

MULTI-OBJECTIVE MULTI-FIDELITY OPTIMIZATION WITH ORDINAL TRANSFORMATION AND OPTIMAL SAMPLING

Haobin Li

Institute of High Performance Computing
Department of Computing Science
1 Fusionopolis Way, 138632, SINGAPORE

Yueqi Li

Loo Hay Lee
Ek Peng Chew

Department of Industrial and Systems Engineering
National University of Singapore
1 Engineering Drive 2
117576, SINGAPORE

Giulia Pedrielli

Centre for Maritime Studies
National University of Singapore
12 Prince George's Park
118411, SINGAPORE

Chun-Hung Chen

Dept. of Systems Engineering & Operations Research
George Mason University
4400 University Drive, MS 4A6
Fairfax, 22030 VA, USA

ABSTRACT

In simulation–optimization, the accurate evaluation of candidate solutions can be obtained by running a high–fidelity model, which is fully featured but time–consuming. Less expensive and lower fidelity models can be particularly useful in simulation–optimization settings. However, the procedure has to account for the inaccuracy of the low fidelity model. Xu et al. (2015) proposed the MO²TOS, a Multi–fidelity Optimization (MO) algorithm, which introduces the concept of ordinal transformation (OT) and uses optimal sampling (OS) to exploit models of multiple fidelities for efficient optimization. In this paper, we propose MO–MO²TOS for the multi–objective case using the concepts of non–dominated sorting and crowding distance to perform OT and OS in this setting. Numerical experiments show the satisfactory performance of the procedure while analyzing the behavior of MO–MO²TOS under different consistency scenarios of the low–fidelity model. This analysis provides insights on future studies in this area.

1 INTRODUCTION

Simulation models have been widely applied for the analysis and optimization of complex system. This has been particularly true in the past decades, also due to the explosive revolution of computing technology, which has greatly facilitated the realization of running simulation models. In this setting, simulation–optimization has received a remarkable attention as an effective set of techniques coupling simulation with search procedures to identify strategies or designs leading to the best performance. Specifically, a search algorithm which selects candidate solution(s) and a simulator which evaluates the objective performance of selected solutions. These two components are iteratively called until a specific stopping criteria is met (Fu et al. 2005).

However, in case simulation models have a high level of detail (i.e., they are *high-fidelity* models), despite the accuracy in the performance estimate, they result in high computational effort. However, the

available simulation budget is limited in nature and this results in the possibility of evaluating only a small fraction of candidates. If less detailed simulation models are available (i.e., *low-fidelity* models), these might be run over a more extended set of solutions and the results might be used in combination and in support of the, few, expensive simulations. Low fidelity models are by construction less accurate than the high fidelity counterpart. Intuitively, the effectiveness of an approach which uses low fidelity models is remarkably influenced by their consistency with the high-fidelity counterpart (Xu et al. 2014, Xu et al. 2015): if the low-fidelity model contains significant bias, misleading results may be obtained. Therefore, Xu et al. (2014) proposed Multi-fidelity Optimization with Ordinal Transformation and Optimal Sampling (MO²TOS), a framework for single objective simulation–optimization problems which combines low and high-fidelity simulation. Specifically, it makes use of ordinal ranking of low-fidelity model observations based on which samples are selected for high-fidelity evaluation. Xu et al. (2015) shows that this procedure can efficiently perform optimization by reducing variances within a group of candidate solutions and enlarged distances between groups.

Nevertheless, currently MO²TOS can manage multi-objective optimization only in case the multiple objectives are scalarized into a single value, which is possible only if the relative preference for each objective is available. However, such preferences are not easy to define. Several algorithms have been proposed for finding the Pareto set, e.g., NSGA-II (Deb et al. 2000), MO-PSO (Lee et al. 2014) and MO-COMPASS (Li et al. 2015), but none them considers the possibility of using evaluations from models with different fidelities.

In this paper, we firstly define the problem, and introduce some preliminary knowledge on the Pareto optimality and the generic framework for the MO²TOS algorithm. Then we develop the multi-objective MO²TOS (MO-MO²TOS) and analyze it through numerical experiments.

2 PROBLEM DEFINITION

We refer to the family of deterministic multi-objective optimization problems defined on any finite solution space Θ , i.e., $\|\Theta\| < \infty$, such that for any solution $\mathbf{x} \in \Theta$, we have a high-fidelity simulation model that can be used to evaluate the performance with sufficient accuracy, i.e., without bias and noise. We denote the high-fidelity results as $\mathbf{g}(\mathbf{x}) = [g^{(1)}(\mathbf{x}), \dots, g^{(H)}(\mathbf{x})]$, where H indicates the number of objectives.

According to the definition of Pareto optimality, we are aiming to find the Pareto set Π on Θ , namely (Li et al. 2015):

$$\Pi(\Theta) \equiv \{\mathbf{x} \in \Theta \mid \forall \mathbf{y} \in \Theta, \mathbf{g}(\mathbf{y}) \prec \mathbf{g}(\mathbf{x})\}, \quad (1)$$

where, ‘ \prec ’ denotes the dominance, i.e., $\mathbf{g}(\mathbf{y}) \prec \mathbf{g}(\mathbf{x})$ if and only if

$$\forall l \in \{1, \dots, H\}, g^{(l)}(\mathbf{y}) \leq g^{(l)}(\mathbf{x}) \text{ and } \exists l \in \{1, \dots, H\}, g^{(l)}(\mathbf{y}) < g^{(l)}(\mathbf{x}). \quad (2)$$

The high-fidelity model is time consuming, i.e., obtaining $\mathbf{g}(\mathbf{x})$ is computationally costly. However, a low-fidelity simulation model is available giving an estimate of the true performance in a remarkably shorter time than the high-fidelity one. We denote the low-fidelity result as $\tilde{\mathbf{g}}(\mathbf{x})$, $\forall \mathbf{x} \in \Theta$, and we model the low-fidelity bias as follows (Xu et al. 2015):

$$\tilde{\mathbf{g}}(\mathbf{x}) = \mathbf{g}(\mathbf{x}) + \boldsymbol{\delta}_{\mathbf{x}}, \quad (3)$$

where $\boldsymbol{\delta}_{\mathbf{x}}$ represents the bias of the low fidelity model corresponding to the feasible solution \mathbf{x} .

Therefore, in this study, we propose an efficient procedure to exploit simulation models at both fidelity levels in order to identify an estimated Pareto set denoted by $\hat{\Pi}$ when the computational budget in terms of number of high-fidelity evaluation is limited and remarkably lower than the number of candidate solutions. Formally, if $S \subseteq \Theta$ is the set of solutions that are sampled for high-fidelity evaluation according to the procedure, the resulting estimated Pareto Set is:

$$\hat{\Pi}(\Theta) = \{\mathbf{x} \in S \mid \forall \mathbf{y} \in S, \mathbf{g}(\mathbf{y}) \prec \mathbf{g}(\mathbf{x})\}. \quad (4)$$

3 MO-MO²TOS FRAMEWORK

In Xu et al. (2015), the MO²TOS algorithm is proposed for solving simulation–optimization problem with single–objective. Grounding on this proposal, we propose the extension in Algorithm 1 that consists of the same two fundamental steps as MO²TOS: Ordinal Transformation (OT) as in Step 4, and the Optimal Sampling (OS) as in Step 8 & 9. In particular, we modify the two essential procedures to make them suitable for solving multi-objective problems.

Algorithm 1: Generic Procedure for MO²TOS Algorithm

```

1 forall the  $\mathbf{x} \in \Theta$  do
2   | Apply low-fidelity model to evaluate  $\mathbf{x}$ , and obtain  $\tilde{\mathbf{g}}(\mathbf{x})$ ;
3 end
4 (OT): Sort all  $\mathbf{x} \in \Theta$  considering  $\tilde{\mathbf{g}}(\mathbf{x})$ , as  $\Theta = \{\mathbf{x}_{(1)}, \dots, \mathbf{x}_{(|\Theta|)}\}$ ;
5 Following the sequence, evenly divide  $\Theta$  into  $m$  groups, i.e., let  $K = \{1, \dots, m\}$  and

$$\Theta_k = \left\{ \mathbf{x}_{(\lfloor \frac{k-1}{m} \cdot |\Theta| + 1 \rfloor)}, \dots, \mathbf{x}_{(\lfloor \frac{k}{m} \cdot |\Theta| \rfloor)} \right\}, \forall k \in K;$$

6 Let  $S = \emptyset$ ;
7 while not stopped do
8   | (OS-1): Sample  $k^* \in K$ , considering  $\mathbf{g}(\mathbf{x}), \forall \mathbf{x} \in \Theta_k \cap S, \forall k \in K$ ;
9   | (OS-2): Sample  $\mathbf{x}^* \in \Theta_{k^*}$ , considering  $\tilde{\mathbf{g}}(\mathbf{x}), \forall \mathbf{x} \in \Theta_{k^*} \setminus S$ ;
10  | Apply high-fidelity model to evaluate  $\mathbf{x}^*$ , and obtain  $\mathbf{g}(\mathbf{x}^*)$ ;
11  |  $S \leftarrow S \cup \{\mathbf{x}^*\}$ ;
12 end
13 With  $S$ , return  $\hat{\Pi}(\Theta)$  by (4).

```

For OT, the low–fidelity model is used to derive a relative relationship (ranking) among candidate solutions. Since the time for low–fidelity model is assumed to be negligible and the solution set is finite, the low-fidelity model is run for all the candidates $\mathbf{x} \in \Theta$, which are subsequently ordered according to increasing values of $\tilde{\mathbf{g}}(\mathbf{x})$. Although affected by bias, we use the low–fidelity to transform the solution space to the 1-dimensional ordinal space according to Xu et al. (2015). In a single–objective case, once $\tilde{\mathbf{g}}(\mathbf{x})$ is computed through the low–fidelity simulator the ordinal transformation ranks \mathbf{x} based on the $\tilde{\mathbf{g}}(\mathbf{x})$ itself. However, in the multi–objective case, such a transformation is complicated by the need to consider the concept of *dominance*. The solution of this issue represents the most important aspect in extending the MO²TOS to its multi–objective version.

Once the dominance–based OT has been performed, we can exploit the efficiency of the optimal sampling scheme in Xu et al. (2015) to select the candidates for high–fidelity evaluations. Specifically, OS is performed by grouping solutions according to the ranking resulting from OT. Compared with a random grouping, the variability within a group of solutions decreases and differences between groups increase. With OS, two decisions are to be made at every iteration, i.e., which group to select, and, within the selected group, which solution to sample for high–fidelity evaluation. Moreover, from iteration to iteration, the evaluated high-fidelity results are used to update the expected quality of each group of solutions.

In the next few sections we detail how to implement OT and OS for the multiple–objective case. Herein, we do not consider scalarizing methods since they lead to the possibility to use MO²TOS and no extension is required in this case. The same holds for the hyper-volume concept in the objective space that is dominated by a multi-objective solution (Bradstreet et al. 2008), and the solutions with larger dominated hyper-volume (DHV) have higher order in the ranking. This is relatively fair comparison between solutions that are non-dominated by each other. But, the DHV does not explicitly reflect the distribution of solutions on the Pareto front. Besides, boundary points need to be identified to make the DHV finite, and the selection of them could also affect the ordering.

3.1 Ordinal Transform

Several ways for ordering solutions with multiple objectives have been proposed in the literature (Swisher et al. 2003, Zitzler et al. 2003).

A particularly common method is the non-dominated sorting (NS), proposed by Deb et al. (2000), based on which NSGA-II is developed (Deb et al. 2000). According to this concept, solutions are first ranked according to the Pareto layers, and one dominating solution has higher rank than the dominated; then, within each Pareto layer where solutions are non-dominated by each other, the crowding distance is used to rank the solutions. The crowding distance reflects the density around a solution in the corresponding Pareto layer, and the solution with lower density has higher rank in the layer.

Algorithm 2: Procedure for calculating the crowding-distance in Pareto layer P

```

1 forall the  $\mathbf{x} \in P$  do
2   |    $d(\mathbf{x}) \leftarrow 0;$ 
3 end
4 forall the  $l \in \{1, \dots, H\}$  do
5   |    $\Delta^{(l)} \leftarrow \max_{\mathbf{x} \in \Theta} \hat{\mathbf{g}}^{(l)}(\mathbf{x}) - \min_{\mathbf{x} \in \Theta} \hat{\mathbf{g}}^{(l)}(\mathbf{x});$ 
6   |   Sort all  $\mathbf{x} \in P$  by  $\hat{\mathbf{g}}^{(l)}(\mathbf{x})$  in ascending order, as  $P = \{\mathbf{x}_{(1)}, \dots, \mathbf{x}_{(|L|)}\};$ 
7   |    $d(\mathbf{x}_{(1)}) \leftarrow \infty$  and  $d(\mathbf{x}_{(|L|)}) \leftarrow \infty;$ 
8   |   forall the  $i \in \{2, \dots, |L|-1\}$  do
9     |     |    $d(\mathbf{x}_{(i)}) \leftarrow d(\mathbf{x}_{(i)}) + \frac{\hat{\mathbf{g}}^{(l)}(\mathbf{x}_{(i+1)}) - \hat{\mathbf{g}}^{(l)}(\mathbf{x}_{(i-1)})}{\Delta^{(l)}}$ 
10  |   end
11 end
12 Return  $d(\mathbf{x}), \forall \mathbf{x} \in P.$ 

```

To be more specific, in MO²TOS, given any solution set Ω , we assume $\hat{\mathbf{g}}(\mathbf{x})$ is the observed objective values for \mathbf{x} , i.e., $\hat{\mathbf{g}}(\mathbf{x}) \leftarrow \mathbf{g}(\mathbf{x})$ for high-fidelity evaluation, and $\hat{\mathbf{g}}(\mathbf{x}) \leftarrow \tilde{\mathbf{g}}(\mathbf{x})$ for low-fidelity evaluation; then P is any Pareto layer that contains solutions that are non-dominated by each other, i.e.,

$$P \subseteq \Omega, \text{ s.t. } \nexists \mathbf{x}, \mathbf{y} \in P, \hat{\mathbf{g}}(\mathbf{x}) \prec \hat{\mathbf{g}}(\mathbf{y}) \quad (5)$$

For each P , the procedure to compute the crowding-distance $d(\mathbf{x}), \forall \mathbf{x} \in P$ works as in Algorithm 2. Subsequently, the NS rule can be described as follows:

$$\begin{aligned} \Omega &= \{\mathbf{x}_{(1)}, \dots, \mathbf{x}_{(|\Omega|)}\}, \\ \text{s.t. } 1 \leq i < j \leq |\Omega| \implies \hat{\mathbf{g}}(\mathbf{x}_{(j)}) &\not\prec \hat{\mathbf{g}}(\mathbf{x}_{(i)}) \text{ and } d(\mathbf{x}_{(i)}) \leq d(\mathbf{x}_{(j)}). \end{aligned} \quad (6)$$

The NS rule can be applied for the Step 4 in Algorithm 1 by letting $\Omega \leftarrow \Theta$ and $\hat{\mathbf{g}}(\mathbf{x}) \leftarrow \tilde{\mathbf{g}}(\mathbf{x}), \forall \mathbf{x} \in \Theta$, so as to form an OT procedure for multi-objective problems.

3.2 Optimal Sampling

Given an ordered set of items, solutions or groups, we propose to sample according to a truncated geometric scheme parametrized over $p \in [0, 1]$. Let Φ be the ordered set containing the elements $a_{(i)}, \forall i \in \{1, \dots, |\Phi|\}$, where i refers to the order. The truncated geometric sampling (TGS) is described in Algorithm 3. With this procedure, we guarantee that the higher rank (smaller i) items have larger probabilities to be sampled. The TGS is used to developed both sub-procedures for the OS. As in Step 8 of Algorithm 1 (OS-1), we need to sample a group Θ_{k^*} from $\{\Theta_1, \dots, \Theta_m\}$. Algorithm 4 shows how TGS is utilized with an algorithm parameter $p_g \in [0, 1]$, in sampling $k^* \in K$ considering $\mathbf{g}(\mathbf{x}), \forall \mathbf{x} \in \Theta_k \cap S, \forall k \in K$. The procedure firstly

Algorithm 3: Procedure of truncated geometric sampling (TGS)

```

1 while true do
2   forall the  $i \in \{1, \dots, |\Phi|\}$  do
3     | Sample  $u$  from a uniform distribution on  $[0, 1]$ ;
4     | if  $u < p$  then
5       |   | Return  $a_{(i)}$ .
6     | end
7   | end
8 end

```

ensures that each group is sampled at least once (Step 1 – 4). Then the groups are sorted according to the averaged ranking performed using the high-fidelity results and TGS is applied to select the k^* (Step 6 – 9).

Algorithm 4: Procedure of sampling $k^* \in K$ (OS-1)

```

1 forall the  $k \in K$  do
2   | if  $\Theta_k \cap S = \emptyset$  then
3     |   | Return  $k^* = k$ .
4   | end
5 end
6 Sort  $S$  according to (6), in which  $\Omega \leftarrow S$  and  $\hat{\mathbf{g}}(\mathbf{x}) \leftarrow \mathbf{g}(\mathbf{x}), \forall \mathbf{x} \in S$ ;
7 Let  $I(\mathbf{x})$  denotes the order of  $\mathbf{x}$  in  $S$ , and  $I_k \leftarrow \sum_{\mathbf{x} \in \Theta_k \cap S} I(\mathbf{x}) / |\Theta_k \cap S|, \forall k$ ;
8 Sort  $K$  according to  $I_k$  in ascending order, as  $K = \{k_{(1)}, \dots, k_{(m)}\}$ ;
9 Sample  $k^*$  from  $K$  by TGS as in Algorithm 3, by  $\Phi \leftarrow S$  and  $p \leftarrow p_g$ ;
10 Return  $k^*$ .

```

With TGS defined in Algorithm 3, the sampling of $\mathbf{x}^* \in \Theta_{k^*}$ as in Step 9 of Algorithm 1 (OS-2) can be straight-forward, by applying TGS with $\Phi \leftarrow \Theta_{k^*} \setminus S$ and $p \leftarrow p_s$, where p_s is another algorithm parameter. The sequence of all $\mathbf{x} \in \Theta_{k^*}$ remains as by OT.

Therefore, we have defined a procedure of OS for multi-objective problems.

4 NUMERICAL RESULTS

Numerical experiments on MO-MO²TOS algorithm were conducted in order to: (1) understand the effect of the different algorithm parameters over the performance, (2) compare the performance of the procedure against random search (i.e., an algorithm which does not use the low-fidelity model information).

Specifically, we focus on an optimization problem defined on the solution space $\Theta = \{1, \dots, N\}$ where $N = 10000$. Each point is associated with the high-fidelity value:

$$\mathbf{g}(\mathbf{x}) = [g^{(1)}(\mathbf{x}), g^{(2)}(\mathbf{x})] \quad (7)$$

where $g^{(1)}(\mathbf{x})$ and $g^{(2)}(\mathbf{x})$ are randomly generated from a standard normal distribution, i.e., $\mathcal{N}(0, 1)$, for all $\mathbf{x} \in \Theta$. Despite the high-fidelity evaluation is trivial, it is good enough to provide a testbed scenario for finding Pareto solutions. Figure 1 shows the distribution of the solutions in the objective space. A spectrum of colors are used to indicate different Pareto Layers, where the red color denotes the Pareto set. Then, each solution has an low-fidelity estimation as $\tilde{\mathbf{g}}(\cdot)$:

$$\tilde{\mathbf{g}}(\mathbf{x}) = [\tilde{g}^{(1)}(\mathbf{x}), \tilde{g}^{(2)}(\mathbf{x})]. \quad (8)$$

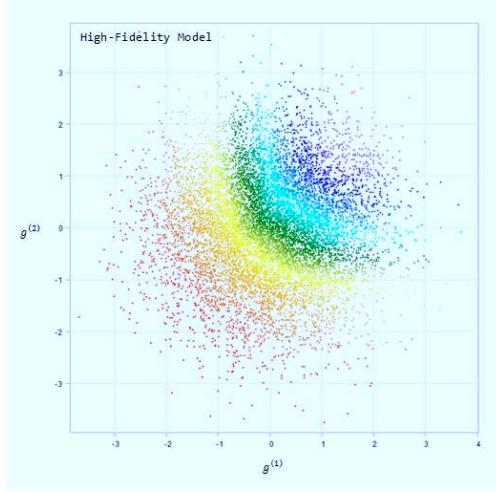


Figure 1: Objective values of high-fidelity model (colored for Pareto layers).

According to the definition in Section 2, we randomly assign the low-fidelity values from the high-fidelity model according to the following:

$$\tilde{g}^{(1)}(\mathbf{x}) = \alpha_1 \cdot g^{(1)}(\mathbf{x}) + \varepsilon \text{ in which } \varepsilon \sim \mathcal{N}(0, \sigma_1^2), \quad (9)$$

$$\tilde{g}^{(2)}(\mathbf{x}) = \alpha_2 \cdot g^{(2)}(\mathbf{x}) + \varepsilon \text{ in which } \varepsilon \sim \mathcal{N}(0, \sigma_2^2). \quad (10)$$

This artificial construction of the low-fidelity response give us the chance, as it will be illustrated later on, to construct scenarios with different ‘‘quality’’ of the low-fidelity model.

Intuitively, a better low-fidelity model has both coefficients α_1, α_2 positive, i.e., the low-fidelity model is indeed consistent with respect to the high-fidelity model. But, in case $\alpha_1 < 0$ or $\alpha_2 < 0$, it means that at least one of the two low-fidelity objective values does not consist with respect to the high-fidelity model and we can expect the ordinal transformation, and, consequently, the optimal sampling, illustrated in Sections 3.1–3.2 to be negatively affected.

The variance component σ_1^2, σ_2^2 , characterizing the bias also hinders the algorithm performance. As previously stated, by controlling the magnitude and variability of the bias, we can create several scenarios which refer to different quality levels.

By varying the parameters of the low-fidelity model, we obtained the experimental conditions in Figure 2. In particular, as shown in the picture, we can separate the entire experiment space into 3 main scenario: (1) positive correlation, i.e., $\alpha_1 = \alpha_2 = 1$, (2) positive and negative correlation, i.e., $\alpha_1 = 1, \alpha_2 = -1$ or $\alpha_1 = -1, \alpha_2 = 1$, and (3) negative correlation, i.e., $\alpha_1 = \alpha_2 = -1$. In the figures, the σ_1 and σ_2 are set to 0.2, but in the experiment we test different possible noise levels.

In the numerical test, we have a limited budget of 1000 evaluations with the high fidelity model, while we performed 100 macro-replications of the algorithm procedure in order to derive the 25, 50, and 75 percentile of the DHV by the observed Pareto set resulting from the macro-replicated experimentation.

4.1 Effect of the parameters in the algorithm performance

As reported in Table 1, there are 32 conditions tested based on the scenarios shown in Figure 2.

4.1.1 Scenario 1: with $\alpha_1 = \alpha_2 = 1$

Not surprisingly, the most relevant factor in this case is represented by the low fidelity model variance. Intuitively, the DHV increases for a low fidelity model characterized by lower variance. Indeed, a large variance results in the diffusion of solutions, leading to groups containing dominated and non-dominated

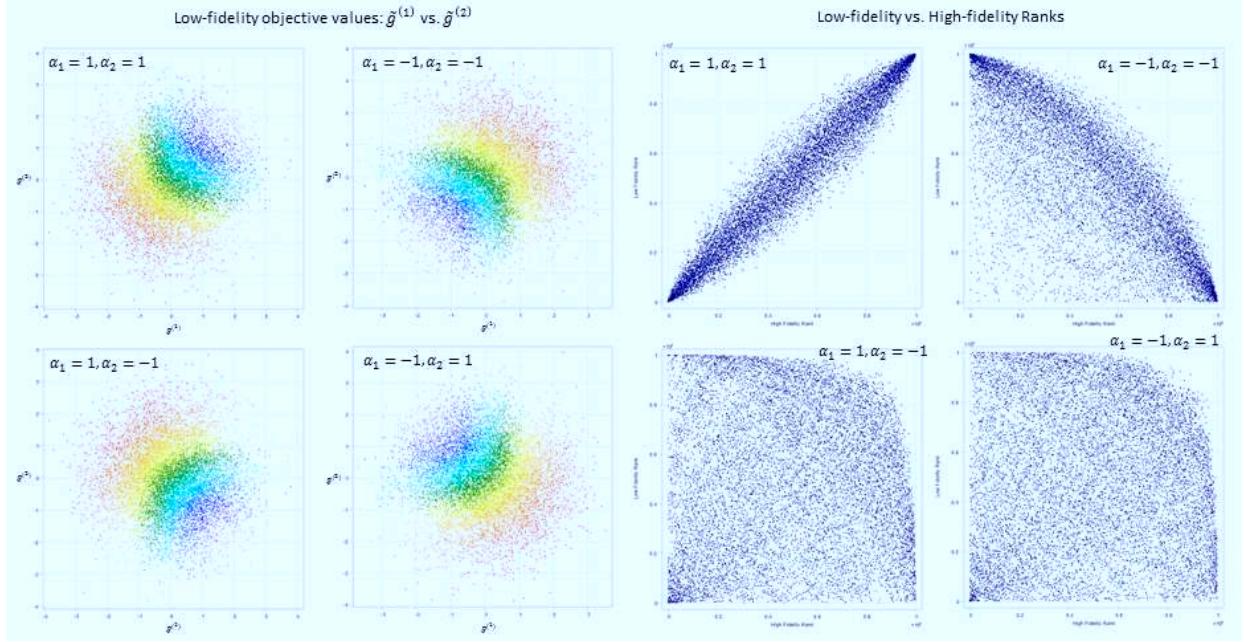


Figure 2: Distortions of Pareto layers & rankings in low-fidelity models under different scenarios.

Table 1: Experimental conditions for each low fidelity level

Level	σ_1	σ_2	m	p_g	p_s
Low	0.2	0.2	10	0.2	0.1
High	0.8	0.8	30	0.8	0.3

solutions. Such a condition, hinders the effectiveness of the ordinal transformation which is indeed based on the group. Some solutions may jump out of its front and drop into other fronts nearby, and this would decrease algorithm efficiency. Under scenario 1, the ranking obtained by OT is close to the true ranking due to the positive correlation characterizing the relationship between the low and the high fidelity function in both objectives.

4.1.2 Scenario 2: with $\alpha_1 = 1, \alpha_2 = -1$ or $\alpha_1 = -1, \alpha_2 = 1$

In these conditions, the number of groups m (as in Step 5 of Algorithm 1) and p_g (as in Step 9 of Algorithm 4) have the most important impact over the DHV. In particular, smaller values of the parameters lead to better performance. Intuitively, the algorithm tends to broadly explore the whole solution space. As we can observe in Figure 2, when one low fidelity model is positively correlated with its high-fidelity model while the other one is negatively correlated with its high-fidelity model, the OT ranking is not effective: solutions from the true Pareto set tend to distribute among different groups, and a low probability of group selection tends to favor a spread search among different groups hence increasing the chance to identify Pareto solutions.

4.1.3 Scenario 3: with $\alpha_1 = \alpha_2 = -1$

It is relevant to notice that the scenario with both low-fidelity models negatively correlated marks an important difference between the single-objective optimization case in Xu et al. (2015) and the multi-objective case of interest in this paper. In fact, when a single objective is of concern, in case the low-fidelity model is “perfectly” negatively correlated with the high-fidelity model (i.e., $\alpha = -1$), at the first iteration of the

algorithm we might simply consider to revert the ordering. Afterwards the procedure works exactly as in the positively correlated case. However, if multiple objectives are considered, the reverting of the Pareto front will not produce the aforementioned results. In fact, solutions in different layers at the boundary of the frontier will be assigned to different layers. This can be observed in Figure 2, where the Pareto Solutions, despite concentrated in the area with smaller high-fidelity and larger low fidelity ranking, tend to distribute in the solution space with several possible values of low-fidelity performance. Given this, reverting the low fidelity evaluations as it would be natural in the single objective case, will not produce the same performance observed in Section 4.1.1.

Specifically, we observed that factor m and p_g have a major impact on the algorithm performance. This is reasonable: larger values of m as well as p_g lead to focus on the “bad observations” hence increasing the chance to recognize a fairly good Pareto set. In fact, as previously stated, a large portion of solutions in the true Pareto set would fall into the last group, before reversion of the OT ranking is performed, while solutions from the last front in true Pareto set would probably fall into the first group, and solutions with non-dominated or near objective performance would still stay close with each other. As a result, after the re-ranking according to the high-fidelity model performance, the obtained groups should be consistent with the high-fidelity model. Therefore, we prefer larger values of p_g to focus on the best solutions and a larger number of groups in order to maximize the chance to identify also misclassified solutions.

4.2 Benchmark Comparison

To show the algorithm performance and demonstrate the efficiency of MO-MO²TOS, we compare it with uniform random search applied to the cases in Figure 2. Based on the analysis in the previous section, we focus on the effect of the parameter p_g , which varies in the set $\{0.1, 0.5, 0.9\}$, with regard to each scenario, p_s , instead, was set to the value 0.1 for all tests. Concerning m , two levels were considered as in the previous section, namely, 10 and 30.

Also in this case, for each experiment setting, 100 macro-replications were performed. The results are reported in Figure 3–7 each corresponding to one of the three scenarios, respectively. Here, the x-axis represents the number of sampled solutions and the y-axis represents the average DHV resulting from the 100 independent macro-replications.

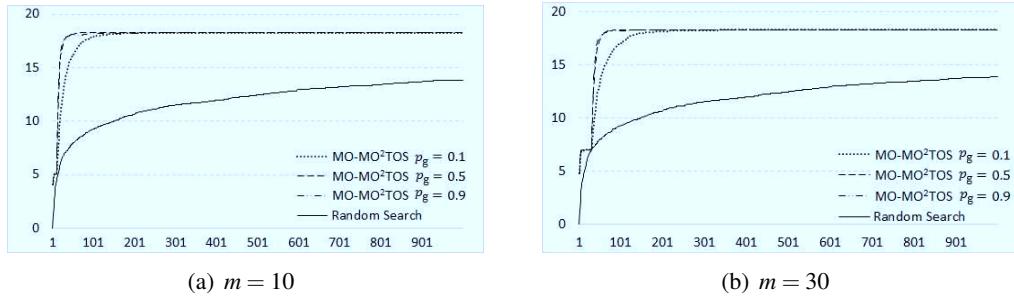


Figure 3: Benchmark comparison for Scenario 1.

From Figure 3, we observe that for Scenario 1, compared to random sampling, MO-MO²TOS performs much better even for small values of sampled solutions (x-axis), which obtains a Pareto set close to the true one with limited samples. This is reasonable since the low fidelity is particularly good under this scenario. After OT, the ranking of solutions, which follows a similar sequence to the true one, facilitates the algorithm to quickly focus on optimal or near optimal solutions. According to the testing result, with 95% acceptance level, the performance of MO-MO²TOS is statistically better than that of random sampling for all cases, i.e., independently from the specific algorithm parameters. Despite this, we observe that when m is low (Figure 3(a)) we prefer larger values of p_g , and vice versa (Figure 3(b)). Figure 4 shows the reason behind the performance of the proposed algorithm. In particular, $m = 10$ is considered. In the

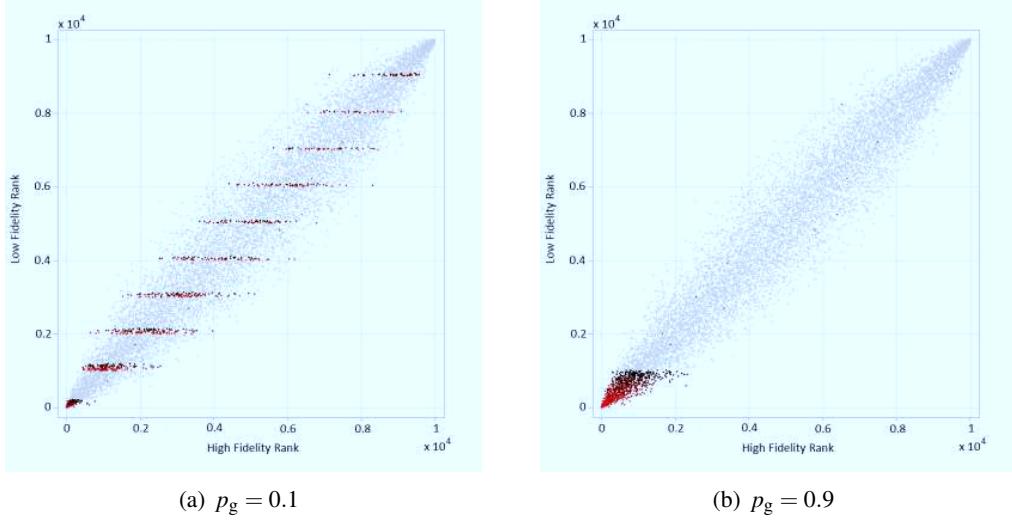


Figure 4: Sampling history for Scenario 1.

Figure, the red color corresponds to the solutions sampled at early stages of the procedure, while darker colors correspond to solutions sampled in later stages. We observe that, when $p_g = 0.1$ (Figure 4(a)), the algorithms spreads the sampling among several groups at least at the first iterations and only towards the end it focuses on the lower left group containing the Pareto solutions. On the contrary, when $p_g = 0.9$, the algorithm will focus on the first group, ending up with a more in-depth exploration of the best groups. As

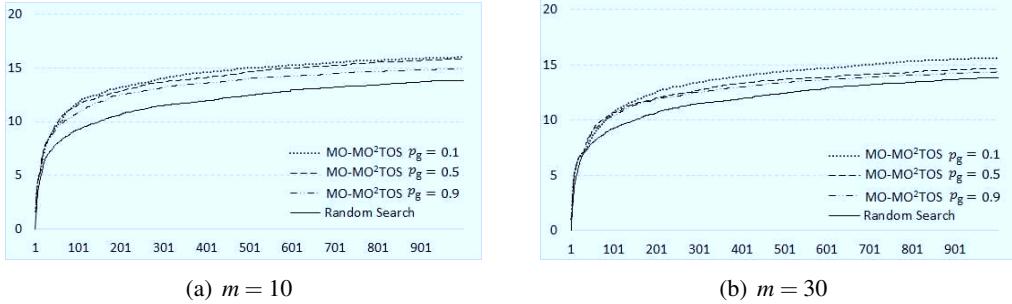


Figure 5: Benchmark comparison for Scenario 2.

already observed in the previous section, Scenario 2 is the most critical for the performance. In particular, we observe a much lower values for DHV which at the end of 1000 iterations compared with Scenario 1. The difference between MO-MO²TOS and random search is relatively small. The reason for this performance resides in the fact that the Pareto solutions are spread among several groups (Figure 6). Hence, as we observe from Figure 5, a lower value for both m and p_g are preferable as they enable a more exploratory search. In the Scenario 3 with both negative correlations, the number of groups plays a remarkable role as we can observe from Figure 7. When $m = 10$, the performance of MO-MO²TOS is worse than random search at the beginning of the algorithm execution, and only if p_g is set to a large value, the algorithm outperform the random search in a later phase. This may be caused by the large diversity of solutions with different performance within each group if m is too small (Figure 8(a)), for which the good solutions do not gain priority (marked in black) in the group since they are badly observed in low-fidelity model. On the contrary, when $m = 30$, it performs better than random search, especially when p_g is large, because once the high-fidelity evaluation re-rank a small group with higher priority, all solutions inside are eventually

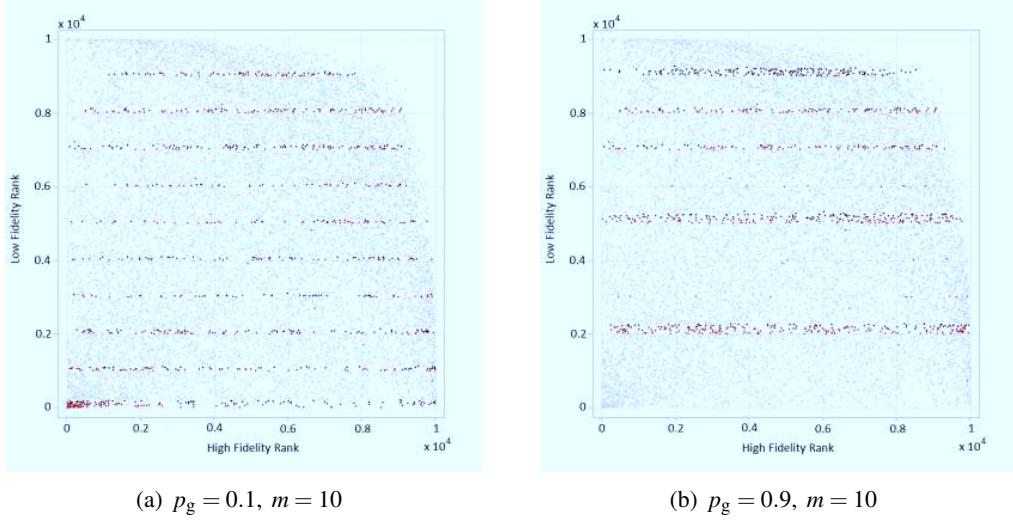


Figure 6: Sampling history for Scenario 2

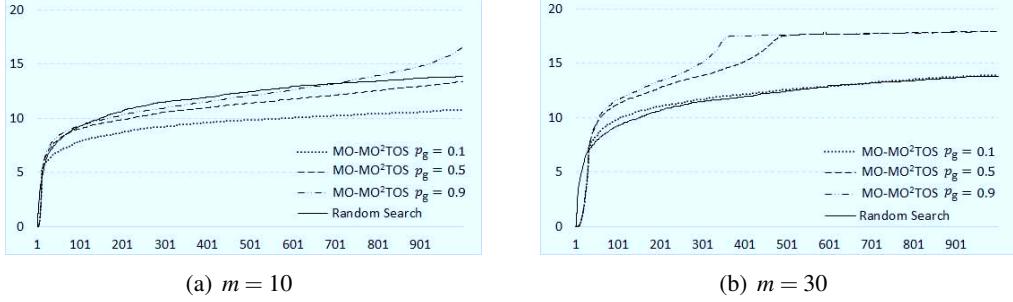


Figure 7: Benchmark comparison for Scenario 3

explored. Therefore, the algorithm is able to identify a good Pareto set under such setting. In conclusion, we suggest large values for both m and p_g are preferred for this scenario.

5 CONCLUSIONS

In this paper, a stochastic search algorithm MO-MO²TOS for multi-objective optimization is developed, and a numerical experiment is designed to test its performance with uniform random search set as the benchmark. It has been shown that MO-MO²TOS can perform more efficiently than random search with proper algorithm parameter setting, for all scenarios. Specifically, when only a small number of iterations can be executed due to computing budget limit, the advantage of the proposed algorithm is obvious.

However, there is still significant difference between the different scenarios depending on the quality of the low-fidelity models. Specifically, when the low-fidelity model is negatively correlated with its high fidelity model, the ranking obtained from OT will be impaired and thereby affects the performance of MO-MO²TOS. In order to improve the algorithm efficiency, except for tuning the algorithm setting, other approaches can be adopted to reduce the bias brought by low fidelity model. One possible approach is to use the negative objective value of a solution if the low and high fidelity models turn out to be negatively correlated. In other words, the first iterations of the algorithm should be used also to learn the correlation and the noise describing the bias of the low-fidelity model in order to improve the algorithm performance by dynamically changing the parameters.

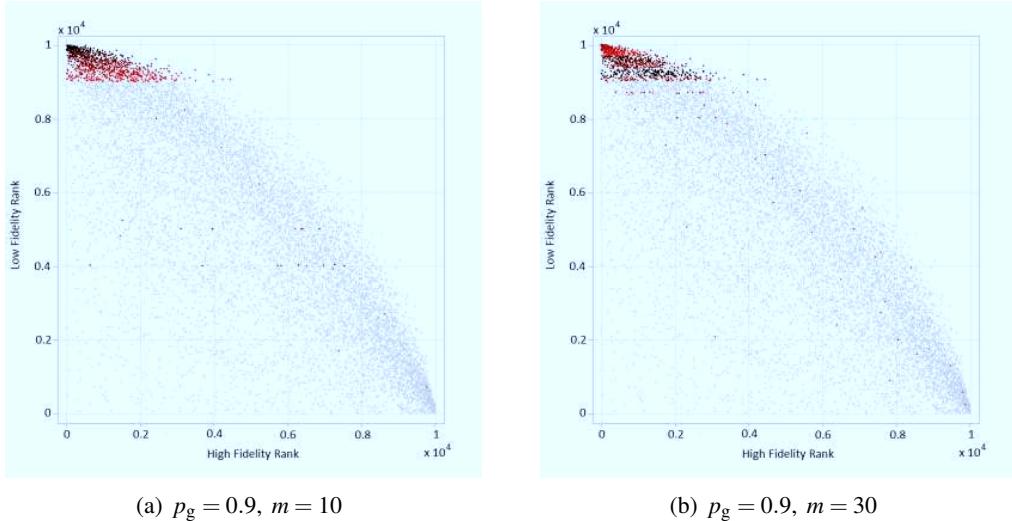


Figure 8: Sampling history for Scenario 3

In this work, we have discussed the application of MO-MO²TOS to the case of deterministic optimization problems. An additional direction would then be to consider the case of stochastic simulations. In this setting, MO-MO²TOS might still prove to be effective by integrating techniques such as Optimal Computing Budget Allocation (Lee et al. 2010) to allocate simulation replications to the sampled configurations.

ACKNOWLEDGMENTS

Research supported in part by the grant (R-SMI-2013-MA-11) funded by the Singapore Maritime Institute.

REFERENCES

- Bradstreet, L., L. While, and L. Barone. 2008. "A fast incremental hypervolume algorithm". *Evolutionary Computation, IEEE Transactions on* 12 (6): 714–723.

Deb, K., S. Agrawal, A. Pratap, and T. Meyarivan. 2000. "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II". *Lecture notes in computer science* 1917:849–858.

Fu, M. C., F. W. Glover, and J. April. 2005. "Simulation optimization: a review, new developments, and applications". In *Proceedings of the 37th conference on Winter simulation*, 83–95. Winter Simulation Conference.

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

Lee, L. H., E. P. Chew, Q. Yu, H. Li, and Y. Liu. 2014. "A study on multi-objective particle swarm optimization with weighted scalarizing functions". In *Proceedings of the 2014 Winter Simulation Conference*, 3718–3729. IEEE Press.

Li, H., L. H. Lee, E. P. Chew, and P. Lendermann. 2015. "MO-COMPASS: A Fast Convergent Search Algorithm for Multi-Objective Discrete Optimization via Simulation". *IIE Transactions* (just-accepted).

Swisher, J. R., S. H. Jacobson, and E. Yücesan. 2003. "Discrete-event simulation optimization using ranking, selection, and multiple comparison procedures: A survey". *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 13 (2): 134–154.

Xu, J., S. Zhang, C. C. Huang, C.-H. Chen, L. H. Lee, and N. Celik. 2015. "MO²TOS: Multi-fidelity Optimization with Ordinal Transformation and Optimal Sampling". submitted to Asia-Pacific Journal of Operational Research.

- Xu, J., S. Zhang, E. Huang, C.-H. Chen, L. H. Lee, and N. Celik. 2014. “Efficient multi-fidelity simulation optimization”. In *Proceedings of the 2014 Winter Simulation Conference*, 3940–3951. IEEE Press.
- Zitzler, E., L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca. 2003. “Performance assessment of multiobjective optimizers: an analysis and review”. *Evolutionary Computation, IEEE Transactions on* 7 (2): 117–132.

AUTHOR BIOGRAPHIES

HAOBIN LI is a scientist in Institute of High Performance Computing, under Agency for Science, Technology and Research (A*STAR) of Singapore. He received his B.Eng. degree (1st Class Honors) in 2009 from the Department of Industrial and Systems Engineering at National University of Singapore, with minor in computer science; and Ph.D. degree from the same department in 2014. He has research interests in operation research, simulation optimization and designing high performance optimization tools which are ready for practical industrial use. His email address is lhb@ihpc.a-star.edu.sg.

YUEQI LI is an undergraduate student in the Department of Industrial and Systems Engineering at National University of Singapore. Her email address is liyueqi@u.nus.edu.

GIULIA PEDRIELLI is Research Fellow for the Centre for Maritime Studies at the National University of Singapore. She received her Ph.D. (honors) from Politecnico di Milano in 2013. Her research focuses on stochastic simulation-optimization in both single and multiple-objectives framework. Her email address is cmsgp@nus.edu.sg.

LOO HAY LEE is Associate Professor and Deputy Head in the Department of Industrial and Systems Engineering, National University of Singapore. He received his B. S. (Electrical Engineering) degree from the National Taiwan University in 1992 and his S. M. and Ph. D. degrees in 1994 and 1997 from Harvard University. He is currently a senior member of IEEE, a committee member of ORSS, and a member of INFORMS. His research interests include production planning and control, logistics and vehicle routing, supply chain modeling, simulation-based optimization, and evolutionary computation. His email address is iseleelh@nus.edu.sg.

EK PENG CHEW is Associate Professor and Deputy Head in the Department of Industrial and Systems Engineering, National University of Singapore. He received his Ph. D. degree from the Georgia Institute of Technology. His research interests include logistics and inventory management, system modeling and simulation, and system optimization. His email address is isecep@nus.edu.sg.

CHUN-HUNG CHEN is a Professor at the Department of Systems Engineering & Operations Research at George Mason University. He received his Ph.D. degree in Engineering Sciences from Harvard University in 1994. He is also affiliated with the National Taiwan University. His work was sponsored by NSF, NIH, DOE, NASA, MDA, Air Force, and FAA. He has received several awards including National Thousand Talents Award (2011), and the Best Automation Paper Award from the IEEE Conference on Robotics and Automation (2003). His email address is cchen9@gmu.edu.