Proceedings of the 2006 Winter Simulation Conference
L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds.

INTEGRATION OF STATISTICAL SELECTION WITH SEARCH MECHANISM FOR SOLVING MULTI-OBJECTIVE SIMULATION-OPTIMIZATION PROBLEMS

Loo Hay Lee
Ek Peng Chew
Suyan Teng

Department of Industrial and Systems Engineering
National University of Singapore
10 Kent Ridge Crescent, 119260, SINGAPORE

ABSTRACT

In this paper, we consider a multi-objective simulation optimization problem with three features: huge solution space, high uncertainty in performance measures, and multi-objective problem which requires a set of non-dominated solutions. Our main purpose is to study how to integrate statistical selection with search mechanism to address the above difficulties, and to present a general solution framework for solving such problems. Here due to the multi-objective nature, statistical selection is done by the multi-objective computing budget allocation (MOCBA) procedure. For illustration, MOCBA is integrated with two meta-heuristics: multi-objective evolutionary algorithm (MOEA) and nested partitions (NP) to identify the non-dominated solutions for two inventory management case study problems. Results show that, the integrated solution framework has improved both search efficiency and simulation efficiency. Moreover, it is capable of identifying a set of non-dominated solutions with high confidence.

1 INTRODUCTION

Discrete event simulation has been a commonly used tool for evaluating the performance of systems which are too complex to be modeled analytically. However, simulation can only evaluate performance measures for a given set of values of the system decision variables, i.e., it lacks the ability of searching optimal values for the decision variables which would maximize or minimize a given response or a vector of responses of the system. This explains the increasing popularity of research in integrating both simulation and optimization, known as simulation optimization: the process of finding the best values of decision variables for a system where the performance is evaluated based on the output of a simulation model of this system.

A simulation optimization problem which minimizes the expected value of the objective function with respect to its constraint set can be expressed as:

$$\min_{\theta \in \Theta} J(\theta). \tag{1}$$

where $J(\theta) = E[L(\theta, \varepsilon)]$ is the performance measure of the problem, $L(\theta, \varepsilon)$ is the sample performance, ε represents the stochastic effects in the system, θ is a p-vector of discrete controllable factors and Θ is the discrete constraint set on θ. If $J(\theta)$ is a scalar function, the problem is single objective optimization; whereas if it is a vector, the problem becomes multi-objective optimization.

The above simulation optimization problem has been extensively studied in the literature. When the search space Θ is finite and relatively small, the problem of selecting the best system is usually known as Ranking and Selection (R&S) problem (Swisher, Jacobson, and Yücesan 2003). Solution approaches proposed for the R&S problem include: indifference-zone ranking and selection (Nelson et al. 2001), optimal computing budget allocation (Chen et al. 2000), and decision theoretic methods (Chick and Inoue 2001). When the search space Θ is either infinite or finite but with a huge number of alternatives, search powers of optimization procedures need to be explored so that we can search efficiently through the given set to find improving solutions. The most straightforward search method is random search, such as the Stochastic Ruler (SR) algorithm (Yan and Mukai 1992), the stochastic comparison (SC) algorithm (Gong, Ho and Zhai 1999), and modified SC or SR algorithms aiming at improving the convergence property (Alrefaei and Andradóttir 1997; Andradóttir 1999). To deal with the randomness of the objective function value, the above solution methods exploit the robustness of order statistics within the random search framework. To search more systematically rather than randomly, most research integrates meta-heuristics with certain statistical analysis techniques: Genetic algorithm (GA) with both a multiple comparison procedure and a R&S procedure (Boesel, Nelson, and Ishii 2003); GA with indifference-zone R&S procedure (Hedlund and Mollaghasemi 2001); modified simulated annealing (SA) algorithm with confidence interval (Alkhamis and Ahmed 2004); SA with R&S procedure (Ahmed and Alkhamis 2002); and nested partitions search with two-stage R&S procedure (Ólafsson 1999).

When the simulation optimization problem has more than one performance measure, the problem is often transformed into a single objective problem so that existing solution techniques can be applied. In Baesler and Sepúlveda (2000), goal programming is incorporated into genetic algorithms to handle the multi-objectives. In papers such as Butler, Morrice, and Mullarkey (2001) and Swisher and Jacobson (2002), the multiple attribute utility (MAU) theory is applied to form a single measure of effectiveness of the objectives. Within such solution frameworks, the optimal solution is a single compromise solution. In case a set of non-dominated solutions are more desirable, the multi-objective problem needs to be solved directly. When search space Θ is finite and relatively small, Lee et al. (2004, 2005) developed a multi-objective computing budget allocation (MOCBA) procedure which incorporates the concept of Pareto optimality into the $R\&S$ scheme to find all non-dominated solutions. When the search space Θ is infinite or finite but very large, MOCBA needs to be integrated with search procedures for efficient exploration of more promising solutions, such as integration with multi-objective evolutionary algorithm (MOEA) in Lee et al. (2006), and integration with nested partitions (NP) algorithm in Chew et al. (2006).

In this study, we consider a simulation optimization problem formulated in (1) with the following features: huge size of solution space (Θ), large uncertainties (ε) in performance measures, and multi-objective problem which requires a non-dominated Pareto set of solutions. These features of the problem make it both challenging and difficult to solve. On one hand, to improve simulation efficiency and to guarantee that final solutions are non-dominated with high confidence, certain statistical selection procedure is needed; on the other hand, to search more efficiently, search mechanism of optimization techniques is also required. Lee et al. (2006) and Chew et al. (2006) presented how to integrate MOCBA (a statistical selection procedure) with MOEA and NP (meta-heuristics). In this paper, we summarize the main ideas from the above two papers with focus on how to incorporate the statistical selection procedure (such as MOCBA) into some meta-heuristic procedures so that either of them will benefit from the other and together they improve both search efficiency and simulation efficiency through the integration. The paper is organized as follows. In Section 2, we first discuss why there is a need to integrate MOCBA with a search procedure for simulation optimization problems. Then based on a brief description of the MOCBA procedure, we present a general solution framework which integrates MOCBA with a search procedure to efficiently allocate simulation replications and identify the non-dominated solutions for the problem. In Section 3, the solution framework is illustrated by integrating MOCBA with two meta-heuristics (MOEA and NP) for solving two inventory management problems. Some computational results are also reported in this section. Finally some conclusions and future research directions are summarized in Section 4.

2 THE INTEGRATION OF STATISTICAL SELECTION WITH SEARCH IN SIMULATION OPTIMIZATION

In this section, we discuss why there is a need to integrate statistical selection with search procedures in simulation optimization, and how to incorporate the selection into the search procedure

2.1 The need to integrate search and selection in simulation optimization

A key feature of simulation optimization problem that makes it difficult is the need to address the search versus selection trade-off: with a limited computing budget, how to allocate the budget between searching over the feasible space for (potentially) better solutions, and determining which of the solutions that have been examined are actually good. Before addressing this issue, we first study the following two extreme problems in simulation optimization.

When the solution space Θ in (1) is finite and small, it is possible to evaluate exhaustively all members from the given (fixed and finite) set of alternatives and compare the performance. In this case, the focus is entirely on selection: the comparison aspect unique to the stochastic setting. Another extreme of the problem is to ignore the stochastic nature of the problem. To evaluate the performance measures, a fixed number of simulation replications are allocated to each design alternative identified as promising solution by the search procedure. In this case, the focus is entirely on search: solution exploration aspect important to deterministic optimization problem.

Obviously, for the general simulation optimization problem such as the one considered in this study, it is neither a pure search nor a pure selection problem. When the solution space is huge, exhaustive search becomes either impractical or impossible. Hence one critical issue is to find out which decision scenarios are the ones desirable to be investigated, and how to identify those good scenarios automatically by a search process designed to find the best set of decisions. This makes the search mechanism of optimization procedures a necessity when exploring the solution space for more promising solutions. On the other hand, when selection aspect is neglected, we may encounter the following difficulties. Firstly there is no statistical guarantee on the quality of the final set of solutions. Secondly, the fixed number of simulation replications are either too few to handle the stochastic noise or too many to be of computational efficiency. In the former case, inaccurate estimations of the performance measures may lead the search to unproductive regions. This is due to the fact that search direction is guided by the evaluation of the objectives in any search mechanism. For example, in MOEA, parents are selected to generate new offspring based on fitness values. In case fitness values are estimated inaccurately, less fit solutions may be selected and survive into the next generation, whereas truly fit solutions may be neglected and lose the chance for further consideration. Similarly in NP, promising index for each region is calculated based on performance measures of the samples picked in that region, and future promising region is selected according to the promising indices calculated. With inaccurate estimation of the performance measures of the samples, the promising region that the future search will focus on may turn out to be less promising. In the latter case, when the search needs to explore a large number of design alternatives each with a large number of simulation replications, which is usually the case for meta-heuristics, the total simulation cost can easily become very high. To avoid unnecessary waste in simulation budget, a more efficient way of allocation than uniform allocation is desirable. Intuitively, for designs with poor performances which are obviously dominated by other designs, it is a waste if further simulation replications are to be allocated to them, though the noise involved in the performance measures maybe still high. Thus a statistical selection procedure is required to single out those designs likely to be competitors for the "best" and to optimally determine the number of simulation replications needed for each of the designs while identifying those non-dominated designs. The MOCBA procedure presented in Lee et. al. (2005) is specially developed for the above purposes.

2.2 The MOCBA procedure and its integration with search mechanisms

Given a set of n design alternatives with H performance measures which are evaluated through simulation, a multi-objective R&S problem is to determine an optimal allocation of the simulation replications to the designs, so that the non-dominated set of designs can be found at the least expense in terms of simulation replications. In this section, we present a brief description of a solution framework for this problem: the Multi-objective Optimal Computing Budget Allocation (MOCBA) algorithm. For more details about how MOCBA works, please refer to Lee et al. (2005).

2.2.1 A Performance Index to Measure the Non-dominated Designs

Suppose we have a set of designs $i\,(i=1,2,...,n)$, each of which is evaluated in terms of H performance measures μ_{ik} ($k=1,2,...,H$) through simulation. Within the Bayesian framework, μ_{ik} is a random variable whose posterior distribution can be derived based on its prior distribution and the simulation output (Lee et al. 2004). We use the following performance index to measure how non-dominated design i is:

$$\psi_i = \prod_{j=1,\, j\neq i}^{n} [1 - P(\boldsymbol{\mu}_j \prec \boldsymbol{\mu}_i)]. \tag{2}$$

where $P(\boldsymbol{\mu}_j \prec \boldsymbol{\mu}_i)$ represents the probability that design j dominates design i. Under the condition that the performance measures are independent from one another and they follow continuous distributions, we have

$$P(\boldsymbol{\mu}_j \prec \boldsymbol{\mu}_i) = \prod_{k=1}^{H} P(\mu_{jk} \le \mu_{ik}). \tag{3}$$

Performance index ψ_i measures the probability that design i is non-dominated by all the other designs. At the end of simulation, all designs in the Pareto set should have ψ_i close to 1, and those designs outside of the Pareto set should have ψ_i close to 0, because they are dominated.

2.2.2 Two Types of Errors of the Selected Pareto Set

During the allocation process, the Pareto set is constructed based on observed performance. Here we call it the selected Pareto set (S_p). The quality of the selected Pareto set depends on whether designs in S_p are all non-dominated and designs outside S_p are all dominated. We evaluate it by two types of errors: Type I error (e_1) and Type II error (e_2) as follows.

Type I error (e_1) is defined as the probability that at least one design in the selected non-Pareto set ($\overline{S}_p$) is non-dominated; while Type II error (e_2) is defined as the probability that at least one design in the selected Pareto set is dominated by other designs. When both types of errors approach 0, the true Pareto set is found. The two types of errors can be bounded by the approximated errors ae_1 and ae_2 respectively as given below.

$$e_1 \le ae_1 = \sum_{i \in \overline{S}_p} \psi_i. \tag{4}$$

$$e_2 \le ae_2 = \sum_{i \in S_p} (1 - \psi_i). \tag{5}$$

When the noise in the simulation output is high, ae_1 and ae_2 can be large. However, once the selected Pareto set approaches the true Pareto set, both ae_1 and ae_2 approach 0, as ψ_i approaches 1 for $i \in S_p$ and ψ_i approaches 0 for $i \in \overline{S}_p$.

2.2.3 Construction of the Selected Pareto Set

Ranking all the designs in descending order of performance index ψ_i, then the selected Pareto set can be constructed according to the criteria below.

C1: Assign a maximum number of k designs with the highest ψ_i into S_p, so that $ae_2 = \sum_{i=1}^{k} (1 - \psi_i) \le \varepsilon$, where ε is a predefined error limit.

C2: Construct S_p so that both ae_1 and ae_2 are minimized by simply putting those designs with $\psi_i \ge 0.5$ into S_p and the rest into $\overline{S}_p$.

2.2.4 The Asymptotic Allocation rules and a Sequential Solution Procedure

To get the true Pareto set with high probability, we need to minimize both Type I and Type II errors. In the MOCBA algorithm of Lee et al. (2005), this is done by iteratively allocating the simulation replications until both ae_1 and ae_2 are within error limit ε^* according to some asymptotic allocation rules as follows:

1). For a design $l \in \overline{S}_p$, $T_l = \dfrac{\alpha_l}{\sum_{l \in \overline{S}_p} \alpha_l + \sum_{d \in S_p} \alpha_d} N_{\max}$

2). For a design $d \in S_p$, $T_d = \dfrac{\alpha_d}{\sum_{l \in \overline{S}_p} \alpha_l + \sum_{d \in S_p} \alpha_d} N_{\max}$

with $\alpha_l = \dfrac{\left(\sigma^2_{lk^l_{j_l}} + \sigma^2_{j_l k^l_{j_l}} \big/ \rho_l\right) \Big/ \delta^2_{l j_l k^l_{j_l}}}{\left(\sigma^2_{mk^m_{j_m}} + \sigma^2_{j_m k^m_{j_m}} \big/ \rho_m\right) \Big/ \delta^2_{m j_m k^m_{j_m}}}$, given m is any fixed design in $\overline{S}_p$; $\alpha_d = \sqrt{\sum_{i \in \Omega_d} \dfrac{\sigma^2_{dk^i_d}}{\sigma^2_{ik^i_d}} \alpha_i^2}$

where $N_{\max}$ is the maximum number of replications available; S_p and $\overline{S}_p$ represent the selected Pareto and non-Pareto set respectively; T_i is the number of replications to be allocated to design i; σ^2_{ik} is the sample variance of kth objective of design i; δ_{ijk} is the difference between sample means of kth objective of design i and design j;

$$j_i \equiv [\text{design } d \mid d \in S, d \ne i, P(\mu_d \prec \mu_i) = \max_{\substack{j \in S \\ j \ne i}} \prod_{k=1}^{H} P(\mu_{jk} \le \mu_{ik})]$$

is the design that dominates design i with the highest probability; $k^i_{j_i}$ is the objective of j_i that dominates the corresponding objective of design i with the lowest probability,

$$k^i_{j_i} \equiv \left[\text{objective } r \text{ of design } j_i \;\middle|\; \begin{array}{l} r \in \{1,\ldots,H\}, \\ P(\mu_{j_i r} \le \mu_{ir}) = \min\limits_{k \in \{1,\ldots,H\}} P(\mu_{j_i k} \le \mu_{ik}) \end{array} \right]$$

with $P(\mu_{jk} \le \mu_{ik}) \sim N(\delta_{ijk}, \dfrac{\sigma^2_{jk}}{N'_j} + \dfrac{\sigma^2_{ik}}{N'_i})$, and T'_i is the number of replications allocated to design i at the immediate previous iteration; $\Omega_d = \{\text{design } i \mid i \in \overline{S}_p, j_i = d\}$;

$$\rho_i = \sqrt{\sum_{r \in \Omega_{j_i}} \frac{\sigma^2_{j_i k^r_{j_i}}}{\sigma^2_{rk^r_{j_i}}}} \text{ or } \rho_i = \frac{T'_{j_i}}{T'_i}.$$

The MOCBA algorithm is now outlined below.

<u>MOCBA algorithm</u>

Step 0: Perform T_0 replications for each design. Calculate the sample mean and variance for each objective of the designs. Set iteration index v: = 0. $T_1^v = T_2^v = \ldots = T_n^v = T_0$.

Step 1: Construct the selected Pareto set S_p according to criterion C2. Calculate ae_1 and ae_2 according to equations (4) and (5). If ($ae_1 < ae_2$), construct S_p according to the criterion C1 with $\varepsilon = ae_1$.

Step 2: If (($ae_1 < \varepsilon^*$) and ($ae_2 < \varepsilon^*$)), go to Step 5.

Step 3: Increase the simulation replications by a certain amount Δ, and calculate the new allocation $T_1^{v+1}, T_2^{v+1}, \ldots, T_n^{v+1}$ according to the asymptotic allocation rules.

Step 4: Perform additional min(δ, $\max(0, T_i^{v+1} - T_i^v)$) replications for design i (i =1,…,n). Update the sample mean and variance of each objective of design i based on cumulative simulation output. Set $v = v + 1$ and go to Step 1.

Step 5: Output designs in the selected Pareto set (S_p).

2.2.5 The integration of MOCBA with search mechanism of optimization procedures

In a general deterministic search procedure, whether it is developed for single objective or multi-objective, whether it is population based or single solution based, the search mechanism works more or less in a similarly way except that the mechanism of generating new solutions may differ significantly (Figure 1). Here "performance evaluation" and "fitness evaluation" are listed separately because, though most search procedures use performance measure of a solution to represent its fitness, in some procedures, such as EA and NP, fitness of a solution (region) is defined separately based on its performance for determining the search direction of the algorithm. Meanwhile, most procedures keep an archive of best solutions ever visited and return it as final solution upon termination. A flow chart showing how general search procedures work is given in Figure 1.

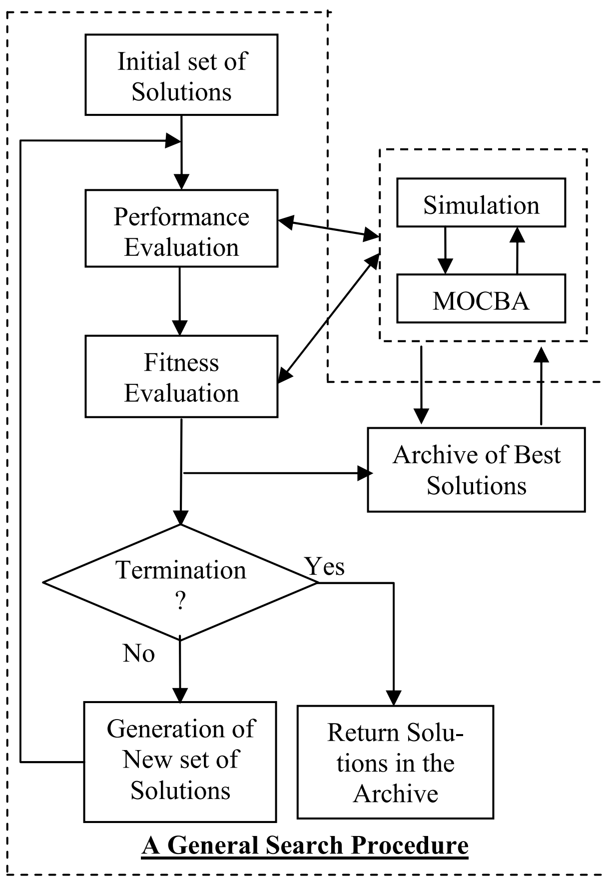

Figure 1: Integration of MOCBA with Search Procedures

However, when we apply such a search procedure to solve multi-objective simulation optimization problems, we need to consider: a). the right number of simulation replications needed for each design to accurately evaluate its performance; b). how to define fitness of a solution considering both variability involved in the performance measures and the multi-objective nature of the problem; c). how to select those best solutions ever visited to form the Pareto set. The MOCBA procedure helps to answer the above questions simultaneously. A general framework of integrating MOCBA (a statistical selection procedure) with search procedures is illustrated in Figure 1.

Specifically, at each iteration, to rank the designs in the current population, performance index defined in (2) is used to measure the fitness (non-dominance) of the designs. A design with larger value of performance index is non-dominated with higher probability. The performance index actually transforms the multi-objective problem into a problem with single objective, as we can now determine the "goodness" of a solution completely based on this single performance index. To determine the right number of simulation replications needed for each design to accurately estimate its finesses, the MOCBA algorithm based on allocation rules presented in 2.2.4 is run on the current set of solutions. At this stage, our main purpose of running MOCBA is not to determine precisely the Pareto set with very high confidence, but to efficiently rank the designs in terms of their fitness values to determine the right direction of the search procedure. Therefore, to save simulation budget and also to avoid loss of truly non-dominated designs during the search procedure, the MOCBA is terminated when each of the performance index inside the Pareto set is above a certain value and each of the performance index in the non-Pareto set is below a certain value. Then designs in the Pareto set are put into the archive. At the final iteration, MOCBA is run again on designs in the archive with very low limits on Type I and Type II errors, so that truly non-dominated designs are identified with least possible simulation replications upon termination of the search procedure.

Though with slight modifications, single solution based search procedure can work with MOCBA for solving multi-objective simulation optimization problems, the ideal search procedures should be able to move with a set of solutions from one iteration to the next, because the set of solutions can provide a basis for generating (approximating) the Pareto set of solutions at each iteration by running MOCBA. Moreover, MOCBA can optimally determine the number of simulation replications needed for each of the set of solutions and therefore can effectively avoid waste of simulation budget. Meta-heuristics such as multi-objective evolutionary algorithm (MOEA) and nested partitions (NP) are nice candidates for the above purposes.

3 INTEGRATION OF MOCBA WITH META-HEURISTIC PROCEDURES -- TWO CASE STUDY PROBLEMS

In this section, we use two example meta-heuristic procedures (MOEA and NP) to illustrate how to integrate a statistical selection procedure with a search procedure. The two integrated frameworks are then applied to solve two inventory management case study problems. This section briefly summarizes the solution procedures and some computational results. For more details regarding problem description and results presentation, refer to Lee et al. (2006) and Chew et al. (2006).

3.1 The integration of MOCBA with MOEA

3.1.1 The integrated MOEA framework

MOEA is an adaptive heuristic search algorithm which simulates the survival of the fittest among individuals over consecutive generations for solving a multi-objective problem. Based on an initial population randomly generated, at each generation, MOEA evaluates the chromosomes and ranks them in terms of their fitness; the fitter solutions will be selected to generate new offspring by recombination and mutation operators. This process of evolution is repeated until the algorithm converges to a population which covers the non-dominated solutions. MOEA has been successfully applied in solving deterministic multi-objective problems (Fonseca and Fleming 1995, Hanne and Nickel 2005).

To make MOEA work well for simulation optimization problems where variability is a main concern, MOCBA is needed mainly in the following two aspects: fitness evaluation and formation of elite population. A detailed description of the integrated solution framework is given below.

Outline of the Integrated MOEA

Step 0: Initialization: Randomly generate an initial feasible population POP_t of size N_t; set elite population $E_t = \varnothing$; set generation index $t = 0$.

Step 1: Run MOCBA (Section 2.2.4) to determine the number of replications for each design in population POP_t, and select designs into the Pareto set.

Step 2: Formation of elite population: Form the elite population E_t with designs in the Pareto set.

Step 3: Check the termination condition. If it is not satisfied, go to Step 5.

Step 4: Termination: Run MOCBA on the Elite Population E_t with both types of errors within error limit ε^*. Output the Pareto set as the final set of non-dominated designs.

Step 5: Evaluation and fitness assignment: Use performance index ψ_i at the termination of MOCBA as the fitness value of chromosome *i*.

Step 6: New population: Set $t = t+1$; let $POP_t = POP_{t-1}$. Create a new population by repeating the following steps until *M* pairs of parents are selected.

1) Selection: select one pair of chromosomes by tournament selection from population POP_{t-1}.
2) Crossover: Perform arithmetic crossover and one-point crossover with probability 0.5 each, resulting in a pair of offspring. Check the feasibility of the offspring.
3) Add the new offspring into population POP_t.

Step 7: Mutation: Run MOCBA on population POP_t to determine fitness value for the new offspring. If $N_t > N_{\max}$, delete $(N_t - N_{\max})$ designs with least fitness value. For each chromosome *i* in POP_t, perform mutation with probability $P_m = 0.1\psi_i$. Check the feasibility of the mutated chromosome. Go to Step 1.

3.1.2 The aircraft spare parts inventory allocation problem — case study 1

When a repairable item on an aircraft becomes defective, it is removed and replaced by another item from the spare stock. The defective part then goes into some repair cycle at the Central Repair Depot (CRD). If the airport does not have the spare part in stock, the aircraft will be grounded until an incoming flight brings a replacement part from the CRD or from a neighboring airport. To reduce departure delays due to unanticipated failures, airlines need to keep inventory of spare parts at the associated airports, as well as at the CRD.

Suppose we have an airport network which consists of S airports and 1 CRD. We assume that *N* repairable spares of the same type are available in the network. At every maintenance check, a failure occurs at the probability of α. Upon a part failure, re-supply of the spare part comes from either the inventory at the airport, the CRD or from the neighboring airports. The repair time for a defective part is assumed to follow uniform distribution between $[R_1, R_2]$. The problem is to determine the allocation of the spare parts among the airports, and the replacement policy (where to get a replacement part: airport's own inventory, neighboring airport or CRD) upon the occurrence of a part failure, so that the average cost involved in the system is minimized and both the average fill-rate and the minimum fill-rate of the entire network are maximized. Here the cost is defined in terms of inventory cost and the transportation cost of shipping the defective part. The fill-rate of an airport is defined as the percentage of failures serviced by its own spare part inventory.

With three objectives evaluated through simulation, the problem has the following difficult features: huge search space, multi-objective, and high variability. To address these difficulties, we apply the integrated MOEA to solve this problem.

In the integrated MOEA, the coding scheme defines each chromosome as $S+1$ genes $L_i (i = 1,2,...,S+1)$, each of which represents the inventory level of spares at airport i (or CRD). At each MOEA generation, crossover and mutation are performed to generate new offspring, and a set of best chromosomes are kept in the elite population. When termination condition (no new solutions are added into the elite population or a maximum number of generations has been executed) is met, MOCBA is run on the final elite population, and the resulted Pareto set is output as the final non-dominated set of solutions. Figures 2 and 3 illustrate the improvement of Pareto front as the number of MOEA generations increases from 1 to 200.

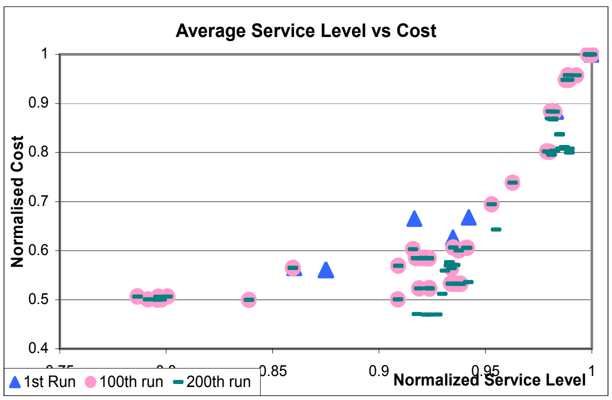

Figure 2 Improvement of Pareto Set in terms of Average Service Level and Cost

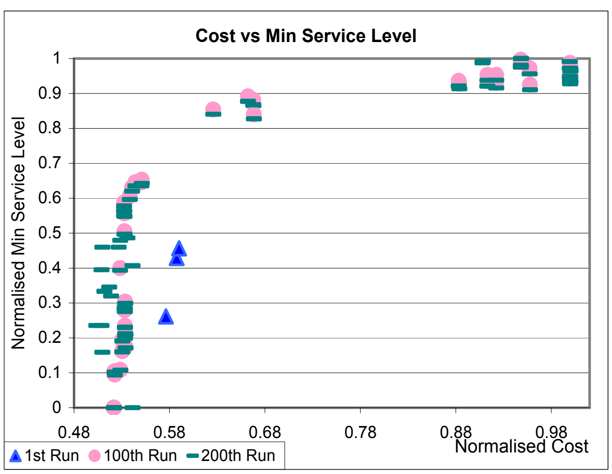

Figure 3 Improvement of Pareto Set in terms of Minimum Service Level and Cost

Figures 2 and 3 show that, all three performance measures of the designs in the Pareto set are improved significantly by the integrated MOEA from the 1st to the 100th generation. The integrated MOEA is capable of identifying those non-dominated designs with a lower cost, and a higher average as well as minimum service level. Meanwhile, from the 100th to 200th generation of the integrated MOEA, the improvement of the Pareto set can be observed but not as apparent as in the first 100 genetic generations. The new designs generated and added into the Pareto set are mostly in the same region as those generated during the first 100 generations, less than 20 new designs have been found are superior to the designs in the Pareto set found at the 100th generation. This implies that, the integrated MOEA has started to converge to a local optimum.

3.2 The integration of MOCBA with NP

3.2.1 The integrated NP framework

The NP method, proposed by Shi and Ólafsson (2000), is a randomized method based on the concept of adaptive sampling for solving single objective global optimization problems. It systematically partitions the feasible region and concentrates the search in regions that are the most promising. It combines partitioning, random sampling, a selection of a promising index, and backtracking to create a Markov chain that converges to a global optimum.

For the single objective deterministic NP to be applicable for solving our problem, the performance index ψ_i introduced in MOCBA is used to calculate the promising index to determine the next promising region. Moreover, MOCBA is also used for effective allocation of simulation replications as well as forming the global Pareto set. The integrated NP framework (described below) is developed for problems with two-dimensional search space: s and Q.

Outline of the Multi-objective Integrated NP

Step 0: Initialization: Determine the feasible region of $[s,Q]$ and denote it as Ω; let it be the current most promising region Ω_b; set the surrounding region as $\Omega_s = \varnothing$; set the global Pareto set $S_p = \varnothing$.

Step 1: Partition and sampling: Partition Ω_b into M sub-regions; take ω samples from each sub-region and the surrounding region Ω_s.

Step 2: Promising index calculation: Run r replications of the simulation model for each sample taken; calculate performance index ψ_i for each sample; calculate the promising index for each region. Denote the region with the highest promising index as Ω_h. Add those samples with $\psi_i > 0.5$ into the global Pareto set S_p.

Step 3: Update of global Pareto set S_p: Run MOCBA on designs in S_p. Find the Pareto set S_p' with both types of errors within error limit ε^*. Replace S_p with S_p'.

Step 4: Check termination condition. If not satisfied, go to Step 6.

Step 5: Termination: Output the global Pareto set S_p as the final set of non-dominated solutions.

Step 6: If $\Omega_h \subseteq \Omega_b$, set $\Omega_b = \Omega_h, \Omega_s = \Omega - \Omega_h$. Go to Step 1.

Step 7: Backtracking: Backtrack to a super region of Ω_b. Regard it as the current most promising region Ω_b, and aggregate the surrounding region as Ω_s. Go to Step 1.

3.2.2 The differentiated service inventory problem — case study 2

In a differentiated service inventory problem, customers are classified into different groups according to their importance to the decision makers. The problem is to determine how to replenish inventories and how to allocate these inventories to different demand classes according to some performance measures such that each demand class is offered with different service level. In this study, we assume that there are m different demand classes where the demand for each class is stochastic, and the inventory is replenished according to a continuous (s, Q) inventory model. Under the dynamic threshold policy developed in Chew, Lee, and Liu (2005) to differentiate demand classes and to offer different services, the problem is to obtain a set of non-dominated reorder point s and order quantity Q with the best combination of cost and service level. Here the cost is defined in terms of average annual cost, which consists of setup cost, inventory holding cost and backorder cost. The service level is defined in terms of average backorder of each customer class, which is calculated as the average number of backorders per year.

This problem involves two objectives which are also evaluated by simulation models. To search for the non-dominated (s, Q) policies within the feasible region which is an area formed by upper bounds of both s and Q, we employ the integrated NP to partition the search space iteratively and focus on the more promising region in searching for better solutions. The upper bound of Q is estimated by the EOQ model. The upper bound of reorder point s is determined by first minimizing the average annual cost and then minimizing the average backorder. With similar termination condition as in the integrated MOEA, designs in the final global Pareto set are illustrated in Figures 4 and 5.

Figures 4 and 5 indicate that, designs in the final run global Pareto set of the integrated NP form a curve which represents the efficient Pareto frontier. All designs from the initial run are above this curve and therefore are dominated. Moreover, we can observe that, in the final run, unlike the initial run, designs are more evenly distributed along the Pareto frontier. This indicates that, the integrated NP is capable of finding a full spectrum of non-dominated designs for the decision makers, which is highly desirable in the practice. Moreover, the algorithm is capable of searching different regions of the search space rather than concentrating on a certain region, therefore it has a higher chance of finding the global optimal set of non-dominated solutions.

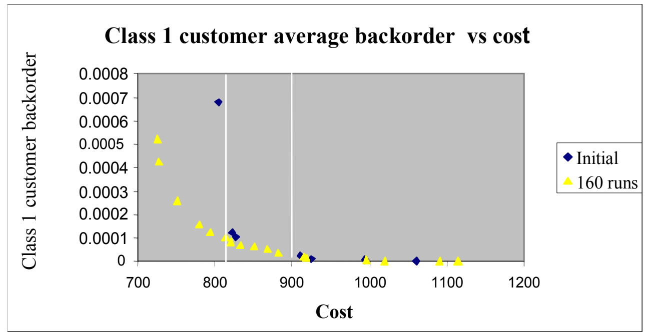

Figure 4 Pareto Front Improvement of Class 1 Customers

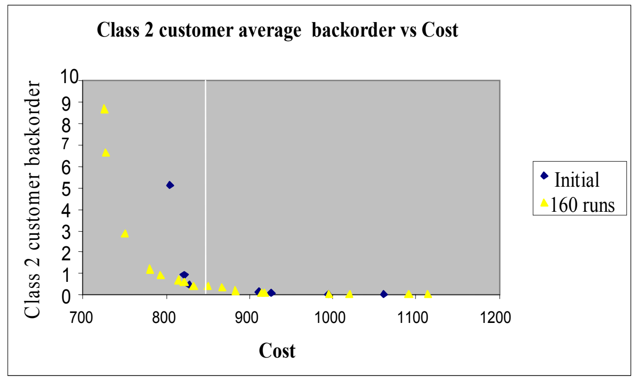

Figure 5 Pareto Front Improvement of Class 2 Customers

4 CONCLUSIONS

For the simulation optimization problem considered in this study, neither a statistical selection procedure nor a search procedure is enough to address the difficult issues involved in the problem: huge solution space, high variability and multi-objective. We therefore propose a solution framework which integrates the two procedures for efficient allocation of simulation replications as well as search and identification of non-dominated solutions. Due to the fact that the performance index introduced in MOCBA transforms the multi-objectives into a single measure of effectiveness which can be used to evaluate the solutions in terms of non-dominance, search procedures developed for both single objective and multi-objective optimization problems are applicable here. We illustrate the solution framework by integrating MOCBA with two metaheuristic procedures which move with a set of solutions at each iteration: MOEA and NP. The two integrated frameworks are then applied to solve two inventory management case study problems. Results show that, throughout the integrated search procedure, the Pareto set is improved greatly in terms of both individual solution quality and the distribution of the solutions along the Pareto frontier. The integrated solution framework is capable of identifying a set of non-dominated solutions with high confidence. In this study, the convergence and the quality of the Pareto set are analyzed by observing how performance measures of the non-dominated designs in the Pareto set change as the search progresses; and the termination condition of the algorithms is based on this observation. In future research, it may be worthwhile to study the convergence property more systematically. Meanwhile, it is also important to investigate how to evaluate the quality of the final Pareto set in comparison with true Pareto set.

REFERENCES

Ahmed, M. A., T. M. Alkhamis. 2002. Simulation-based optimization using simulated annealing with ranking and selection. *Computers & Operations Research* 29, 387–402.

Alkhamis, T. M., M. A. Ahmed. 2004. Simulation-based optimization using simulated annealing with confidence interval. *Proceedings of the 2004 Winter Simulation Conference*, 514–519.

Alrefaei, M. H., S. Andradóttir. 1997. Accelerating the convergence of the stochastic ruler method for discrete stochastic optimization. *Proceedings of the 1997 Winter Simulation Conference*, 352–357.

Andradóttir, S. 1999. Accelerating the convergence of random search methods for discrete stochastic optimization. *ACM Transactions on Modeling and Computer Simulation* 9 (4), 349–380.

Baesler, F. F., J. A. Sepúlveda. 2000. Multi-response simulation optimization using stochastic genetic search within a goal programming framework. *Proceedings of the 2000 Winter Simulation Conference,* 788–794.

Boesel, J., B. L. Nelson, and N. Ishii. 2003. A framework for simulation-optimization software. *IIE Transactions* 35, 221–229.

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

Chen, C. H., J. W. Lin, E. Yücesan, and S. E. Chick. 2000. Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Discrete Event Dynamic Systems: Theory and Applications* 10: 251–270.

Chew, E. P., L. H. Lee, and S. D. Liu. 2005. Dynamic rationing policies for inventory systems with multiple demand classes. *Working paper*, Department of Industrial & Systems Engineering, National University of Singapore.

Chew E. P., L. H. Lee, S. Y. Teng, and C. H. Koh. 2006. Multi-objective simulation optimization for differentiated service inventory problems. Submitted to *Journal of the Operational Research Society*.

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

Fonseca, C. M. and P. J. Fleming. 1995. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation* 3(1), 1–16.

Gong, W. B., Y. C. Ho, and W. G. Zhai. 1999. Stochastic comparison algorithm for discrete optimization with estimation. *SIAM Journal on control and optimization* 10, 384–404.

Hanne, T. and S. Nickel. 2005. A multiobjective evolutionary algorithm for scheduling and inspection planning in software development projects. *European Journal of Operational Research* 167, 663–678.

Hedlund, H. E., and M. Mollaghasemi. 2001. A genetic algorithm and an indifference-zone ranking and selection framework for simulation optimization. *Proceedings of the 2001 Winter Simulation Conference*, 417–421.

Lee, L. H., E. P. Chew, S. Y. Teng, and D. Goldsman. 2004. Optimal computing budget allocation for multi-objective simulation models. In *Proceedings of the 2004 Winter Simulation Conference* (ed. R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters), 586–594.

Lee, L. H., E. P. Chew, S. Y. Teng, and D. Goldsman. 2005. Finding the non-dominated Pareto set for multi-objective simulation models. Submitted to *IIE Transactions.*

Lee, L. H., E. P. Chew, S. Y. Teng, and Y. K. Chen 2006. Multi-objective simulation-based evolutionary algorithm for an aircraft spare parts allocation problem. Submitted to *European Journal of Operational Research.*

Nelson, B. L., J. Swann, D. Goldsman, and W. M. Song. 2001. Simple procedures for selecting the best simulated system when the number of alternatives is large. *Operations Research* 49: 950–963.

Ólafsson, S. 1999. Iterative ranking-and-selection for large-scale optimization. *Proceedings of the 1999 Winter Simulation Conference*, 479–485.

Shi, L. and S. Ólafsson. 2000. Nested partitions method for global optimization. *Operations Research* 48 (3), 390–407.

Swisher, J. R. and S. H. Jacobson. 2002. Evaluating the design of a family practice healthcare clinic using discrete-event simulation. *Health Care Management Science* 5 (2): 75–88.

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* 13 (2): 134–154.

Yan D., and H. Mukai. 1992. Stochastic discrete optimization. *SIAM Journal on control and optimization* 30, 594–612.

AUTHOR BIOGRAPHIES

LOO HAY LEE is an Associate Professor 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, committee member of ORSS, and a member of INFORMS. His research interests include simulation-based optimization, production scheduling and sequencing, logistics and supply chain planning, and vehicle routing. His email address is <iseleelh@nus.edu.sg> and his website is <www.ise.nus.edu.sg/staff/leelh/>.

EK PENG CHEW is an Associate Professor and Deputy Head (Academic) 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> and his web address is <www.ise.nus.edu.sg/staff/chewep/>.

SUYAN TENG is currently a Research Fellow in the Department of Industrial and Systems Engineering, National University of Singapore. She received her Ph.D. degree from National University of Singapore in 2004. Her research interests include simulation-based optimization and heuristic algorithms development for solving vehicle routing and scheduling problems. Her e-mail address is <isetsy@nus.edu.sg>