Efficient Multi-Objective Ranking and Selection in the Presence of Uncertainty

by

Wen Zhang

Thesis

Submitted to the University of Warwick

for the degree of Doctor of Philosophy

Doctor of Philosophy

Warwick Business School

September 2018

THE UNIVERSITY OF WARWICK

Contents

List of Tables

List of Figures

Acknowledgments

Thanks for Professor Juergen Branke, Doctor Wenjuan Zhang, Doctor Nalan Gulpinar, Professor Amos Ng, my parents, Mr. Yang Tao, Professor Henry Wynn and all other colleagues in the University of Warwick for their kind help during these tough but interesting four years. I have never thought I will be able to get a title of Doctor in my whole life, but with this experience, I believe I can do anything I want in the future.

Declarations

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis. This thesis has also not been submitted for any degree in any university previously. I further declare that one paper titled *A new myopic sequential sampling algorithm for multi-objective problems*, drawn from Chapter 3 of this thesis, was co-authored with Juergen Branke, has been published in *Winter Simulation Conference 2015*. Furthermore, the paper titled *Multiobjective ranking and selection based on hypervolume*, drawn from Chapter 5 of this thesis, was co-authored with Juergen Branke and Yang Tao, has been published in *Winter Simulation Conference 2016*. Finally, the paper titled *Identifying Efficient Solutions via Simulation: Myopic Multi-Objective Budget Allocation for the Bi-objective Case*, drawn from Chapter 3, 4, 5 of this thesis, was co-authored with Juergen Branke, has been submitted to *OR Spectrum* during the autumn of 2018.

Abstract

We consider the problem of ranking and selection with multiple-objectives in the presence of uncertainty. Simulation optimisation offers great opportunities in the design and optimisation of complex systems. In the presence of multiple objectives there is usually no single solution that performs best on all the objectives. Instead, there are several Pareto-optimal (efficient) solutions with different trade-offs which cannot be improved in any objective without sacrificing performance in another objective. For the case where alternatives are evaluated on multiple stochastic criteria, and the performance of an alternative can only be estimated via simulation, we consider the problem of efficiently identifying the Pareto optimal designs out of a (small) given set of alternatives. We develop a simple myopic budget allocation algorithm and propose several variants for different settings. In particular, this myopic method only allocates one simulation sample to one alternative in each iteration. Empirical tests show that the proposed algorithm can significantly reduce the necessary simulation budget and perform better than some existing well known algorithms in certain settings.

Abbreviations

PCS	Probability of Correct Selection
$P(CS)$	Estimated Probability of Correct Selection
PGS	Probability of Good Selection
$P(GS)$	Estimated Probability of Gorrect Selection
EOC	Expected Opportunity Cost
$MORS$	Multi-Objective Ranking and Selection
$M - MOBA$	Myopic Multi-Objective Budget Allocation
$MOCBA$	Multiobjective Optimal Computing Budget Allocation
$MOEA$	Multi-Objective Evolutionary Algorithm
HVD	Hypervolume Difference
MHV	Marginal Hypervolume
HVD	hypervolume difference
ZDT	Zitzler, Deb and Thiele

Chapter 1

Introduction

1.1 Motivation

This thesis considers the problem of Multi-Objective Ranking and Selection (MORS), i.e., how to allocate sampling across alternatives efficiently in simulation where the selection criteria is multiple and conflictive.

Decision makers (DM) in the real world are often faced with optimisation scenarios where they need to find the systems of interest from a number of alternatives. These alternative systems are usually dynamic and complex in nature, making it difficult or even impossible to build analytical models for evaluation. One of the most popular methods to tackle these complex system problems is simulation. Simulation is a powerful tool to improve the performance of systems in both industry and business practices. It contributes to the efficient management of processes and systems, the main focus of many industrial engineers, by providing a what-if analysis and the general process of simulation is to change current parameter settings in the operation of the system and see whether the result is better or worse [Yoon and Bekker, 2017]. It is usually not easy to estimate the effect of the changes because the system is often complex and stochastic, simulation helps the decision maker by simulating the real system and providing the estimates of system performance. Examples of the application of simulation in complex systems include

1. Supply chain management. We would like to make inventory decisions in a supply chain. The suppliers are located in different geographic regions and the supplies are subject to multi-level local disruptions of each supplier individually and to two-level regional disruptions of all suppliers in the same region. The two conflicting objectives are to minimise expected cost and maximise expected service level [Sawik, 2015]. We want to satisfy as much as possible

customers' demands and minimise the cost. Since the distribution of customer demand is stochastic and changing over time, it is not easy to find an analytical solution and simulation will provide useful insights to the DM.

2. Chemical process optimisation problem. We would like to choose the inputs to an industrial chemical process to maximise the quality of the output. For example, Lomelí-Rodríguez et al. [2017] present a process simulation, multi objective optimisation, and sensitivity analysis were performed in a simulation software for a library of biomass-derived renewable polyesters. This comprehensive simulation and optimisation work then provides a preliminary work for the process design, process intensification and industrialisation of fundamental unit operations.

3. Bike-sharing system. A very common problem in a bike-sharing system is how to optimise both bike and dock allocations for each station at the beginning of the day, so that the expected number of customers who cannot find a bike, or cannot find a dock to return a bike is minimised [Jian et al., 2016]. Jian et al. [2016] propose a gradient-like heuristic methods that can improve any given allocation based on a discrete-event simulation model of the system.

4. Fire prediction. Forest fires are time evolving disasters that consume environmental and financial resources, endangering the rescue units that try to mitigate them. As such, a simulator that can predict the fire propagation is essential to the placement of control systems, in order to reduce the loss of natural resources without risking the firefighters [Morales et al., 2014].

5. Financial problems. Simulation is also widely used in the financial area including financial forecasting, valuation analysis, financial planning and debt analysis. For instance, a fixed-income specialist might invest in fixed-rate products, however, the specialist might be funded by floating rate debt returns. Basis risk exists in such a system, and the evolution of interest rates is the real-life event that the specialist would worry about. A simulation of interest rates could greatly help the specialist to design a portfolio to reduce risk [Allman et al., 2011].

6. Healthcare. Simulation has been used for modelling healthcare systems for decades and Brailsford [2007] presents a review of applications of simulation in healthcare. Examples include screening diabetic patients for eye complications and mother-to-child transmission of HIV in developing countries or

organisational issues of resource allocation and capacity planning [Brailsford, 2007].

7. Six Sigma and Simulation. In the work of El-Haik and Al-Aomar [2006] and Joines and Roberts [2013], simulation is used in the implementation of Six Sigma. For example, the cost of performing a design of experiments with replications is too high (e.g., raw material cost, cost of shutting down current process) and researchers have worked with companies in developing process and Monte Carlo simulation models that could be used to determine their capabilities and ascertain the potential improvement in their changes.

8. Biomedical. Since it has become increasingly difficult to perform animal experiments because of issues related to the procurement of animals, and strict regulations and ethical issues related to their use [Badyal et al., 2009], biomedical industry already uses computer modelling and simulation in the development, and to a lesser extent in the assessment process [Viceconti et al., 2016].

As simulation plays an important role in a wide range of aspects in the modern society as the above examples show, how to improve the efficiency of simulation has been a hot topic in the simulation realm in recent years.

Simulation optimisation aims to efficiently identify the best possible alternative, where best is defined as best expected performance. Since an alternative's true performance is unknown and can only be evaluated by stochastic simulation, it is usually necessary to average over several simulation runs in order to obtain accurate performance estimates. As the process of testing an alternative might either involve a time-consuming computer simulation or a physical experiment, to be efficient, we need to decide how many simulations need to be done for each alternative. Ranking and selection (R&S) can be viewed as the most fundamental class of simulation optimisation problems and it focuses on the situation when the feasible region is a small, finite set and the goal is to identify the best alternative [Chau et al., 2014]. R&S methods aim to allocate simulation replications more efficiently, and this research area has received substantial interest in recent years, e.g., Chau et al. [2014]. A straightforward example of R&S problem is displayed in Figure 1.1. In this scenario, saying we want to select the alternative has the true maximum value. Current performance of each alternative is shown as Figure 1.1 after initial three samplings. We observed that solution circle is obviously inferior to others, but for triangle and square alternatives, we cannot decide which one is better. Therefore, R&S methods will spend more simulation replications of triangle and square alternatives to distinguish which one is better rather than the circle alternative which we have already been confident to a certain extent that is not as good as others.

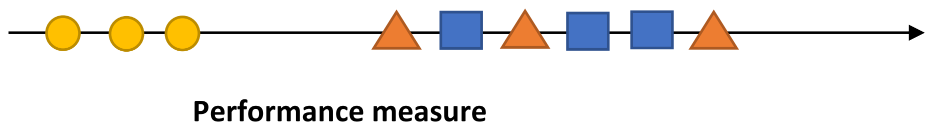

Figure 1.1: Simple example of R&S problem

However, many real-world simulation optimisation problems require the consideration of multiple conflicting objectives. In this case, there is usually no single solution that performs best in all objectives, but a set of Pareto-optimal solutions with different trade-offs. A solution is called *Pareto-optimal* or *efficient* if there is no other solution that performs better in all objectives. A Pareto front or Pareto set is the set of solutions that are all Pareto efficient. For instance, different staffing levels at a call centre will incur different costs and different customer waiting times, and a solution is Pareto optimal, if there is no better solution that has lower cost as well as lower customer waiting times. In the presence of multiple stochastic criteria, the R&S problem becomes a multi-objective ranking and selection problem where the goal is to efficiently identify the set of Pareto optimal solutions.

Although plenty of research has been published on single objective R&S, there is little research on MORS. In this thesis, we propose a simple yet powerful Myopic Multi-Objective Budget Allocation (M-MOBA) framework. M-MOBA is myopic in the sense that it only allocates simulation samples to one alternative in each iteration. It is an extension of the small sample procedures proposed by Chick et al. [2010] to the multi-objective case, which is the first and only attempt to the best our knowledge so far. M-MOBA is easy to compute and avoids some of the approximations necessary for other methods. We show how this framework can be adapted to different bi-objective problem settings.

There are a variety of goals in R&S. The simplest goal is to maximise the probability of correct selection (PCS). For a minimisation problem, the true PCS is defined mathematically as

$$PCS = P(\mu_{x_s} \leq \mu_{x^*}),$$

where μ_{x^*} is the mean performance of true best solution x^* and μ_{x_s} is the mean performance of the selected solution x_s.

The PCS reported in empirical experiments in the literature is usually the estimated PCS. For Q replications of an experiment, the PCS can be estimated as

$$P(CS) = \left(\frac{Q_c}{Q}\right)$$

where Q_c is the number of replications for which the method correctly identified the best alternative.

To tackle the MORS problem with PCS measure, we develop a M-MOBA PCS procedure, which is described in Chapter 3.

However, if two alternatives have almost identical performance, even a large number of samples may not be able to correctly identify the better one, and anyway the decision maker might not care about very small differences. So it seems natural to introduce an *indifference zone*, the smallest difference δ that deserves to be discerned. Then the goal is to maximise the Probability of Good Selection (PGS), which is the probability that the selected alternative is not worse by more than δ compared to the true best. For a minimisation problem, PGS is calculated as

$$PGS = P(\mu_{x_s} \leq \mu_{x^*} + \delta),$$

where μ_{x^*} is the mean performance of true best solution x^* and μ_{x_s} is the mean performance of the selected solution x_s. The estimated PGS can be defined similarly to the estimated PCS. Although some research have been taken for the close performance alternatives in the single objective R&S problem, research in MORS in this realm is rare due to the difficulty of defining indifference zone and PGS in the multi-objective setting.

To tackle this problem, a M-MOBA variant to solve the problem when there are similar performance alternatives is proposed in Chapter 4.

Another commonly used goal is to minimise the expected opportunity cost (EOC), defined as the true difference in performance between the true best and the selected system. Expected opportunity cost (EOC) is of practical concern in business, engineering and other applications where design performance represents economic value and is particularly useful for risk-neutral decision makers [Chick and Wu, 2005]. While PCS only cares about whether a solution is correct, opportunity cost intuitively describes how far away the selected alternative is from the true best system [Lee et al., 2007]. Again, how to define EOC in the multi-objective is not straightforward and we introduce hypervolume (HV) to tackle this problem.

The M-MOBA variant using hypervolume difference (HVD) as the performance measure is introduced in Chapter 5.

In addition, it is worthwhile to explore how to implement M-MOBA in different settings. Chapter 6 describes an instance where M-MOBA is integrated in Multi-objective Evolutionary Algorithms (MOEA) to increase the efficiency of fitness function evaluation.

1.2 Objectives and significance of the study

In this study, we consider the multi-objective ranking and selection problem and focus specially on sampling allocation across alternatives. Since simulation in reality is often time consuming or costly, one of the areas that has seen substantial interest in the realm of simulation in recent years is Ranking and Selection. R&S aims to efficiently identify the best of a finite set of alternatives, where best is defined as having the best expected performance, and the performance can only be estimated from stochastic samples. At the same time, in practice, the criteria of selection are often multiple and conflicting. Therefore, the problem of ranking and selection in the context of multiple objectives deserves to be studied. So far, at least to the best of knowledge, there are only a few works to solve MORS problems, which is reviewed later, and this study aims to tackle the problem of MORS, developing new, simple and more efficient techniques. More specifically, we can summarise the significance of this research in parallel to main objectives as follows:

1. A comprehensive review on existing literature on MORS is provided.

2. We develop a simple but efficient method which uses probability of correct selection as performance criterion to tackle MORS problems. This research extends the myopic expected value of information (EVI) method [Chick et al., 2010] to the bi-objective problem for the first time and proposes a M-MOBA method [Branke and Zhang, 2015] to tackle the MORS problem. We demonstrate empirically that M-MOBA substantially improves efficiency compared to the naive Equal allocation method and other existing methods in certain

situations.

3. A variant of M-MOBA that allows different objectives to be sampled independently is introduced. This may be relevant in problems where the different criteria are determined by different simulation runs.

4. A variant using hypervolume change is developed. The advantage of using the hypervolume measure is that it could find solutions close to the true Pareto front, as well as a good spread of solutions along the true Pareto front [Beume et al., 2007]. We propose a new performance measure for MORS, the hypervolume difference, based on the HV measure that is commonly used to evaluate results in multiobjective optimization [Emmerich and Klinkenberg, 2008]. As far as we know, this paper is the first attempt to use hypervolume as the performance measure to the MORS problem. We then continue to derive a new myopic ranking and selection procedure similar to our M-MOBA [Branke and Zhang, 2015] but based on the new HV difference criterion rather than the probability of correct selection.

5. We also study an alternative way of measuring performance by taking into account an indifference zone, which is useful when tiny difference exists among certain alternatives. We propose a new definition of indifference zone and Probability of Good Selection (PGS) to overcome the deficiencies of the existing method. We demonstrate empirically that this variant can substantially improve efficiency.

6. The developed M-MOBA method is integrated into the Evolutionary Algorithm (EA) in the ranking and selection phrase and empirically demonstrate that M-MOBA can help to improve the efficiency of EA.

7. Empirical evaluations of our approach are presented. Empirical results show that M-MOBA is able to substantially reduce the number of simulation runs needed to obtain a desired performance, when compared to equal allocation or other methods from the literature.

In summary, findings of this study constitute an important contribution to the MORS literature. It provides guidelines to optimally allocate computing budget for practitioners carrying out real world simulation experiments. The proposed approach may have great potential in application since it is easy to implement in different settings.

1.3 Thesis organisation

The rest of the thesis is organised as follows.

- Chapter 2 provides a comprehensive review of the existing literature of ranking and selection. Existing main basic approaches to R&S and MORS are summarised.

- Chapter 3 describes the proposed M-MOBA method with the PCS performance measure and a variant of independent sampling. Numerical experimental results show that M-MOBA performs as well as other well known methods and even better in certain settings. This chapter is based on [Branke and Zhang, 2015].

- Chapter 4 considers the MORS problem when some systems may have very similar objective values and a decision maker might not be too concerned with small differences between these systems. In order to tackle this problem, we introduce a new concept of indifference zone and good selection, and develop a corresponding M-MOBA indifference zone (M-MOBA IZ) algorithm. Numerical experimental results demonstrate that M-MOBA IZ is able to significantly save simulation samples in this situation.

- Chapter 5 shows that the M-MOBA method can be extended to a variant with Hypervolume performance measure. Empirical tests show that the proposed method performs well with respect to the stated hypervolume objective. This chapter is based on [Branke et al., 2016].

- Chapter 6 provides an application of the M-MOBA method. We develop a solution framework which integrates a multi-objective evolutionary algorithm (MOEA) with the M-MOBA method for multi-objective simulation optimisation problems.

- Chapter 7 concludes this study by summarising significance and contributions of this research. Limitations to this study, including the specific assumptions made and the solution approaches employed, are further discussed, suggesting future research directions to enriching and enhancing the work reported in this thesis.

Chapter 2

Literature review

Simulation optimisation combines two well-established paradigms simulation and optimisation [Fu et al., 2008]. Simulation modelling is a powerful tool for capturing the complexity of real-world systems and optimisation software packages now have the ability to consider the number of variables in the millions [Chau et al., 2014]. The combination of the two most useful paradigms in the operations research realm is still in its relative infancy, which makes it an extremely fertile area for research [Chau et al., 2014]. When the solution space is finite and small enough, simulating each alternative is possible and the simulation optimisation problem becomes a ranking and selection problem. This chapter provides a comprehensive overview of relevant R&S and literature. Section 2.1 formulates the problem of R&S and MORS. Section 2.2 reviews major single-objective R&S methods, whereas Section 2.3 reviews the main methods to MORS problems. Section 2.4 provides a brief summary of further related work and Section 2.5 introduces the multi-objective evolutionary algorithms. Section 2.6 summarises the literature and justify the work of this thesis.

2.1 General formulation of R&S problem

Consider a set of m designs with the true unknown performance of each design i, without loss of generality, the single objective R&S problem can be formulated as

$$\begin{aligned} \underset{n_i}{\text{maximise}} \quad & \mathcal{P} \\ \text{subject to} \quad & \sum_{i=1}^{m} n_i \leq N_t. \end{aligned}$$

where $\mathcal{P}$ is DM's selected performance measure (e.g., PCS, PGS, etc.), n_i is the number of samples taken for alternative i and N_t is the total simulation budget.

When the objective is multiple and conflicting, the R&S problem becomes MORS problem where more complex performance measures will be employed, which is explained in Section 2.3.1. This research focuses on the MORS problem and a more detailed formulation of the MORS problem is in Section 3.1.

2.2 Major R&S methods

Sampling each alternative an equal number of times is inefficient since it will waste a lot of simulation runs on the obviously inferior alternatives. In recent years, a number of R&S techniques have been developed to improve the efficiency of sampling. The state-of-the-art procedures allocate the sampling budget sequentially, based on observations made so far. There are two categories of statistical models for R&S, frequentist and Bayesian. Frequentist models construct estimates based purely on the observed simulation output. This view generally assumes that there is some unknown but fixed underlying parameter for a population. In contrast, the Bayesian approach assumes prior knowledge about the performance of each alternative and regards the unknown performance as a random variable whose distribution encodes our own uncertainty about the exact value [Chau et al., 2014]. Currently, there are five main basic approaches to R&S summarised in Table 2.1.

- The indifference zone methods such as KN^{++} [Kim and Nelson, 2006] which aim at identifying an alternative that is not worse by more than δ compared to the true best. KN^{++} maintains a set of possibly best solutions and drops solutions from this set when it detects clear evidence that an alternative is unlikely to be best. The procedure iterates until only one solution remains.
- The expected value of information (EVI) procedure [Chick and Inoue, 2001] which maximises the expected value of information in the next samples.
- The small sample EVI procedures that include the Knowledge Gradient (KG) method [Frazier et al., 2008] and the myopic method proposed by Chick et al. [2010]. In each iteration, these methods only allocate samples to one alternative.
- The optimal computing budget allocation (OCBA) [Chen, 1996] approach that maximises the overall simulation efficiency for finding an optimal decision. Different from the small sample EVI procedure, OCBA is an asymptotic approach. For a comprehensive introduction of OCBA method, see the work of Fu et al. [2008], Chen and Lee [2010] and Fu et al. [2007].

Table 2.1: Five main basic approaches to R&S and some exampling references

		Objectives		
		PCS	EOC	PGS
OCBA	Frequentist	Chen and Lee [2010]	-	-
	Bayesian	Chen and Lee [2010]	He et al. [2007]	Branke et al. [2005]
Indifference zone	Frequentist	-	Chick and Wu [2005]	Lee and Nelson [2015] Kim and Nelson [2006]
	Bayesian	Frazier [2014]	-	-
Small EVI	Bayesian	Chick et al. [2010]	Frazier et al. [2008] Ryzhov et al. [2012] Chick et al. [2010]	-
Racing	Bayesian	Birattari et al. [2010]	-	-
EVI	Bayesian	Chick and Inoue [2001]	Chick and Inoue [2001]	-

- The racing method such as F-race that is based on the non-parametric Friedman's two-way analysis of variance by ranks [Birattari et al., 2010]. Similar to KN^{++}, racing methods drop alternatives from sampling that are unlikely to be the best based on the observations so far, until only one alternative remains. However, racing methods have no performance guarantee.

As summarised by Chau et al. [2014], the indifference zone method is generally from a frequentist view although Frazier [2014] proposed a Bayesian-inspired method to correct the indifference zone method's tendency to over-deliver, i.e. ,produce better performance than what is actually required at the expense of many more samples. EVI is a Bayesian statistical model based approach and OCBA can be adapted to both frequentist and Bayesian models [Chen and Lee, 2010]. A comparison of the performance of indifference-zone, EVI and OCBA methods can be found in the work of Branke et al. [2007].

2.2.1 Other R&S methods

In addition to methods mentioned above which aim to select a single best solution, there exist R&S techniques trying to identify a best subset of alternatives. Koenig and Law [1985] developed a two-stage procedure for selecting the top m designs with best performance, Chen et al. [2008] developed an OCBA based algorithm to find m best designs and Yan et al. [2010] proposed the OCBA-bSG algorithm to identify a best subset of m simplest and good enough designs. In many settings this goal is associated with other goals. For example, Chingcuanco and Osorio [2013] developed a procedure that selects the best m out of k stochastic systems with an EOC goal.

2.3 Overview of multi-objective ranking and selection

In this section, we review the main methods to MORS problems from different perspectives.

2.3.1 MORS performance measures

In the presence of multiple, conflicting objectives, it is difficult to decide which alternative is best. Thus, in a Multi-Objective Ranking and Selection (MORS) problem, the objective is usually to find the Pareto optimal set, that is the set of alternatives for which no other feasible alternative is better on all objectives. The image of the Pareto optimal set in objective space is often called the Pareto front. For a minimisation problem, a solution y is called *dominated* by another solution x (denoted by $x \prec y$), if $\mu_{x,h} \leqslant \mu_{y,h}$ for all objectives and $\mu_{x,h} < \mu_{y,h}$ for at least one objective h.

Similar to the single objective R&S problem, one of the most widely used goals is PCS, which is defined as correctly identifying the entire set, and only this set, of Pareto optimal solutions (see also Section 2.3.2 for details). It is not entirely obvious how to define an indifference zone for multiple objectives, but one attempt has been made by Teng et al. [2010] which for a minimisation problem defines a solution x to be non-dominated if $\nexists y | \mu_{y,h} \leq \mu_{x,h} + \delta_h \forall h \wedge \exists h : \mu_{y,h} < \mu_{x,h} + \delta_h$ and PGS is then the probability to identify all the solutions that are non-dominated according to this definition. In Section 4 we will discuss the drawbacks of this definition and propose an alternative. Lee et al. [2007] define the opportunity cost (OC) in a multi-objective setting as follows. For a truly dominated solution that is wrongly classified as non-dominated, the OC is defined as the minimum amount this solution would need to improve in each objective for it to become non-dominated. Correspondingly, for a truly non dominated solution that is classified as dominated, the OC is the minimum amount this solution would need to deteriorate in each objective to become dominated.

Outside R&S such as in multiobjective optimisation or multiobjective reinforcement learning, hypervolume is often used as performance measure. Hypervolume is the area dominated by a set of solutions and bounded by a user-defined reference point. Zitzler and Thiele [1999] present hypervolume as the only quality indicator known to be fully compliant to Pareto dominance, i.e.,whenever a set A dominates another set B (an objective vector z_1 is not worse than objective vector z_2 in all objectives and better in at least one objective and every $z_2 \in B$ is dominated by at least one $z_1 \in A$), then the measure yields a strictly better quality value for the former [Zitzler et al., 2003]. For a comprehensive literature review of the hypervolume measurement see the work of Bader and Zitzler [2011]. We have proposed to use hypervolume difference in the context of R&S [Branke et al., 2016], which will be discussed in more detail in Section 5.

2.3.2 MORS methods

Compared with the single objective R&S problem, the literature on MORS is relatively limited.

One of the most widely used approaches is converting performance over multiple objectives into a scalar measure using costs or multiple attribute utility theory (MAUT). By combining with an indifference zone R&S method, Morrice et al. [1998] provide a MAUT approach to MORS. Butler et al. [2001] show applications for the procedure and conduct sensitivity analysis for the weights via Monte Carlo simulation. Morrice and Butler [2006] have also extended the approach to model constraints using value functions. The MAUT approach transforms the problem into a single-objective one and cannot fully characterise the trade-off among the multiple performance measures. Although Butler et al. [2001] use a mechanism to assess the relative importance of each criterion, an accurate model of the DM's preferences is difficult to construct in practice.

Instead of using a single utility function, Branke and Gamer [2007] use a distribution of utility functions, and aim to minimise the expected opportunity cost over this distribution of utility functions using a variant of OCBA [He et al., 2007]. Frazier and Kazachkov [2011] developed a similar procedure based on the KG policy.

Most MORS procedures aim at maximising the probability of exactly identifying the set of Pareto optimal solutions. Examples include the multi-objective optimal computing budget allocation (MOBCA) proposed by Lee et al. [2010b], which is a multi-objective version of the OCBA algorithm. MOCBA has also been extended to allow for other measures of selection quality such as EOC [Lee et al., 2007, 2010a], and PGS [Teng et al., 2010]. The approach by Hunter and Feldman [2015] and Feldman et al. [2015] allocates samples to maximise the rate of decay of the probability that a misclassification event occurs. It is asymptotically optimal, and can take into account correlation between objectives. The myopic M-MOBA [Branke and Zhang, 2015] has been derived from the Small EVI paradigm [Chick et al., 2010]. Mattila and Virtanen [2015] combined the multiple attribute utility (MAU) method and Pareto domination idea. In the proposed procedure, the objectives are aggregated with a MAU function and incomplete preference information regarding the weights that reflect the relative importance of the objectives.

There are two procedures, MOCBA-p which identifies the non-dominated designs according to pairwise dominance and OCBA-a which identifies the designs that are non-dominated according to absolute dominance [Mattila and Virtanen, 2015]. The dominance relationship is defined by the estimation of the expected utilities over feasible weights [Mattila and Virtanen, 2015]. The advantage of this combination is that it doesn't require strict preference statements and has reduced computational burden as with MOCBA-p, a smaller set of designs is obtained since any pairwise non-dominated design is also absolutely non-dominated and OCBA-a, on the other hand, is a somewhat simpler procedure and more straightforward to implement.

Another possibility of solving MORS is to regard one performance measure as primary objective and the rest as stochastic constraints. The general aim is then to efficiently identify the system having the best objective function value from among those systems whose constraint values are above a specified threshold [Hunter and Pasupathy, 2013]. Research in this category includes the work by Andradottir and Kim [2010], in which they provide indifference-zone frameworks with statistical performance guarantee consisting of two phases: identification and removal of infeasible systems, and removal of systems whose primary performance measure is dominated by that of other feasible systems. These phases can be executed sequentially or simultaneously. Park and Kim [2011] propose a penalty function with memory which determines a penalty value for a solution based on the history of feasibility checks on the solution and converts the problem into a series of new optimisation problems without stochastic constraints. Hunter and Pasupathy [2013] present the first complete characterisation of the optimal sampling plan relying on the large deviation framework, a consistent estimator for the optimal allocation and a corresponding sequential algorithm. Pujowidianto et al. [2012] and Pasupathy et al. [2014] focus on asymptotic theory in the context of stochastically constrained simulation optimisation problems on large finite (many thousands) sets of alternatives and provide a sampling framework called SCORE (Sampling Criteria for Optimisation using Rate Estimators) that approximates the optimal simulation budget allocation.

Finally, Zhang et al. [2013] present a multi-objective S-Race algorithm which attempts to eliminate alternatives as soon as there is sufficient statistical evidence of them being dominated (worse in all objectives compared to another solution). Experimental results suggest that S-Race typically offers cogent computational advantages in exchange for relatively small discrepancies in the set of Pareto-optimal alternatives identified. However, S-Race has limitations including Type II errors not being strictly controlled, unnecessary computational cost on comparing non-dominated models and the sign test employed not being an optimal test procedure.

Zhang et al. [2015] overcome these limitations by introducing a multi-objective racing algorithm based on the Sequential Probability Ratio Test (SPRT) with an Indifference Zone that is able to reduce the possibility of misclassifying Pareto optimal alternatives by chance and also reduce the computational cost.

As it is not easily to clarify current MORS algorithms into categories like Table 2.1 due to mix methods applied within each algorithm, we categorise the literature of MORS methods based on three aspects: DM's preference, stopping rule, and whether dropping it is permanently alternatives during the selection procedure. A posteriori MORS methods generate an entire Pareto set and then allow the DM to express a preference after the optimisation is conducted, instead of before or during while a priori MORS methods incorporate the decision-maker's preferences before the optimisation is conducted [Miettinen, 2012]. Stopping rules include the total fixed budget (TFB) with which the algorithm will stop as soon as the predefined budget is exhausted; the flexible budget (FB) and the flexible budget with guarantee (FBG). The third is whether dropping alternatives (DA) during selection.Table 2.2 summarises the literature and some papers are placed in the same row if they are quite similar. For papers without numerical experiments, does not mention stopping rule clearly or using other rules, we classify them as "other stopping rule" (OSR).

2.4 Other related methods

Other related methods include evolutionary algorithm (EA) based techniques that vary the number of samples used per solution based on the amount of noise in combination with a user-defined confidence level [Syberfeldt et al., 2010]. Cervantes et al. [2016] propose a quite similar resampling technique using Welch test to decide the domination relationship. A trust-region based method for approximating the Pareto front of a bi-objective stochastic optimisation problem has been proposed by Kim and Ryu [2011]. Marceau-Caron and Schoenauer [2014] propose to directly estimate the probability of each individual to survive to the next generation, and use Hoeffding races to drop alternatives early; a novel multi-objective optimisation approach proposed for hybrid renewable energy system which combines the uses of ε-constraint method, i.e., a technique that can be used where one objective is chosen to be optimised and the remaining objectives are considered as constraints bound by given target levels, and a particle swarm optimisation based approach [Sharafi and ELMekkawy, 2014].

Cervantes et al. [2016] integrate dominance based statistical testing methods as part of the selection mechanism of evolutionary multi-objective genetic algorithms

Table 2.2: MORS algorithms

Paper	A posteriori	A priori	FTB	FB	FBG	DA	OSR
Dudewicz and Taneja [1978, 1981]		✓		✓			
Morrice et al. [1998]; Butler et al. [2001]		✓			✓		
Morrice and Butler [2006]		✓			✓		
Branke and Gamer [2007][a]	✓			✓			
Frazier and Kazachkov [2011][a]	✓		✓				
Lee et al. [2007, 2010a]	✓		✓				
Lee et al. [2010b]	✓		✓				
Teng et al. [2010]	✓		✓				
Hunter and Feldman [2015][a]	✓						✓
Feldman et al. [2015][a]	✓						✓
Branke and Zhang [2015]	✓		✓				
Branke et al. [2016]	✓		✓				
Mattila and Virtanen [2015]	✓	✓					
Andradottir and Kim [2010]		✓			✓		
Park and Kim [2011]		✓					✓
Pujowidianto et al. [2012]		✓					✓
Hunter and Pasupathy [2013]		✓		✓			
Pasupathy et al. [2014]		✓					✓
Zhang et al. [2013]	✓				✓	✓	
Zhang et al. [2015]	✓				✓	✓	

[a] These methods use a set of utility functions for choosing the Pareto optimal solutions.

and reduce the number of fitness evaluations. In addition, a recent trend in the literature is to combine different approaches in the algorithm design. For example, Mattila and Virtanen [2015] combined the MAU method and Pareto domination idea. In the newly proposed procedure, the performance measures are aggregated with a MAU function and incomplete preference information regarding the weights that reflect the relative importance of the measures. There are two procedures, MOCBA-p which identifies the non-dominated designs according to pairwise dominance and OCBA-a which identifies the designs that are non-dominated according to absolute dominance Mattila and Virtanen [2015]. The dominance relationship is defined by the estimation of the expected utilities over feasible weights Mattila and Virtanen [2015]. The advantage of this combination is that it doesn't require strict preference statements and has deduced computational burden

2.5 Brief introduction to multi-objective evolutionary algorithms

Many real-world problems can be formulated as optimisation problems and these problems often include two or more objectives to be optimised, and are normally called multi-objective optimisation problems (MOPs) [Fan et al., 2016]. Similar to MORS problem, objectives are often conflict with each other. Improvement of one objective may lead to deterioration of another. Thus, a single solution, which can optimise all objectives simultaneously, does not exist and the best trade-off solutions, namely the Pareto optimal solutions, are important to a decision maker [Zhou et al., 2011]. Evolutionary algorithms (EAs), as population-based search methods, are believed to be well suited for solving MOPs in that they can achieve a set of non-dominated solutions in one run [Wang et al., 2017]. There has been a growing interest in applying EAs in the past more than 20 years to deal with MOPs and these EAs are called multiobjective evolutionary algorithms (MOEAs)[Zhou et al., 2011]. Generally speaking, existing MOEAs can be divided into three categories according to their selection criteria, namely Pareto-, indicator-, and reference-based MOEAs [Wang et al., 2017].

1. Pareto-based MOEAs use the Pareto dominance as their main selection methodology for convergence [Wang et al., 2017]. One of the famous typical methods in this category is non-dominated sorting genetic algorithm II (NSGA-II). a majority of MOEAs in both the research and the application areas share more or less the same framework as that of NSGA-II: a selection operator based

on Pareto domination and a reproduction operator are used iteratively [Zhou et al., 2011]. However, Praditwong and Yao [2007] show that Pareto-based MOEAs fail to solve many objective optimisation problems (MaOPs) that are defined to be MOPs with more than three objectives, mainly due to the dominance comparison becomes less effective when the number of objectives increases for a limited population size [Ishibuchi et al., 2008]. Other typical methods include SPEA-II [Zitzler et al., 2001]and PAES-II [Corne et al., 2001].

2. Indicator-based MOEAs use an indicator as the selection criterion to replace the Pareto dominance in Pareto-based MOEAs. Performance metrics includes IGD [Bosman and Thierens, 2003]and HV [Zitzler and Thiele, 1999]. Fan et al. [2016] summarise that representative methods in this category include IBEA [Zitzler and Künzli, 2004], R2IBEA [Phan and Suzuki, 2013], SMS- EMOA [Beume et al., 2007] and HypE [Bader and Zitzler, 2011].

3. Reference-based MOEAs decompose an MOP into a set of sub-problems according to the predefined references, such as weights Zhang and Li [2007], reference points [Deb and Jain, 2014], reference vectors [Cheng et al., 2016], and direction vectors [Jiao et al., 2013], [Liu et al., 2014]. Different aggregation functions have been suggested to convert an MOP into a set of single-objective optimisation problems, including weighted sum, Tchebycheff approach, and penalty-based boundary intersection (PBI) approach [Fan et al., 2016]. Among methods in this category, MOEA/D is very popular and MOEA/D variants have been proposed such as the work of Liu et al. [2014], Li and Zhang [2009], Li et al. [2014], Cai et al. [2015], etc.. For a comprehensive review for many-objective evolutionary algorithms (MaOEAs) for MaOPs, see the work of Li et al. [2015].

Since MORS techniques will avoid wasting simulation samples on irrelevant alternatives, the performance of MOEA will be improved by combing it with MORS algorithms. There are few works in this area and one study is the work of Lee et al. [2008] in which MOCBA is integrated into the evaluation procedure to reduce the computing cost for an aircraft spare parts allocation problem. Syberfeldt et al. [2010] proposed a noise-handling method by using an iterative resampling procedure that reduces the noise until the likelihood of selecting the correct solution reaches a given confidence level. This technique is able to prevent the propagation of inferior solutions in the selection process due to noisy objective values. Different from other methods, the proposed technique varies the number of samples used per solution based on the amount of noise in the local area of the search space [Zhou et al., 2011]. In this way, the algorithm avoids wasteful samplings when the benefit of additional samplings is insignificant. A Dynamic Resampling strategy is proposed which identifies the solutions closest to the reference point which guides the population of the Evolutionary Algorithm [Siegmund et al., 2016]. Experiment results show that distance-based D-OCBA-m will support the preference-based multi-objective Evolutionary Algorithm R-NSGA-II for problems with variable noise when the Dynamic Best Sampling variant is used.

2.6 Summary

In summary, most R&S studies focus on problems with a single objective measure, or a single objective measure with one or more constraint measures [Branke et al., 2007], [Kim and Nelson, 2007]. Research for problems with multiple objectives are still in the infant stage. At the same time, the majority research in MORS try to transform the multi-objective problem into single objective problem, which, however are not justified to provide desired solutions [Deb, 2001]. Furthermore, existing methods that are able to find the Pareto optimal solutions are mostly from asymptotic view where some approximations (e.g., Bonferroni's or Slepian's inequality) are not avoidable. Therefore, the exploration of solving the MORS problem without approximation is necessary and valuable. This research shows how to tackle the MORS problem with different performance measures, especially the bi-objective problem, from a small EVI view in the following chapters. We also show how to integrate the developed MORS method to improve the efficiency of MOEA.

Chapter 3

M-MOBA PCS procedure and independent sampling

Based on the small-sample EVI procedure derived by Chick et al. [2010] and Frazier et al. [2008], we proposed a simple but efficient myopic multi-objective budget allocation (M-MOBA) algorithm for MORS problems [Branke and Zhang, 2015]. By being myopic and only allocating a few additional samples to one alternative, small sample procedures can avoid various asymptotic approximations. More specifically, in each iteration of sample allocation, we only allocate samples to the alternative that is expected to provide the maximum value of information.

3.1 Introduction

In this study, we will discuss a small sample ranking and selection procedure for the multi-objective case. The basic idea of the small sample or myopic approach is what is the expected impact of one additional sample for one system on the performance difference before/after, as in a Bayesian setting we will assume the information after the additional sample will be the correct one. We consider the problem of efficiently identifying the Pareto optimal designs out of a given set of alternatives, for the case where alternatives are evaluated on multiple stochastic criteria, in other words, the question is how to allocate budget efficiently to each alternative i optimise objective (PCS, EOC, etc.) given the total simulation budget N_t. Throughout this research, the allocation rules are explained by assuming that there are two objectives for each alternative so that the Pareto set and the dominance relationship can be visualised in a two-dimensional coordinate system. However, extending the basic ideas to more than two objectives should be possible. The problem of MORS can be formulated as follows.

Consider H objectives and a set of m designs with the true unknown performance of each design i in objective h being denoted by $\mu_{i,h}$. Assuming minimisation throughout this research, a design i is said to dominate design j ($i \prec j$) if and only if $\mu_{i,h} \leq \mu_{j,h}$ for all objectives and $\mu_{i,h} < \mu_{j,h}$ for at least one objective. A design that is not dominated by any other design is called Pareto optimal.

The performance of each design in each objective needs to be estimated via sampling. Vectors are written in boldface, e.g., $\mathbf{X}_i$=(X_{ihn}) is a matrix that contains the simulation output for design i, objective h and simulation replication n. Let furthermore $\mu_{i,h}$ and $\sigma^2_{i,h}$ be the unknown (true) mean and variance of alternative i, which can only be estimated using the simulation outputs X_{ihn}. We assume that

$$\{X_{ihn} : n = 1, 2, ...\} \overset{iid}{\sim} \mathcal{N}(\mu_{i,h}, \sigma^2_{i,h}), \text{for } i = 1, 2, ..., m \text{ and } h = 1, 2, ...H.$$

Similar to the work of Chick et al. [2010], we will first describe random variables with Student t distributions. If T_ν is a random variable with standard t distribution with ν degrees of freedom, we denote the distribution of $\mu + \frac{1}{\sqrt{\kappa}} T_\nu$ by $St(\mu, \kappa, \nu)$ [Bernardo and Smith, 1994]. If $\nu > 2$, then the variance is $\nu/(\nu - 2)\kappa$. As $\kappa \to \infty$, $St(\mu, \kappa, \nu)$ converges in distribution to a point mass at μ. The cumulative distribution function (cdf) of the standard t distribution is denoted by $\Phi()$ and the probability density function (pdf) by $\phi()$.

Let n_i be the number of samples taken for alternative i so far, $\bar{x}_{i,h}$ the sample mean and $\hat{\sigma}^2_{i,h}$ the sample variance. Then, we will get an observed Pareto set based on the $N = \sum_i n_i$ simulations. As n_i increases, $\bar{x}_{i,h}$ and $\hat{\sigma}^2_{i,h}$ will be updated and the observed Pareto front may change accordingly. If alternative i is to receive another τ_i samples, let $\mathbf{Y}_i$=(Y_{ihn}) denote the data to be collected in the next stage of sampling, $\mathbf{y}_i$=(y_{ihn}) be the realisation of $\mathbf{Y}_i$ and $\bar{y}_{i,h}$ the average of the new samples in objective h, then the new overall sample mean in each objective can be calculated as

$$\bar{z}_{i,h} = \frac{n_i \bar{x}_{i,h} + \tau_i \bar{y}_{i,h}}{n_i + \tau_i}. \tag{3.1}$$

Before the new samples are observed, the sample average that will arise after sampling, denoted as $Z_{i,h}$, is a random variable, and we can use the predictive distribution for the new samples and get [DeGroot, 2005]

$$Z_{i,h} \sim St(\bar{x}_{i,h}, n_i * (n_i + \tau_i)/(\tau_i * \hat{\sigma}^2_{i,h}), n_i - 1)$$

.

A correct selection occurs when the selected set of alternatives, $\mathbf{S}(\mathbf{Y})$, is the true Pareto set $\mathbf{P}$. The objective of a selection procedure can be defined in terms of a loss function. Let $\mathbf{W} = (\mu_{ihn})$. Similar to Chick et al. [2010], we define a zero-one loss function $L_{0-1}(\mathbf{S}(\mathbf{Y}), \mathbf{W}) = \mathbb{1}\{\mathbf{S}(\mathbf{Y}) \neq \mathbf{P}\}$, where the indicator function $\mathbb{1}\{\cdot\}$ equals 1 if its argument is true, and is 0 otherwise. We can now define the problem of allocating $\tau > 0$ samples to m alternatives in order to maximise the EVI of a single additional stage of sampling:

$$\begin{aligned} \min_{\tau_1,\tau_2,\tau_3\ldots\tau_m \geqslant 0} \quad & E[L_{0-1}(\mathbf{S}(\mathbf{Y}), \mathbf{W})] \\ \text{such that} \quad & \tau = \sum_{i=1}^{m} \tau_i \end{aligned} \tag{3.2}$$

Minimising the predicted expected loss in Equation(3.2) is equivalent to maximising the EVI.

Since there are different performance measures and stopping rules for MORS, we will define the MORS problem mathematically as follows by using PCS and total budget stopping rule, i.e., we will stop when the predefined budget N_t simulations was been used up. A common method of defining the probability of correct selection (PCS) is the probability that the observed Pareto set $\mathbf{S}(\mathbf{Y})$ is equal to the true Pareto set $\mathbf{P}$. Therefore, PCS can be defined as:

$$PCS = P(\mathbf{S}(\mathbf{Y}) = \mathbf{P})$$

The MORS problem is then given the known means and variances and the total budget N_t, determine the optimal allocation of the replications to the designs such that the PCS is maximised

$$\begin{aligned} \underset{n_i}{\text{maximise}} \quad & PCS \\ \text{subject to} \quad & \sum_{i=1}^{m} n_i \leq N_t. \end{aligned}$$

3.2 Problem statement

We will first consider the problem with PCS measurement, where a selection is defined as correct if and only if the selected set of alternatives is identical to the true Pareto set. M-MOBA, in each iteration, will only allocate one sample to one alternative – the alternative that has the highest probability of changing the observed

Pareto set. The advantage of this method is that we will not waste the precious simulation runs on irrelevant solutions for which we are already sufficiently confident that they are Pareto optimal or not. We are more interested in solutions that with more simulations, are likely to change our decision. This algorithm has first been proposed by Branke and Zhang [2015] and serves as basis of all other extended versions we will present later.

Assume that after an initial n_0 samples for each alternative, the current Pareto set consists of a set of alternatives a_i, where $i = 1, 2, ..., k_1$. We will consider each alternative a_c in turn and estimate the expected value of information, i.e., the probability that the Pareto set will change if one additional sample is allocated to a_c. If the particular alternative under consideration is removed, some previously dominated alternatives may become Pareto optimal, denoted by b_j, with $j = 1, 2, ..., k_2$. We further denote the newly formed Pareto set when the particular alternative under consideration is removed as p_r, with $r = 1, 2, ..., k_3$. For each alternative a_i, there are three possible situations and each of them will be explained below.

The first situation is depicted in Figure 3.1, where a_c is on the observed Pareto set composed of points a_1, a_c, a_2, a_3 and indicated by the dashed line. Alternatives a_1 and b_1 are the nearest neighbours of a_c in the direction of objective f_1 and alternatives b_3 and a_2 are the nearest neighbours of a_c in the direction of objective f_2. We want to calculate the probability that the current Pareto set will change if we allocate τ additional simulation samples to a_c. If we only allocate samples to a_c, all other alternatives can be considered deterministic in the immediate one-step look-ahead. Then, the Pareto set changes if and only if the new mean estimate for alternative a_c after sampling

1. dominates one of the previously non-dominated solutions (a_1, a_2, a_3 in Figure 3.2)
2. becomes dominated itself, or
3. exposes a previously dominated solution (b_1, b_2, b_3 in Figure 3.2).

In the example in Figure 3.2, a change happens if the new mean estimate falls outside the shaded area.

Since we assume that the samples in the two objectives are independent, we can calculate the probability for a_c to remain in the shaded area separately for each objective, and multiply them to get the probability P that the new mean estimate for a_c remains in the shaded area, and $1 - P$ is the probability that with one additional sample, a_c will move out of the area and hence a new observed Pareto

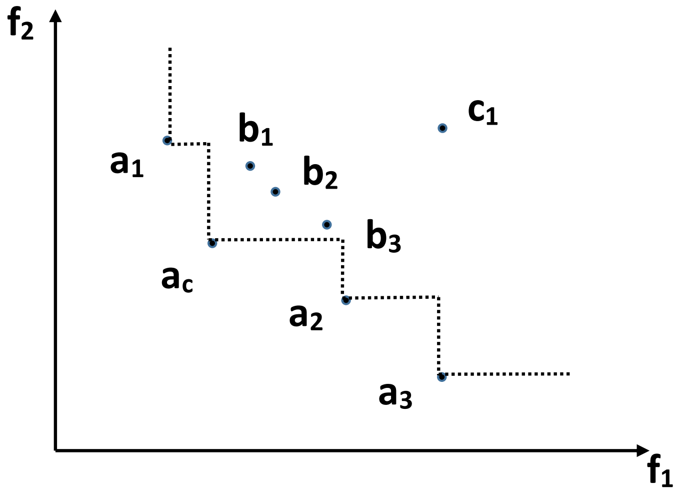

Figure 3.1: a_c solely dominates another alternative

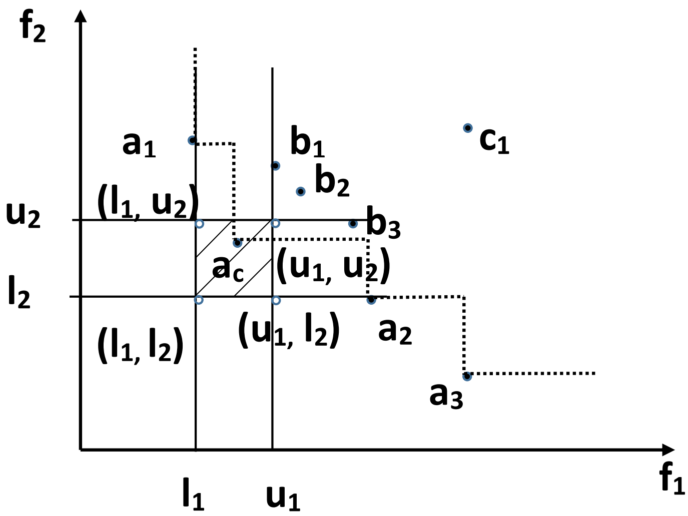

Figure 3.2: The Pareto set will change if and only if the estimated mean of alternative a_c will fall outside the shaded area

front will be obtained. Let us assume that the two objectives f_1 and f_2 values of nearest neighbours of a_c are (l_1, u_1) and (l_2, u_2), i.e.,

$$l_1 = max\{\bar{x}_{p_r,1} < \bar{x}_{a_c,1} | r = 1, 2, ..., k_3\}$$

$$l_2 = max\{\bar{x}_{p_r,2} < \bar{x}_{a_c,2} | r = 1, 2, ..., k_3\}$$

$$u_1 = min\{\bar{x}_{p_r,1} > \bar{x}_{a_c,1} | r = 1, 2, ..., k_3\}$$

$$u_2 = min\{\bar{x}_{p_r,2} > \bar{x}_{a_c,2} | r = 1, 2, ..., k_3\}$$

then the probability P is

$$\int_{l_2}^{u_2} \int_{l_1}^{u_1} \phi_{a_{c,1}}(x) \cdot \phi_{a_{c,2}}(y) dx dy \quad (3.3)$$

where $\phi_{a_{c,h}}$ is the predictive probability distribution of the new location of a_c in dimension h.

If a_c doesn't expose any new solutions when it is removed, then the Pareto set will only change if the new estimated mean will become dominated, or dominates a previously non-dominated alternative. Figure 3.3 shows an example, with the area in which a_c may fall without causing a change highlighted.

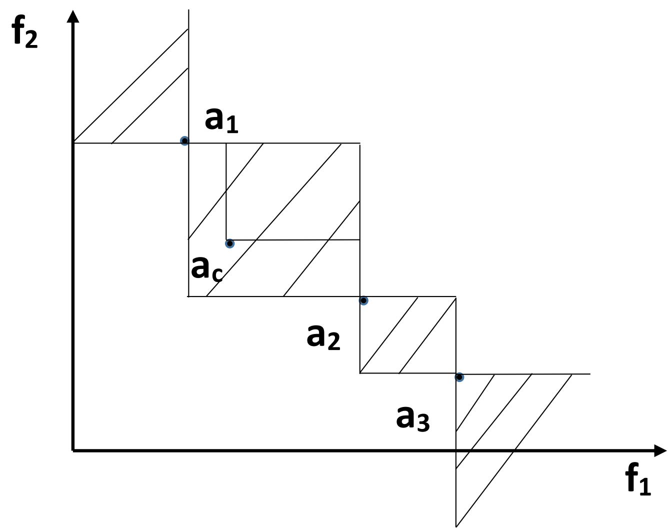

Figure 3.3: The Pareto set will change if and only if the estimated mean of alternative a_c will fall outside the shaded area

Assume there are k Pareto optimal alternatives after a_c has been removed and they are sorted from small to large based on f_1, with an additional virtual

0-th solution at $(-\infty, \infty)$ and a virtual $(k+1)$-th solution at $(\infty, -\infty)$, then the probability P can be calculated as

$$\sum_{i=0}^{k} \int_{a_{i+1,2}}^{a_{i,2}} \int_{a_{i,1}}^{a_{i+1,1}} \phi_{a_{c,1}}(x) \cdot \phi_{a_{c,2}}(y) dx dy \tag{3.4}$$

where alternative i with objective values $(a_{i,1}, a_{i,2})$ is Pareto optimal if a_c is removed.

When a_c is not in the Pareto set, a change happens if and only if a_c becomes non-dominated. An example can be seen in Figure 3.4.

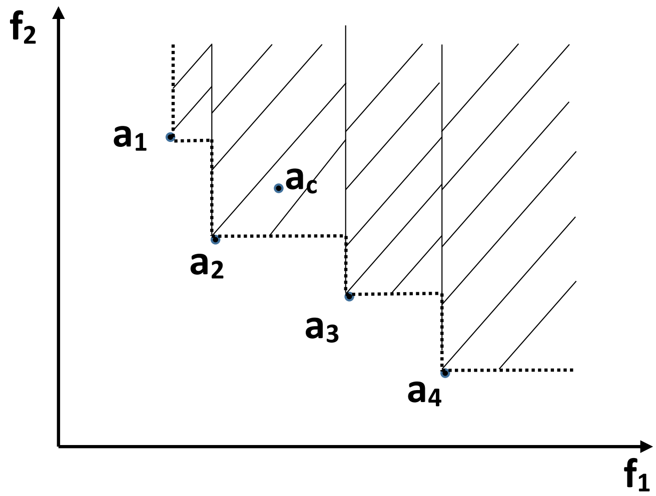

Figure 3.4: The Pareto set will change if and only if the estimated mean of alternative a_c will fall outside the shaded area when a_c is not in the Pareto set

In this scenario, the shaded area is defined by all current Pareto optimal alternatives. Similar to the above scenario, if there are k Pareto optimal alternatives, the probability P can be computed as

$$\sum_{i=1}^{k} \int_{a_{i,2}}^{\infty} \int_{a_{i,1}}^{a_{i+1,1}} \phi_{a_{c,1}}(x) \cdot \phi_{a_{c,2}}(y) dx dy \tag{3.5}$$

where alternative i is Pareto optimal and $a_{k+1,1} = \infty$.

Sometimes, objectives can be evaluated independently, e.g., if different simulation models are used to evaluate different objectives. For example, in order to test the performance of vehicle, we can test their fuel consumption at various loads, braking distance, top speed, etc.. In this case, in order to further improve the efficiency of sampling, it is possible to regard the sampling allocation process for each objective independently. This independent sampling procedure can be employed with different measures and without loss of generality we use PCS here. Instead of evaluating all objectives of an alternative simultaneously as in the M-MOBA PCS procedure, we will evaluate only one objective of one alternative in each iteration. We calculate P_i using the same methods as in M-MOBA PCS, and allocate the simulation sample to the solution and objective that has the biggest probability to change the current Pareto front. For comparison purposes, for a two objective problem, we assume the M-MOBA PCS procedure will allocate one sample for each objective of a solution in every iteration, while the M-MOBA Differential Sampling PCS (M-MOBA DS PCS) procedure will only allocate one sample to the selected objective. We suggest that by allowing to evaluate objectives independently, the efficiency of the algorithm may be improved substantially. This would be even more the case if evaluating different objectives would take different time or involve different cost, because it would allow the algorithm to focus on the cheaper objectives. Different costs could be easily integrated into M-MOBA DS by using the quotient of probability of change and computational cost to decide which solution and objective to evaluate next. In order to test the effect of using different cost, for M-MOBA DS PCS, we calculate R_h as

$$R_h = \frac{P_h}{c_h} \tag{3.6}$$

where P_h is the probability of change if one more simulation replication is allocated for each solution on objective h and c_h is the cost of simulating this objective and simulate the corresponding objective h that has the maximum R_h.

3.3 M-MOBA PCS Algorithm

Based on the above analysis, we can formulate the small-sample multi-objective budget allocation procedure as summarised in Algorithm 1. Generally, the first-stage sample size n_0 should be larger than 2 to get sufficiently accurate estimate of standard deviation. We will discuss how to choose the value of n_0 in detail in the following section. Through this research, the stopping rule adopted is fixed budget, i.e., the algorithm will stop as soon as the predefined budget is exhausted. However, different stopping rules (as discussed before) can be easily employed.

ALGORITHM 1: Procedure M-MOBA PCS

1: Specify a first-stage sample size n_0, and a number of samples $\tau = 1$ to allocate per subsequent stage. Specify stopping rule parameters
2: Sample $X_{ihn}, i = 1, \ldots, m; h = 1, \ldots, H; n = 1, \ldots, n_0$ independently, and initialise the number of samples $n_i \leftarrow n_0$
3: Determine the sample statistics $\bar{x}_{i,h}$ and $\hat{\sigma}^2_{i,h}$, and the observed Pareto front
4: **while** stopping rule not satisfied **do**
5:     For each alternative i, calculate the probability P_i that the new samples will lead to a change in the Pareto set by Equation 3.3, 3.4 and 3.5 respectively according to the different situation of a_c
6:     Allocate τ samples to the alternative that has the largest P_i
7:     Update sample statistics n_i, $\bar{x}_{i,h}$ and $\hat{\sigma}^2_{i,h}$ and observe a new Pareto front
8: **end while**
9: Select alternatives on the observed Pareto front

3.4 Numerical experiments

Throughout this research, all numerical experiments are implemented on Matlab. Different configurations are constructed by building a set of alternatives using predefined mean and variance value. Random numbers are then generated in each iteration based on every alternative's mean and variance in order to test the performance of algorithms stochastically. All numerical experiments are computed on a machine with 2.4 GHz Intel Core i7 CPU and 8GB memory.

In this section, we compared the performance of M-MOBA PCS with MOCBA [Chen and Lee, 2010] by using two configurations from Chen and Lee [2010] and Equal allocation (which simply allocates an equal number of samples to each alternative). For each method, each design is sampled $n_0 = 5$ times during initialisation, and then additional samples are allocated one at a time (τ=1) until a pre-set budget has been used up. On the one hand, n_0 should be setting as larger than 2 to get sufficiently accurate estimate of standard deviation. On the other hand, if n_0 is too large, the power of the algorithm is not apparently. $n_0 = 5$ is an empirical value to set. Results are averaged over 1000 runs. In case we run into problems of numerical precision, τ is changed to 10 for the expected information change calculation, but still only one sample is allocated. This technique is useful since if we do not use it, the algorithm will keep allocating samples on the first alternative as all alternatives have the 0 probability to change due to the precision limitation of the software. If the numerical precision problem persists, we will use Equal allocation until the problem disappears and τ is then set back to 1. We use the same set up for all other numerical experiments in other chapters.

In an earlier paper [Branke and Zhang, 2015], we compared the performance of M-MOBA PCS with MOCBA [Chen and Lee, 2010] by using two configurations from Chen and Lee [2010]. In Branke and Zhang [2015], as we didn't have access to an implementation of MOCBA at the time, we just compared with results read approximately from figures provided in Chen and Lee [2010]. For this paper, Dr. Haobin Li has kindly provided us with his code of MOCBA, and so we are able to compare MOCBA and M-MOBA directly and under identical settings.

In the first benchmark problem, there are 3 designs and each of them is evaluated according to 2 objectives. Objective values of the designs are shown in Table 3.1.

Table 3.1: True expected performance in each objective, std. is 5.

Index	Obj. 1	Std. dev. 1	Obj. 2	Std. dev. 2
0	1	5	2	5
1	3	5	1	5
2	5	5	5	5

The resulting PCS over the budget allocated is shown in Figure 3.5.

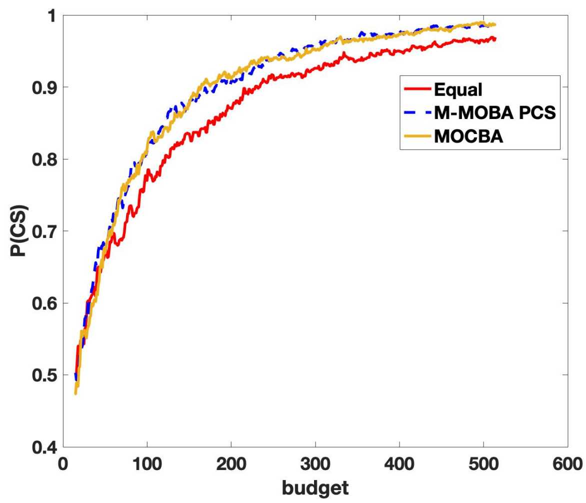

Figure 3.5: Comparison of PCS for different algorithms on the 3-alternative case

The results show that the probability of correct selection using our algorithm is generally higher than Equal allocation with the same simulation budget. Also, P(CS) using the M-MOBA PCS converges faster to 1 than Equal allocation. M-MOBA PCS performs very similar to MOCBA in this situation since the problem tested is very simple and the performance difference between different algorithms is not obvious.

Table 3.2: 16-alternative configuration with two objectives. Standard deviation for all designs is set as 2 in each objective.

Index	Obj. 1	Obj. 2	Index	Obj. 1	Obj. 2
1	0.5	5.5	9	4.8	5.5
2	1.9	4.2	10	5.2	5
3	2.8	3.3	11	5.9	4.1
4	3	3	12	6.3	3.8
5	3.9	2.1	13	6.7	7.2
6	4.3	1.8	14	7	7
7	4.6	1.5	15	7.9	6.1
8	3.8	6.3	16	9	9

In the second configuration, the expected values of each design are shown in Table 3.2 and visualised in Figure 3.6. In this configuration, there are 16 alternative designs, 2 objectives, and the standard deviation of each alternative in each objective is set to 2.

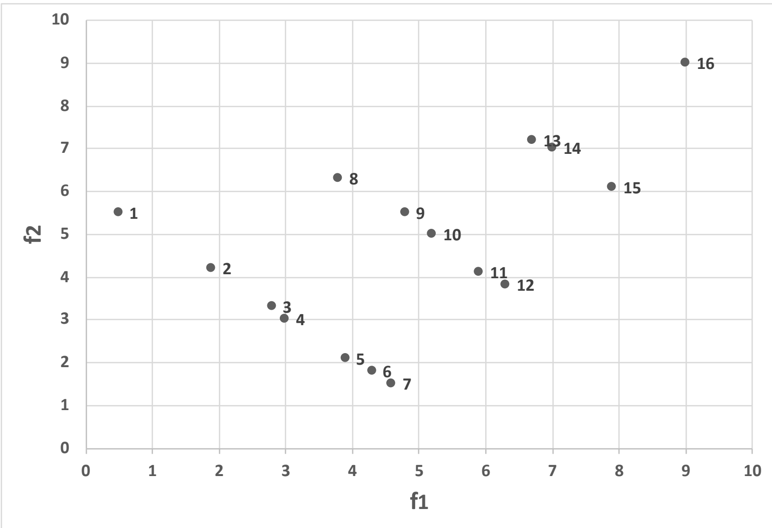

Figure 3.6: Configuration with 16 alternatives

Results are summarised in Figure 3.7. Comparing our algorithm, M-MOBA PCS ($\tau = 1$), MOCBA and Equal allocation, it can be seen that both M-MOBA PCS and MOCBA work much better than Equal allocation and M-MOBA PCS also works better than MOCBA. The difference of performance between the latter two methods reaches a peak when the total simulation budget is around 1600. When the simulation budget continues increasing, the difference between M-MOBA PCS and MOCBA reduces again. The very good performance of M-MOBA PCS for small samples makes sense as M-MOBA PCS has been designed from a myopic perspective, whereas MOCBA is based on asymptotic considerations.

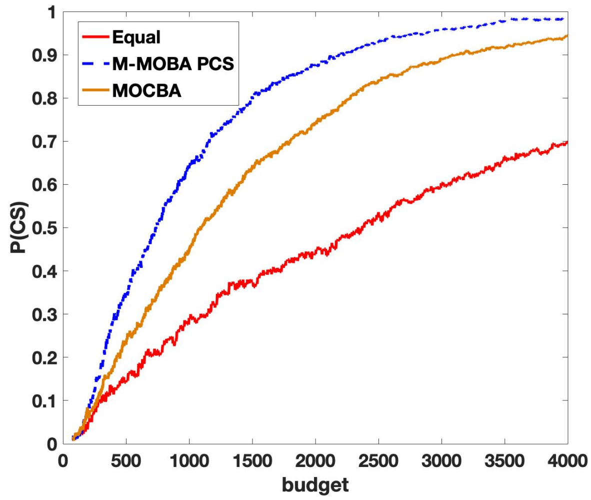

Figure 3.7: Comparison of P(CS) for different algorithms on the 16-alternative case

Still using the 16 alternatives configuration used by Chen and Lee [2010], we test the M-MOBA DS PCS procedure and compare it with the original M-MOBA PCS procedure and Equal allocation.

Figure 3.8 shows P(CS) as the number of samples allocated increases. It can be seen that both M-MOBA PCS and M-MOBA DS PCS perform much better than Equal allocation and M-MOBA DS PCS performs better than M-MOBA PCS throughout the entire run. This matches our expectation because the M-MOBA DS PCS allocates the sampling budget more precisely to the objectives where they provide the highest value of information. This procedure is valuable when the simulation budget is quite limited and the objectives can be evaluated independently.

For testing the effect of cost in the experiment, we simply assume that the cost of simulating objective 1 is 1 and 2 for objective 2. Again, using the 16 alternatives configuration, we test the M-MOBA DS PCS procedure and compare it with the original M-MOBA PCS procedure and Equal allocation.

Figure 3.9 shows how P(CS) increases as the number of samples allocated increases. It can be seen that both M-MOBA DS PCS and M-MOBA PCS perform better than Equal allocation throughout the entire run and M-MOBA DS PCS performs better than M-MOBA PCS.

In order to check how different methods spend the simulation samples, Figure 3.10 and Figure 3.11 show how M-MOBA PCS and M-MOBA DS PCS allocate

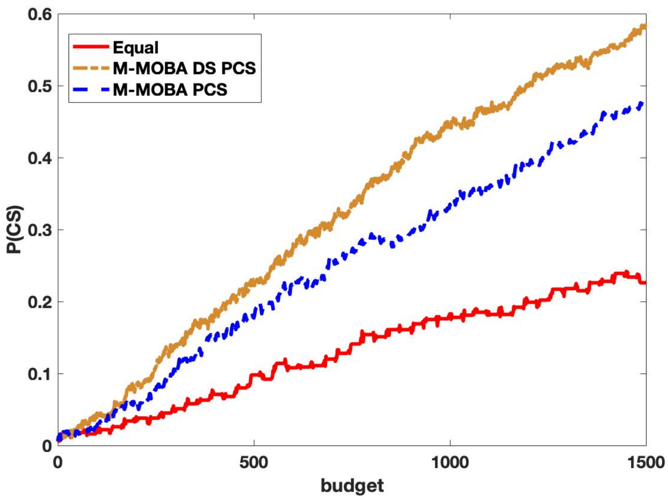

Figure 3.8: Comparison of P(CS) for M-MOBA DS PCS, M-MOBA PCS and Equal allocation on the 16-alternative case

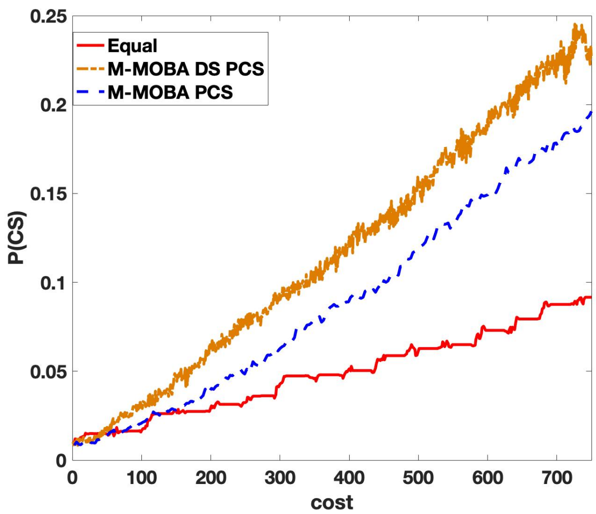

Figure 3.9: M-MOBA DS PCS procedure

the simulation samples respectively. In general, they both focus on alternatives 1˜7, which are true Pareto optimal solutions. The only difference is that M-MOBA DS PCS focuses more on objective 1 due to the lower simulation cost.

To further investigate the effect of cost, we designed an experimental configuration as shown in Figure 3.12. There are 6 alternatives in total. Alternative 1 and alternative 4 are quite similar in objective 2 while alternative 3 and alternative

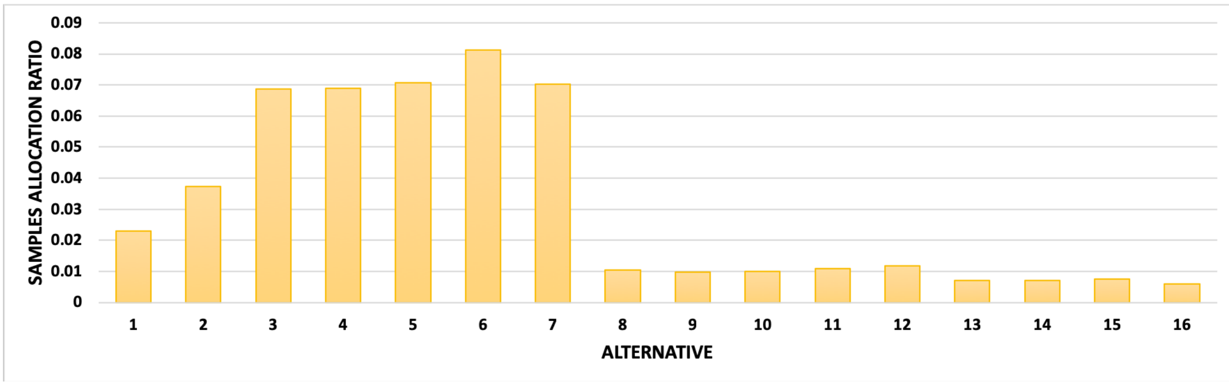

Figure 3.10: Allocation of samples to the different alternatives for 16-alternatives case of M-MOBA PCS

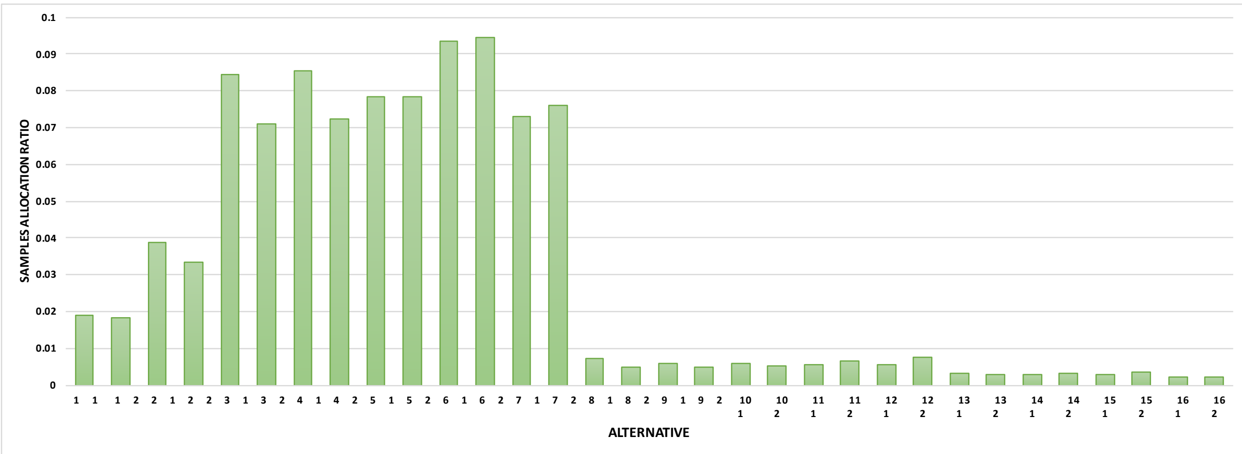

Figure 3.11: Allocation of samples to the different alternatives and objectives for 16-alternative case of M-MOBA DS PCS

5 are very close in objective 1. We assume that M-MOBA DS PCS will allocate more simulation samples on objective 2 for alternative 1 and 4 and on objective 1 for alternative 3 and 5 in order to distinguish the small difference that affects the correct selection.

Figure 3.13 shows the allocation frequency. As expected, M-MOBA DS PCS spends most simulation samples on the first objective of alternative 3 and 5. Although it still spends more simulation samples on alternative 1 and 5's first objective because of the higher cost of simulating objective 2, the sampling difference between the two objectives is much smaller compared to the alternative 3 and 5 group due to the tiny difference between alternative 1 and alternative 4 on the second objective.

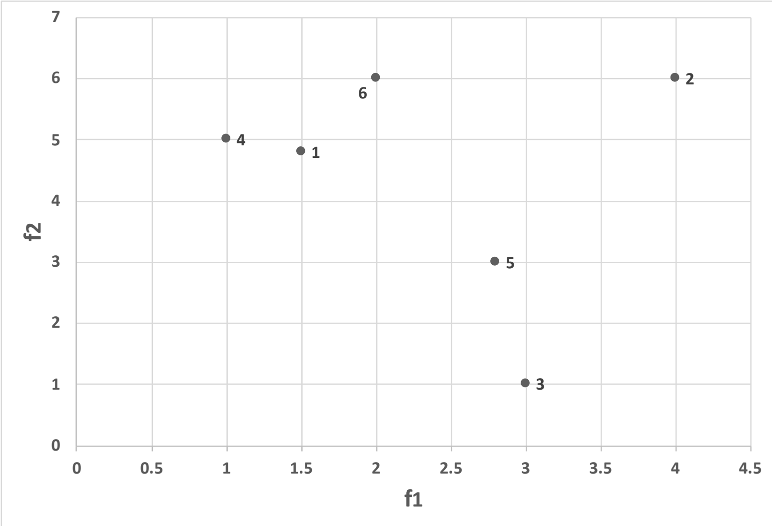

Figure 3.12: 6-alternative case

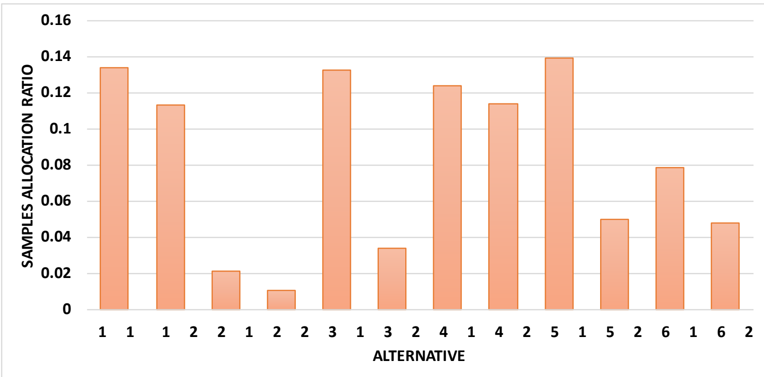

Figure 3.13: Allocation of samples to the different alternatives and objectives in the 6-alternative case of M-MOBA DS PCS

3.5 Conclusion

In this chapter, a new simple myopic budget allocation algorithm named M-MOBA PCS, for solving multi-objective problems using the probability of correct selection as the performance measure has been proposed. This method is an extension of the small-sample EVI procedure proposed by Chick et al. [2010] to multi-objective problems. Empirical comparisons to Equal allocation and MOCBA show that the new method works very well, especially in situations where the total budget that can be allocated is small. We also developed a variant when objectives can be evaluated independently. Instead of evaluating all objectives of an alternative simultaneously as in the M-MOBA PCS procedure, we will evaluate only one objective of one alternative in each iteration. Empirical results show that by allowing to evaluate objectives independently, the efficiency of the algorithm improve substantially. However, if two alternatives have almost identical performance, even a large number of samples may not be able to correctly identify the better one, and anyway the decision maker might not care about very small differences. Therefore, a M-MOBA variant with PGS as performance measure is proposed in the following chapter.

Chapter 4

M-MOBA indifference zone procedure

4.1 Introduction

In practice, some systems may have very similar objective values and a DM might not be too concerned with small differences between these systems, hence we should treat these designs as equally acceptable [Teng et al., 2010]. Furthermore, if the difference is very small, even a large number of samples would not allow us to decide with confidence which system is better. As discussed in Section 2.3.1, one way to deal with this is to introduce an indifference zone, and use the probability of good selection as performance criterion. However, it is not obvious how to define an indifference zone in the case of multiple objectives. In this chapter, we introduce a new concept of indifference zone and good selection, and develop a corresponding M-MOBA indifference zone (M-MOBA IZ) algorithm.

Teng et al. [2010] have proposed an indifference zone concept for multi-objective problems as follows. A DM is indifferent between system j and system i in objective h, denoted by $\mu_{j,h} \simeq \mu_{i,h}$ if and only if $|\delta_{ijh}| \leq \delta_h$, where $\delta_{ijh} = \mu_{j,h} - \mu_{i,h}$ and δ_h is the indifference-zone of the hth objective. Based on this definition, any solution located within the indifference zone area of solution m is indifferent to m and so the dominance relationship can be visualised as shown in Figure 4.1. PGS has been defined as the probability that exactly all the solutions that are not dominated by any other solution have been identified correctly. However, with this definition small differences can still switch a solution between being in the desired set or not. For example, in the scenario shown in Figure 4.1, if solution n is observed as n', it will be incomparable to m while if it is observed as n'', it will dominate m. Thus

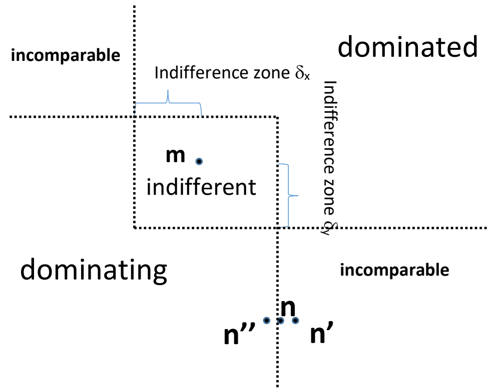

Figure 4.1: Indifference zone definition of Teng et al. [2010], and dominance of a solution relative to solution m.

an algorithm optimising under this definition is likely to spend a lot of simulation samples to distinguish the domination relationship between m and n, although the small difference may not be relevant to the DM.

This is why in the following, we will introduce an alternative definition of indifference zone for multi-objective problems.

4.2 New definition of indifference zone and good selection

The key idea of our new indifference zone definition is to extend the number of categories. In addition to a system being either dominated or non-dominated, we introduce the categories of "indifference-zone dominated", "borderline non-dominated", "borderline dominated" and "indifference-zone non-dominated" as illustrated in Figure 4.2.

A system is

- *indifference-zone dominated* if there is another solution that is at least δ_h better in each objective h
- *borderline dominated*, if it would become non-dominated by improving each objective h by δ_h
- *borderline non-dominated*, if it is non-dominated, but would become dominated by worsening each objective h by δ_h

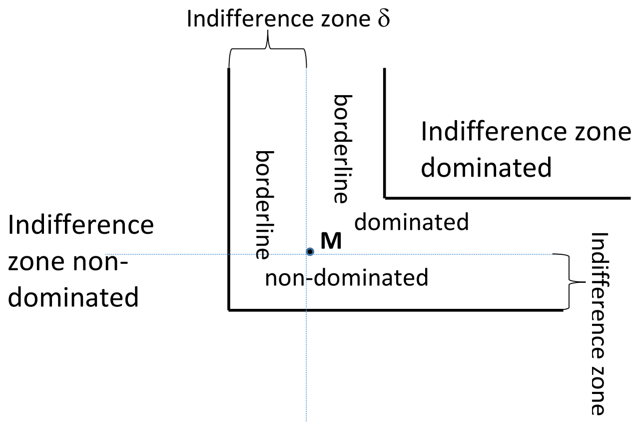

Figure 4.2: New indifference zone definition

- *indifference-zone non-dominated* if it remains non-dominated even if all objectives are worsened by δ_h

  More formally,

- system i indifference-zone dominates system j, denoted by $i \prec_{IZ} j$, if $\mu_{i,h} < \mu_{j,h} - \delta_h, \forall h = 1, 2, ..., H$
- system i borderline dominates system j, denoted by $i \precsim_{IZ} j$, if $\mu_{i,h} < \mu_{j,h}$, $\forall h = 1, 2, ..., H$ and $\exists h \in \{1, 2, ..., H\}$, $|\delta_{ijh}| \leqslant \delta_h$

  Therefore, a system j is categorised as

- *indifference-zone dominated* if $\exists i \in \{1, 2, ..., m\}$, $i \prec_{IZ} j$
- *borderline dominated* if $\nexists i \in \{1, 2, ..., m\}$, $i \prec_{IZ} j$ and $\exists i \in \{1, 2, ..., m\}$, $i \precsim_{IZ} j$
- *borderline non-dominated* if $\nexists i \in \{1, 2, ..., m\}$, $i \prec_{IZ} j$, $\nexists i \in \{1, 2, ..., n\}$, $i \precsim_{IZ} j$ and $\exists i \in \{1, 2, ..., m\}, h \in \{1, 2, ..., H\} \mu_{i,h} > \mu_{j,h} - \delta_h$
- *indifference-zone non-dominated* if $\nexists i \in \{1, 2, ..., m\}$, $i \prec_{IZ} j$ or $i \precsim_{IZ} j$ and $\nexists i \in \{1, 2, ..., m\}, \nexists h \in \{1, 2, ..., H\} \mu_{i,h} > \mu_{j,h} - \delta_h$

For example, in Figure 4.3, we have a set of indifference-zone non-dominated systems a, b, c, which are still Pareto optimal if both objectives increase by a small amount δ (a', b', c' are still Pareto non-dominated). By contrast, d will be

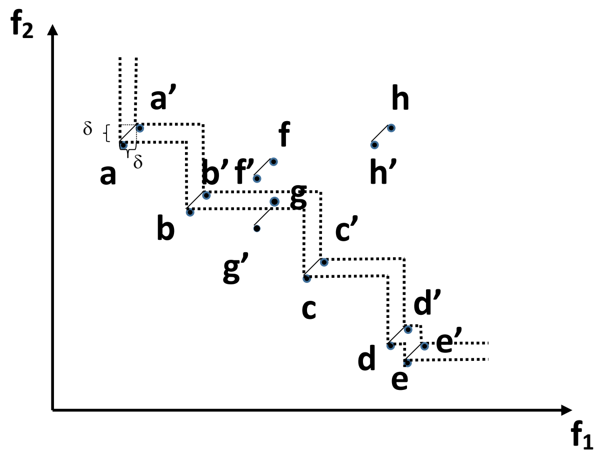

Figure 4.3: An example of solutions in different dominance categories

dominated by e if its objective values increase by δ and vice versa, and thus d and e are borderline non-dominated. Similarly, solution f and h are indifference-zone dominated as they would still be Pareto dominated even if both objectives are improved by δ while g is borderline dominated as it would become non-dominated decreasing its objective values by δ.

Based on the above definitions, we propose a definition of “good selection”. If c_i is the “true” category of alternative i, we still count the solution as correctly classified if based on the observed objective values, the category is “similar” to the true category, as defined in Table 4.1. For example, we accept if a borderline dominated solution is classified as borderline non-dominated or as dominated, but we do not accept if it is classified as indifference-zone non-dominated. This solves the issue of classifying n in Figure 4.1, as there is a tolerance for classification in adjacent categories.

Table 4.1: Difference between observed and true classification is still considered as “good” for cases marked with checkmark.

True / Observed	Indifference-zone Dominated	Borderline dominated	Borderline non-dominated	Indifference-zone Non-dominated
Indifference-zone Dominated	✓	✓	✗	✗
Borderline dominated	✓	✓	✓	✗
Borderline non-dominated	✗	✓	✓	✓
Indifference-zone Non-dominated	✗	✗	✓	✓

4.3 M-MOBA IZ procedure

We use the above definition of PGS to design an M-MOBA procedure that can work with indifference zones (M-MOBA IZ). Similar to the original M-MOBA, we will calculate the probability that a solution, if re-sampled, will change its category by more than one grade. Similar to the M-MOBA PCS procedure, we discuss the calculation of the probability based on the current domination situation of each alternative.

For a solution that is indifference-zone dominated or borderline dominated:

- For a solution that is indifference-zone dominated, the area that a_c needs to move out to change the selected set is exemplified in Figure 3.4 and the probability P can be calculated with Equation(3.5).
- For a solution that is borderline dominated, if all other solutions on the observed Pareto front are indifference-zone non-dominated, an example for the area that a_c needs to move out is shown in Figure 4.4, i.e., the original area plus the striped area that allows a_c to become borderline non-dominated.

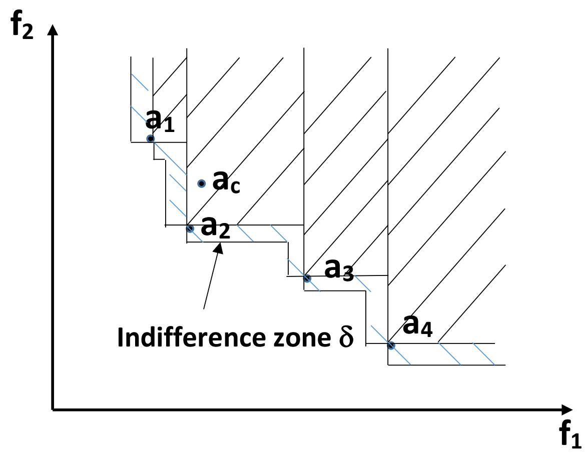

Figure 4.4: Indifference zone for a borderline dominated solution a_c if all other solutions on the observed Pareto front are non-dominated

- For a solution that is borderline dominated, if a solution on the observed Pareto front is borderline non-dominated, the area that a_c needs to leave is the area discussed above plus the small rectangle around the borderline non-dominated solution. For example, if solution a_2 shown in Figure 4.5 is borderline non-dominated (with respect to a_c), the area with indifference zone for a_c is the shaded part.

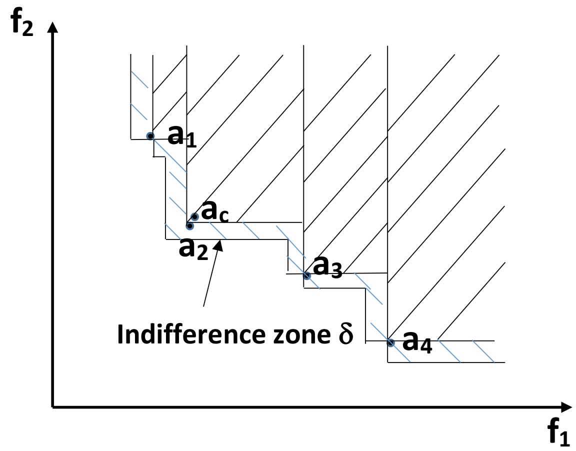

Figure 4.5: Indifference zone for a borderline dominated solution a_c if a borderline non-dominated solution exists

For a solution that is on the observed Pareto front and no new solutions become indifference-zone non-dominated or borderline non-dominated when this solution is removed:

- For an indifference-zone non-dominated solution, if all solutions on the observed Pareto front are indifference-zone non-dominated, the area that a_c needs to move out of is exemplified in Figure 3.3 and the probability P can be calculated with Equation (3.4).

- For an indifference-zone non-dominated solution, if a solution on the observed Pareto front is borderline non-dominated, the area that a_c needs to move out is the area in Figure 3.3 plus the stripe areas around the borderline non-dominated solution shown as Figure 4.6. Here, a_1 is borderline non-dominated (due to a_4), the area a_c needs to leave in order to bring a change is the shaded part. Furthermore, if two borderline non-dominated solutions are neighbours on the Pareto front, the small square area between the two stripe areas also needs to be added. For example, in Figure 4.7, a_1 and a_2 are both borderline non-dominated (due to a_4 and a_5, respectively), the area that a_c needs to leave in order to bring a change is the shaded part shown in Figure 4.7.

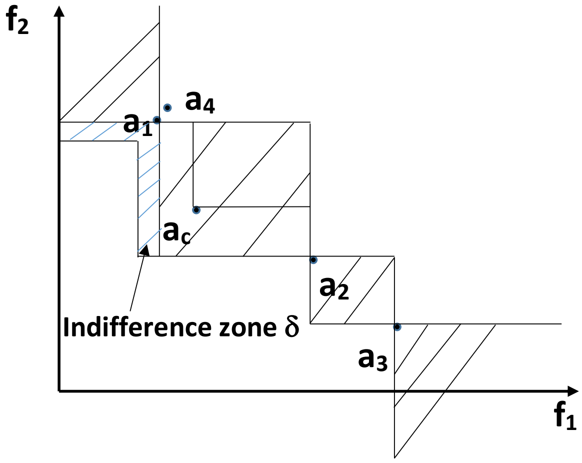

Figure 4.6: Indifference zone for a borderline non-dominated solution if a borderline non-dominated solution exists

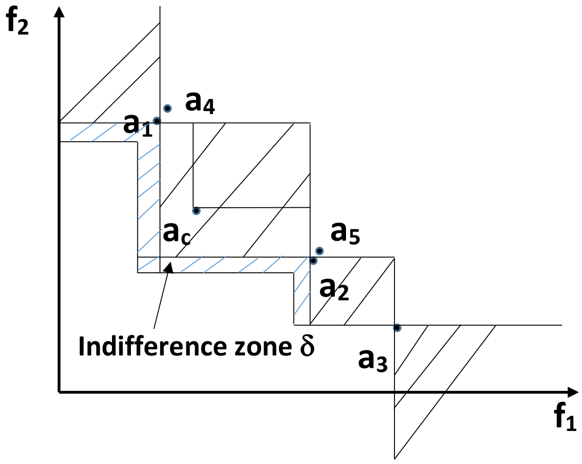

Figure 4.7: Indifference zone for an indifference-zone non-dominated solution if neighboured borderline non-dominated solutions exist

- For a borderline non-dominated solution, if all other solutions on the observed Pareto front are indifference-zone non-dominated, the shaded area that a_c needs to move out is shown in Fig. 4.8, which is the original shaded area from Figure 3.3 plus a stripe area on the upper right side.
- For a borderline non-dominated solution, if a solution on the observed Pareto

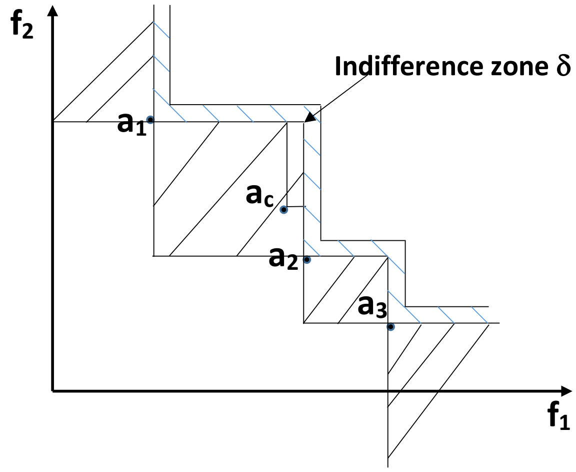

Figure 4.8: Indifference zone for a borderline non-dominated solution if all solutions on the observed Pareto front are indifference-zone non-dominated

front is borderline non-dominated, the shaded area that a_c needs to leave is the area discussed in Figure 4.7 plus the stripe area on the upper right side. For example, similar with the situation in Figure 4.7 where a_1 and a_2 are both borderline non-dominated, the area that a_c needs to leave in order to bring a change is the shaded part shown in Figure 4.9.

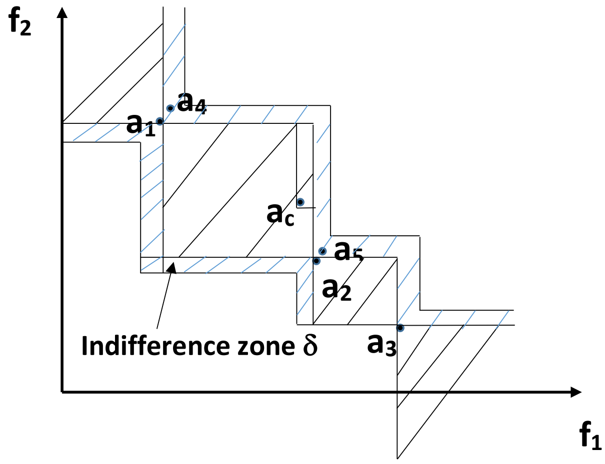

Figure 4.9: Indifference zone for a borderline non-dominated solution if a borderline non-dominated solution exists

For a solution that is on the observed Pareto front and new solutions become indifference-zone non-dominated or borderline non-dominated when this solution is removed:

- If the new Pareto optimal solutions after the solution under consideration is removed are all indifference-zone non-dominated, we only need to check solutions that define the shaded area shown in Figure 3.2. If some solutions that define the left and down side of the shaded area are borderline non-dominated, the shaded area can be extended accordingly. For example, in Figure 4.10, since a_2 is borderline non-dominated, the area that a_c needs to leave is as the figure shows.

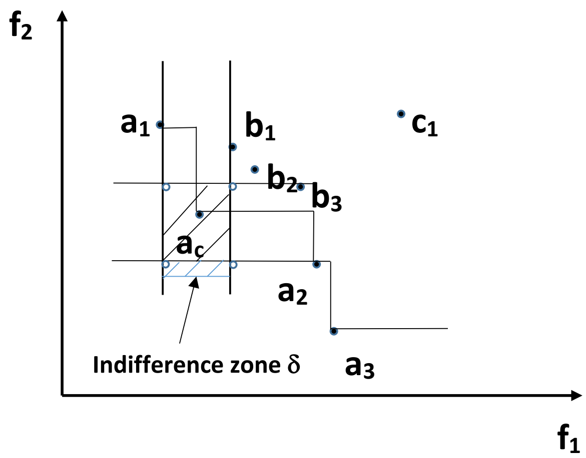

Figure 4.10: Indifference zone for a solution that, if removed, reveals a set of non-dominated solutions

- If the new Pareto optimal solution after the solution under consideration is removed is borderline non-dominated, the situation is so complex that we have not found a good method to summarise. For this situation, we use a brute-force method that divides the whole plane into different cells based on each solution's objective values and the indifference zone in each objective accordingly, and checks for each cell whether it would change the current Pareto front in case the currently considered solution were to fall into this cell. For example, if we have 4 solutions in total as in Figure 4.11, the number of cells that need to be considered is $(4 * 3)^2 = 144$. Please note that for the sake of clear demonstration, the domination relationship in this figure does not exactly conform to the situation that the new Pareto optimal solution after the solution under consideration is removed is borderline non-dominated.

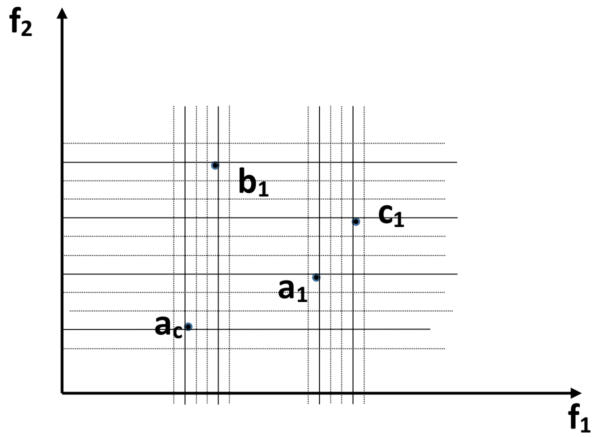

Figure 4.11: Cells created to compute probability of change

4.4 M-MOBA IZ Algorithm

Similar to the M-MOBA PCS process, we can formulate the M-MOBA IZ procedure as summarised in Algorithm 2. The first-stage sample size n_0, the value of τ and stopping rule is identical with those of M-MOBA PCS. The indifference zone value of δ needs to be specified. The way of calculating probability P_i that the new samples will lead to a change on current domination situation is according to the discussion in section 4.3. In the end of the algorithm, we will select alternatives that correctly categorised based on Table 4.1.

4.5 Numerical experiments

In order to test the performance of M-MOBA IZ, we construct a configuration that includes four categories of solutions mentioned before, namely IZ dominated, borderline dominated, borderline non-dominated and IZ non-dominated as shown in Figure 4.12. Expected values of each design are listed in Table 4.2 and the indifference zone δ is 0.2 in both objectives. The performance in terms of PCS and PGS measures is shown in Figure 4.13 and 4.14, respectively. In terms of PCS (Figure 4.13), as expected, M-MOBA PCS performs best and the difference

ALGORITHM 2: Procedure M-MOBA IZ

```
Specify a first-stage sample size n_0, and a number of samples τ = 1 to allocate per
   subsequent stage and the indifference zone δ. Specify stopping rule parameters
Sample X_{ihn}, i = 1, ..., m; h = 1, ..., H; n = 1, ..., n_0 independently, and initialise the
   number of samples n_i ← n_0
Determine the sample statistics x̄_{i,h} and σ̂²_{i,h}, and the observed Pareto front
while stopping rule not satisfied do
   For each alternative i, calculate the probability P_i that the new samples will lead to
      a change on current domination situation respectively according to the different
      situation of a_c discussed in section 4.3
   Allocate τ samples to the alternative that has the largest P_i
   Update sample statistics n_i, x̄_{i,h} and σ̂²_{i,h} and observe a new Pareto front
end while
Select alternatives that are correctly categorised according to Table 4.1
```

between its performance and Equal allocation is quite large. Both M-MOBA IZ and M-MOBA PCS work better than Equal allocation throughout the run. In terms of PGS, the highest PGS reached by M-MOBA IZ is more than five times higher than the highest PCS reached by any algorithm within the same budget since PGS is a less strict criterion. M-MOBA PCS performs even worse than Equal allocation in terms of PGS, which confirms that focusing too much on PCS may be detrimental if the user has an indifference zone. Our proposed M-MOBA IZ, on the other hand, works very well. To further investigate how the different methods spend the simulation samples, Figure 4.15 shows the percentage of samples allocated to a particular design. M-MOBA spends quite a lot of samples on alternatives 2, 4, 7, 9,10 and 11 in order to distinguish the small differences between these alternatives. By contrast, the samples spent by M-MOBA IZ are more evenly distributed except the apparently dominated solutions of 3, 5, and 6.

Table 4.2: Configuration with 13 alternatives and two objectives. Standard deviation for all designs is 1.5 in each objective.

Index	Obj. 1	Obj. 2	Index	Obj. 1	Obj. 2
1	1	8	8	1.5	6
2	2	5	9	2.1	5.2
3	3.5	5.01	10	2.5	4
4	3	2	11	2.6	3.9
5	2.5	8	12	2	7
6	3	7	13	2.5	6
7	3.05	2.2			

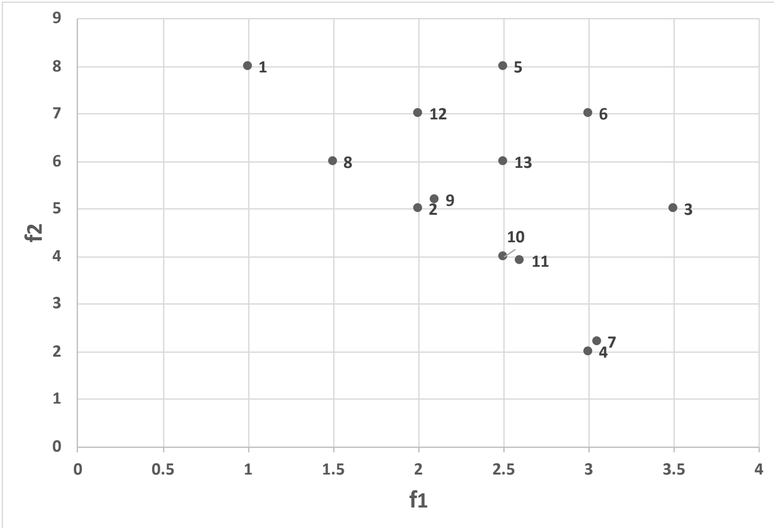

Figure 4.12: Similar solution configuration with 13 alternatives

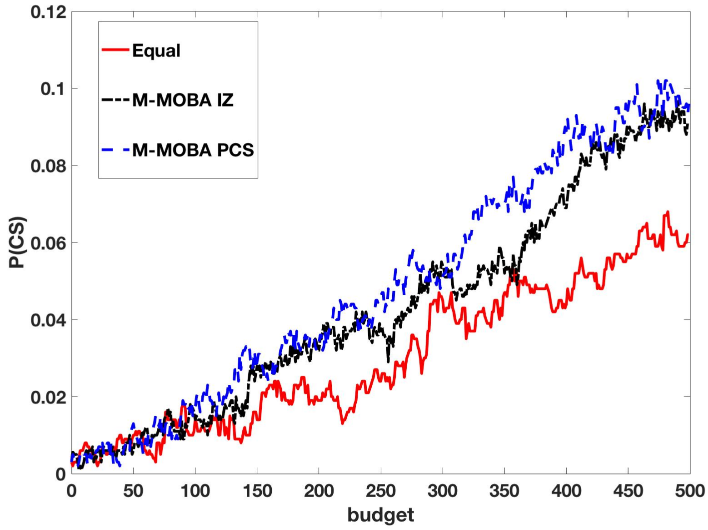

Figure 4.13: Similar solution configuration PCS performance comparison

4.6 Conclusion

When some systems have very similar objective values and a DM do not be concerned with small differences between these systems, M-MOBA PCS is not proper as it aims to find all true Pareto optimal solutions. Thus, in this chapter, a M-MOBA variant with PGS as performance measure is proposed. We introduce a novel definition

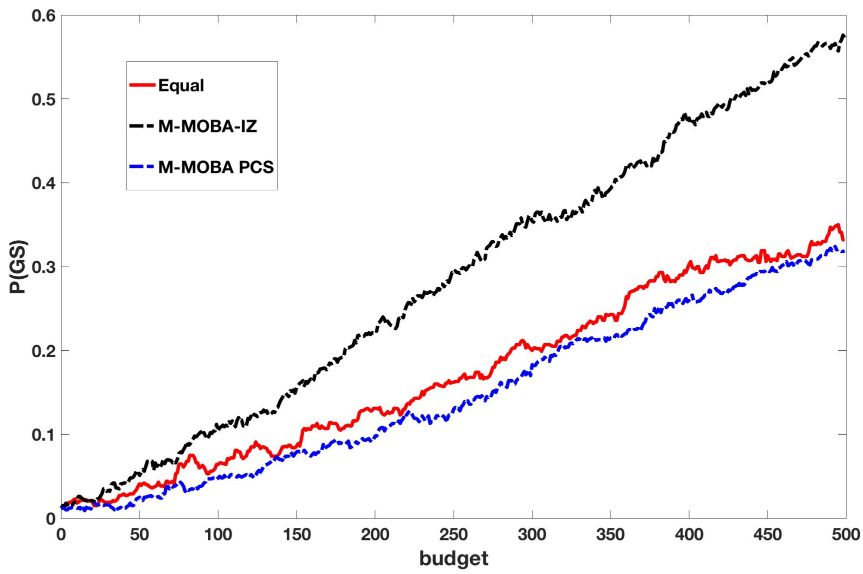

Figure 4.14: Similar solution configuration PGS performance comparison

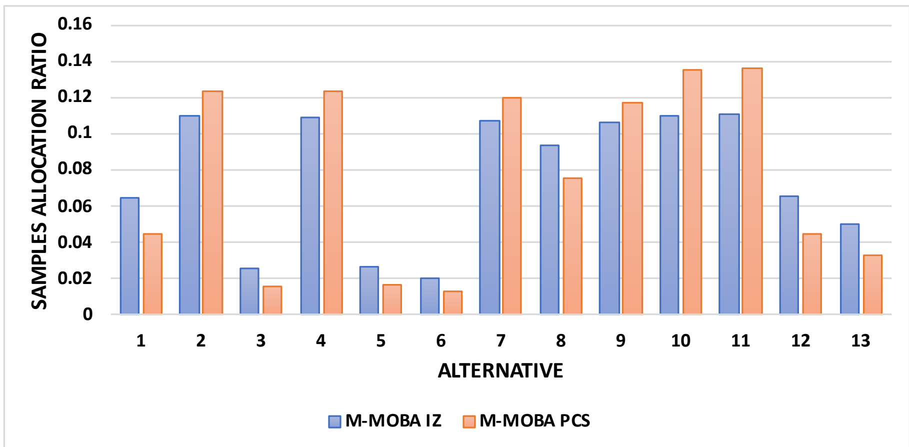

Figure 4.15: Allocation of samples to different alternatives for 13 similar alternatives configuration

of indifference zone and good selection, and develop a corresponding M-MOBA IZ algorithm. Empirical comparisons to Equal allocation and M-MOBA PCS demonstrate the advantage of applying M-MOBA IZ in situations where similar systems exist.

In spite of the difference, both M-MOBA PCS and M-MOBA IZ aim to find true (or very close to the true) Pareto optimal systems. This zero-one loss performance measure is not adequate in all scenarios. In the following chapter, we will discuss this problem explicitly and propose a solution correspondingly.

Chapter 5

M-MOBA Hypervolume procedure

5.1 Introduction

Although PCS is useful to identify the true Pareto optimal set, there are some disadvantages. Consider a scenario shown in Figure 5.1, with the true values of a set of Pareto optimal solutions a, b, c and d depicted, and an iso-utility curve corresponding to a specific DM. Solution b will be correctly identified as the most preferred solution for this DM. However, if solution c would be observed as c', the domination relationships among all solutions remain the same, and thus this deviation from the true mean would not impact the PCS measure. The DM, however, would now falsely select c' as best solution, and suffer a loss in utility. Another disadvantage of PCS is illustrated in Figure 5.2. Intuitively, solutions a and c are much more likely to be picked by a DM than solution b, since they are much better than b in one objective but just a little worse in the other objective. So, misclassifying b is probably not as bad as misclassifying a and c, but PCS does not make this distinction.

Given these drawbacks of the PCS measure for multi-objective problems, we propose *hypervolume difference* as an alternative measure.

Let Λ denote the Lebesgue measure, then the hypervolume (HV) is defined as

$$HV(B,R) := \Lambda(\bigcup_{y\in B}\{y' \mid y \prec y' \prec R\}), \qquad B \subseteq \mathbb{R}^m \tag{5.1}$$

where B is a set of solutions and $R\in\mathbb{R}^m$ denotes a reference point that is usually user defined and chosen such that it is dominated by all other solutions. In the context of ranking and selection, the reference point could also be determined based

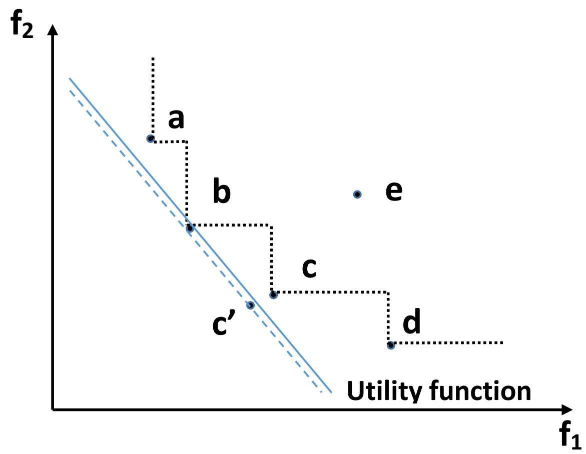

Figure 5.1: Even though all dominance relations are correct if solution c is observed as c', the DM may pick the wrong solution.

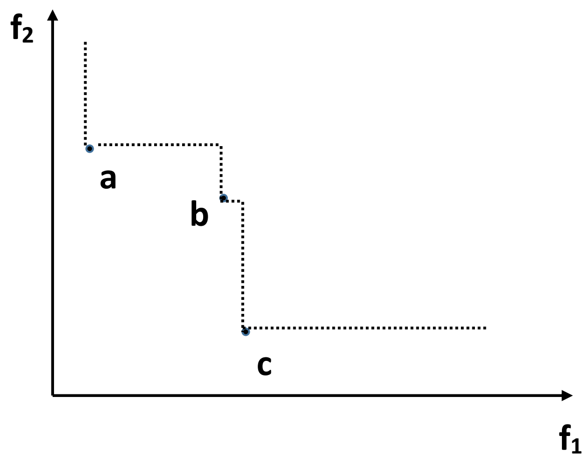

Figure 5.2: Solutions a and c are more likely to be preferred by a DM.

on the initial observations for each system.

Figure 5.3 shows a set of 5 solutions in 2-objective space. Four of the solutions are Pareto-optimal, and the HV is the shaded area, defined by the Pareto optimal solutions and the reference point R. The dominated solutions (solution e in this configuration) do not contribute to the HV. HV is a standard metric to judge the performance in multi-objective optimization [Beume et al., 2007]. It rewards

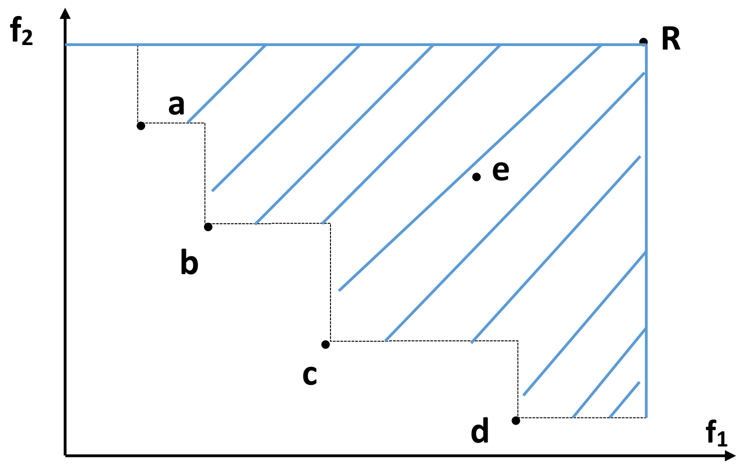

Figure 5.3: Hypervolume of a set of solutions

solutions close to the true Pareto front, as well as a good spread of solutions along the true Pareto front [Beume et al., 2007].

For the case of ranking and selection where evaluations are stochastic, we need a metric that penalises over-estimation as well as under-estimation of objective values, and thus propose the hypervolume difference (HVD). Given two sets of Pareto-optimal solutions A and B,

$$\begin{aligned} HVD(A,B,R) &:= HV(A,R) + HV(B,R) - 2*(IHV(A,B,R), \\ \text{where } IHV(A,B,R) &= \Lambda\{y' \prec R \mid \exists(y \in A, z \in B) : (y \prec y') \wedge (z \prec y')\} \end{aligned} \tag{5.2}$$

Basically, HVD is the difference between the hypervolume of two set of solutions according to the chosen reference point. Figure 5.4 provides an example for the proposed HVD in a two objective configuration.

HVD is able to overcome the drawbacks of PCS-based metrics discussed above. For the scenario shown in Figure 5.1, HVD will penalise deviations from the true fitness values of Pareto-optimal solutions, even if all dominance relations are correct, see Figure 5.5. And for the scenario shown in Figure 5.2, while PCS fails to reflect the higher importance of a and c, hypervolume does pay more attention to these solutions. This is illustrated in Figure 5.6: If distorting solutions a and b by the same distance and direction, the HVD between the new and old Pareto front

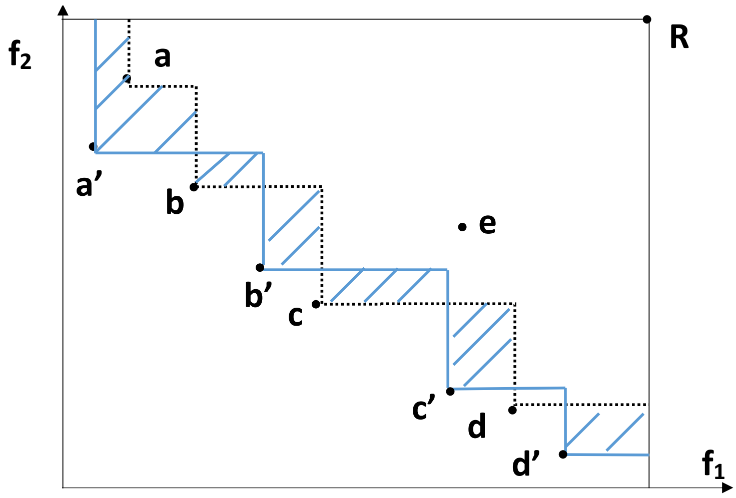

Figure 5.4: Hypervolume difference of two sets of solutions

made by a distortion to a is larger than by the same distortion to b.

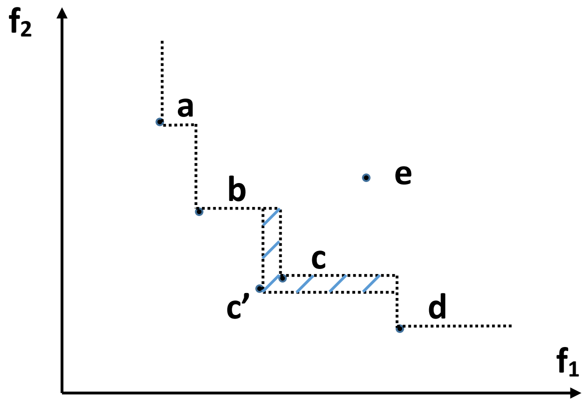

Figure 5.5: Hypervolume difference penalises any deviation from the true front

As additional advantage, it should be noted that HVD also allows straightforward incorporation of partial user preferences. If a DM already has a rough idea of the region in which the desired solutions are likely to be, the reference point can be set to reflect this preference by setting it to the maximum acceptable value in each objective. For example, if the reference point is defined as R shown in Fig-

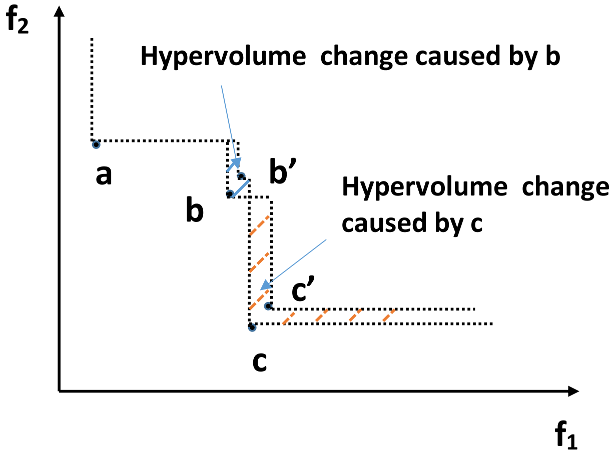

Figure 5.6: Hypervolume change caused by distorting different solutions is different

ure 5.7, solutions a and d will have little influence on HVD, even if their values are disturbed, and thus ranking and selection will focus its sampling effort on the more relevant solutions b and c.

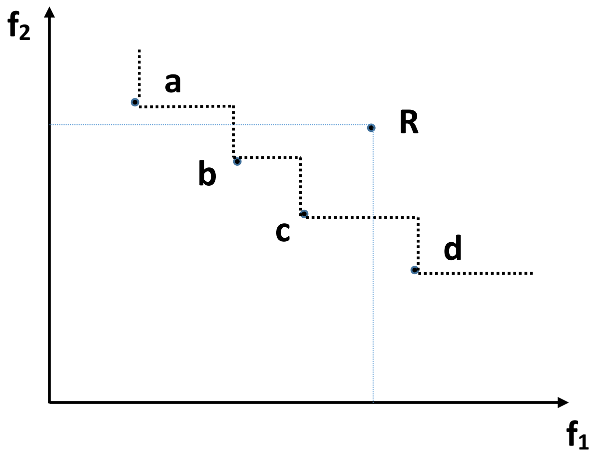

Figure 5.7: Effect of choosing reference point

Following the general M-MOBA framework, we will sample where we expect the sample will lead to the biggest change in HV, i.e., where the expected HVD between the Pareto fronts before and after sampling is maximal.

Our derivation of the HVD is partly based on the expected hypervolume computation by Emmerich and Klinkenberg [2008]. In the following parts, we will describe a few illustrative examples of the HV change in Section 5.2, discuss how to approach this computationally, and provide the mathematical derivation of a closed formula in Section 5.3.

5.2 Determining the hypervolume change

Similar to the M-MOBA PCS procedure, we will discuss the calculation of hypervolume change based on the current situation of each solution.

For a currently dominated solution, b_c shown in Figure 5.8, if it is updated due to a new sample but remains dominated, it doesn't change the hypervolume. If it becomes non-dominated, for example, if it moves to a new position b_c' in Figure 5.8, it will cause the HV to increase by the shaded area.

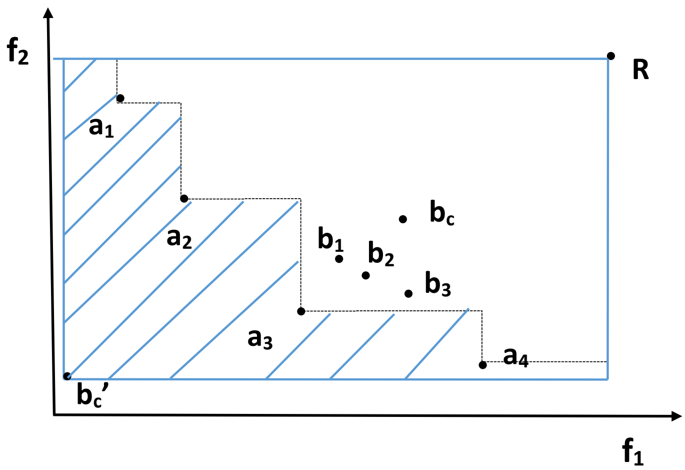

Figure 5.8: Increased HV if b_c moves to b_c'

For solution a_c in Figure 5.9, which is currently non-dominated, the situation is more complex. If the alternative becomes dominated by allocating a new sample to it, such as if it moves to a_c', the HV will decrease by the shaded area between a_c and the new Pareto optimal solutions b_1, b_2, b_3. This area is constant as long as a_c' becomes dominated, and can be easily calculated.

If a_c shifts to a new position a_c' above and left of its left neighbour on the Pareto front (a_2 in Figure 5.10 or a_c'' below and right of the right neighbour a_3), it

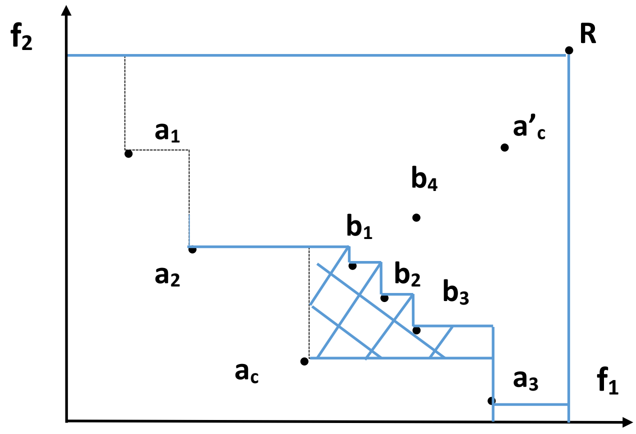

Figure 5.9: HV change if a non-dominated solution a_c moves to dominated location a'_c

will cause an increase in HV in one area (near a'_c or a''_c), and a decrease in another (the part dominated by the original a_c).

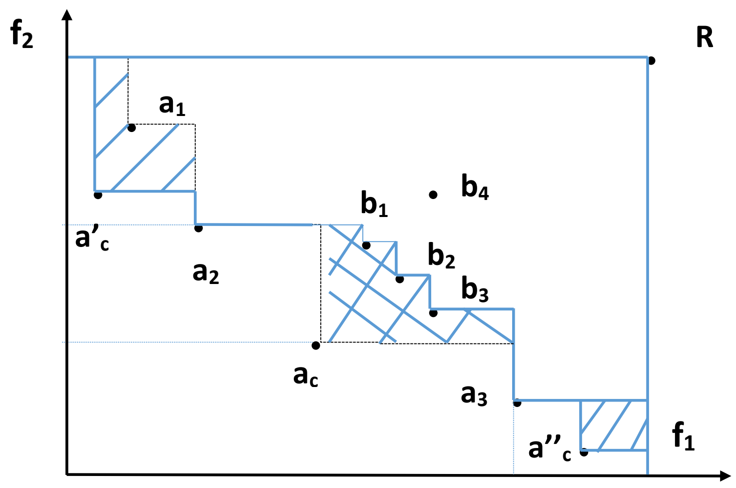

Figure 5.10: HV change if a_c moves to a'_c or a''_c

If a_c moves to a location that dominates its previous location, such as a'_c in Figure 5.11, this change will only increase the HV by the shaded area, there is no reduction of HV.

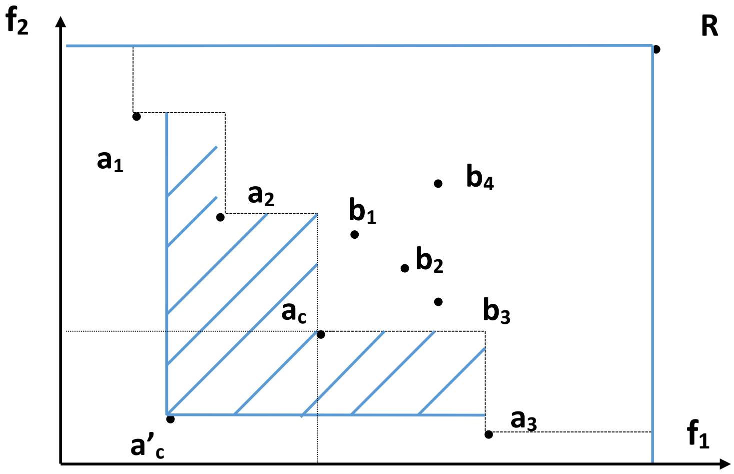

Figure 5.11: Increased part if a_c to a_{c3}

If a_c moves into the area originally dominated by a_c, but remains non-dominated, there is only a reduction of HV, an example is provided in Figure 5.12, which is a zoom-in of the particular area of interest.

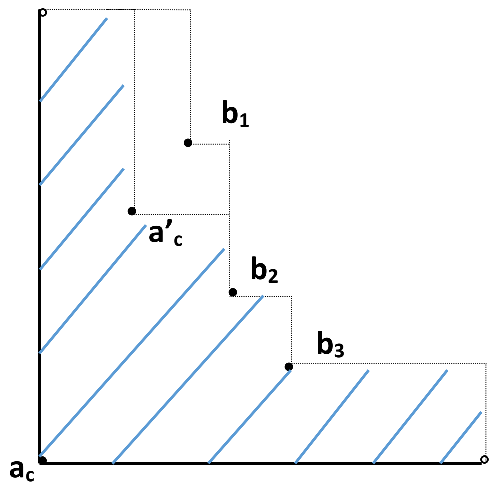

Figure 5.12: Increased part when a_c to a_{c4}

Finally, in the remaining area, there is again a part that is increased and a part that is decreased, see Figure 5.13 for an example.

As we have seen above, drawing an additional sample for a particular alter-

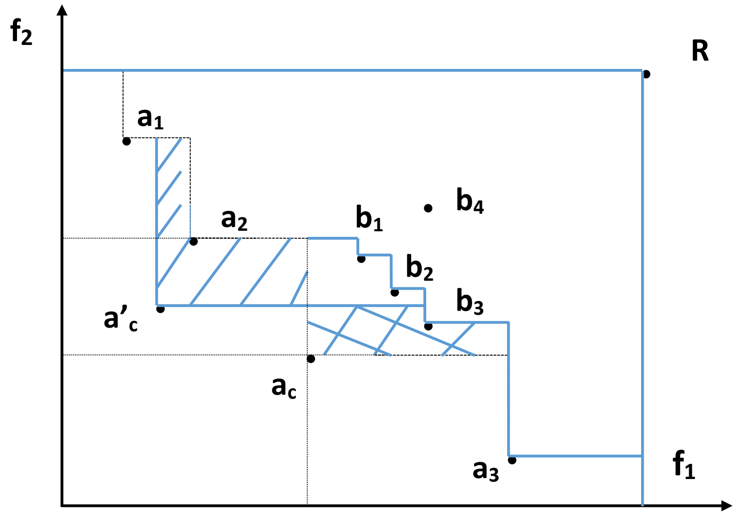

Figure 5.13: Change lead by a_c to a_{c5}

native and the subsequent change in this alternative's mean value can lead to an increase of HV in some area and a decrease in some other area. In the following section, we will introduce how to calculate the expected HV change mathematically.

5.3 Calculation of the expected HV change

Calculating the expected HV change requires to break down the calculation into different cells, but for each cell, we can find a closed form expression. Then, these expected changes can be added up to result in the overall expected HV change. In the following, we will explain the computation for one particular cell, with other cells computed analogously.

Consider Figure 5.14, where all solutions on the current Pareto front are labelled $a_1, \ldots, a_k$, with coordinates $a_{i,h}$ for alternative i and objective h, and the solutions are sorted in increasing order of objective 1. For technical reasons, let us define $a_{0,1} = -\infty, a_{0,2} = a_{r,2}, a_{k+1,1} = a_{r,1}, a_{k+1,2} = -\infty$, where r is the reference point. We consider another sample for design a_c, and the calculation for one particular cell that is outlined in bold and defined by upper right corner u with coordinates (u_1, u_2) and lower left corner l with coordinates (l_1, l_2). Let us assume that these two corners are defined by the Pareto optimal solutions a_p and a_q, by $u = (a_{p+1,1}, a_{q-1,2})$ and $l = (a_{p,1}, a_{q,2})$.

Then, the contribution of the cell to the expectation of the HV change when

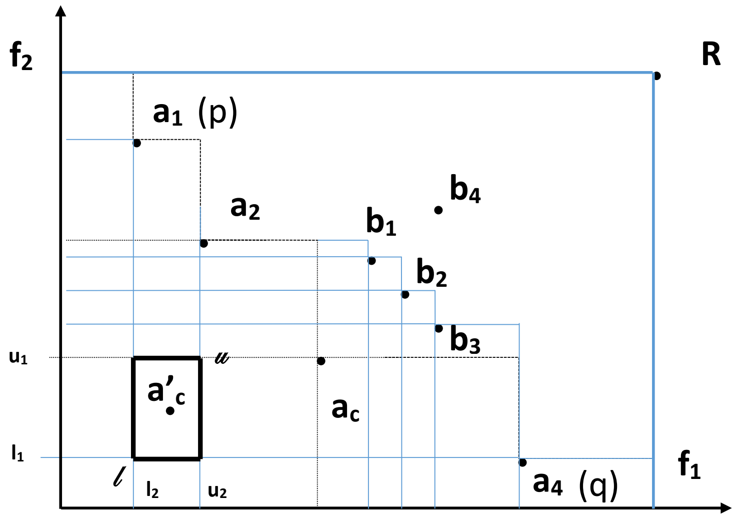

Figure 5.14: Different cells that need to be considered when calculating the expected HV change from re-sampling

sampling design a_c is computed as

$$\int_{l_2}^{u_2}\int_{l_1}^{u_1}\left[(a_{p+1,1}-x)(a_{p,2}-y)+\sum_{p<i<q}(a_{i+1,1}-a_{i,1})(a_{i,2}-y)\right]\cdot\phi_{c,1}(x)\cdot\phi_{c,2}(y)dxdy \tag{5.3}$$

where $\phi_{c,h}$ is the predictive probability distribution of the new location of x_c in dimension h.

For efficient computation, we derive a closed form for calculating the expected HV change in one cell.

Let $\phi(x;\mu,\kappa,\nu)$ denote the distribution of $\mu+\frac{1}{\sqrt{\kappa}}T_\nu$, where T_ν is a random variable with standard t distribution with ν degrees of freedom, i.e., the t distribution we estimate for the new location of an alternative's mean values after having taken another sample, with mean μ, precision κ and ν degrees of freedom. The cumulative density function is then

$$\Phi(x;\mu,\kappa,\nu)=\Phi_t(\sqrt{\kappa}(x-\mu);\nu). \tag{5.4}$$

with $\Phi_t(x;\nu)$ the cumulative standard t-distribution, and the probability density function is [DeGroot and Schervish, 2012]

$$
\begin{aligned}
\phi(x;\mu,\kappa,\nu) &= \sqrt{\kappa}\cdot\phi_t(\sqrt{\kappa}(x-\mu);\nu)\\
&= \sqrt{\frac{\kappa}{\nu\pi}}\frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})}\cdot\left(1+\frac{\kappa(x-\mu)^2}{\nu}\right)^{-\frac{\nu+1}{2}}
\end{aligned}
\tag{5.5}
$$

with $\phi_t(x;\nu)$ the standard t-distribution. The HV change, due to the point we are considering moving to a new position (x, y), is always a function in the form $axy + bx + cy + d$. The constant coefficients a, b, c, d are different in different areas and some of the coefficients could be 0 sometimes. The contribution of the area $[l_1, u_1] \times [l_2, u_2]$ (e.g, the small cell highlighted in Figure 5.14) to the expectation of the HV change is

$$
\begin{aligned}
&\int_{l_1}^{u_1}\int_{l_2}^{u_2}(axy+bx+cy+d)\cdot\phi_{i1}(x)\cdot\phi_{i2}(y)dxdy\\
=&a\int_{l_1}^{u_1}x\phi_{i1}(x)dx\int_{l_2}^{u_2}y\phi_{i2}(y)dy + b\cdot\Phi_{i2}(y)|_{l_2}^{u_2}\cdot\int_{l_1}^{u_1}x\phi_{i1}(x)dx\\
&+c\cdot\Phi_{i1}(x)|_{l_1}^{u_1}\cdot\int_{l_2}^{u_2}y\phi_{i2}(y)dy + d\cdot\Phi_{i1}(x)|_{l_1}^{u_1}\cdot\Phi_{i2}(y)|_{l_2}^{u_2}
\end{aligned}
\tag{5.6}
$$

where $\phi_{ih}(x) = \phi(x;\mu_{ih},\kappa_{ih},\nu_i)$, $\Phi_{ih}(x) = \Phi(x;\mu_{ih},\kappa_{ih},\nu_i)$, $\mu_{ih} = \bar{x}_{ih}$, $\kappa_{ih} = n_i(n_i+\tau_i)/\tau_i\hat{\sigma}^2_{ih}$ and $\nu_i = n_i - 1$. On the right hand side of Equation (5.6), the most critical part is solving the integrals, and it can be done by calculating the corresponding indefinite integral, which is

$$
\begin{aligned}
\int x\phi(x;\mu,\kappa,\nu)dx &= \int(x-\mu)\phi(x;\mu,\kappa,\nu)dx + \mu\Phi(x;\mu,\kappa,\nu)dx\\
&= \psi(x;\mu,\kappa,\nu) + \mu\Phi(x;\mu,\kappa,\nu)
\end{aligned}
\tag{5.7}
$$

with

$$
\begin{aligned}
\psi(x;\mu,\kappa,\nu) :=& \int(x-\mu)\phi(x;\mu,\kappa,\nu)dx\\
=&\sqrt{\frac{\nu}{\kappa\pi}}\cdot\frac{\Gamma(\frac{\nu+1}{2})}{(1-\nu)\Gamma(\frac{\nu}{2})}\left(1+\frac{\kappa(x-\mu)^2}{\nu}\right)^{\frac{1-\nu}{2}}\\
=&\frac{\nu+\kappa(x-\mu)^2}{(1-\nu)\sqrt{\kappa}}\phi(x;\mu,\kappa,\nu).
\end{aligned}
\tag{5.8}
$$

In the rest of this section, for convenience, we will denote $\psi(x;\mu_{ih},\kappa_{ih},\nu_i)$ as $\psi_{ih}(x)$. Using the above results and gathering the terms with same integrals, Equation (5.6) can be rewritten as

$$\begin{aligned}
&\int_{l_1}^{u_1}\int_{l_2}^{u_2}(axy+bx+cy+d)\cdot\phi_{i1}(x)\cdot\phi_{i2}(y)dxdy\\
=&a\Psi_{i1}(x)|_{l_1}^{u_1}\Psi_{i2}(y)|_{l_2}^{u_2}+(b+a\mu_{i2})\Psi_{i1}(x)|_{l_1}^{u_1}\Phi_{i2}(y)|_{l_2}^{u_2}\\
+&(c+a\mu_{i1})\Phi_{i1}(x)|_{l_1}^{u_1}\Psi_{i2}(y)|_{l_2}^{u_2}+(a\mu_{i1}\mu_{i2}+b\mu_{i1}+c\mu_{i2}+d)\Phi_{i1}(x)|_{l_1}^{u_1}\Phi_{i2}(y)|_{l_2}^{u_2},
\end{aligned} \tag{5.9}$$

where Ψ is the integral of ψ.

For example, considering the integral (5.3), the value of coefficient a, b, c and d will be

$$\begin{aligned}
&a=1, \qquad b=-a_{p2},\\
&c=-a_{p+1,1}-\sum_{p<i<q}(a_{i+1,1}-a_{i1}),\\
&d=a_{p+1,1}a_{p2}+\sum_{p<i<q}(a_{i+1,1}-a_{i1})a_{i2},
\end{aligned}$$

and then we can substitute them, in addition to

$$\begin{aligned}
\mu_{ch}&=\bar{x}_{ch},\\
\kappa_{ch}&=n_c(n_c+\tau_c)/\tau_c\hat{\sigma}_{ch}^2,\\
\nu_c&=n_c-1,
\end{aligned}$$

where $h=1$ or 2, into Equation(5.9) to solve the integral (5.3).

5.4 M-MOBA HV Algorithm

The overall M-MOBA procedure based on the HV change criterion is denoted as M-MOBA HV and the procedure is almost identical to that of M-MOBA PCS. The only difference is that for each alternative i, M-MOBA HV will calculate the expected hypervolume change that would result from allocating τ additional sample to alternative i and allocate those τ samples to the alternative i that has the largest expected hypervolume change. The M-M-MOBA HV procedure is summarised as Algorithm 3.

ALGORITHM 3: Procedure M-MOBA HV

1: Specify a first-stage sample size n_0, and a number of samples $\tau = 1$ to allocate per subsequent stage and the indifference zone δ. Specify stopping rule parameters
2: Sample $X_{ihn}, i = 1, \ldots, m; h = 1, \ldots, H; n = 1, \ldots, n_0$ independently, and initialise the number of samples $n_i \leftarrow n_0$
3: Determine the sample statistics $\bar{x}_{i,h}$ and $\hat{\sigma}^2_{i,h}$, and the observed Pareto front
4: **while** stopping rule not satisfied **do**
5:     For each alternative i, calculate the expected hypervolume change VC_i that would result from allocating τ additional sample to alternative i according to Equation 5.9
6:     Allocate τ samples to the alternative that has the largest VC_i
7:     Update sample statistics n_i, $\bar{x}_{i,h}$ and $\hat{\sigma}^2_{i,h}$ and observe a new Pareto front
8: **end while**
9: Select alternatives on the observed Pareto set and calculate the difference between the selected and true Preto set.

5.5 Numerical experiments

In this section, we test M-MOBA HV on three configurations, and compare it with two other methods, the original M-MOBA PCS procedure, and Equal allocation that allocates the same number of samples to each design. For each method, each design is sampled $n0 = 5$ times during initialisation, and then additional samples are allocated one at a time ($\tau = 1$) until a pre-set budget has been reached. Results are averaged over 1000 runs. In case we run into problems of numerical precision, τ is changed to 10 for the expected HV change calculation, but still only one sample is calculated. If the numerical precision problem persists, we will use Equal allocation until the problem stops and τ is then set back to 1.

The first configuration is still the 16 alternatives configuration proposed by Chen and Lee [2010]. Figure 5.15 demonstrates the reduction of the HV difference as the number of samples allocated increases. It can be seen that the M-MOBA-HV method performs much better than both the Equal and M-MOBA PCS methods in terms of HVD between the selected and true Pareto set. Although M-MOBA PCS has been shown to identify the Pareto optimal solutions much more quickly than Equal allocation on this problem in Chapter 3, in terms of HVD it is actually only slightly better than Equal allocation. In contrast, Figure 5.16 shows that M-MOBA PCS still works much better than M-MOBA HV and Equal allocation when P(CS) is used as the performance measure.

The second configuration is designed to demonstrate the impact of solutions that are close to being dominated or non-dominated. These points have a small influence on the resulting HV, and whether they are actually identified as dominated or non-dominated may not matter so much to a decision maker. The configuration has 10 designs, 2 objectives, and the standard deviation of each alternative in each

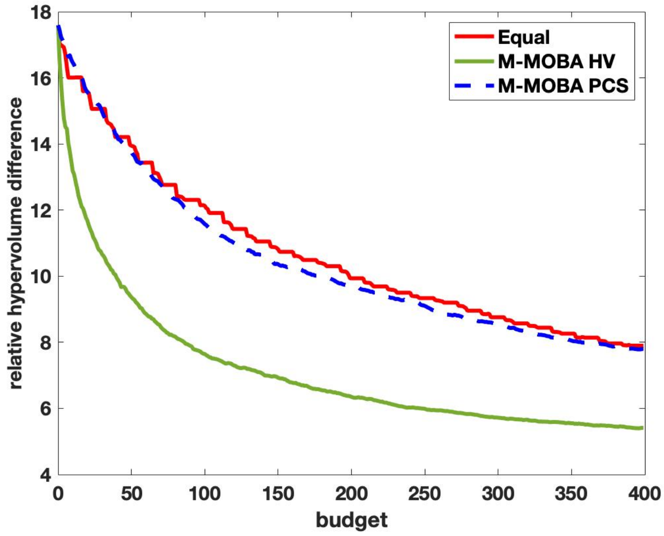

Figure 5.15: Comparison of hypervolume difference with M-MOBA HV, M-MOBA PCS and Equal allocation by using 16 alternatives configuration

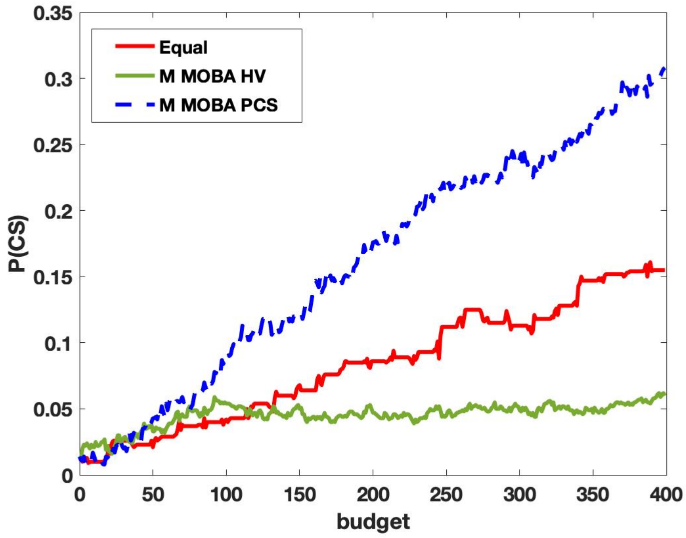

Figure 5.16: Comparison of P(CS) for standard configuration with 16 alternatives

objective is set to 2. The reference point is (10,10) in this case. Expected values of each design are shown in Table 5.1 and visualised in Figure 5.17. Designs 6 and 7 are dominated, but close to being non-dominated, and design 3 is non-dominated, but close to being dominated.

Table 5.1: Borderline configuration with 10 alternatives.

Index	Obj. 1	Obj. 2
1	1	5
2	5	1
3	3	3
4	3.1	2
5	2	3.1
6	4	2.1
7	2.1	4
8	5.5	5
9	3.5	5
10	6	6

Table 5.2: Similar solution configuration with 8 alternatives.

Index	Obj. 1	Obj. 2
1	1	5
2	5	1
3	3.2	2.1
4	3	2
5	2	3.1
6	6	4
7	5	5
8	4	6

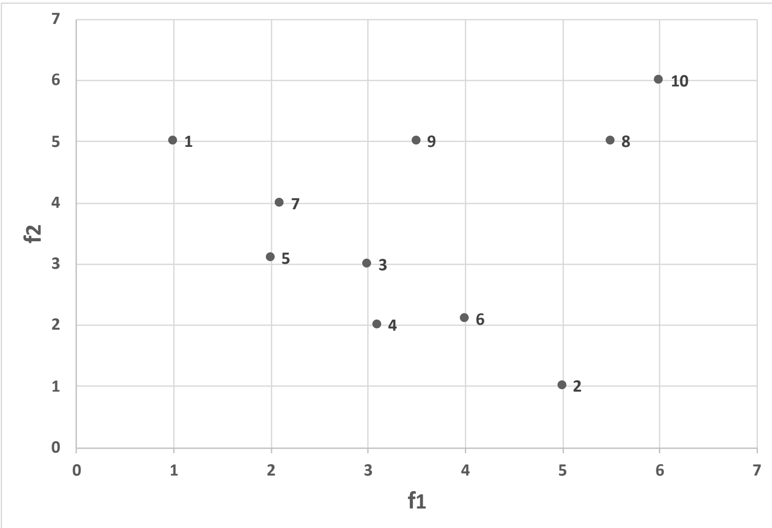

Figure 5.17: Borderline configuration with 10 alternatives.

The result is depicted in Figure 5.18. Again, M-MOBA HV outperforms the other two competitors through the whole process. The PCS-based version of M-MOBA now is even worse than Equal allocation. On the opposite side, Figure 5.19 shows M-MOBA PCS still works much better than others when P(CS) is used as the performance measure and M-MOBA HV works even a lot worse than Equal allocation. To investigate this further, Figure 5.20 shows the percentage of samples allocated to a particular design. M-MOBA PCS allocates quite a few samples to the borderline designs 3, 6 and 7, because it aims to improve the probability of correct selection, and for these designs the classification is most difficult. For a decision maker, however, these designs are probably less relevant. M-MOBA HV instead focuses on the designs 1,2,4 and 5, which are the Pareto optimal solutions probably most relevant to a decision maker. Thus, it creates reliable performance estimates where it is most relevant.

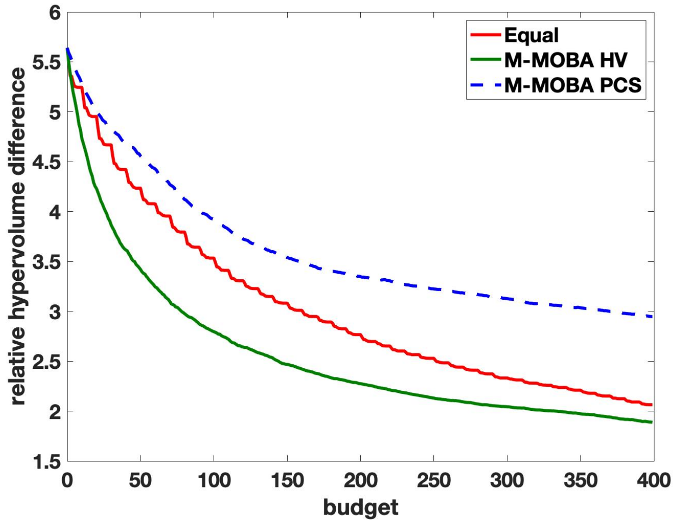

Figure 5.18: Comparison of hypervolume difference of borderline configuration

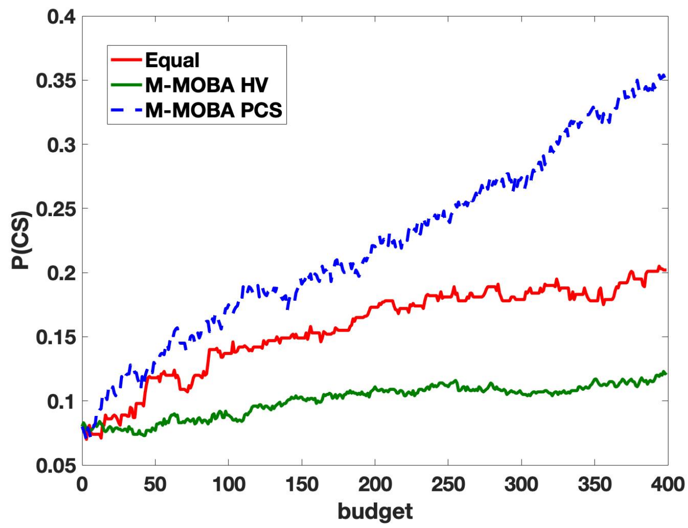

Figure 5.19: Comparison of P(CS) of borderline configuration

The third configuration is designed to show the impact of very similar designs. There are 8 designs, 2 objectives, and the standard deviation of each alternative in

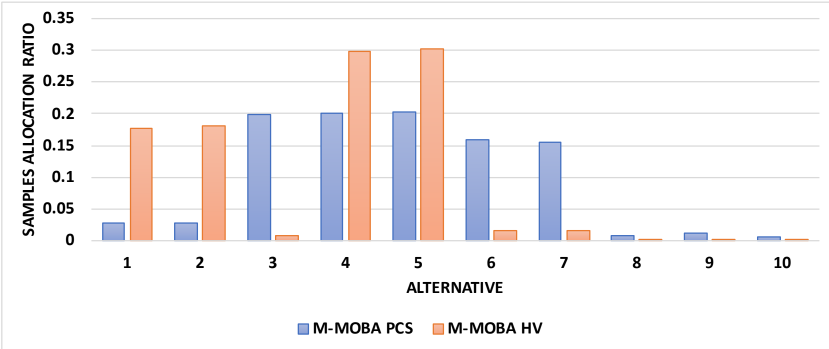

Figure 5.20: Allocation of samples to the different alternatives for borderline configuration

each objective is set to 2. Expected values of each design are reported in Table 5.2, with a visualisation in Figure 5.21. Still, for PCS-based MORS algorithms, it is difficult to distinguish between them. On the other hand, the distinction is probably not very relevant for a decision maker. The results displayed in Figure 5.22 are similar to Configuration 2 in the sense that M-MOBA HV performs best, and the PCS-based M-MOBA is worse than Equal allocation. While M-MOBA PCS still works better than others in correctly selecting the true Pareto optimal solutions shown as Figure 5.23. Again, Figure 5.24 provides further details on the distribution of samples onto the different alternatives.

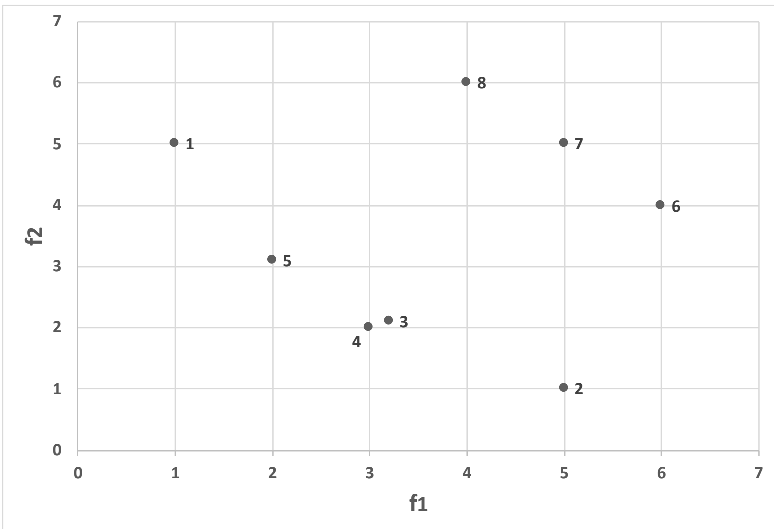

Figure 5.21: Similar solution configuration with 8 alternatives.

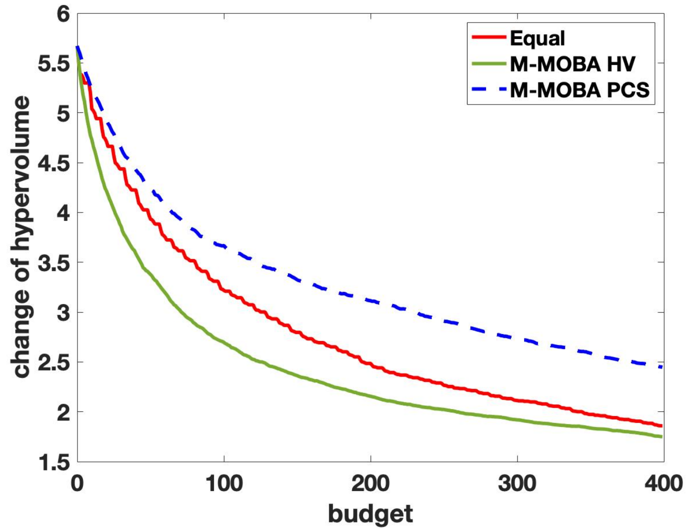

Figure 5.22: Comparison of hypervolume difference of similar solution configuration

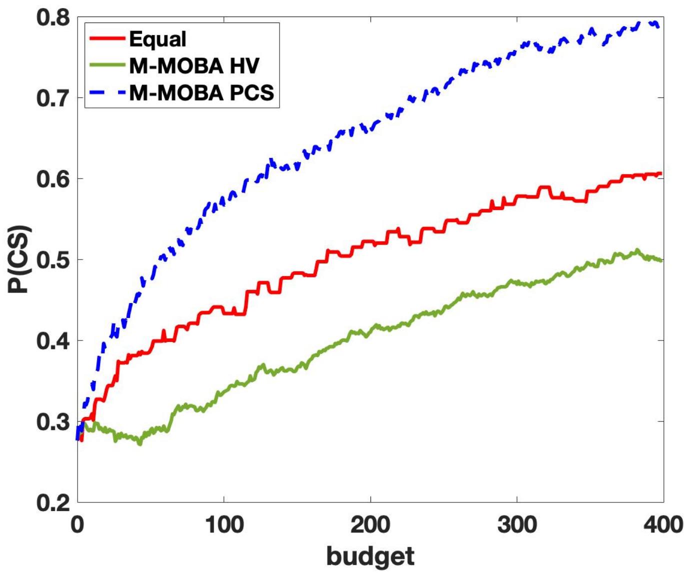

Figure 5.23: Comparison of P(CS) of similar solution configuration

5.6 Conclusion

In this chapter, we proposed a new performance measure for MORS, the hypervolume difference, based on the HV measure that is commonly used to evaluate results in multiobjective optimisation. We also derived a closed form for calculating the

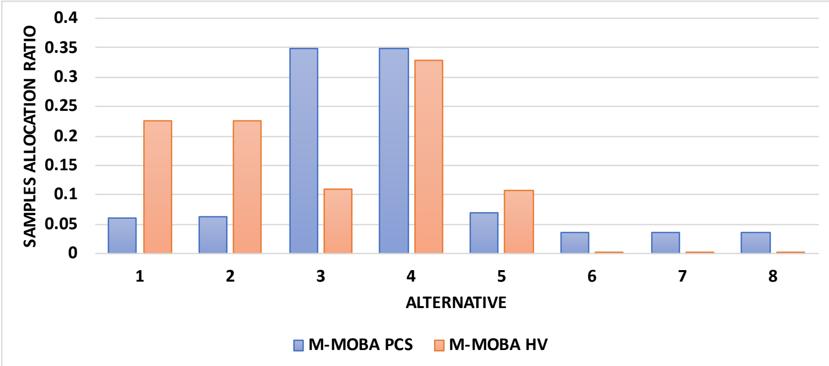

Figure 5.24: Allocation of samples to the different alternatives for similar solution configuration

expected HV change in one cell. To the best our knowledge, this constitutes the first approach based on hypervolume. We argue that the hypervolume difference criterion leads to information collection that is more relevant to a decision maker, taking more samples from the more interesting areas of the Pareto front. Empirical comparisons to Equal allocation and the PCS-based M-MOBA show that the new method indeed works very well with respect to the hypervolume difference performance measure.

Chapter 6

M-MOBA integration into Multi-objective Evolutionary Algorithms

6.1 Introduction

In many real-life applications, a decision maker has several conflicting objectives to consider and wants to determine an optimal tradeoff among them. This problem is called multiobjective optimisation problem (MOP) [Zhang and Li, 2007]. Many evolutionary algorithms (EAs) have been developed for MOPs in recent years [Deb, 2001]. The major advantage of these multiobjective EAs (MOEAs) over other methods is that they work with a population of candidate solutions and thus can produce a set of Pareto-optimal solutions to approximate the Pareto front in a single run [Ke et al., 2013].

Since MORS techniques will avoid wasting simulation samples on irrelevant alternatives, the performance of MOEA will be improved by combing it with MORS algorithms. However, there only few works in this area and one study is the work of Lee et al. [2008] in which MOCBA is integrated into the evaluation procedure to reduce the computing cost for an aircraft spare parts allocation problem.

This chapter considers the multi-objective simulation optimisation problem with infinite or finite but very big search space. We develop a solution framework that can explicitly and efficiently deal with the stochastic noise involved in the performance measures while searching the solution space. The solution framework employs MOEA to search the solution space, and applies the M-MOBA Marginal Hypervolume (M-MOBA MHV) method to efficiently allocate the simulation replications and to identify the non-dominated Pareto set of alternatives. Different from the widely used the NSGA II framework which identifies non-dominated solutions on different fronts sequentially, we use marginal hypervolume as the measurement, i.e., if a solution is Pareto optimal, it will have a positive marginal hypervolume, otherwise, the marginal hypervolume of this solution is zero. As shown in Figure 6.1, where alternatives a, b, c, d have positive marginal hypervolume and e, f have zero marginal hypervolume according to reference point R.

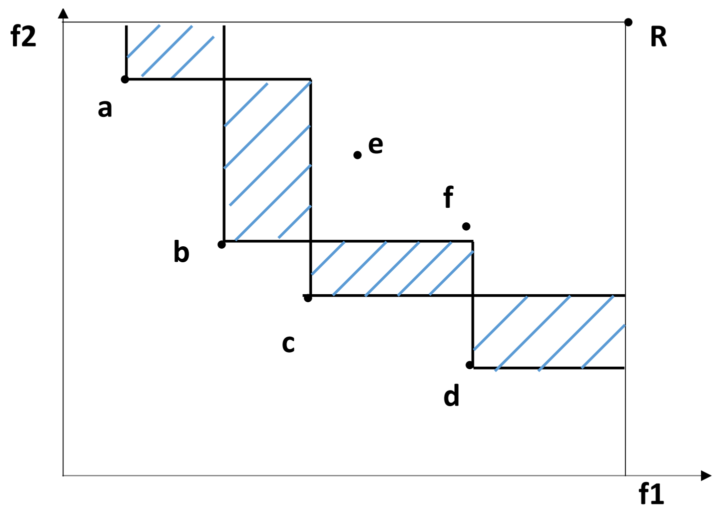

Figure 6.1: Marginal hypervolume for different alternatives

6.2 A multi-objective algorithm integrating MOEA and M-MOBA

This section presents a framework that applies MOEA in conjunction with the M-MOBA. MOEAs are adaptive heuristic search algorithms which simulate the survival of the fittest among individuals over consecutive generations for solving a problem [Lee et al., 2008]. We first generate an initial population randomly. Then, at each generation, MOEA evaluates and ranks alternatives in terms of their fitness. The fitter alternatives will be selected to generate new offspring by recombination and mutation operators. Alternatives with higher fitness values will have a higher probability to be selected into the next generation. This process of evolution is repeated until the algorithm converges to a population which covers the non-dominated solutions.

6.2.1 Coding scheme, Fitness assignment and M-MOBA

We use the real-valued coding scheme to represent a candidate solution as a chromosome. For example, for the ZDT1 problem [Deb and Agrawal, 1999], each chromosome contains 30 decision variables, each of which is known as a gene. We will generate genes randomly, calculate the objective value according to the ZDT1 function and add an i.i.d. normally distributed noise to each objective value whenever the function is called.

The initial performance of the population in this algorithm is the mean value of each alternative after the first stage n_0 sampling and the fitness value of each alternative is its marginal hypervolume based the mean value. We develop a M-MOBA variant named M-MOBA MHV to allocate simulation budget to each alternative, i.e., for each alternative, we will calculate the probability that its marginal hypervolume will exceed the predefined threshhold with one more simulation and only allocate simulation samples to the alternative that has the highest probability. In this study, we set the threshold using solutions' marginal hypervolume as follows. For an instance, if the marginal hypervolume of a set of six solutions is shown as in Table 6.1, for all alternatives with a marginal hypervolume larger than that of D, namely, A, B and C, the threshhold is 3.8 and we will allocate the simulation sample to the alternative that has the largest probability to be smaller than 3.8 with one more sample. Similarly, for D, E and F, which have a marginal hypervolume smaller than 4, and we will allocate the simulation sample to the alternative that has the largest probability to be bigger than 4 with one more sampling.

The motivation of designing the algorithm in this way is that the probability to correctly select the better half is maximised.

As we have not found an analytical method to calculate the expected marginal hypervolume after one more sample, in this study, we construct an empirical distribution function (empirical CDF) for each alternative by using 1000 samples and find the probability of exceeding the threshold accordingly. Figure 6.2 shows an example of an empirical CDF of one alternative.

If we have more than half of alternatives that have a 0 MHV, we just rank them sequentially using their sequential number.

6.2.2 Selection, Crossover and Mutation operator

The standard method to do parent selection is tournament. The probability of being selected is linearly decreasing with rank. Parent selection is done by tournament selection as stated below. First, we rank the whole set of solutions based on every

Table 6.1: Explanation of threshhold.

Alternative	Marginal Hypervolume
A	10
B	7.5
C	4
D	3.8
E	0
F	0

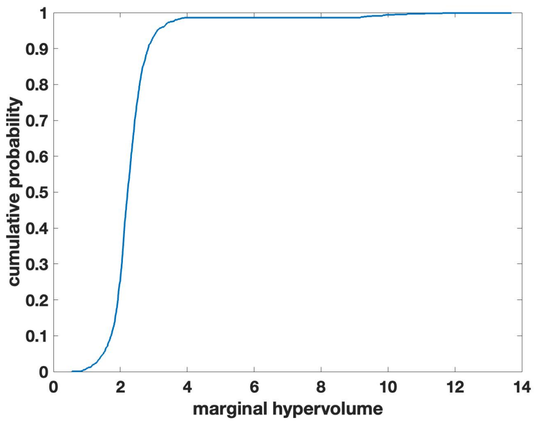

Figure 6.2: An example of empirical CDF

alternative's current marginal hypervolume and construct a random ranked set. Then, we will compare alternatives in pairs and select the one that has a higher marginal hypervolume. For instance, still using the previous example shown in Table 6.1, the selected parents are shown as Table 6.2.

Crossover is a recombination operator that combines two chromosomes (parents) to produce a new chromosome (offspring) [Lee et al., 2008]. The crossover operator we use here is Simulated Binary Crossover (SBX) [Deb, 2001] since simulation results show that real-coded GAs with the SBX operator can overcome the Hamming cliff problem, precision problem, and fixed mapping problem in difficult problems [Deb and Agrawal, 1995]. SBX simulates the binary crossover observed in nature and is given as below [Seshadri, 2006].

$$c_{1,k} = 0.5[(1 - \beta_k)p_{1,k} + (1 + \beta_k)p_{2,k}], and \\ c_{2,k} = 0.5[(1 + \beta_k)p_{1,k} + (1 - \beta_k)p_{2,k}]; \tag{6.1}$$

where $c_{i,k}$ is the ith child with kth component, $p_{i,k}$ is the selected parent and

Table 6.2: Parents selection.

Alternative	MHV	Alternative	MHV	Selected Parent
A	10	B	7.5	A
B	7.5	F	0	B
C	4	E	0	C
D	3.8	A	10	A
E	0	C	4	C
F	0	D	3.8	D

$\beta_k (k \geq 0)$ is a sample from a random number generated having the density

$$\begin{aligned} p(\beta) &= 0.5(\eta_c + 1)\beta^{\eta_c}, \quad \text{if} \quad 0 \leq \beta \leq 1 \\ p(\beta) &= 0.5(\eta_c + 1)\frac{1}{\beta^{\eta_c+2}}, \quad \text{if} \quad \beta > 1. \end{aligned} \tag{6.2}$$

This distribution can be obtained from a uniformly sampled random number u between (0, 1). η_c is the distribution index for crossover. Then

$$\begin{aligned} \beta(u) &= (2u)^{\frac{1}{\eta+1}}, \quad \text{if} \quad u < 0.5 \\ \beta(u) &= \frac{1}{[2(1-u)]^{\frac{1}{\eta+1}}}, \quad \text{if} \quad 0.5 \leq u. \end{aligned} \tag{6.3}$$

Polynomial mutation [Deb and Agrawal, 1999] is done for all children.

$$c_k = p_k + (p_k^u - p_k^l)\delta_k, \tag{6.4}$$

where c_k is the δth child component and p_k is the δth parent component with p_k^u being the upper bound on the parent component, p_k^l the lower bound and δ_k a small variation which is calculated from a polynomial distribution by using

$$\begin{aligned} \delta_k &= (2r_k)^{\frac{1}{\eta_m+1}} - 1, \quad \text{if} \quad r_k < 0.5 \\ \delta_k &= 1 - [2(1-r_k)]^{\frac{1}{\eta_m+1}}, \quad \text{if} \quad 0.5 \leq r_k \end{aligned} \tag{6.5}$$

where r_k is uniformly sampled random number between (0,1) and η_m is the mutation distribution parameter.

6.2.3 Elite population

When children has been created by crossover and mutation, the new population is formed by combining children and parents. M-MOBA will then be used again to rank the new population. An elite population is obtained by selecting the top half solutions based on their marginal hypervolume.

6.2.4 Determination of the final non-dominated Pareto set and performance measure

After the maximum number of generations has been reached, a set of solutions is returned. Here, we use utility function as the performance measure. Assume we have Q utility functions and R Pareto optimal solutions, for each pre-defined $\lambda_q \in [0 : 1]$, we will calculate the observed utility for each solution r according to the linear utility function of

$$u_{r_q} = \lambda_q f_{r1} + (1 - \lambda_q) f_{r2} \tag{6.6}$$

where u_{r_q} is the utility of solution r using λ_q and f_{r_n} is the nth objective value for each solution r. We then select the alternative a_q which has the smallest u_{r_q} and form a new set of $\mathbf{A}$ that has Q elements. Add up the true utility of the selected solutions

$$U = \sum_{q=1}^{Q} (\lambda_q f_{ta_{1_1}} + (1 - \lambda_q) f_{ta_{1_2}}) \tag{6.7}$$

where $a_q \in \mathbf{A}$ and $f_{ta_{qn}}$ is the true nth objective value for each solution a_q.

The true utility U is set as the performance measure. Utility function is used as the performance measure because the true Pareto optimal set is unknown, which makes it difficult to use PCS or HVD as performance measure.

6.3 Algorithm

The flow chart of the multi-objective simulation optimisation framework is shown in Figure 6.3. The framework is a standard MOEA process except M-MOBA is used in the fitness evaluation process to improve the efficiency. The pseudo code of the algorithm is summarised as Algorithm 4. DM need to specify a maximum generation number T, a first-stage sample size n_0, and a number of samples $\tau > 0$ to allocate per subsequent stage. Then performs the MOEA procedure with M-MOBA until the final generation reaches. Finally select the elite solutions and evaluate the selection by DM defined tchebycheff utility function.

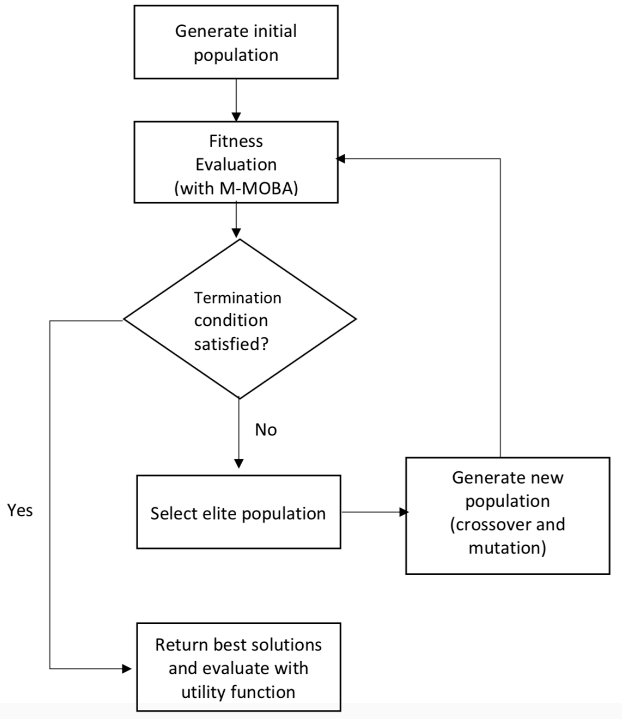

Figure 6.3: Flow chart of the multi-objective simulation optimisation framework.

ALGORITHM 4: MOEA with M-MOBA integrated

1: Specify a maximum generation number T, a first-stage sample size $n_0 \geq 5$, and a number of samples $\tau > 0$ to allocate per subsequent stage.
2: Randomly generate an initial feasible population; set generation index $t = 0$.
3: **while** $t \leqslant T$ **do**
4: Evaluate the population based on the marginal hypervolume; run M-MOBA to determine the number of replications for each design in population.
5: Select parents based on every alternative's current marginal hypervolume.
6: Generate children by performing crossover and mutation using Equation 6.1-6.5; form the new population by combining children and parents.
7: **end while**
8: Select elite solutions and evaluate the selection by using tchebycheff utility function.

6.4 Numerical experiments

We test the proposed M-MOBA MHV algorithm with the ZDT1 and ZDT2 problems as they are two typical test problems for MOEA. Zitzler et al. [2000] suggested a set of benchmark functions that have been extensively used in the literature for the analysis and comparison of multi-objective EAs. These problems have properties that are known to cause difficulties in converging to the true Pareto-optimal front and reflect characteristics of real-world problems, such as multimodality, non-separability, and high dimensionality. Each of the test functions defined is structured in the same manner and consists itself of three functions f_1, g, h [Deb and Agrawal, 1999]:

$$\begin{aligned} &\text{Minimise} && T(\mathbf{x}) = (f_1(x_1), f_2(\mathbf{x})) \\ &\text{subject to} && f_2(\mathbf{x}) = g(x_2, ..., x_m)h(f_1(x), g(x_2, ..., x_m)) \\ &\text{where} && \mathbf{x} = (x_1, ..., x_m) \end{aligned} \tag{6.8}$$

The function f_1 is a function of the first decision variable only, g is a function of the remaining $m-1$ variables, and the parameters of h are the function values of f_1 and g.

ZDT1 test function has a convex Pareto-optimal front:

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2, ..., x_m) &= 1 + 9 * \sum_{i=2}^{m} x_i/(m-1) \\ h(f_1, g) &= 1 - \sqrt{f_1/g} \end{aligned} \tag{6.9}$$

where $m = 30$ and $x_i \in [0, 1]$. The Pareto-optimal front is formed with $g(x) = 1$.

The Pareto front (with population of 100) of ZDT1 is shown in Figure 6.4.

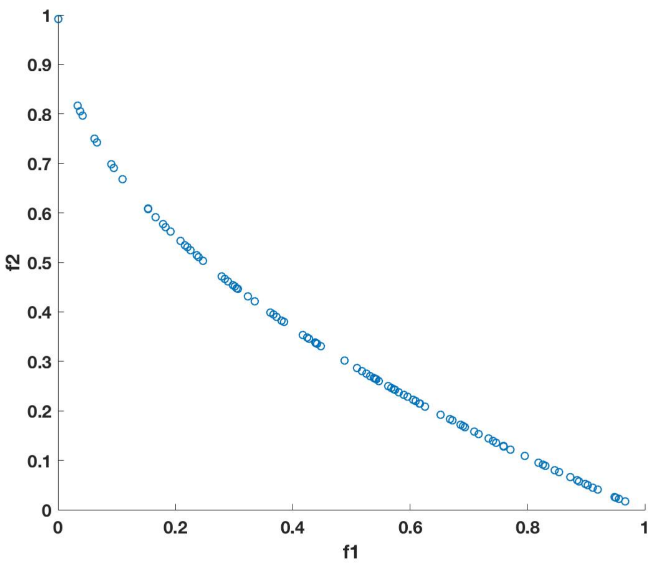

Figure 6.4: Pareto front of ZDT1 function

ZDT2 test function is a nonconvex Pareto-optimal front:

$$
\begin{aligned}
&f_1(x_1) = x_1 \\
&g(x_2, ..., x_m) = 1 + 9 * \sum_{i=2}^{m} x_i/(m-1) \\
&h(f_1, g) = 1 - (f_1/g)^2
\end{aligned}
\tag{6.10}
$$

where $m = 30$ and $x_i \in [0, 1]$. The Pareto-optimal front is formed with $g(x) = 1$.

The Pareto front (with population of 100) of ZDT2 is shown in Figure 6.5.

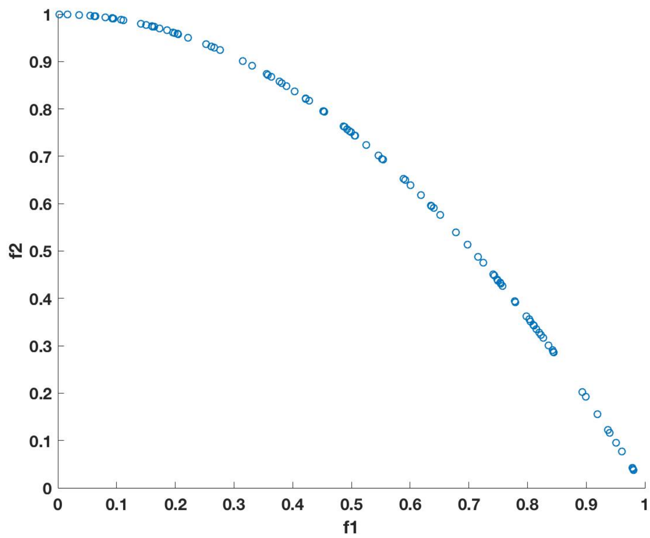

Figure 6.5: Pareto front of ZDT2 function

The algorithm parameters are set as follows. The population size is set as 10, total number of simulation samples in each generation is 20, total generation is 30, the distribution index for crossover η_c is and the mutation distribution parameter η_m are both set as 20. λ is set as 0.1 to 0.9 with a step of 0.1.The reference point is set based on the initial observed value of each alternative. Generally, the value of reference point should be large enough to cover all solutions.

We test M-MOBA MHV by comparing its performance with Equal allocation, M-MOBA PCS and M-MOBA HV. We use the above methods in the ranking and selection stage of MOEA as Figure 6.3 shows and evaluate the true linear utility of the selected solutions. The test result of using ZDT1 is shown in Figure 6.6. M-MOBA HV performs very well in selecting the true best alternatives. It saves about 500 simulation replications to achieve the utility of 13 compared with the

Equal allocation. In contrast, M-MOBA MHV and M-MOBA PCS exhibit similar convergence patterns which is just slightly better than the naive Equal allocation.

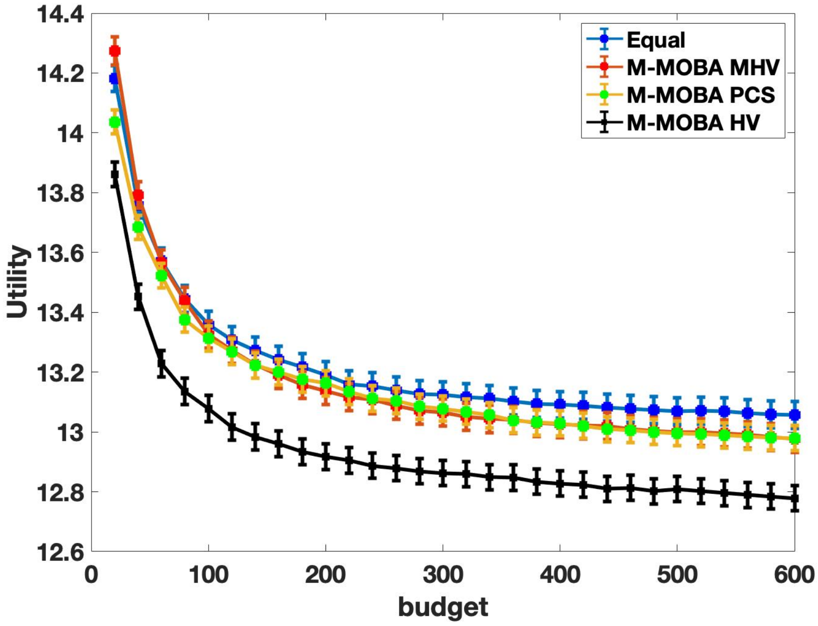

Figure 6.6: Test result of using ZDT1 function

Similarly, the test result of using ZDT2 function displayed in Figure 6.7 also indicates that M-MOBA HV is the best among these algorithms. Again, it saves around 500 samples to achieve the utility of 13 compared with the Equal allocation and the difference between Equal allocation, M-MOBA MHV and M-MOBA PCS is not obvious.

The poor performance of M-MOBA MHV is a little surprising as this algorithm is designed to maximise probability that solutions with maximised MHV are selected and it should return the alternatives with the largest marginal hypervolume which should be the Pareto optimal solutions as well since only alternatives on the Pareto front have positive marginal hypervolume. One potential reason why M-MOBA MHV does not perform as well as expected maybe that M-MOBA MHV focuses too much on solutions near the threshold of e.g., solution C and D in Table 6.1 and it does not evaluate solutions with upper or lower values well. As a result, the probability of selecting wrong parents in each generation is high. In order to test whether this conjecture is true or not, we select parents randomly rather than using the tournament selection described above in the ZDT1 function and the result is shown in Figure 6.8. It can be found that although M-MOBA HV

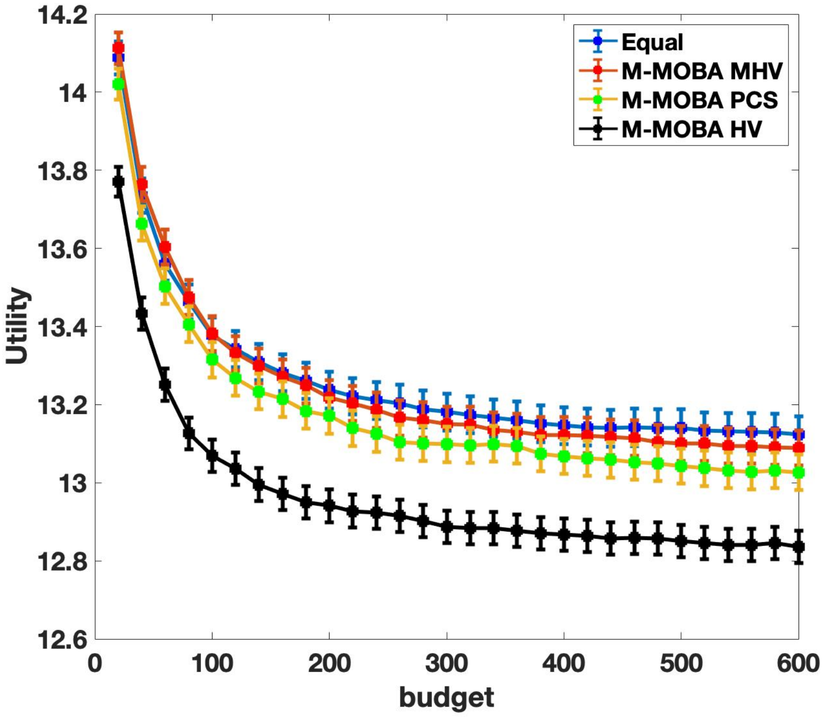

Figure 6.7: Test result of using ZDT2 function

still works best among all algorithms, the difference between M-MOBA MHV and M-MOBA HV as well as M-MOBA PCS and M-MOBA HV becomes apparently smaller. When 600 simulation replications are used, the utility difference between M-MOBA HV and M-MOBA MHV is around 0.1, smaller than the 0.3 difference between M-MOBA HV and M-MOBA MHV in Figure 6.6. Both M-MOBA MHV and M-MOBA PCS work better compared to their previous edition. For example, when using 300 simulation replications, M-MOBA HV and M-MOBA PCS can only reach a utility of around 13.2 but they are able to achieve less than 13 and around 12.8 respectively. This performance difference indicates that M-MOBA MHV does not do well in recognising the right parents.

We further explore whether we can improve the performance M-MOBA MHV by changing the algorithm of the final generation, i.e., using M-MOBA HV rather than M-MOBA MHV in the final generation as we suppose M-MOBA HV will help to identify the true Pareto optimal solutions and estimate their true fitness values more accurately. Figure 6.9 exhibits test result and it can be seen that M-MOBA HV still outperforms others while there is no obvious difference between M-MOBA

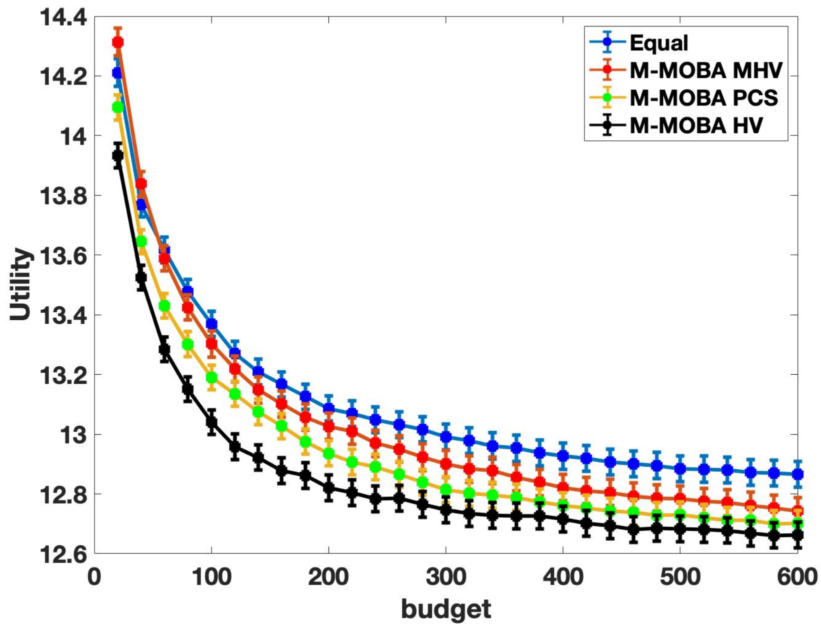

Figure 6.8: Test result of using random parents in ZDT1 problem

MHV's performance comparing with the result in Figure 6.6.

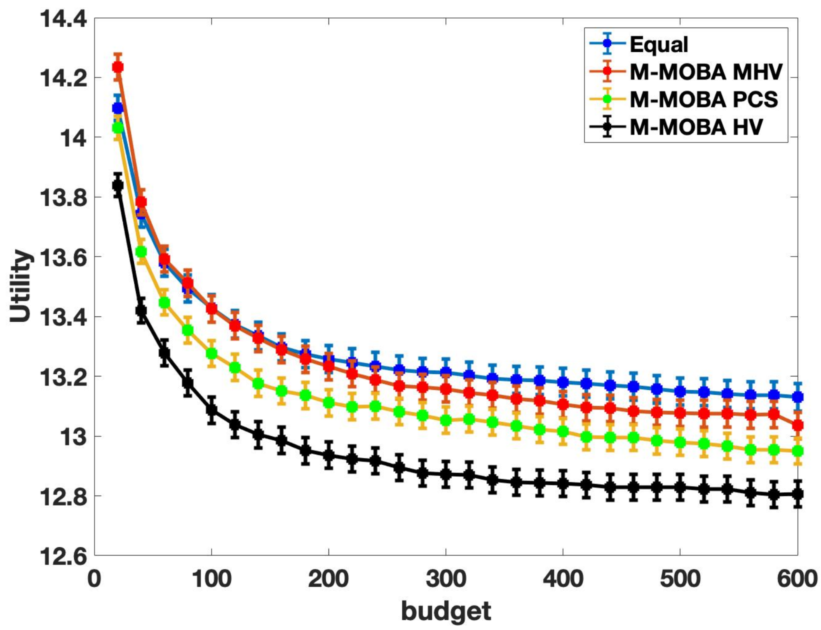

Figure 6.9: Test result of changing the final generation algorithm in ZDT1 problem

6.5 Conclusion

In this chapter, we explored how to integrate M-MOBA into an MOEA to improve the efficiency of allocating simulation replications. We designed a new M-MOBA MHV method and compare its performance with Equal allocation, M-MOBA PCS and M-MOBA HV. Empirical results indicate that M-MOBA HV works much better than others in both ZDT1 and ZDT2 problems. The performance of Equal allocation, M-MOBA PCS and M-MOBA MHV is quite similar. We suppose one potential reason why M-MOBA MHV does not work as well as expected may be that M-MOBA MHV focuses too much on solutions near the threshold. As a result, the probability of selecting wrong parents high. We then change the scheme of parents selection and test results show that although M-MOBA HV still works best among all algorithms, the difference between M-MOBA MHV and M-MOBA HV as well as M-MOBA PCS and M-MOBA HV becomes apparently smaller. This performance difference implies that M-MOBA MHV does not do well in recognising the right parents.

Chapter 7

Conclusions and Future work

7.1 Summary of findings

In this thesis, we explored the problem of ranking and selection with multiple-objectives in the presence of uncertainty, especially on the bi-objective case. In view of the practical needs within a multi-objective simulation optimisation context, the desired systems are several Pareto-optimal (efficient) solutions with different trade-offs which cannot be improved in any objective without sacrificing performance in another objective. For the case where alternatives are evaluated on multiple stochastic criteria, and the performance of an alternative can only be estimated via simulation, we consider the problem of efficiently identifying the Pareto optimal designs out of a (small) given set of alternatives.

We developed a myopic budget allocation algorithm M-MOBA and proposed several variants for different settings. This myopic method is a multi-objective extension of the single objective small EVI procedure proposed by Chick et al. [2010]. In particular, it only allocates one simulation sample to one alternative in each iteration. We firstly developed a PCS-based M-MOBA which aims at finding the whole set of Pareto optimal solutions correctly. PCS is the most common performance measure in MORS as the initial objective of ranking and selection in practice is to select the true Pareto optimal solutions and other typical measures such as opportunity cost are not easily extended to multiple objectives. Numerical experiments illustrate that the proposed M-MOBA PCS procedure can generally obtain a significantly higher empirical probability of correct selection than the naive Equal allocation scheme, and perform as well as the existing MOCBA method in a simple 3 alternatives configuration and even better than MOCBA in the 16 alternatives configuration. In other words, to reach the same pre-specified probability of correct selection, M-MOBA PCS can save a substantial amount of computing budget compared to alternative allocations.

We then considered the scenario that in reality, sometimes objectives can be evaluated independently, e.g., in case where different simulation models are used to evaluate different objectives. For instance, when testing a vehicle's performance, different simulations are conducted to test different parameters including fuel consumption, emission, etc. In this case, in order to further improve the efficiency of sampling, it is possible to regard the sampling allocation process for each objective independently. Instead of evaluating all objectives of an alternative simultaneously as in the M-MOBA PCS procedure, we developed a M-MOBA PCS variant, M-MOBA DS PCS, which will evaluate only one objective of one alternative in each iteration. Empirical results show that by allowing to evaluate objectives independently, the efficiency of the algorithm can be further improved. We then explored how to tackle the problem when similar performance solutions exist. In practice, some systems may have very similar objective values and a DM might not be too concerned with small differences between these systems, hence we should treat these designs as equally acceptable [Teng et al., 2010]. Furthermore, if the difference is very small, even a large number of samples would not allow us to decide with confidence which system is better. We introduce a new definition of indifference zone and PGS for multi-objective problems which overcome the shortage of the existing MOCBA indifference zone definition. Numerical experiments show that the proposed M-MOBA IZ procedure performs much better than M-MOBA PCS and Equal allocation when the performance measure is PGS.

Another contribution of this study is that we proposed a new performance measure for MORS, the hypervolume difference, based on the HV measure that is commonly used to evaluate results in multiobjective optimisation [Emmerich and Klinkenberg, 2008]. As far as we know, this paper is the first attempt to use hypervolume as the performance measure to the MORS problem. We also derived a closed form calculation of the expected HV change in one cell. Unlie most publications for multi-objective problems aiming to maximising the probability of correctly selecting all Pareto optimal solutions, we suggest minimising the difference in hypervolume between the observed means of the perceived Pareto front and the true Pareto front as a new performance measure. We argue that this hypervolume difference is often more relevant for a decision maker as described in Chapter 5. Empirical tests show that the proposed M-MOBA HV method performs well with respect to the stated hypervolume objective in different settings.

We also explored how to integrate M-MOBA into an MOEA to improve the efficiency of allocating simulation replications. We designed a new M-MOBA MHV method and compare its performance with Equal allocation, M-MOBA PCS and M-MOBA HV by using some typical MOEA test configurations. Empirical results imply that M-MOBA HV outperforms other competitors substantially through the whole generations. The performance of Equal allocation, M-MOBA PCS and M-MOBA MHV is quite similar. In order to find reasons of this unexpected result, we change the scheme of parents selection of M-MOBA MHV. Although M-MOBA HV still beats other algorithms, the difference between M-MOBA MHV and M-MOBA HV as well as M-MOBA PCS and M-MOBA HV becomes apparently smaller. This performance difference indicates that M-MOBA MHV does not do well in recognising the right parents. Further research of how to improve M-MOBA MHV's performance is deserved to conduct in the future.

In conclusion, M-MOBA is a powerful algorithm to tackle the MORS problem that can be adapted relatively easily to different problem settings. We suggest different M-MOBA variants are used in different situations according to Table 7.1.

Table 7.1: Different methods for different scenarios.

Scenario	Method	Explanation
PCS is the objective	M-MOBA PCS	M-MOBA PCS is the chosen method if PCS is the objective
Small differences are irrelevant to the DM	M-MOBA IZ	M-MOBA IZ is used if the configuration contains similar alternatives and PGS is the objective
Different simulation models are used to evaluate different criteria	M-MOBA DS	M-MOBA DS is valuable when the simulation budget is quite limited and the objectives can be evaluated independently
A good approximation of the Pareto front is desired	M-MOBA HV	M-MOBA HV is preferred if the DM's main purpose is to minimise the HVD between the true and selected Pareto front

7.2 Limitations of this research and future work

Being an exploratory study, there are a number of limitations in this research inevitably. Due to time limitations, some problems have not been solved during this

PhD study. In the following, we will discuss these limitations and propose potential solutions accordingly.

1. This research aims to solve MORS problems but the current research is limited to bi-objective. A developed M-MOBA to solve more than two objectives is the primary potential work for future research and it will help to make this method more universal. The difficulty of extending this algorithm to more than two objectives is that current analysis relies a lot of on graphing and when more than two objectives are considered, it is not straightforward to analyse. Potential solutions could be more advanced graphical model analysis techniques, new methods that do not rely on graphing, etc., or using Monte Carlo sampling, but this would be computationally expensive.

2. To test M-MOBA on real-world simulation optimisation problems will be interesting. Lee et al. [2008] have employed MOEA to search the solution space and applied MOCBA to efficiently allocate the simulation replications and to identify the non-dominated Pareto set of solutions in a real aircraft spare parts allocation problem. Similarly, a practical test of M-MOBA in a real world problem can be conducted in the future.

3. Other M-MOBA variants with different stopping rules rather than fixed budget could be developed. In this research, the stopping rule for all M-MOBA variants has been a fixed budget, i.e., the algorithm stops as soon as the simulation budget has been used up. However, it would be straightforward to change the stopping rule to be more flexible. For instance, we can use the same approximation for Type I and Type II error as Chen and Lee [2010] and stop when the predefined error level has been reached.

4. It could be desirable to further inspect the effect of reference point chosen. M-MOBA HV and M-MOBA MHV both assume that the reference point is large enough to be dominated by all solutions. However, a large reference point will affect the algorithm's performance as the larger the reference point is, the larger the marginal hypervolume of the extreme solutions (i.e., solutions on the ends of the Pareto front) are and as a result, the hypervolume difference brought by extreme solutions when simulation replications are allocated is larger as well which forces the algorithm to focus more on these solutions. Therefore, how to select a proper value for the reference point is worth to investigate. Further, the choice of the reference point may also help to focus on a specific part of the Pareto front if the DM has a pre-specified preference.

5. The M-MOBA allocation developed for integration into MOEAs has not performed as well as the M-MOBA-HV that was developed for a different purpose. There should thus still be scope to further developing an M-MOBA variant specifically for the selection phase of MOEAs.

In summary, although different variants of M-MOBA have been discussed in this research, a few of potential extensions can be envisaged to the proposed approaches in the future.

Bibliography

KA. Allman, J. Laurito, and M. Loh. *Financial Simulation Modeling in Excel: a step-by-step guide*. John Wiley & Sons, 2011.

S. Andradottir and S.-H. Kim. Fully sequential procedures for comparing constrained systems via simulation. *Naval Research Logistics*, 57(5):403–421, 2010.

J. Bader and E. Zitzler. Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19(1):45–76, 2011.

DK. Badyal, V. Modgill, and J. Kaur. Computer simulation models are implementable as replacements for animal experiments. *Alternatives to laboratory animals: ATLA*, 37(2):191–195, 2009.

José M Bernardo and Adrian FM Smith. Bayesian theory. 1994. *John Willey and Sons. Valencia (España)*, 1994.

N. Beume, B. Naujoks, and M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 161 (3):1663–1669, 2007.

M. Birattari, Z. Yuan, P. Balaprakash, and T. Stützle. F-race and iterated f-race: An overview. In *Experimental methods for the analysis of optimization algorithms*, pages 311–336. Springer, 2010.

P.A. Bosman and D. Thierens. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7(2):174–188, 2003.

SC. Brailsford. Advances and challenges in healthcare simulation modeling: tutorial. In *Winter Simulation Conference*, pages 1436–1448. IEEE, 2007.

J. Branke and J. Gamer. Efficient sampling in interactive multi-criteria selection. In *INFORMS Simulation Society Research Workshop*, pages 42–46, 2007.

J. Branke and W. Zhang. A new myopic sequential sampling algorithm for multi-objective problems. In *Winter Simulation Conference*, pages 3589–3598. IEEE, 2015.

J. Branke, S.E. Chick, and C. Schmidt. New developments in ranking and selection: an empirical comparison of the three main approaches. In *Winter Simulation Conference*, pages 708–717. IEEE, 2005.

J. Branke, S.E. Chick, and C. Schmidt. Selecting a selection procedure. *Management Science*, 53(12):1916–1932, 2007.

J. Branke, W. Zhang, and Y. Tao. Multiobjective ranking and selection based on hypervolume. In *Winter Simulation Conference*, pages 859–870. IEEE, 2016.

J. Butler, D.J. Morrice, and P.W. Mullarkey. A multiple attribute utility theory approach to ranking and selection. *Management Science*, 47(6):800–816, 2001.

X. Cai, Y. Li, Z. Fan, and Q. Zhang. An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization. *IEEE Transactions on Evolutionary Computation*, 19(4):508–523, 2015.

A. Cervantes, D. Quintana, and G. Recio. Efficient dynamic resampling for dominance-based multiobjective evolutionary optimization. *Engineering Optimization*, 49(2):1–17, 2016.

M. Chau, M.C. Fu, H. Qu, and IO. Ryzhov. Simulation optimization: a tutorial overview and recent developments in gradient-based methods. In *Winter Simulation Conference*, pages 21–35. IEEE, 2014.

C.-H. Chen. A lower bound for the correct subset-selection probability and its application to discrete-event system simulations. *Automatic Control, IEEE Transactions on*, 41(81):1227–1231, 1996.

C.-H. Chen and L. H. Lee. *Stochastic Simulation Optimization: An Optimal Computing Budget Allocation.* World Scientific, 2010.

C.-H. Chen, D. He, and M. Fu. Efficient simulation budget allocation for selecting an optimal subset. *INFORMS Journal on Computing*, 20(4):579–595, 2008.

R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff. A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 20(5):773–791, 2016.

S.E. Chick and K. Inoue. New two-stage and sequential procedures for selecting the best simulated system. *Operations Research*, 49(5):732–743, 2001.

S.E. Chick and Y. Wu. Selection procedures with frequentist expected opportunity cost bounds. *Operations Research*, 53(5):867–878, 2005.

S.E. Chick, J. Branke, and C. Schmidt. Sequential sampling to myopically maximize the expected value of information. *INFORMS Journal on Computing*, 22(1):71–80, 2010.

F. Chingcuanco and C. Osorio. A procedure to select the best subset among simulated systems using economic opportunity cost. In *Winter Simulations Conference*, pages 452–562. IEEE, 2013.

D.W Corne, N.R Jerram, J.D Knowles, and M.J Oates. Pesa-ii: Region-based selection in evolutionary multiobjective optimization. In *Genetic and Evolutionary Computation Conference*, pages 283–290. Morgan Kaufmann Publishers Inc., 2001.

K. Deb. *Multi-objective optimization using evolutionary algorithms.* John Wiley & Sons, 2001.

K. Deb and RB. Agrawal. Simulated binary crossover for continuous search space. *Complex systems*, 9(2):115–148, 1995.

K. Deb and S. Agrawal. A niched-penalty approach for constraint handling in genetic algorithms. In *International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA-99)*, pages 235–243, 1999.

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.

M.H. DeGroot. *Optimal Statistical Decisions*, volume 82. John Wiley & Sons, 2005.

Morris H DeGroot and Mark J Schervish. *Probability and statistics.* Pearson Education, 2012.

E.J. Dudewicz and V.S. Taneja. Multivariate ranking and selection without reduction to a univariate problem. In *Winter Simulation Conference*, pages 207–210. IEEE, 1978.

E.J. J Dudewicz and V.S. S Taneja. A multivariate solution of the multivariate ranking and selection problem. *Communications in Statistics-Theory and Methods*, 10 (18):1849–1868, 1981.

B. El-Haik and R. Al-Aomar. *Simulation-based lean six-sigma and design for six-sigma*. John Wiley & Sons, 2006.

M. Emmerich and J.W. Klinkenberg. The computation of the expected improvement in dominated hypervolume of pareto front approximations. *Rapport technique, Leiden University*, 34, 2008.

Z. Fan, K. Hu, F. Li, Y. Rong, W. Li, and H. Lin. Multi-objective evolutionary algorithms embedded with machine learning—a survey. In *Evolutionary Computation (CEC), 2016 IEEE Congress on*, pages 1262–1266, 2016.

G. Feldman, S. R. Hunter, and R. Pasupathy. Multi-objective simulation optimization on finite sets: optimal allocation via scalarization. In *Winter Simulation Conference*, pages 3610–3621. IEEE, 2015.

P.I. Frazier. A fully sequential elimination procedure for indifference-zone ranking and selection with tight bounds on probability of correct selection. *Operations Research*, 62(4):926–942, 2014.

P.I. Frazier and K. Kazachkov. Guessing preferences: A new approach to multi-attribute ranking and selection. In *Winter Simulation Conference*, pages 4319–4331. IEEE, 2011.

P.I. Frazier, W.B. Powell, and S. Dayanik. A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization*, 47(5): 2410–2439, 2008.

M. Fu, J.-Q. Hu, C.-H. Chen, and X. Xiong. Simulation allocation for determining the best design in the presence of correlated sampling. *INFORMS Journal on Computing*, 19(1):101–111, 2007.

M.C. Fu, C.-H. Chen, and L. Shi. Some topics for simulation optimization. In *Winter Simulation Conference*, pages 27–38. IEEE, 2008.

D. He, S.E. Chick, and C.-H. Chen. The opportunity cost and ocba selection procedures in ordinal optimization for a fixed number of alternative systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(5):951–961, 2007.

S. R. Hunter and G. Feldman. Optimal sampling laws for bi-objective simulation optimization on finite sets. In *Winter Simulation Conference*, pages 3749–3757. IEEE, 2015.

S.R. Hunter and R. Pasupathy. Optimal sampling laws for stochastically constrained simulation optimization on finite sets. *INFORMS Journal on Computing*, 25(3): 527–542, 2013.

H. Ishibuchi, N. Tsukamoto, and Y. Nojima. Evolutionary many-objective optimization: A short review. In *IEEE Congress on Evolutionary Computation.*, pages 2419–2426. IEEE, 2008.

N. Jian, D. Freund, H.M Wiberg, and S.G Henderson. Simulation optimization for a large-scale bike-sharing system. In *Winter Simulation Conference*. IEEE, 2016.

L. Jiao, H. Wang, R. Shang, and F. Liu. A co-evolutionary multi-objective optimization algorithm based on direction vectors. *Information Sciences*, 228:90–112, 2013.

J.A Joines and S.D Roberts. *Simulation modeling with SIMIO: a workbook*. Simio LLC Sewickley, PA, 2013.

Liangjun Ke, Qingfu Zhang, and Roberto Battiti. Moea/d-aco: A multiobjective evolutionary algorithm using decomposition and antcolony. *IEEE transactions on cybernetics*, 43(6):1845–1859, 2013.

S. Kim and J.-H. Ryu. The sample average approximation method for multi-objective stochastic optimization. In *Winter Simulation Conference*, pages 4026–4037, 2011.

S.-H. Kim and B.L. Nelson. Selecting the best system. *Handbooks in operations research and management science*, 13:501–534, 2006.

Seong-Hee Kim and Barry L Nelson. Recent advances in ranking and selection. In *Winter Simulation Conference*, pages 162–172. IEEE, 2007.

L.W. Koenig and A.M. Law. A procedure for selecting a subset of size m containing the l best of k independent normal populations, with applications to simulation. *Communications in Statistics-Simulation and Computation*, 14(3):719–734, 1985.

L. H. Lee, E. P. Chew, and S. Teng. Finding the pareto set for multi-objective simulation models by minimization of expected opportunity cost. In *Winter Simulation Conference*, pages 513–521, 2007.

L. H. Lee, E. P. Chew, and S. Teng. Computing budget allocation rules for multi-objective simulation models based on different measures of selection quality. *Automatica*, 46(12):1935–1950, 2010a.

L. H. Lee, E. P. Chew, S. Teng, and D. Goldsman. Finding the non-dominated pareto set for multi-objective simulation models. *IIE Transactions*, 42(9):656–674, 2010b.

L.H. Lee, E.P. Chew, S. Teng, and Y. Chen. Multi-objective simulation-based evolutionary algorithm for an aircraft spare parts allocation problem. *European Journal of Operational Research*, 189(2):476–491, 2008.

S. Lee and B.L. Nelson. Computational improvements in bootstrap ranking & selection procedures via multiple comparison with the best. In *Winter Simulation Conference*, pages 3758–3767. IEEE, 2015.

B. Li, J. Li, K. Tang, and X. Yao. Many-objective evolutionary algorithms: A survey. *ACM Computing Surveys (CSUR)*, 48(1):13, 2015.

H. Li and Q. Zhang. Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii. *IEEE Transactions on Evolutionary Computation*, 13 (2):284–302, 2009.

K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang. Stable matching-based selection in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 18(6):909–923, 2014.

H. Liu, F. Gu, and Q. Zhang. Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. *IEEE Transactions on Evolutionary Computation*, 18(3):450–455, 2014.

Mónica Lomelí-Rodríguez, Martín Rivera-Toledo, and José A López-Sánchez. Process intensification of the synthesis of biomass-derived renewable polyesters: reactive distillation and divided wall column polyesterification. *Industrial & Engineering Chemistry Research*, 56(11):3017–3032, 2017.

G. Marceau-Caron and M. Schoenauer. Racing multi-objective selection probabilities. In *Parallel Problem Solving from Nature (PPSN)*, number 8672 in LNCS, pages 631–640. Springer, 2014.

V. Mattila and K. Virtanen. Ranking and selection for multiple performance measures using incomplete preference information. *European Journal of Operational Research*, 242:568–579, 2015.

K. Miettinen. *Nonlinear Multiobjective Optimization*, volume 12. Springer Science & Business Media, 2012.

G.A. Morales, R.S. Morales, C.F. Valencia, and R. Akhavan-Tabatabaei. A forest fire propagation simulator for bogotá. In *Winter Simulations Conference*, pages 1505–1515. IEEE, 2014.

D.J. Morrice and J.C. Butler. Ranking and selection with multiple targets. In *Winter Simulation Conference*, pages 222–230. IEEE, 2006.

D.J. Morrice, J. Butler, and P.W. Mullarkey. An approach to ranking and selection for multiple performance measures. In *Winter Simulation Conference*, pages 719–726, 1998.

C. Park and S.-H. Kim. Handling stochastic constraints in discrete optimization via simulation. In *Winter Simulation Conference*, pages 4217–4226, 2011.

R. Pasupathy, S. R. Hunter, N. A. Pujowidianto, L. H. Lee, and C. Chen. Stochastically constrained ranking and selection via SCORE. *ACM Transactions on Modeling and Computer Simulation*, 25(1):1–26, 2014.

D.H. Phan and J. Suzuki. R2-ibea: R2 indicator based evolutionary algorithm for multiobjective optimization. In *2013 IEEE Congress on Evolutionary Computation (CEC)*, pages 1836–1845. IEEE, 2013.

K. Praditwong and X. Yao. How well do multi-objective evolutionary algorithms scale to large problems. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*. IEEE, 2007.

N. A. Pujowidianto, R. Pasupathy, S.R. Hunter, L. H. Lee, and C.-H. Chen. Closed-form sampling laws for stochastically constrained simulation optimization on large finite sets. In *Winter Simulation Conference*, pages 1–10. IEEE, 2012.

I.O. Ryzhov, W.B. Powell, and P.I. Frazier. The knowledge gradient algorithm for a general class of online learning problems. *Operations Research*, 60(1):180–195, 2012.

T. Sawik. Integrated supply chain scheduling under multi-level disruptions. *IFAC-PapersOnLine*, 48(3):1515–1520, 2015.

A. Seshadri. Multi-objective optimization using evolutionary algorithms (moea). *Matlab Website: Http://www. Mathworks. com/matlabcentral/fileexchange/10429, by*, 19, 2006.

M. Sharafi and T.Y. ELMekkawy. Multi-objective optimal design of hybrid renewable energy systems using PSO-simulation based approach. *Renewable Energy*, 68:67–79, 2014.

F. Siegmund, A. Ng, and K. Deb. A ranking and selection strategy for preference-based evolutionary multi-objective optimization of variable-noise problems. In *IEEE Congress on Evolutionary Computation*, pages 3035–3044. IEEE, 2016.

A. Syberfeldt, A. Ng, R.I. John, and P. Moore. Evolutionary optimisation of noisy multi-objective problems using confidence-based dynamic resampling. *European Journal of Operational Research*, 204(3):533–544, 2010.

S. Teng, L.H. Lee, and E. P. Chew. Integration of indifference-zone with multi-objective OCBA. *European Journal of Operational Research*, 203:419–429, 2010.

M. Viceconti, A. Henney, and E. Morley-Fletcher. In silico clinical trials: how computer simulation will transform the biomedical industry. *International Journal of Clinical Trials*, 3(2), 2016.

H. Wang, M. Olhofer, and Y. Jin. A mini-review on preference modeling and articulation in multi-objective optimization: current status and challenges. *Complex & Intelligent Systems*, 3(4):233–245, 2017.

S. Yan, E. Zhou, and C.-H. Chen. Efficient simulation budget allocation for selecting the best set of simplest good enough designs. pages 1152–1159. Winter Simulation Conference, 2010.

M. Yoon and J. Bekker. Single-and multi-objective ranking and selection procedures in simulation: A historical review. *South African Journal of Industrial Engineering*, 28(2):37–45, 2017.

Q. Zhang and H. Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.

T. Zhang, M. Georgiopoulos, and G.C. Anagnostopoulos. S-Race: A multi-objective racing algorithm. In *Genetic and Evolutionary Computation Conference*, pages 1565–1572. ACM, 2013.

T. Zhang, M. Georgiopoulos, and G.C.Anagnostopoulos. Sprint multi-objective model racing. In *Genetic and Evolutionary Computation Conference*, pages 1383–1390. ACM, 2015.

A. Zhou, B. Qu, H. Li, S. Zhao, P. N. Suganthan, and Q. Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, 2011.

E. Zitzler and S. Künzli. Indicator-based selection in multiobjective search. In *International Conference on Parallel Problem Solving from Nature*, pages 832–842. Springer, 2004.

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.

E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.

E. Zitzler, M. Laumanns, and L. Thiele. Spea2: Improving the strength pareto evolutionary algorithm. *TIK-report*, 2001.

E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, and F. Viviane Grunert. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.