Mat-2.4108 Independent Research Project in Applied Mathematics

Simulation budget allocation with incomplete preference information

Heikki Vesterinen

Aalto University School of Science

August 19, 2012

Contents

1 Introduction

Ranking and selection (R&S) procedures aim to find the best system design or a subset that contains the best system designs among all feasible designs as efficiently as possible in terms of computing time [5]. Ranking of the designs is based on the expected values of one or more performance measures, which are estimated by simulation. Most of the existing R&S procedures are designed for problems with a single performance measure, although many applications in practice involve two or more measures (e.g. the problem studied in [6]). Multiple performance measures can be aggregated to a single performance measure with the help of multi-attribute utility theory and procedures for a single performance measure can be thus applied, as done in [1, 9]. However, if the decision maker (DM) is not able or willing to provide exact information about his preferences, utilities cannot be expressed unambiguously. Then procedures for a single performance measures are not applicable.

One branch of current research focuses on finding the set of Pareto designs, i.e., the designs for which there is no other design that is at least as good with respect to all performance measures and superior with respect to at least one performance measure. Lee et al. [7] present a procedure called the multi-objective optimal computing budget allocation (MOCBA) procedure, which has been developed further and applied in practice in [2, 6, 8]. The MOCBA procedure is an extension of the optimal computing budget allocation (OCBA) procedure presented by Chen et al. [3].

This paper presents a technique for incorporating incomplete preference information into the MOCBA procedure. The technique utilizes multi-attribute utility theory with incomplete preference information [11] for allocating computing budget efficiently in a situation where incomplete preference information is related to the weights of a multi-attribute utility function. The allocation of computing budget is optimized for finding the set of pairwise non-dominated designs, instead of the Pareto set. Pairwise non-dominated designs form a subset of the Pareto set. Thus, computing budget can be concentrated to a smaller set of designs, which still contains the best one according to the preference information.

2 Ranking and selection

Let us consider a ranking and selection problem over a finite set of alternatives, referred to as designs. The best design or a subset containing the best design is determined based on several possibly conflicting simulated performance measures, i.e.,

$$\max_{i\in\Theta}(E[X_{i1}], ..., E[X_{iH}]), \tag{1}$$

where $\Theta = \{1, ...K\}$ is the set of all designs, H is the number of performance measures, $\mathbf{X}_i = (X_{i1}, ..., X_{iH})$ is a vector of random variables representing the simulated performance measures of ith design. For simplicity, we denote J_{il} as $E[X_{il}]$.

2.1 Pareto optimality

In R&S problems with multiple performance measures, it is usually not possible to find a design which is the best one with respect to all performance measures. Instead, we can search for non-dominated Pareto designs, which are designs that are optimized to the extent that no improvement can be made in any performance measure without making some other performance measure worse. Formally, a design i is said to be dominated by design j, if design j is not worse than design i with respect to any performance measure and design j is better than design i with respect to at least one performance measure.

Definition 1. *Design i is dominated by design j, i.e., $i \prec j$ if and only if,*

$$\begin{aligned} &\forall l \in \{1, .., H\}, J_{il} \leq J_{jl} \qquad \text{and} \\ &\exists l \in \{1, .., H\}, J_{il} < J_{jl} \end{aligned}$$

If design i is not dominated by design j, we denote it as $i \nprec j$. The Pareto set is the set of all non-dominated designs in Θ, i.e.,

$$S_p = \{i \in \Theta \mid i \nprec j, \forall j \in \Theta\}. \tag{2}$$

2.2 Aggregation of performance measures with a multi-attribute utility function

One approach to ranking and selection problems with multiple performance measures is to aggregate the performance measures by a multi-attribute utility (MAU) function [4]. Aggregating the performance measures converts the R&S problem into one with a single performance measure. The ranking of the designs is then determined based on the expected values of the MAU function, i.e., design i is preferred to design j if and only if $E[u(\mathbf{X}_i)] > E[u(\mathbf{X}_i)]$.

The functional form of the MAU function u depends on the structure of the set of attributes, i.e., the performance measures. The assumption that the attributes are additive independent leads to an additive MAU function[4]

$$u(\mathbf{X}_i) = \sum_{l=1}^{H} w_l u_l(X_{il}), \tag{3}$$

where u_l is a single-attribute utility function that describes DM's preferences of the lth attribute as a scalar value and w_l is a weight representing the relative importance of the lth attribute compared to the other attributes.

2.3 Incompletely specified preferences

A disadvantage of aggregating the performance measures with a MAU function is that the DM has to provide all information of the decision situation, which may be too restricting requirement in practical applications. The reason that the DM cannot provide exact information may be that he is unwilling to provide the information or incapable to express his preferences. The reason might be also that he has not made up his mind, i.e., DM's preferences are not structured enough or the preferences are unstable. To overcome these difficulties methodologies have been established that allow incompletely defined utility function, while still basing the analysis on the subjective expected utility theory.[11]

A decision situation is called *incomplete* if either the MAU function or the distributions of the performance measures of the designs cannot be expressed unambiguously. In this paper, we consider a situation where the values of the performance measures are determined by a simulation model and the DM does not make any subjective assessments about their distributions. Furthermore, the incomplete information is related to the weights of the additive MAU function (3), whereas the single-attribute utility functions are uniquely determined.

Instead of exact weights of the MAU function, a set of feasible weights $\mathbf{W}$ is elicited from the DM. Thus, the designs do not have an exact expected utility but a range of possible values for the expected utility. Several dominance relations can be defined, which the ranking of the designs can be based on. The pairwise dominance relation is defined as follows [11].

Definition 2. *Design i is pairwise dominated by design j in $\mathbf{W}$, i.e., $i \prec_p j$ if and only if,*

$$\begin{aligned}&\forall \mathbf{w} \in \mathbf{W},\ E[u(\mathbf{X}_i|\mathbf{w})] \leq E[u(\mathbf{X}_j|\mathbf{w})] \qquad \text{and}\\ &\exists \mathbf{w} \in \mathbf{W},\ E[u(\mathbf{X}_i|\mathbf{w})] < E[u(\mathbf{X}_j|\mathbf{w})]\end{aligned}$$

The pairwise dominance relation of designs i and j can be checked by studying the difference of expected utilities

$$h_{ij}(\mathbf{w}) = E[u(\mathbf{X}_j|\mathbf{w})] - E[u(\mathbf{X}_i|\mathbf{w})] = \sum_{l=1}^{H} w_l \Big(E[u_l(X_{jl})] - E[u_l(X_{il})] \Big). \quad (4)$$

The difference is a linear function with respect to the weights. If the set of feasible weights $\mathbf{W}$ is defined by a set of linear equalities and inequalities, the pairwise dominance can be checked by solving two LP problem: Find the maximum and the minimum of the difference $h_{ij}(\mathbf{w})$ subject to the linear constraints, i.e, $\mathbf{w} \in \mathbf{W}$. Design j dominates design i if and only if the minimum is at least zero and the maximum is greater than zero. Because we are dealing with LP problems, it suffices to check only the extreme points of the feasible weights, denoted as $\mathbf{W}_{\text{ext}} = \{\mathbf{w}_1, ..., \mathbf{w}_M\}$.[11]

3 Simulation budget allocation with incomplete preference information

If the performance measures are not aggregated, one may try to determine the set of Pareto designs first and then the pairwise non-dominated designs among the Pareto set. However, a weakness of this approach is that there are usually many designs in the Pareto set that are not even close to being pairwise non-dominated according to the incomplete preference information. Hence, computing budget is wasted to solve the Pareto status of such designs accurately.

We present a technique, which allows us to take the advantage of incomplete preference information already in the computing budget allocation stage. The technique is based on studying the Pareto status of the utilities of the designs at the extreme points of $\mathbf{W}$, instead of the performance measures. It is applicable when the set of feasible weights is defined by linear equalities and inequalities.

3.1 Correspondence of dominance and pairwise dominance relations

Let us define the expected utility of ith design at extreme point $\mathbf{w}_m$ of the set of feasible weights $\mathbf{W}$ as

$$U_{im} = E[u(\mathbf{X}_i|\mathbf{w}_m)] = \sum_{l=1}^{H} w_{ml} E[u_l(X_{il})], \tag{5}$$

where w_{ml} is the weight of lth performance measure at the mth extreme point of $\mathbf{W}$.

Instead of maximizing the performance measures $J_1, J_2, ..., J_H$ we can consider the expected utilities at the extreme points $\mathbf{W}_{\text{ext}}$ as the new objectives $U_1, U_2, ..., U_M$. The new R&S problem is then

$$\max_{i \in \Theta} (U_{i1}, ..., U_{iM}), \tag{6}$$

which has a nice property that its Pareto set is the set of pairwise non-dominated designs of the original problem.

Theorem 1. *Assume the utilities of designs are determined according to equation (3) and the set of feasible weights is defined by a set of linear equalities and inequalities. If design i is a non-dominated Pareto design of problem (6), then i is pairwise non-dominated.*

Proof. If i is a non-dominated in problem (6), then there is no j such that

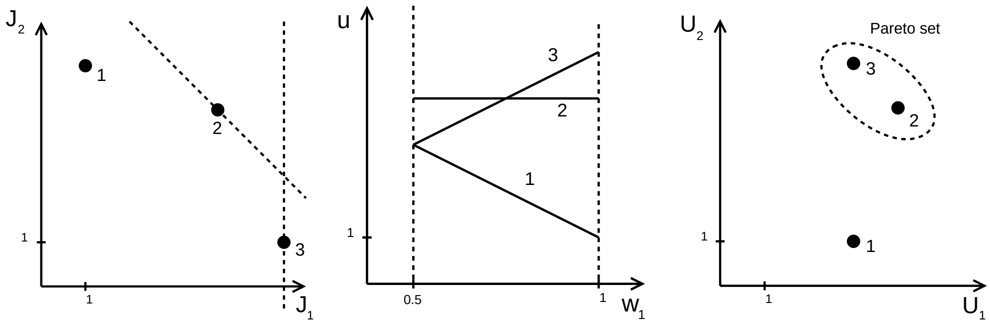
$$\begin{aligned} &\forall m \in \{1,..,M\}, U_{im} \leq U_{jm} \qquad \text{and} \\ &\exists m \in \{1,..,M\}, U_{im} < U_{jm}, \end{aligned}$$

which is equivalent to

$$\begin{aligned} &\forall m \in \{1,..,M\}, h_{ij}(\mathbf{w}_m) = E[u(\mathbf{X}_j|\mathbf{w}_m)] - E[u(\mathbf{X}_i|\mathbf{w}_m)] \geq 0 \qquad \text{and} \\ &\exists m \in \{1,..,M\}, h_{ij}(\mathbf{w}_m) = E[u(\mathbf{X}_j|\mathbf{w}_m)] - E[u(\mathbf{X}_i|\mathbf{w}_m)] > 0, \end{aligned}$$

i.e., design i is pairwise non-dominated, as discussed in Section 2.3. □

The principle is illustrated with an example in Figure 1. In this example, there are three designs for which $(J_{11}, J_{12}) = (1, 5)$, $(J_{21}, J_{22}) = (4, 4)$ and $(J_{31}, J_{32}) = (5, 1)$ that are all non-dominated. The incomplete preference information is that $w_1 \geq \frac{1}{2}$, thus the extreme points of feasible weights are $\mathbf{w}_1 = (1, 0)$ and $\mathbf{w}_2 = (\frac{1}{2}, \frac{1}{2})$, and utility functions are identities, i.e., $u_1(x) = u_2(x) = x$. In Figure 1a, the dots illustrate the designs and the dashed lines are contours of the MAU function with extreme weights. In Figure 1b, the utilities of the designs are illustrated as functions of weight w_1. If the utility of one design is better than another with all feasible weights the design pairwise dominates the other. Because the utilities are linear with respect to the weights, it suffices to check only the extreme points of the feasible weights, i.e., the end points of the lines. If a design is pairwise non-dominated, it appears non-dominated in the coordinate system of the utilities of the extreme weights, as seen in Figure 1c.



(a) (b) (c)

Figure 1: (a) Performance measures of three designs. (b) Utilities as a function of weight w_1. (c) Utilities at the extreme points of feasiable weights of the same three designs.

3.2 Multi-objective computing budget allocation procedure

Multi-objective optimal computing budget allocation (MOCBA) procedure is a method for allocating computing budget in order to select the correct Pareto set with a high probability. The MOCBA procedure that is presented in this paper is the same as in [2], with the exception that a maximization problem is considered instead of a minimization problem, which results in some changes in the formulas.

The construction of the observed Pareto set S_{op} is based on the means of simulated performances, denoted as $\bar{J}_{il}$. A design is considered non-dominated, i.e. $i \in S_{op}$, if condition (2) holds for the means of the designs. There are two types of errors that can occur when determining the observed Pareto set.

Type I error:

Type I error occurs when at least one of the designs in the observed non-Pareto set $\bar{S}_{op} = \Theta \backslash S_{op}$ is actually non-dominated. The probability of type I error is denoted by

$$e_1 = 1 - P\Big[\bigcap_{i \in \bar{S}_{op}} E_i^c \Big], \tag{7}$$

where E_i^c is the event that design i is dominated.

Type II error:

Type II error occurs when at least one of the designs in the observed Pareto set S_{op} is actually dominated. The probability of type II error is denoted by

$$e_2 = 1 - P\Big[\bigcap_{i \in S_{op}} E_i \Big], \tag{8}$$

where E_i is the event that design i is non-dominated.

The following lemma provides upper bounds for both error types [2].

Lemma 1. *Type I and Type II errors have the following upper bounds.*

$$e_1 \le ub_1 = H|\bar{S}_{op}| - H \sum_{i \in \bar{S}_{op}} P(\tilde{J}_{j_i l^i_{j_i}} \ge \tilde{J}_{i l^i_{j_i}}),$$
$$e_2 \le ub_2 = (K-1) \sum_{i \in S_{op}} P(\tilde{J}_{j_i l^i_{j_i}} \ge \tilde{J}_{i l^i_{j_i}}),$$

where $\tilde{J}_{il}$ is a random variable representing the lth performance measure of ith design ($\tilde{J}_{il} \sim \mathcal{N}\left(\bar{J}_{il}, \frac{\sigma^2_{il}}{N_i}\right)$, where σ^2_{il} is the variance of lth performance measure of ith design and N_i is the number of replications for design i.) and

$$j_i = \arg \max_{\substack{j \in \Theta \\ j \ne i}} \prod_{l=1}^{H} P(\tilde{J}_{jl} \ge \tilde{J}_{il})$$
$$l^i_{j_i} = \arg \min_{l \in \{1,\ldots,H\}} P(\tilde{J}_{jl} \ge \tilde{J}_{il}).$$

j_i is the design that most likely dominates design i and $l^i_{j_i}$ is the objective of j_i that is least likely better than the corresponding objective of design i.

The objective of the computing budget allocation procedure is to maximize the probability that the determined Pareto set S_{op} is correct, denoted as P_{CS}. However, the probability P_{CS} cannot be expressed with an explicit equation, but we can still establish a lower bound for it [2].

Lemma 2. *APCS-M is the lower bound for P_{CS}, where*

$$APCS{-}M = 1 - ub_1 - ub_2$$

Instead of maximizing the probability of correct selection P_{CS} we maximize the lower bound of P_{CS}. We consider the following approximate multi-objective computing budget allocation problem.

$$\max_{N_1,\ldots,N_K} APCS{-}M \tag{9}$$
$$\text{s.t. } N_1 + N_2 + \ldots + N_K = T \quad \text{and} \quad N_i \ge 0,$$

where N_i is the number of replications allocated for design i and T is the total computing budget.

An allocation rule is derived in [2], which asymptotically maximizes the $APCS{-}M$. The performance measures are assumed to be independently distributed across different replications, different designs and different performance measures of the same design. The derivation of the allocation rule is based on applying the method of Lagrange multipliers to problem (9) and assuming $T \to \infty$. The allocation rule provides asymptotically optimal proportions of computing budget for each design, denoted as α_i. The allocation quantities are then given by $N_i = \alpha_i T$. The following lemma is a simplified version of the allocation rule.[2] (Note that the allocation rule is modified for a maximization problem.)

Lemma 3. *The asymptotic allocation rule can be approximated as follows.*

$$\text{For } h, m \in S_A,\ \frac{\alpha_h}{\alpha_m} = \left(\frac{\sigma_{hl^h_{j_h}} / \delta_{hj_hl^h_{j_h}}}{\sigma_{ml^m_{j_m}} / \delta_{mj_ml^m_{j_m}}} \right)^2. \tag{10}$$

$$\text{For } d \in S_B,\ \alpha_d^2 = \sum_{h \in \Theta_d^*} \frac{\sigma^2_{dl^h_d}}{\sigma^2_{hl^h_d}} \alpha_h^2, \tag{11}$$

where

$$\delta_{ijl} = \bar{J}_{jl} - \bar{J}_{il}, \tag{12}$$

$$l^i_j = \arg \min_{l \in \{1,\dots,H\}} P(\tilde{J}_{jl} \geq \tilde{J}_{il}) = \arg \min_{l \in \{1,\dots,H\}} \frac{\delta_{ijl}|\delta_{ijl}|}{\sigma^2_{il} + \sigma^2_{jl}}, \tag{13}$$

$$j_i = \arg \max_{\substack{j \in \Theta \\ j \neq i}} \prod_{l=1}^{H} P(\tilde{J}_{jl} \geq \tilde{J}_{il}) = \arg \max_{\substack{j \in \Theta \\ j \neq i}} \frac{\delta_{ijl^i_j}|\delta_{ijl^i_j}|}{\sigma^2_{il^i_j} + \sigma^2_{jl^i_j}}, \tag{14}$$

$$S_A = \left\{ h | h \in S, \frac{\delta^2_{hj_hl^h_{j_h}}}{\sigma^2_{hl^h_{j_h}} + \sigma^2_{j_hl^h_{j_h}}} < \min_{i \in \Theta_h} \frac{\delta^2_{ihl^i_h}}{\sigma^2_{il^i_h} + \sigma^2_{hl^i_h}} \right\}, \tag{15}$$

$$S_B = S \backslash S_A, \tag{16}$$

$$\Theta_h = \{i | i \in S, j_i = h\}, \quad \Theta_d^* = \{h | h \in S_A, j_h = d\}, \tag{17}$$

where m in Eq. (10) is any fixed design in S_A. The simplified allocation rule classifies the designs into two sets S_A and S_B. The designs in S_A play the role of being dominated and the designs in S_B play the role of dominating. The computing budget allocation of a design depends on the role that it plays.

3.3 MOCBA procedure with incomplete information

The set of pairwise non-dominated designs can be solved by considering the utilities of the designs at the extreme points of $\mathbf{W}$ instead of the performance measures, as described in Section 3.1. The utilities at the extreme weights U_{im} are the new objectives that correspond to J_{il} in the MOCBA procedure. We can simulate the performance measures of a design and evaluate the corresponding simulated value of utility $u(\mathbf{X}_i|\mathbf{w}_m)$. The mean of simulated utilities, denoted as $\bar{U}_{im}$, is the estimate of the expected utility U_{im}. The MOCBA procedure is then applied simply by replacing $\bar{J}_{il}$ by $\bar{U}_{im}$ in the allocation rule formulas and taking into account that we now have M objectives instead of H (Index m in $\bar{U}_{im}$ corresponds to l in $\bar{J}_{il}$).

In the derivation of the allocation rule, it is assumed that the different performance measures of the same design are independent. Unfortunately, the utilities $(U_{i1}, ..., U_{iM})$ corresponding to the performance measures are not independent. The utilities U_{im} of the same design depend on the same independently distributed random variables $(X_{i1}, ..., X_{iH})$, which implies that the utilities are correlated. The more accurate the preference information is, the closer $\mathbf{w}_m$ are to each other and the more correlated the utilities are.

If there is no preference information at all, then $M = H$ and $\forall\, m : \ U_{im} = J_{im}$, which means that the computing budget allocation procedure is the normal MOCBA procedure. In case of exact preference information, the set of feasible weights consists of a single weight and the computing budget allocation procedure is the same as the OCBA procedure.

4 Numerical results

We study the performance of the simplified version of the MOCBA procedure (presented in Lemma 3) in determining the pairwise non-dominated designs of a test problem. The procedure is compared to equal computing budget allocation to determine how much the probability of the correct selection of pairwise non-dominated designs improves. We also study how the procedure allocates the computing budget.

4.1 Test problem

Let us consider a R&S problem with two performance measures, in which the expected performances of fifteen designs are located on the four arches of circles, as illustrated in Figure 2 on the left. The standard deviation of the simulated performance measures of each design is 0.5 for both performance measures. The utility functions of both performance measures are identities, i.e., $u_1(x) = u_2(x) = x$. The set of feasible weights of the test problem is chosen so that the set of pairwise non-dominated designs consists of designs 3 and 4 as clearly as possible. The extreme weights are $\mathbf{w}_1 = (0.5, 0.5)$ and $\mathbf{w}_2 = (0.634, 0.366)$. The utilities of the designs at the extreme weights are illustrated in Figure 2 on the right.

4.2 Performance of simplified MOCBA with incomplete information

The performance of the simplified MOCBA procedure is compared to the equal computing budget allocation by solving the test problem described in Section 4.1 with both methods. The probability of correct selection of pairwise non-dominated designs is estimated by simulating the computing budget allocation procedure 1000 times and calculating the number of cases where correct selection was made.

The simplified MOCBA procedure is applied iteratively with different computing budgets. After each iteration the allocation quantities are evalu-

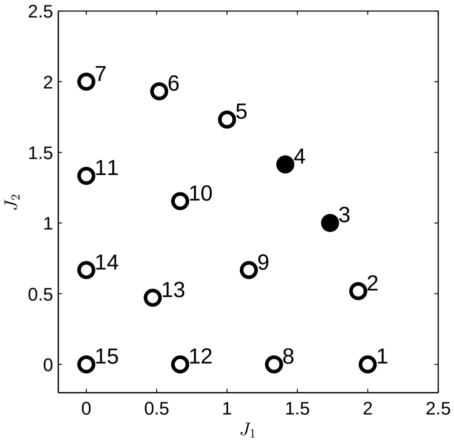


Figure 2: Performance measures of designs on the left. Utilities of extreme weights on the right. Pairwise non-dominated designs are denoted by filled circles.

ated with new means and variances of the performance measures. Initially, the performance measures of the designs are simulated with 20 replications, totaling 300 replications altogether. Then the computing budget is increased with increments of 100 replications, which are allocated according to the allocation rule with the limitation that no design gets more than 40 replications. If the whole computing budget of the iteration step is not used due to the 40 replication per design limit, new iterations with updated means and variances are carried out with the leftover computing budget until the whole computing budget of 100 replications is used.

New allocations in some iterations may be such that some designs are supposed to get less replications than they already have. Then the total number of suggested new replications for the rest of the designs exceeds the total computing budget of the iteration. If this happens, the computing budget of the iteration is decreased so that the total number of new replications suggested by the allocation rule equals the real computing budget.

The estimated probabilities of correct selection are illustrated in Figure

3. The success probability of the simplified MOCBA procedure was 93.6% with a 10000 replication total computing budget. It outperforms the equal allocation in small computing budgets, but surprisingly the equal allocation is better with larger computing budgets. With the equal allocation, the probability of correct selection approaches 100 percent. This does not seem to be the case with the simplified MOCBA procedure. In some cases, some of the critical designs do not get replications at all, no matter how large the computing budget is.

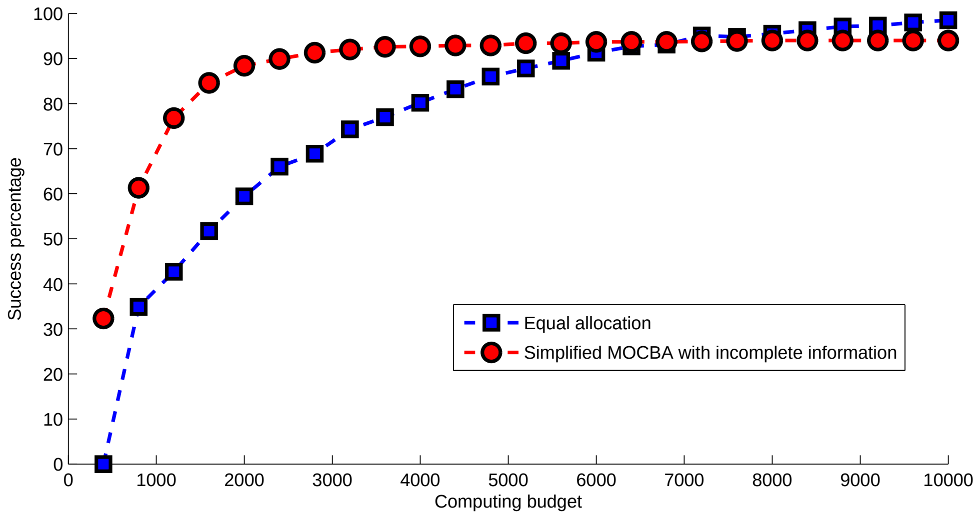


Figure 3: Success probability of correct selection for equal allocation and simplified MOCBA procedure with incomplete information.

4.3 Computing budget allocations

We next study how the simplified MOCBA procedure allocates the computing budget among the designs, when it is applied with incomplete information to determine the pairwise non-dominated designs. The average computing budget allocation of the final iteration round is calculated over 1000 simulations, which is illustrated in Figure 4.

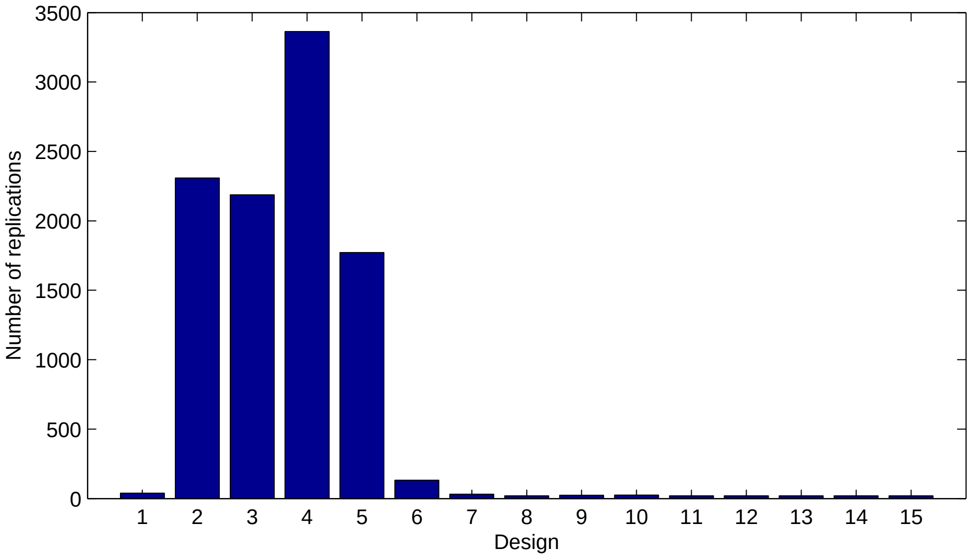


Figure 4: Average computing budget allocation

96.2 percent of the total computing budget is allocated to designs 2, 3, 4 and 5, while not a single additional replication after the 20 initial replications is allocated for designs 12, 13, 14 and 15 in any of the 1000 simulations. Most replications are allocated to the pairwise non-dominated designs 3 and 4 and to designs 2 and 5 that are close to being pairwise non-dominated. A very small proportion of the total computing budget is allocated for designs 1, 6 and 7, although they are Pareto designs. Our technique concentrates the computing budget to those designs that the DM is interested in, whereas the normal MOCBA procedure without preference information would have allocated considerably more computing budget for designs 1, 6 and 7.

4.4 Considerations for improvement

The computing budget allocations of unsuccessful simulations, are examined to find out why the simplified MOCBA determines the set of pairwise dominated designs incorrectly relatively often with extensive computing budgets.

Table 1: Computing budgets of ten unsuccessful simulations.

	Designs 1	2	3	4	5	6
1	20	20	4841	20	4895	24
2	80	4617	26	4572	465	60
3	20	4596	20	4620	544	20
4	20	4871	23	4844	32	30
5	20	4877	20	4861	20	22
6	60	141	4765	27	4768	59
7	20	4540	21	4554	665	20
8	20	4808	20	4847	105	20
9	20	4173	40	4120	1447	20
10	160	4767	34	4818	21	20

The computing budgets of designs 1-6 in ten unsuccessful simulations are listed in Table 1. In each of the simulations, either design 3 or 4 has a very low number of allocated replications. The design with few replications is misclassified in each of the simulations and usually some of the pairwise dominated designs are also misclassified. When studied with a larger number of simulations than the sample of ten simulations presented here, it seemed that design 3 tends to be misclassified more often than design 4, which might be the reason why less replications are allocated for design 3 than for design 2 on average.

Sometimes the simplified MOCBA procedure suggested exactly zero allocation quantity α_i for either of the non-dominated designs. Further examination revealed that it is possible in the MOCBA framework that some design i belongs to S_b, the set of designs that play the role of dominating other designs, but Θ_i^* is empty, meaning that there is no design for which it is the most likely dominator. In that case α_i equals zero, because there is no terms in the right side of Eq. (11). However, this only accounts for roughly half of the misclassifications.

In the other half of the unsuccessful simulations the suggested allocation quantity α_i of the misclassified design was very low but not zero. It is supposed that this happens when, for example, design 3 is far enough from being in the set of pairwise non-dominated after the initial replications. Then designs 2 and 4 have equal expected utility with weights $\mathbf{w}_2$ and supposedly the whole computing budget is used for finding out if design 2 is pairwise non-dominated. The issue may be addressed by setting a minimum for the amount of new replications for each design in each iteration, which would guarantee that as the total computing budget and the number of iterations increase also the number of allocated replications for each design increases and the means of the performance measures will approach their expected values. It should be also studied if the "full" version of MOCBA procedure (presented in [2]) performs better than the simplified version.

To find out the impact of the correlation of the utilities, the following test is done. The MOCBA procedure is compared to equal allocation in another test problem, where the expected performances were at the same coordinates as the utilities in the original test problem (See the diagram on the right in Figure 2) and the standard deviations of the performances were the same as the standard deviations of the utilities in the original problem, i.e., they are calculated as

$$\mathrm{Var}(U_{im}) = \mathrm{Var}(w_{m1}X_{i1} + w_{m2}X_{i2}) = w_{m1}^2\mathrm{Var}(X_{i1}) + w_{m2}^2\mathrm{Var}(X_{i2}), \quad (18)$$

which results in variances 0.125 and 0.134 for the utilities at $\mathbf{w}_1$ and $\mathbf{w}_2$ respectively. This is the same problem as the original test problem with the exception that the performance measures of the same design are uncorrelated.

The estimated probabilities of correct selection for this test problem are illustrated in Figure 5. The equal computing budget allocation performs equally well in both problems, while the simplified MOCBA procedure performs significantly better in this problem than in the original problem, which implies that the correlations of the performances have an impact on the performance of the simplified MOCBA procedure.

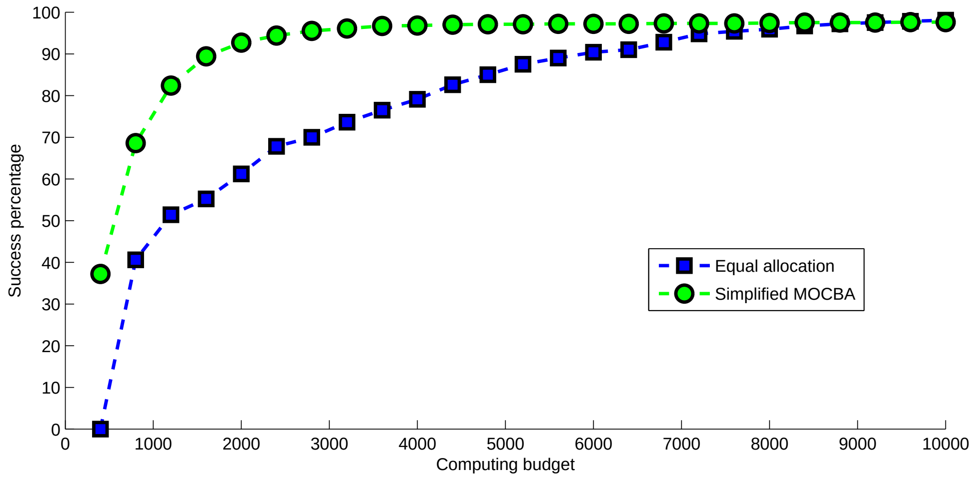


Figure 5: Success probability of correct selection for equal allocation and MOCBA procedure in the test problem with uncorrelated performances.

5 Conclusions

In this paper, we presented a technique for incorporating incomplete preference information into the optimal computing budget allocation framework. Our technique is applicable, when the preferences of a DM are expressed with an additive MAU function and the incomplete information is related to the weights of the MAU function such that the set of feasible weights is determined by linear equalities and inequalities. The technique allows us to utilize existing multi-objective computing budget allocation procedures, which are designed for determining the Pareto set, for finding the set of pairwise non-dominated designs.

The technique was tested with the simplified MOCBA procedure for finding two pairwise non-dominated designs among fifteen designs. The results were not completely satisfactory as our technique was outperformed by equal allocation with extensive computing budgets, though it was better with smaller computing budgets. However, this issue is not related to our technique for incorporating incomplete information into the computing budget allocation procedure, but to the simplified MOCBA procedure. The simplified MOCBA procedure is prone to fail in the early stage of the iterative budget allocation, in which case no additional replications are allocated to some potentially non-dominated designs, no matter how large the computing budget is.

If the performance measures of a given design are correlated, the simplified MOCBA procedure succeeds with smaller probability. Our technique requires solving the Pareto set, in which the objectives of any given design are highly correlated. The more exact the incomplete information is the more correlated the utilities are. Hence, a computing budget allocation procedure that can deal with correlated performance measures is necessary for our techinque to perform well.

In further research our technique should be tested with other computing budget allocation procedures that are less distracted by the correlations of the performance measures. One of the promising methods is the indifference zone method presented in [8]. It does not take into account the correlations of the performance measures, but it is generally more robust. Also the performance of the "full" MOCBA procedure should be studied with correlated performance measures.

References

[1] J. C. Butler, D. J. Morrice and P. W. Mullarkey. A Multiple Utility Theory Approach to Ranking and Selection. *Management Science*, 47(6):800-816, 2001

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

[3] C. H. Chen, E. Yücesan and S. E. Chick. Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Discrete Event Dynamics: Theory and Applications*, 10:251-270, 2000

[4] R. L. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Wiley, 1976

[5] S.-H. Kim, B. L. Nelson. Recent advances in ranking and selection. In *Proceedings of the 2007 Winter Simulation Conference*, pages 162-172, Piscataway, NJ, 2007. Institute of Electrical and Electronics Engineers Inc.

[6] 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*, 189:476-491, 2008

[7] L. H. Lee, E. P. Chew, S. Teng and D. Goldsman. Optimal Computing Budget Allocation for Multi-objective simulation models, In *Proceedings of the 2004 Winter Simulation Conference*

[8] S. Teng, L. H. Lee and E. P. Chew. Integration of indifference-zone with multi-objective computing budget allocation. *European Journal of Operational Research*, 203(2):419-429, 2010

[9] D. J. Morrice and J. C. Butler. Ranking and selection with multiple "targets". In *Proceedings of the 2006 Winter Simulation Conference*

[10] S. Teng, L. H. Lee and E. P. Chew. Multi-objective ordinal optimization for simulation optimization problems. *Autonomica*, 43:1884-1895, 2007

[11] M. Weber. Decision making with incomplete information. *European Journal of Operational Research*, 28(1):44-57, 1987