	0.027
3	0.006	0.019	0.135	0.023	0.081	0.050	0.038	0.051	0.044	0.042	0.023	0.033	0.042	0.043	0.066	0.042	0.034	0.041	0.036
4	0.786	0.787	0.530	0.045	0.115	0.043	0.016	0.069	0.043	0.053	0.782	0.538	0.534	0.386	0.128	0.351	0.822	0.691	0.705
5	0.792	0.109	0.519	0.057	0.067	0.027	0.025	0.068	0.018	0.053	0.611	0.128	0.367	0.154	0.089	0.155	0.492	0.172	0.323

Table 12.4: Average values of the standard deviation reliability index error for all examples. The green colour represents the best results; the red colour is for the worst results.

	MC-CoV			AS	eMC	FORM	IS	MC	SS	SSS	AS	IS	SS	AS	IS	SS	AS	IS	SS
Example	LMM	GMM	SGMM	AM							GMM			LMM			SGMM		
1	354	472	483	$5.3{\cdot}10^7$	$4.7{\cdot}10^7$	$1.3{\cdot}10^5$	$4.4{\cdot}10^7$	$4.7{\cdot}10^7$	$3.3{\cdot}10^7$	$4.7{\cdot}10^7$	418	358	369	413	367	418	404	349	358
2	308	424	428	$6.3{\cdot}10^7$	$4.7{\cdot}10^7$	$6.2{\cdot}10^4$	$5.9{\cdot}10^7$	$4.7{\cdot}10^7$	$4.4{\cdot}10^7$	$4.1{\cdot}10^7$	420	783	347	396	409	412	433	849	351
3	438	350	346	$5.7{\cdot}10^7$	$4.7{\cdot}10^7$	$9.9{\cdot}10^4$	$4.5{\cdot}10^7$	$4.7{\cdot}10^7$	$3.9{\cdot}10^7$	$4.1{\cdot}10^7$	387	1364	350	418	503	446	380	1251	350
4	428	373	346	$7.1{\cdot}10^7$	$4.7{\cdot}10^7$	$1.8{\cdot}10^5$	$4.5{\cdot}10^7$	$4.7{\cdot}10^7$	$4.9{\cdot}10^7$	$4.4{\cdot}10^7$	435	447	420	280	311	300	405	448	419
5	387	373	366	$6.3{\cdot}10^7$	$4.7{\cdot}10^7$	$3.2{\cdot}10^5$	$4.4{\cdot}10^7$	$4.7{\cdot}10^7$	$4.8{\cdot}10^7$	$4.4{\cdot}10^7$	464	443	419	338	334	330	450	437	412

Table 12.5: Average number of analytical model evaluations for all examples. The green colour represents the best results; the red colour is for the worst results.

	GMM			LMM			SGMM		
Example 1	AS	IS	SS	AS	IS	SS	AS	IS	SS
elapsed time [sec]	369	618	226	202	462	210	353	584	211
elapsed time [hours]	0.103	0.172	0.063	0.056	0.128	0.058	0.098	0.162	0.059
initial DoE	50	50	50	200	200	200	50	50	50
added samples	368	308	319	213	167	218	354	299	308
analytical g(x)-calls	418	358	369	413	367	418	404	349	358
MM built for opt.	31	31	31	3050	3050	3050	31	31	31
MM built for update	0	0	0	606.6	662	601.8	0	0	0
MM-calls	$5.34\cdot10^7$	$8.95\cdot10^7$	$3.62\cdot10^7$	$4.25\cdot10^7$	$8.90\cdot10^7$	$4.31\cdot10^7$	$5.20\cdot10^7$	$8.94\cdot10^7$	$3.42\cdot10^7$

	GMM			LMM			SGMM		
Example 2	AS	IS	SS	AS	IS	SS	AS	IS	SS
elapsed time [sec]	393	1183	224	251	383	175	444	1155	237
elapsed time [min]	6.5	19.7	3.7	4.2	6.4	2.9	7.4	19.3	3.9
elapsed time [hours]	0.109	0.329	0.062	0.070	0.106	0.049	0.123	0.321	0.066
initial DoE	50	50	50	200	200	200	50	50	50
added samples	370	733	297	196	209	212	383	799	301
analytical g(x)-calls	420	783	347	396	409	412	433	849	351
MM built for opt.	31	31	31	3050	3050	3050	31	31	31
MM built for update	0	0	0	764	3595	1345	0	0	0
MM-calls	$6.2\cdot10^7$	1.2.E+08	$4.3\cdot10^7$	6.28E+07	5.94E+07	4.26E+07	$6.2\cdot10^7$	1.2.E+08	$4.3\cdot10^7$

	GMM			LMM			SGMM		
Example 3	AS	IS	SS	AS	IS	SS	AS	IS	SS
elapsed time [sec]	384	1567	209	215	544	153	364	1745	203
elapsed time [hours]	0.107	0.435	0.058	0.060	0.151	0.043	0.101	0.485	0.056
initial DoE	50	50	50	200	200	200	50	50	50
added samples	337	1314	300	218	303	246	330	1201	300
analytical g(x)-calls	387	1364	350	418	503	446	380	1251	350
MM built for opt.	31	31	31	3050	3050	3050	31	31	31
MM built for update	0	0	0	788	3714	1237	0	0	0
MM-calls	$5.61\cdot10^7$	$9.05\cdot10^7$	$3.68\cdot10^7$	$5.98\cdot10^7$	$9.05\cdot10^7$	$3.39\cdot10^7$	$5.55\cdot10^7$	$9.06\cdot10^7$	$3.52\cdot10^7$

	GMM			LMM			SGMM		
Example 4	AS	IS	SS	AS	IS	SS	AS	IS	SS
elapsed time [sec]	169	1317	175	442	706	404	116	1364	158
elapsed time [hours]	0.047	0.366	0.049	0.123	0.196	0.112	0.032	0.379	0.044
initial DoE	50	50	50	200	200	200	50	50	50
added samples	385	397	370	80	111	100	355	398	369
analytical g(x)-calls	435	447	420	280	311	300	405	448	419
MM built for opt.	31	31	31	3050	3050	3050	31	31	31
MM built for update	0	0	0	322	544	430	0	0	0
MM-calls	$1.60\cdot10^7$	$8.67\cdot10^7$	$1.95\cdot10^7$	$6.11\cdot10^7$	$8.65\cdot10^7$	$4.59\cdot10^7$	$1.18\cdot10^7$	$8.41\cdot10^7$	$1.93\cdot10^7$

	GMM			LMM			SGMM		
Example 5	AS	IS	SS	AS	IS	SS	AS	IS	SS
elapsed time [sec]	714	1757	492	554	1299	546	718	2198	496
elapsed time [hours]	0.198	0.488	0.137	0.154	0.361	0.152	0.199	0.610	0.138
initial DoE	50	50	50	200	200	200	50	50	50
added samples	414	393	369	138	134	130	400	387	362
analytical g(x)-calls	464	443	419	338	334	330	450	437	412
MM built for opt.	31	31	31	3050	3050	3050	31	31	31
MM built for update	0	0	0	545	547	489	0	0	0
MM-calls	$6.22\cdot10^7$	$8.95\cdot10^7$	$4.38\cdot10^7$	$6.25\cdot10^7$	$8.92\cdot10^7$	$4.89\cdot10^7$	$6.01\cdot10^7$	$8.95\cdot10^7$	$4.31\cdot10^7$

Table 12.6: Comparison of statistics for RBDO using an Asymptotic sampling, Importance sampling, and Subset simulation with a global meta-model, local meta-models, and sparse global meta-model.