Multiobjective ranking and selection with incomplete preference information

Bachelor's Thesis

Ville Koponen

Systems Analysis Laboratory
Aalto University School of Science

May 4, 2014

ABSTRACT
Author : Ville Koponen Title: Multiobjective Ranking and Selection With Incomplete Preference Information Date: May 4, 2014 Language: English Number of pages : 29
Professorship: Systems and Operations Research
Supervisor: Professor Raimo P. Hämäläinen Instructor: M.Sc. Ville Mattila
In this thesis, a multiobjective simulation optimization procedure is presented. The procedure is based on optimal computing budget allocation, which is modified to work on multiobjective problems by using multi-attribute utility function and incomplete preference information. This procedure is compared against an established multiobjective computing allocation procedure and simulation experiments show that significant computational savings can be achieved with a wide variety of problems. Results show that if the decision maker is able or willing to give preference information, the proposed procedure may save computational time in simulations.
Keywords: multiobjective ranking and selection, simulation optimization, incomplete preference information

TIIVISTELMÄ
Tekijä : Ville Koponen Työn nimi: Monitavoitteinen simulointi-optimointi epätäydellisillä preferensseillä Päiväys: 4. toukokuuta 2014 Kieli: Englanti Sivuäärä: 29
Professuuri: Systeemi- ja operaatiotutkimus
Valvoja: Professori Raimo P. Hämäläinen Ohjaaja: Diplomi-insinööri Ville Mattila
Tässä työssä esitellään monitavoitteinen simulointi-optimointi menetelmä. Menetelmä perustuu olemassa olevaan optimal computing budget allocation menetelmään, jota on muutettu toimimaan monitavoitteisissa ongelmissa. Muutokset perustuvat moniatribuuttiseen hyötyfunktioon ja epätäydelliseen preferenssi-infromaatioon. Menetelmää verrataan toiseen olemassa olevaan monitavoiteoptimointiin tarkoitettuun menetelmään. Simulointikokeiden perusteella uudella menetelmällä voidaan saavuttaa merkittäviä säästöjä laskenta-ajassa, jos menetelmää käyttävällä päätöksentekijä voi tai haluaa ilmoittaa hyötynsä.
Asiasanat: monitavoitteinen ranking and selection, simulointi-optimointi, epätäydellinen preferenssi-informaatio

Contents

1 Introduction

Ranking and selection (R&S) procedures in discrete-event simulation optimization are statistical methods for selecting the best simulated system design or a subset containing the best design from a set of competing designs (Swisher et al., 2003; Kim and Nelson, 2007). Most existing R&S procedures are concerned with a single measure of performance. Although frequently assessed in practical settings only few R&S procedures allow multiple performance measures (Morrice et al.; Butler et al., 2001; Swisher and Jacobson, 2002; Teng et al., 2007; Lee et al., 2008; Teng et al., 2010; Lee et al., 2010). This thesis presents a new procedure for multiple performance measure R&S based on multi-attribute utility theory (Keeney and Raiffa, 1976) and an existing R&S procedure designed for a single performance measure (Chen et al., 2000). In particular, the presented procedure incorporates incomplete preference information from a decision-maker (DM) seeking the best design.

The existing literature on R&S for multiple performance measures is twofold. First, Morrice et al.; Butler et al. (2001); Swisher and Jacobson (2002) combine multiple performance measures into a single one through a multi-attribute utility function. Then, R&S procedures for a single performance measure can be utilized for determining the design with the maximum expected utility. Second, Teng et al. (2007); Lee et al. (2008); Teng et al. (2010); Lee et al. (2010) develop procedures for determining all non-dominated designs. A design is non-dominated if it is non-inferior to any other design with respect to all performance measures. These procedures are extensions of the optimal computing budget allocation (OCBA) procedure developed in (Chen et al., 2000). In OCBA, an expression for the probability of correctly selecting the best design given an allocation of computing budget, i.e., simulation replications for determining the performance of the designs is found. Further simulation replications are allocated to the designs by maximizing the probability of correct selection assuming an infinite number of replications. By updating the allocation when an incremental number or replications have been performed, a sequential procedure for efficiently determining the best design is obtained. The versions for multiple performance measures are similar, but maximize a probability of correctly identifying the non-dominated designs.

The R&S procedure presented in this thesis lies methodologically in the intersection of the procedures described in (Morrice et al.; Butler et al., 2001; Swisher and Jacobson, 2002) and in (Teng et al., 2007; Lee et al., 2008; Teng et al., 2010; Lee et al., 2010). The multiple performance measures are combined into single one through a MAU utility function, but the procedure additionally incorporates incomplete preference information (White et al., 1984; Kirkwood and Sarin, 1985; Hazen, 1986; Weber, 1987; Salo and Hämäläinen, 1992). The utility of a design does not therefore obtain a unique scalar value, but a range of values. The objective of the presented procedure is to determine the designs that are non-dominated based on the utilities, i.e., the incomplete preference information. Thus, instead of seeking a single utility maximizing design, a subset of designs is obtained. This is accomplished by expressing the probability of correctly identifying the preferentially non-dominated designs given an allocation of simulation replications among the designs. Further replications are sequentially allocated by maximizing the probability under the assumption that the computing budget is unlimited, similar to the OCBA procedure. The presented procedure can thus be regarded as an extension of OCBA to multiple performance measure R&S with incomplete preference information.

The DM may be unwilling or incapable of giving a complete specification of preferences. In the procedures described in (Morrice et al.; Butler et al., 2001; Swisher and Jacobson, 2002), for instance, the DM states his preferences prior to evaluating the alternative designs. The difficulty is that the DM is required to provide a complete specification of preferences without accurate information of the ranges of the values of the performance corresponding to the designs. In the procedure presented in this thesis, the DM also gives preference statement prior to the evaluation of the designs. Since incomplete statements are allowed, however, the DM has room for preferential uncertainty and should be more confident in giving the statements. In comparison to the multiple performance measure OCBA procedures (Teng et al., 2007; Lee et al., 2008; Teng et al., 2010; Lee et al., 2010), the benefit of the presented procedure is the possibility for computational savings that come with the prior expression of preferences. Since all non-dominated designs do not have to be identified, simulation replications can be primarily allocated to a subset of these designs that are preferentially non-dominated or nearly preferentially non-dominated. The preferentially non-dominated designs may thus be identified with the same level of confidence as the non-dominated designs, but with fewer simulation replications. Although the magnitude of the saving in computing effort depends on the preferences and the designs to be compared, the illustrative examples presented in this thesis imply that such benefits can be significant.

The thesis is organized as follows. Section 2 introduces the use of the MAU function incorporating incomplete preference information for comparing the alternative designs. Section 3 presents the R&S procedure based on OCBA for sequentially allocating simulation replications among the designs and identifying the absolutely non-dominated designs with high level of confidence. Section 4 illustrates the application of the procedure through several example problems and analyses the computational savings compared to OCBA procedures for multiple performance measures. Concluding remarks are given in Section 5.

2 Multi-attribute utilities and incompletely specified preferences

Multiobjective ranking and selection is about determining the best design from a finite set of alternatives, which all have multiple performance measures that are evaluated with stochastic simulation. $\mathbf{X}_k = (X_{k1}, \ldots, X_{kn})$ denotes the random variables corresponding to n performance measures of the design k. This section describes how designs with multiple performance measures can be compared by the means of multi-attribute utility theory. The goal is to determine a subset of preferred designs that is consistent with preferential information given by a decision-maker.

2.1 Multi-attribute utility function

The comparison of the designs is based on a multi-attribute utility (MAU) function. The MAU function determines a utility value that describes the performance of a design. The value is based on performance measures of the design and the preferences of the decision-maker. Value of a single performance measure is determined with a single-attribute utility function that maps the value to a range $[0, 1]$. The decision-maker determines the relative importance of the performance measures with weights. Additive MAU function is

$$u(\mathbf{X}_k) = \sum_{i=1}^{n} w_i u_i(X_{ki}), \tag{1}$$

where $u_i, i = 1, \ldots, n$ are single-attribute utility functions and $w_i \in [0, 1], l = 1, \ldots, n$ are weights, for which holds $\sum_{i=1}^{n} w_i = 1$.

Additive MAU function can be used if attributes are assumed to be additively independent, which means that the preference of one attribute is not dependent on the values of other attributes (Keeney and Raiffa, 1976). Additive MAU function is one of several different MAU functions and it is chosen in this work because of its simplicity and easy implementation to the R&S problem. It has also been found to be robust even if the additive independence assumption does not completely hold (Keeney and Raiffa, 1976).

When comparing designs, design k is preferred to design l if and only if $E[u(\mathbf{X}_k)] > E[u(\mathbf{X}_l)]$, that is, the the expected utility of design k is higher than that of design l.

2.2 Incomplete information

The description of the preferences of the DM through the MAU function (1) is widely accepted and applied. However, it requires the DM to evaluate exactly the single-attribute utility functions and the weights in the MAU function. In some occasions, the DM may be unwilling or incapable of doing such evaluations. In the decision theoretic literature, this difficulty has been addressed in several instances by methodology that allows incomplete evaluations by the DM.

The incomplete information can relate to both the MAU function and the probabilities of the distributions of the measures of the design ($X_{ki}, k = 1, \ldots, K, i = 1, n$). With no loss of generality, we restrict the consideration of incomplete information to concern the weights of the additive MAU function.

A wealth of studies have considered techniques for eliciting incomplete preferences from the DM for the construction of the utility function, i.e., single-attribute utility functions and the weights. In this thesis, these techniques are not considered, since the focus is on making the strongest possible inferences given a set of preference information and a limited computing budget for determining the performance of the designs. For discussion of preference elicitation, the reader is referred to (e.g. Keeney and Raiffa, 1976; von Winterfeldt and Edwards, 1986)

Incompletely specified weights are presented as intervals, instead of exact values. As a result, there are several different ways to determine the dominance relations between the designs. First, design k is said to dominate design l according to pairwise dominance if the expected utility of k is higher for all feasible weights. Formally, $E[u(\mathbf{X}_k|\mathbf{w})] > E[u(\mathbf{X}_l|\mathbf{w})]\ \forall \mathbf{w} \in \mathbf{W}$, where $u(\mathbf{X}_k|\mathbf{w})$ is the additive MAU function (1) conditional on the weights $\mathbf{w} = (w_1, \ldots, w_n)$ and $\mathbf{W}$ the set of all feasible weight vectors. Second, design k dominates design l according to absolute dominance if the expected utility of k over all feasible weights is higher than the expected utility over all feasible weights, i.e., $E[u(\mathbf{X}_k|\mathbf{w})] > E[u(\mathbf{X}_l|\mathbf{w}')]\ \forall \mathbf{w}, \mathbf{w}' \in \mathbf{W}$. Absolute dominance is a more restrictive condition in the sense that an absolutely dominated design is also pairwise dominated but the opposite is not true.

Figure 1 shows an example illustrating different dominance relations. Figure shows possible values of MAU functions (1) with different weights for five different designs with two performance measures. Weights for both measures are limited between 0.3 and 0.7. Pairwise dominance relations in this example are: design 5 dominates designs 1, 2, and

4, design 4 dominates designs 1 and 2 and design 3 dominates designs 1 and 2. Absolute dominance relations are otherwise the same, except design 3 does not dominate design 1.

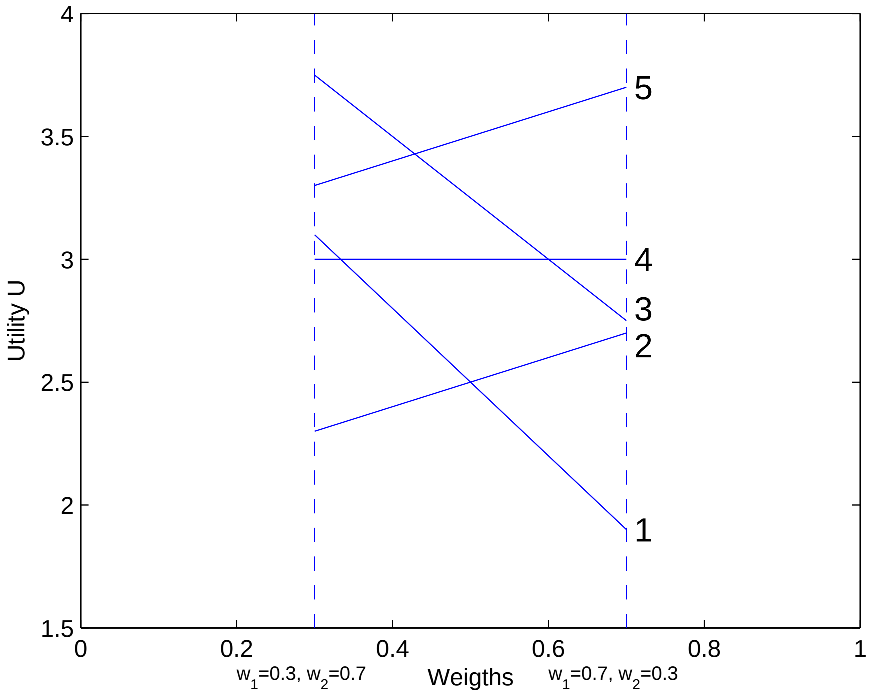

Figure 1: Relation of dominance definitions.

The dominance relations do not yield a complete ranking for the designs and there is no unique way to determine the ranking. Several decision rules utilizing the preference information as well as the information on the performance of the designs are available though (Weber, 1987). According to Weber (1987), the rule that is applied should incorporate all information that is gathered so far in the analysis. Sorting of the designs with respect to highest or lowest expected utilities may therefore be undesirable, since some of the preference information is omitted (i.e., only the most favourable or unfavourable outcomes are considered).Weber (1987) suggest decision rules that are based on expressing the strength of preference between pairs of solutions. Let the difference in expected utilities for the kth and the lth design with weights $\mathbf{w}$ be denoted with $h(\mathbf{X}_k, \mathbf{X}_l|\mathbf{w}) = E[u(\mathbf{X}_k|\mathbf{w})] - E[u(\mathbf{X}_l|\mathbf{w})]$. If one assumes a given probability distribution for the weights over the set of all feasible weights, the strength of preference for the kth design over the lth design can be expressed as the probability that k has higher expected utility:

$$d(\mathbf{X}_k, \mathbf{X}_l|\mathbf{W}) = P\left(h(\mathbf{X}_k, \mathbf{X}_l|\tilde{\mathbf{w}}) \geq 0\right), \tag{2}$$

where $\tilde{\mathbf{w}}$ are random weights that belong to the set $\mathbf{W}$. If $d(\mathbf{X}_k, \mathbf{X}_l|\mathbf{W})$ is larger than 0.5 one can regard that the kth design is preferred to the lth design.

2.3 Dominance and ranking with simulated performance

In this thesis, techniques for identifying absolutely dominated designs by utilizing stochastic simulation to determine the performance of the designs are considered. These techniques allocate simulation replications among the designs so that the non-dominated designs can most likely be identified. Absolute dominance is considered because it allows existing and proven R&S techniques to be used in the allocation of the replications. For pairwise dominance, the question of how to allocate computing effort becomes considerable more difficult and there is no apparent way to apply existing techniques for the task. Although the use of absolute dominance prohibits the identification of pairwise dominated designs, it still allows to eliminate several poor designs from consideration and thus benefits the DM. For the purpose of identifying the absolutely dominated designs, the lowest and highest utilities of design k are defined as:

$$\underline{u}(\mathbf{X}_k|\mathbf{W}) = \min_{\mathbf{w}\in\mathbf{W}} u(\mathbf{X}_k|\mathbf{w}), \tag{3}$$

$$\overline{u}(\mathbf{X}_k|\mathbf{W}) = \max_{\mathbf{w}\in\mathbf{W}} u(\mathbf{X}_k|\mathbf{w}). \tag{4}$$

Based on m independent simulation replications, the lowest and highest expected utilities of design k are estimated through:

$$\hat{\underline{u}}(\mathbf{X}_{k1},\ldots,\mathbf{X}_{km}|\mathbf{W}) = \min_{\mathbf{w}\in\mathbf{W}} \frac{1}{m}\sum_{j=1}^{m}\sum_{i=1}^{n} w_i u_i(X_{kij}), \tag{5}$$

$$\hat{\overline{u}}(\mathbf{X}_{k1},\ldots,\mathbf{X}_{km}|\mathbf{W}) = \max_{\mathbf{w}\in\mathbf{W}} \frac{1}{m}\sum_{j=1}^{m}\sum_{i=1}^{n} w_i u_i(X_{kij}), \tag{6}$$

where $\mathbf{X}_{kj} = (X_{k1j},\ldots,X_{knj})$ are random variables representing the performance of the kth design in the jth simulation replication. Design k is regarded as dominating design l if $\hat{\underline{u}}(\mathbf{X}_{k1},\ldots,\mathbf{X}_{km}|\mathbf{W})] > \hat{\overline{u}}(\mathbf{X}_{l1},\ldots,\mathbf{X}_{lm}|\mathbf{W})$.

Simulations are allocated to different designs based on absolute dominance. Pairwise dominance and the ranking of the designs can be determined based on the performed simulations. Pairwise dominance and ranking based on simulated performance of the designs is shortly described. First, the expected utility of the kth design with weights $\mathbf{w}$ based on m independent simulation replications is estimated through:

$$\hat{u}(\mathbf{X}_{k1}, \ldots, \mathbf{X}_{km}|\mathbf{w}) = \frac{1}{m} \sum_{j=1}^{m} \left(\sum_{i=1}^{n} w_i u_i(X_{kij}) \right). \tag{7}$$

The kth design is regarded as dominating the lth design according to the pairwise dominance if $\hat{u}(\mathbf{X}_{k1}, \ldots, \mathbf{X}_{km}|\mathbf{w}) > \hat{u}(\mathbf{X}_{l1}, \ldots, \mathbf{X}_{lm}|\mathbf{w})\ \forall \mathbf{w} \in \mathbf{W}$.

Finally, the strength of preference for the kth design over the lth design (2) is estimated through

$$\begin{aligned} &\hat{d}(\mathbf{X}_{k1}, \ldots, \mathbf{X}_{km}, \mathbf{X}_{l1}, \ldots, \mathbf{X}_{lm}|\mathbf{W}) = \\ &\frac{1}{M} \sum_{j=1}^{M} \mathbf{1} \left(\hat{u}(\mathbf{X}_{k1}, \ldots, \mathbf{X}_{km}|\tilde{\mathbf{w}}_j) - \hat{u}(\mathbf{X}_{l1}, \ldots, \mathbf{X}_{lm}|\tilde{\mathbf{w}}_j) \geq 0 \right), \end{aligned} \tag{8}$$

where $\tilde{\mathbf{w}}_1, \ldots, \tilde{\mathbf{w}}_M$ are drawn randomly from $\mathbf{W}$ according to their assumed distribution.

3 Optimal computing budget allocation procedure

This section describes a procedure for determining the absolutely non-dominated designs with high level of confidence using a limited total computing budget, i.e., number of simulation replications. This procedure is based on the optimal computing budget allocation (OCBA) procedure presented in (Chen et al., 2000). In the original OCBA, the goal is to determine the design that minimizes a single performance measure. Chen et al. (2000) determine an expression for the probability that the design for which the estimated value of the performance measure (based on given allocation of simulation replications) is actually the one with the minimum expected performance. Further simulation replications are allocated among the designs by finding an allocation that asymptotically maximizes this probability referred to as the probability of correct selection. By determining a new allocation after a given amount of additional replications have been performed a sequential R&S procedure is obtained.

Here, OCBA is applied for multiobjective R&S with incomplete preference information. An expression for the probability of correctly selecting the non-dominated designs with a given allocation of replications is given. Further, we show that the rules for allocating further replication in OCBA can be applied for the maximization of this probability with minor modification. As the result, a procedure largely similar to OCBA is obtained for efficiently determining the absolutely non-dominated designs.

3.1 Probability of correct selection

The absolutely non-dominated designs have highest expected utility that is higher than the maximum lowest expected utility of all designs. The other designs are dominated. Let us denote the non-dominated designs, based on given allocation of simulation replications, with S and the dominated designs with $\bar{S}$. Since the estimators (6) of the lowest and highest expected utilities are sample averages of independent random variables they can be treated as approximately normally distributed. In order to simplify notation, we use $\underline{\hat{u}}_k = \underline{\hat{u}}(\mathbf{X}_{k1}, \ldots, \mathbf{X}_{km}|\mathbf{W})$ to denote the estimator for the lowest expected utility of design k and denote the estimator for the highest expected utility similarly. Further let b denote the design with maximum lowest expected utility, i.e., $b = \arg\max_{i\in\{1,\ldots,K\}} \underline{\hat{u}}_i$. Based on the Bonferroni inequality, the probability of correct selection, i.e., that each design in S is actually non-dominated and each design in $\bar{S}$ is actually dominated can be expressed as

$$P_{\text{cs}} \geq 1 - \sum_{k \in S, k \neq b} P\left(\underline{\hat{u}}_b > \hat{\bar{u}}_k\right) - \sum_{k \in \bar{S}, k \neq b} P\left(\hat{\bar{u}}_k > \underline{\hat{u}}_b\right). \tag{9}$$

3.2 Rule for allocation of simulation replications

The derivation of the rule for allocating simulation replications among the designs to maximize the probability of correct selection is similar to the derivation of the OCBA procedure presented in (Chen et al., 2000). The steps of the reasoning are repeated here for clarity and convenience. The optimization problem to be solved is

$$\begin{aligned}
&\max_{m_1,\dots,m_K} 1 - \sum_{k \in S, k \neq b} P\left(\underline{\hat{u}}_b > \hat{\bar{u}}_k\right) - \sum_{k \in \bar{S}, k \neq b} P\left(\hat{\bar{u}}_k > \underline{\hat{u}}_b\right) \\
&\text{s.t.} \sum_{k=1}^{K} m_k = T,
\end{aligned} \tag{10}$$

where m_k is the number of simulation replications allocated to the kth design and T the total computing budget. In (Chen et al., 2000), the strategy of solving (10) is to first assume $m_k, k = 1, \dots, K$ continuous and ignore all associated non-negativity constraints, express the Karush-Kuhn-Tucker (KKT) conditions of the Lagrangian relaxation of the problem, and to investigate the relationships of $m_k, k = 1, \dots, K$ when T is assumed infinite.

First, some additional notation is introduced. Let the negative absolute value of the difference in the estimates for the lowest expected utility for design b and the highest expected utility for design k be $\delta_{bk} = -\left|\underline{\hat{u}}(\mathbf{x}_{b1}, \dots, \mathbf{x}_{bm_b}|\mathbf{W}) - \underline{\hat{u}}(\mathbf{x}_{k1}, \dots, \mathbf{x}_{km_k}|\mathbf{W})\right|$ where the $\mathbf{x}_{k1}, \dots, \mathbf{x}_{km_k}$ refer to realizations of the performance measures for the kth design. Further, the variance of the estimator for this difference is $\sigma_{bk}^2 = \sigma_b^2/m_b + \sigma_k^2/m_k$, where σ_b^2 and σ_k^2 are the variances of the estimators for the lowest expected utility of design b and the highest expected utility of design k. The above quantities are unknown but they can be estimated from the realizations of the performance measures and the resulting utilities. The summation terms in the objective function in (10) are now approximated by:

$$\sum_{k\in S,k\neq b} P\left(\underline{\hat{u}}_b > \hat{\overline{u}}_k\right) - \sum_{k\in \bar{S},k\neq b} P\left(\hat{\overline{u}}_k > \underline{\hat{u}}_b\right) \approx \sum_{k\in\{1,\dots,K\},k\neq b} \int_0^\infty \frac{1}{\sqrt{2\pi}\sigma_{bk}} e^{-\frac{(x-\delta_{bk})^2}{2\sigma_{bk}^2}} dx$$
$$= \sum_{k\in\{1,\dots,K\},k\neq b} \int_{-\frac{\delta_{bk}}{\sigma_{bk}}}^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$$

The Lagrangian relaxation of (10) becomes

$$F = 1 - \sum_{k\in\{1,\dots,K\},k\neq b} \int_{-\frac{\delta_{bk}}{\sigma_{bk}}}^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt - \lambda\left(\sum_{k=1}^K m_k - T\right). \tag{11}$$

It should be emphasized that this expression is exactly the same as in the derivation of OCBA (Chen et al., 2000). The only difference is that $\delta_{bk}, k = 1,\dots,K, k\neq b$ refer to the difference in the lowest expected utility of design b and highest expected utility of design k instead of difference in expected value of a performance measure between the best performing design and the other designs. Thus, the remaining steps in the derivation of the allocation rule are exactly the same as in (Chen et al., 2000).

The KKT conditions for (11) are

$$\frac{\partial F}{\partial m_k} = \frac{\partial F}{\partial\left(-\frac{\delta_{bk}}{\sigma_{bk}}\right)} \frac{\partial\left(-\frac{\delta_b k}{\sigma_{bk}}\right)}{\partial \delta_{bk}} \frac{\partial \delta_{bk}}{\partial m_k} - \lambda$$
$$= \frac{-1}{2\sqrt{2\pi}} \exp\left[\frac{-\delta_{bk}^2}{2\sigma_{bk}^2}\right] \frac{\delta_{bk}\sigma_k^2}{m_k^2(\sigma_{bk}^2)^{3/2}} - \lambda = 0, k = 1,\dots,K, k\neq b, \tag{12}$$
$$\frac{\partial F}{\partial m_b} = \frac{-1}{2\sqrt{2\pi}} \sum_{k\in\{1,\dots,K\},k\neq b} \exp\left[\frac{-\delta_{bk}^2}{2\sigma_{bk}^2}\right] \frac{\delta_{bk}\sigma_b^2}{m_b^2(\sigma_{bk}^2)^{3/2}} - \lambda = 0, \tag{13}$$
$$\lambda\left(\sum_{k=1}^K m_k - T\right) = 0, \lambda \geq 0.$$

With the KKT conditions the relationship of m_b and $m_k, k = 1,\dots,K, k\neq b$ can be investigated. First, Equation (12) gives

$$\frac{-1}{2\sqrt{2\pi}} \exp\left[\frac{-\delta_{bk}^2}{2\sigma_{bk}^2}\right] \frac{\delta_{bk}}{(\sigma_{bk}^2)^{3/2}} = \lambda\frac{m_k^2}{\sigma_k^2}, k = 1,\dots,K, k\neq b. \tag{14}$$

Substituting (14) into (13) yields

$$\sum_{k\in\{1,\ldots,K\},k\neq b}\frac{\lambda m_k^2\sigma_b^2}{m_b^2\sigma_k^2}-\lambda=0, \tag{15}$$

which further gives

$$m_b=\sigma_b\sqrt{\sum_{k\in\{1,\ldots,K\},k\neq b}\frac{m_k^2}{\sigma_b^2}}. \tag{16}$$

Moreover, the relationship between m_k and m_l, $k,l\in\{1,\ldots,K\}\,,k\neq l\neq b$ needs to be considered. From Equation (12),

$$\exp\left(\frac{-\delta_{bk}}{2\left(\frac{\sigma_b^2}{m_b}+\frac{\sigma_k^2}{m_k}\right)}\right)\cdot\frac{\delta_{bk}\sigma_k^2/m_k^2}{\left(\frac{\sigma_b^2}{m_b}+\frac{\sigma_k^2}{m_k}\right)^{3/2}}=\exp\left(\frac{-\delta_{bl}}{2\left(\frac{\sigma_b^2}{m_b}+\frac{\sigma_l^2}{m_l}\right)}\right)\cdot\frac{\delta_{bl}\sigma_l^2/m_l^2}{\left(\frac{\sigma_b^2}{m_b}+\frac{\sigma_l^2}{m_l}\right)^{3/2}}. \tag{17}$$

If the variances $\sigma_1,\ldots,\sigma_K$ are assumed equal Equation (16) implies

$$m_b=\sqrt{\sum_{k\in\{1,\ldots,K\},k\neq b}m_k^2}. \tag{18}$$

Thus, it appears that the number of replications allocated to the design b with the maximum lowest expected utility is notably higher compared to the other designs. It should be noted that this may not be true for the final allocation, since b might actually correspond to different designs in different stages of budget allocation (recall that the allocation is done sequentially) as the the utilities of the designs are estimated with increasing accuracy. During one stage, however, design b is allocated the most replications since its utility may potentially affect the status of dominance for several designs. It is thus assumed that $m_b >> m_k, k\in\{1,\ldots,K\}\,,k\neq b$ which allows to write Equation (17) as

$$\exp\left(\frac{-\delta_{bk}}{2\left(\frac{\sigma_k^2}{m_k}\right)}\right)\cdot\frac{\delta_{bk}\sigma_k^2/m_k^2}{\left(\frac{\sigma_k^2}{m_k}\right)^{3/2}}=\exp\left(\frac{-\delta_{bl}}{2\left(\frac{\sigma_l^2}{m_l}\right)}\right)\cdot\frac{\delta_{bl}\sigma_l^2/m_l^2}{\left(\frac{\sigma_l^2}{m_l}\right)^{3/2}}. \tag{19}$$

Rearrangement further gives

$$\exp\left(\frac{1}{2}\left(\frac{\delta_{bl}}{\frac{\sigma_l^2}{m_l}}-\frac{\delta_{bk}}{\frac{\sigma_k^2}{m_k}}\right)\right)\sqrt{\frac{m_l}{m_k}}=\frac{\delta_{bl}\sigma_k}{\delta_{bk}\sigma_l}, \tag{20}$$

and taking natural logarithm yields

$$\frac{\delta_{bl}^2}{\sigma_l^2}m_l+\log(m_l)=\frac{\delta_{bk}^2}{\sigma_k^2}m_k+\log(m_k)+2\log\left(\frac{\delta_{bl}\sigma_k}{\delta_{bk}\sigma_l}\right). \tag{21}$$

Letting the total computing budget tend to infinity, i.e., $T \to \infty$, the logarithm terms become negligible compared to other terms. Thus, with minor rearrangement

$$\frac{m_k}{m_l}=\left(\frac{\sigma_k/\delta_{bk}}{\sigma_l/\delta_{bl}}\right)^2, k,l\in\{1,\ldots,K\}, k\neq l\neq b. \tag{22}$$

In the beginning of the derivation, all non-negativity constraints on $m_k, k=1,\ldots,K$ were ignored. According to Equations (16) and (22), $m_k, k=1,\ldots,K$ are, however, non-negative since they all have the same sign and must sum up to the total computing budget T. Further, if $m_k, k=1,\ldots,K$ satisfy Equations (16) and (22), the KKT conditions hold and a local optimal solution is obtained. The result is that the approximate probability of correctly identifying the non-dominated solutions is asymptotically maximized when computing budget is allocated according to the equations.

3.3 Summary of the multiobjective R&S procedure

With the computing budget allocation rules implied by Equations (16) and (22), the complete allocation procedure is described as follows.

0. Determine the total computing budget T, the number of additional replications Δ and the number of initial replications m_0. Set the iteration counter to $j \leftarrow 0$. Perform m_0 simulation replications for each design such that $m_1^j = m_2^j = \ldots = m_K^j = m_0$, where m_k^j is the number of replications performed for the kth design after the jth iteration.

1. If $\sum_{k=1}^{K} m_k^j \geq T$, go to step 4.

2. Calculate the new allocation of computing budget $m_1^{j+1}, \dots, m_K^{j+1}$ according to Equations (16) and (22) by using $\sum_{k=1}^{K} m_k^j + \Delta$ as the intermediate total computing budget at iteration $j+1$.

3. Perform $\max(0, m_k^{j+1})$ additional replications for each design $k \in \{1, \dots, K\}$. Set the iteration counter to $j \leftarrow j+1$. Go to step 1.

4. Determine the set of non-dominated solutions based on the final allocation $m_1^j, \dots, m_K^j$.

4 Numerical experiments

OCBA procedure's probability of correctly determining the absolutely non-dominated designs is tested with numerical experiments. OCBA procedure is compared against multiobjective computing budget allocation procedure (MOCBA) (Chen and Lee, 2010). MOCBA allocates simulation replications among the designs so that non-dominated designs, in terms of traditional dominance, are identified with a high level of confidence.

Procedures are tested with two different problems. First problem has nine predetermined designs which all have two performance measures. In the second problem procedures are tested with randomly generated problems. These problems include 50 different designs that have two performance measures with means that are randomly selected.

4.1 Example problem

OCBA and MOCBA procedures are compared in the case where the decision maker wants to find the absolutely non-dominated designs by using the MAU function and imprecise information. First, performance estimates are gathered from the procedures. Second, absolutely non-dominated designs are found using these estimates. These estimated dominances are then compared against dominances calculated from the actual means of the performance measures.

Problem of nine designs with two performance measures is shown in Table 1 and Figure 2. There are five non-dominated designs, which are marked with a red circle and three absolutely non-dominated designs, which are marked with a black cross. Variance of 2^2 is used for all performance measures. Weights of the performance measures are bounded [0.3 0.6] and [0.4 0.7]. Lowest and highest utilities calculated with these weights are shown in table 2.

Table 1: Mean values of the performance measures of the designs.

design #	1	2	3	4	5	6	7	8	9
Measure 1	0.0	0.67	2.5	5.0	1.1	2.2	4.0	2.4	3.5
Measure 2	5.0	2.5	0.67	0	4.0	2.2	1.1	3.	2.4

Figure 3 shows the probability of correct selection of the preferentially dominated set for both procedures with computing budgets from 400 to 2000. OCBA finds the absolutely non-dominated sets with a higher probability. MOCBA is more accurate when

Table 2: Highest and lowest utilities of the designs.

design #	1	2	3	4	5	6	7	8	9
Highest utility	3.5	2.0	1.8	3.0	3.1	2.2	2.8	3.2	3.1
Lowest utility	2.0	1.4	1.2	1.5	2.3	2.2	2.0	2.8	2.7

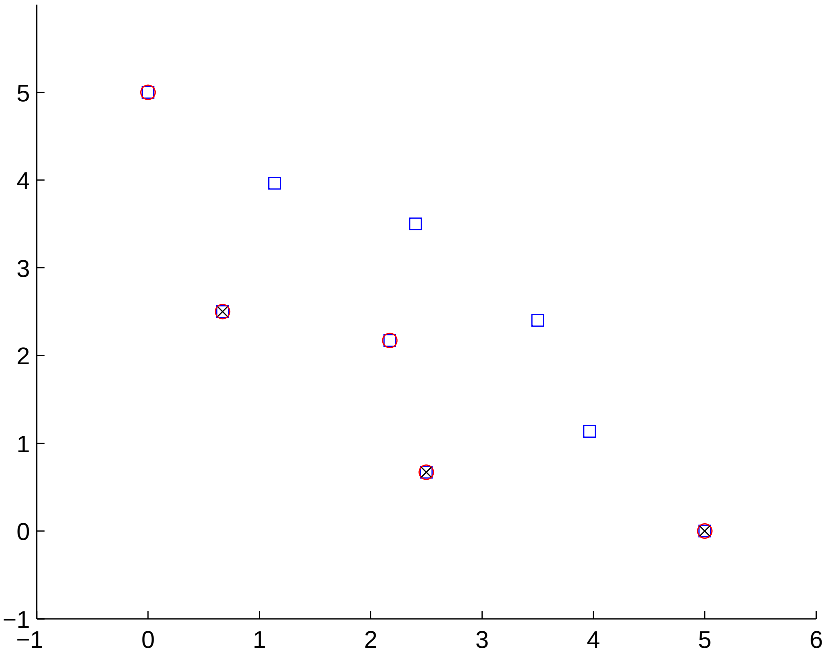

Figure 2: Means of the performance measures of the designs. Non-dominated designs are marked with red circles, dominated designs with blue squares and absolutely non-dominated designs with black crosses.

determining the non-dominated set. This is not surprising, as these procedures aim to do different things. OCBA procedure tries to find the absolutely non-dominated designs and MOCBA tries to find the non-dominated set. OCBA is more accurate in determining the absolutely non-dominated designs than the MOCBA is at determining the non-dominated set.

Table 3 shows how the procedures allocated replications between different designs. MOCBA procedure allocates replications to designs that are non-dominated, or close to being non-dominated symmetrically(its a symmetrical problem), designs 2 and 3, 5 and 7 and 8 and 9 receive almost equal amounts. OCBA allocates to designs that are absolutely non-dominated, or close to being absolutely non-dominated. For example design 7 gets more replications than design 6, which is different from MOCBA.

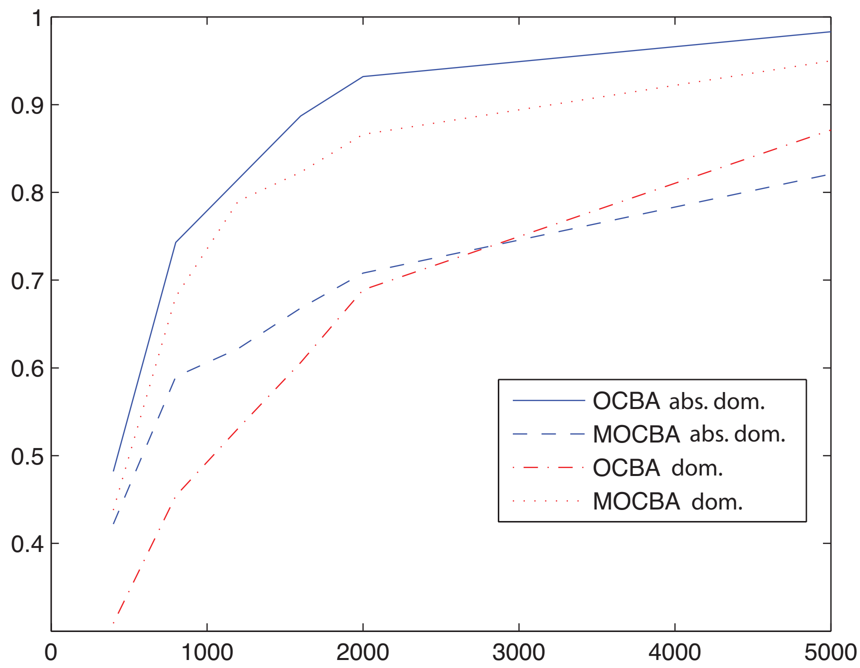

Figure 3: PCS for absolutely non-dominated and non-dominated sets of OCBA and MOCBA procedures.

Table 3: Percentage of replications allocated between designs %.

design #	1	2	3	4	5	6	7	8	9
OCBA	13.0	14.7	26.1	13.9	4.6	7.8	16.8	1.3	1.8
MOCBA	8.0	26.4	25.8	3.0	4.2	21.0	6.2	2.7	2.7

Table 4 shows which designs were classified wrong, absolutely non-dominated designs were classified as absolutely dominated, or the other way around. Designs 1, 4 and 7 seem to be the most difficult to determine correctly. These are also the designs whose lowest utilities are closest to the highest utility of the dominating design 3.

Table 4: Number of times which each design was classified wrongly.

design #	1	2	3	4	5	6	7	8	9	total
OCBA	430	114	31	257	98	66	443	8	5	1452
MOCBA	790	188	19	656	70	244	537	1	8	2513

Table 5: Preference weights, simulations budgets, and OCBA and MOCBA PCS with randomized problems.

w_1	0.3-0.6	0.3-0.6	0.4-0.5	0.4-0.5	0.1-0.9	0.1-0.9
w_2	0.4-0.7	0.4-0.7	0.5-0.6	0.5-0.6	0.1-0.9	0.1-0.9
T	1000	2000	1000	2000	1000	2000
OCBA PCS	65.8	77.2	80.2	88.4	18.1	33.3
MOCBA PCS	50.5	59.2	66.9	73.7	9.7	12.2

4.2 Randomly generated problems

In this section, OCBA and MOCBA procedures are compared with randomly generated test problems and with different weight combinations. Each problem consist of 50 different designs, each with two performance measures. Both procedures use same designs and new replications are made according to the procedures. Probability of correct selection is estimated by finding the absolutely dominated and absolutely non-dominated designs from the simulation outcomes.

True performance measures of the designs are generated from a uniform distribution on the range of from 0 to 10. Values of the performance measures in simulation replications are generated from a normal distribution using true performance measures as means and with variances of 2^2. Results from using 1000 different designs with different weight intervals and computing budgets are shown in table 5.

OCBA procedure is more accurate than MOCBA in finding the absolutely non-dominated sets. If Computing budget is increased from 1000 to 2000 both procedures have a better change of finding all the absolutely non-dominated designs. When weight intervals are defined with narrow weight intervals, both procedures perform better.

5 Conclusions

Ranking and selection (R&S) methods are used to compare different designs, whose performance are measured with stochastic simulation. The goal is to find the best or a set of best designs, when computational resources are limited. These methods are automatic procedures which determine how simulation replications are allocated between different designs. Unlikely candidates for the best design receive less computational time than the likely candidates.

In this thesis, a new multiobjective R&S method is proposed. The method is based on an existing optimal computing budget allocation (OCBA) procedure. This procedure is modified so that incomplete preference information can be used. Instead of finding the best design, modified method aims to find all of the preferentially non-dominated designs. Elimination of the dominated designs should make the decision of choosing the best design easier for the decision maker (DM). In this thesis absolute dominance is used. There exists other dominance relations, but absolute dominance is used because it is compatible with existing R&S methods.

Incomplete preference information means that the DM is either unwilling or unable to state his preference completely. Instead of stating his absolute preferences between different performance measures, like time or money, the DM states his preferences on intervals.

The procedure presented in this thesis is tested with numerical experiments. Experiments show that the procedure allocates most simulations to designs that are close to being either dominated or non-dominated. This way the performance of these designs are most accurately determined. This can mean that, in some cases, designs that are clearly non-dominated receive only a little simulations and therefore their performance is measured inaccurately. This maybe unfortunate, because the DM might be interested in relative performances on the non-dominated designs.

The modified OCBA procedure is compared against multiobjective computing allocation (MOCBA) procedure. The MOCBA procedure allocates simulations so that all of the non-dominated designs are determined. Results show that the OCBA procedure is more accurate in determining the absolutely non-dominated designs. The DM can therefore save computational resources by allocating his simulations with the new procedure.

One drawback of the proposed procedure is that it requires the incomplete information from the DM. It might not always be worth the trouble to determine the preference information just to save computational time. But especially in decision making cases where the information is already available, this procedure may lead to significant computational savings.

References

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

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

Chun-Hung Chen, Jainwu Lin, Enver Ýucesan, and Stephen E. Chick. Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Discrete Event Dynamic Systems: Theory and Applications*, 10:251–270, 2000.

Gordon B. Hazen. Partial information, dominance, and potential optimality in multiattribute utility theory. *Operations Research*, 34(2):296–310, 1986.

Ralph L. Keeney and Howard Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs.* Wiley, 1976.

Seong-Hee Kim and Barry L. Nelson. Recent advances in ranking and selection. In *Proceedings of the 2007 Winter Simulation Conference The best is yet to come*, pages 162–172, Piscataway, NJ, 2007. Institute of Electrical and Electronics Engineers Inc.

Craig W. Kirkwood and Rakesh K. Sarin. Ranking with partial information - a method and an application. *Operations Research*, 33(1):38–48, 1985.

Loo Hay Lee, Ek Peng Chew, Suyan Teng, and Yankai Chen. Multi-objective simulation-based evolutionary algorithm for an aircraft spare parts allocation problem. *European Journal of Operational Research*, 189:476–491, 2008.

Loo Hay Lee, Ek Peng Chew, Suyan Teng, and David Goldsman. Finding the non-dominated pareto set for multi-objective simulation models. *IEE Transactions*, 42(9): 656–674, 2010.

Douglas J. Morrice, John Butler, , and Peter W. Mullarkey.

Ahti A. Salo and Raimo P. Hämäläinen. Preference assessment by imprecise ratio statements. *Operations Research*, 40(6):1053–1061, 1992.

James R. Swisher and Sheldon H. Jacobson. Evaluating the design of a family practice health care clinic using discrete-event simulation. *Health Care Management Science*, 5:75–88, 2002.

James R. Swisher, Sheldon H. Jacobson, and Enver Ýucesan. Discrete-event simulation optimization using ranking, selection, and multiple comparison procedures: A survey. *ACM Transactions on Modeling and Computer Simulation*, 13(2):134–154, 2003.

Suyan Teng, Loo Hay Lee, and Ek Peng Chew. Multi-objective ordinal optimization for simulation optimization problems. *Automatica*, 43:1884–1895, 2007.

Suyan Teng, Loo Hay Lee, and Ek Peng Chew. Integration of indifference-zone with multi-objective computing budget allocation. *European Journal of Operational Research*, 203 (2):419–429, 2010.

Detlof von Winterfeldt and Ward Edwards. *Decision Analysis and Behavioral Research.* Cambridge University Press, 1986.

Martin Weber. Decision making with incomplete information. *European Journal of Operational Research*, 28(1):44–57, 1987.

C. C. White, A. P. Sage, and S. Dozono. A model of multiattribute decisionmaking and trade-off weight determination under uncertainty. *IEEE Transactions on Systems, Man, and Cybernetics*, 14(2):223–229, 1984.

A Summary in Finnish

Tietokonesimulointia voidaan käyttää systeemien käyttäytymisen tutkimiseen. Simulointi on hyödyllistä erityisesti tapauksissa, joissa mittausten suorittaminen oikealla systeemillä ei ole mahdollista tai se on kallista. Simulointien suorittaminen voi kuluttaa paljon laskenta-aikaa, mitä voidaan säästää suorittamalla vain tarpeellinen määrä simulointeja.

Ranking & selection-menetelmiä (R&S) käytetään eri systeemien vertaamiseen, kun systeemien suoritusta mitataan stokastisella simuloinnilla. Tavoitteena on löytää paras systeemi tai parhaiden systeemien joukko, kun käytössä on vain rajallinen määrä laskentaresursseja. R&S-menetelmät määrittelevät, kuinka monta kertaa mitäkin systeemiä simuloidaan, jotta paras systeemi löytyy halutulla tarkkuudella, tai miten ennalta määrätty määrä simulointeja kannatta jakaa systeemien kesken. Selvästi huonompia systeemejä simuloidaan vähemmän kuin hyviä systeemejä.

Monitavoitteisella simullointioptimoinnilla tarkoitetaan tilanteita, kun simuloitavilla systeemeillä on useampia eri mitattavia attribuutteja. Esimerkiksi voidaan olla kiinnostuneita projektin kestosta ja hinnasta. Monitavoitteisia ongelmia on kirjallisuudessa lähestytty kahdella eri tavalla. Ensimmäinen tapa on etsiä kaikki pareto-optimaaliset systeemit. Tällä tarkoitetaan niitä systeemejä, joille ei löydy toista systeemiä, joka on parempi kaikissa attribuuteissa. Toinen tapa on muuntaa monitavoitteinen ongelma hyötyfunktion avulla yksitavoitteiseksi. Tähän ongelmaan voidaan sitten soveltaa yksitavoitteisia R&S-menetelmiä.

Hyötyfunktio painottaa eri attribuutteja toistensa suhteen painoilla. Hyötyfunktion arvo saadaan kertomalla jokaista attribuuttia sen painolla ja laskemalla kaikki yhteen. Päätöksentekijä kertoo preferenssinsä eri attribuuttien suhteen määrittelemällä painot. Joissain tilanteissa päätöksentekijä ei pysty tai halua määritellä painoille yhtä tarkkaa arvoa. Tällöin päätöksentekijä voi määritellä jokaiselle painolle välin, jolle paino todellisuudessa sijoittuu. Tätä kutsutaan epätäydelliseksi preferenssi-informaatioksi. Kun hyötyfunktion painojen arvot on esitetty väleillä, myös itse hyötyfunktion arvo on jollain välillä. Vaihtoehto on preferentiaalisesti absoluuttisesti dominoitu, kun löytyy yksikin toinen vaihtoehto, jolle hyötyfunktio antaa kaikilla mahdollisillä painoilla paremman hyödyn, päätöksen tekijän asettamien rajojen puitteissa. Tässä työssä käytetään absoluuttista dominanssia. Muitakin dominanassirelaatioita on olemassa, mutta absoluuttista käytetään, koska se on yhteensopiva käytettävän menetelmän kanssa.

Tässä työssä on esitetty uusi monitavoitteinen R&S-menetelmä. Menetelmä perustuu olemassa olevaan yksitavoitteiseen menetelmään. Tätä menetelmää on muutettu siten, että epätäydellistä preferenssi-informaatiota voidaan hyödyntää. Moniattribuuttien ongelma muutetaan yksiatribuuttiseksi hyötyfunktion avulla. Attribuuttien oikeiden arvojen sijasta käsitellään systeemin hyödyn arvoa. Parhaan systeemin etsimisen sijaan tämä menetelmä pyrkii etsimään kaikki preferentiaalisesti ei dominoidut systeemit.

Menetelmän toimii vaiheittain. Ensin jokaista systeemiä simuloidaan muutaman kerran, jolloin saadaan jokaisen systeemin suoritukselle ensimmäiset estimaatit. Seuraavaksi näiden estimaattien perusteella lasketaan menetelmän tuottamien sääntöjen mukaisesti, kuinka paljon jokaista systeemiä simuloidaan seuraavassa vaiheessa. Näiden sääntöjen perusteella suoritetaan pieni määrä uusia simulaatioita, joiden perusteella saadaan uudet, tarkemmat estimaatit. Näiden uusien estimaattien perusteella päätellään, miten seuraavat simulaatiot jaetaan. Näin jatketaan, kunnes koko simulaatiobudjetti on käytetty.

Simulaatioiden jakamissäännöt on johdettu todennäköisyydestä, jolla estimaattien perusteella dominoidut systeemit ovat oikeasti dominoituja ja ei-dominoidut ei-dominoituja. Säännöt painottavat lisäsimulointeja systeemeille, jotka ovat lähellä dominoinnin rajaa ja systeemeille, joiden estimaatti on epätarkka.

Tämän työn menetelmää kokeiltiin simulointikokeilla. Kokeet osoittivat, että menetelmä jakaa eniten simulointitoistoja niille systeemeille, jotka ovat melkein dominoituja tai ei-dominoituja. Tällöin näiden systeemien suoritus mitataan tarkiten. Tämä tarkoittaa, että joissakin tapauksissa selvästi ei-dominoituja systeemejä simuloidaan vähän ja niiden suorituksen arvo jää epätarkaksi. Tämä voi olla ongelma, jos päätöksentekijä haluaa verrata ei-dominoitujen systeemien suoritusta keskenään.

Tämän työn menetelmä verrattiin toiseen monitavoitteiseen menetelmään, joka pyrkii etsimään kaikki pareto-optimaaliset systeemit. Simulointien perusteella tämän työn menetelmä vaikuttaisi löytävän preferentiaalisesti ei-dominoidut systeemit toista menetelmää tarkemmin useimmissa tapauksissa. Tarkkuuden lisäys perustuu siihen, ettei simulointitoistoja tarvitse tuhlata sellaisiin systeemeihin, jotka ovat lähellä pareto-optimaalisuutta, mutta kaukana preferentiaalisesta dominanssista. Päätöksentekijä voi siis säästää lasketa-aikaa käyttämällä uutta menetelmää.

Esitetyn menetelmän yhtenä heikkoutena on se, että se vaatii preferenssi-informaatiota päätöksentekijältä. Joissakin tapauksissa preferenssien määrittäminen voi olla hankalampaa kuin laskenta-ajan lisääminen. Mutta niissä tapauksissa joissa päätöksentekijä määrittelee päätösprosessissa epätaydelliset preferenssit joka tapauksessa, voi uuden menetelmän käyttäminen aiheuttaa säästöjä laskenta-ajassa.