Pareto-Optimal Prioritization Scheme Design for Cadaveric Liver Allocation: A Multi-objective Stochastic Genetic Algorithm Approach

Wen-Hsin Feng[a], Zhouyang Lou[a], Nan Kong[b], Hong Wan[a]

[a] *School of Industrial Engineering, Purdue University, 315 N. Grant Street, West Lafayette, IN 47907*
[b] *Weldon School of Biomedical Engineering, Purdue University, 206 Martin Jischke Dr., West Lafaytte, IN 47907*

Abstract

Cadaveric liver is a scarce life-saving resource for end-stage liver disease patients. It is critical to design optimal schemes to prioritize the wait-list patients for receiving liver transplants. Due to the complexity of the United States liver allocation system, we use simulation-based optimization for its policy improvement. We consider multiple system outcomes simultaneously and incorporate a previously developed discrete-event simulation model into a multi-objective genetic algorithm framework to obtain Pareto-optimal policies, consisting of desirable weights placed on distinct prioritization criteria. To accommodate the stochastic nature of the system, we adapt a ranking-and-selection procedure to construct an elite chromosome set in each generation of the genetic algorithm, and use the elite chromosome set to improve the offspring generation. To ensure sufficient diversity in the population, we apply clustering to identify representative elite chromosomes. In the algorithm tuning phase, we further alleviate the computational burden of evaluating each candidate allocation policy by embedding a response surface based surrogate model in the algorithm framework. Through computational experiments, we identify promising parameter settings with the surrogate model and then obtain a set of Pareto solutions for the multi-objective stochastic optimization problem over the full-edged liver allocation simulation. Our proof-of-the-concept study on pre-transplant patient mortality rate and post-transplant graft survival rate verifies the viability of our approach. We are able to efficiently identify good balances among the allocation prioritization weights with our proposed algorithm.

Keywords: cadaveric liver allocation, simulation optimization, multi-objective optimization, genetic algorithm

1. Introduction

It is reported that more than 30,000 end-stage liver disease patients die in the United States while waiting for liver transplants [1]. To almost all these patients, the only viable option is transplant of a cadaveric donor liver, i.e., the liver is procured after the donor is dead. However, allocating cadaveric livers has been a contentious issue in the United States for decades due to significant imbalance between liver supply and patient demand [1]. Given this resource-deprived environment, it is critical to identify the *most appropriate* patients to receive each liver offer. From a policy viewpoint, it is thus important to develop promising allocation guidelines such that certain system outcomes can be optimized. In this paper, we apply simulation optimization to address the issue of optimal allocation guideline development.

Multiple criteria are used in the current allocation guideline to rank waiting list patients. These criteria include donor-recipient blood type compatibility, geographic proximity, recipient disease severity, and recipient waiting time. With the current guideline, a cadaveric liver allocator first checks whether a waiting list patient's blood type is compatible to the donor's. Then it stratifies the compatible patients into nine classes based on their disease severity and geographic proximity to the donor (i.e., three classes based on disease severity and three classes based on geographic proximity). Finally, the allocator ranks the patients within each class based on their waiting times. Once the ranking sequence of the waiting list patients is determined, the donor liver is offered sequentially. For more information on the current guideline, we refer to the website of United Network for Organ Sharing (UNOS) at *www.unos.org*.

A number of metrics are important when assessing allocation guidelines. They can be categorized into pre-transplant and post-transplant metrics. In the pre-transplant category, one may use pre-transplant queue length and pre-transplant waiting time. In the post-transplant category, one may use post-transplant organ and patient survival rates. Note that these metrics are often in disagreement when come to improving the allocation guideline. For example, if an allocation decision maker gives more preference to waiting list patients with more acute condition, then these patients tend to wait shorter to receive transplants and thus fewer patients would die while waiting. However, more acute waiting list patients tend to have higher risk of graft failure or patient mortality. As a result, there is a potential increase in post-transplant survival rates. Hence, to improve the

current guideline, we need to deal with a multi-objective optimization problem. Furthermore, since the current liver allocation system is complex, it is impossible to derive closed-form expressions for the aforementioned system outcomes. We thus resort to simulation-based optimization for the policy improvement.

In this paper, we apply a single-score ranking formula developed in Feng et al. [1] that combines the four main allocation criteria and incorporates many features in the current allocation guideline. Such a ranking formula is intended to facilitate the analysis and optimization of the allocation guideline. Note that at present, there is no clear emphasis on the direction of the policy improvement. This single-score ranking formula at least provides an efficient way to systematically investigate all the alternative prioritization criteria.

Given a donor liver l, the single-score ranking formula assigns a ranking score $Q(l,p)$ for each waiting list patient p as:

$$Q(l,p) = B(l,p)[\omega_S S(p) + \omega_M M(p)(1 - S(p)) + \omega_L L(l,p) + \omega_T T(p)]. \tag{1}$$

In Equation (1), $B(l,p)$ indicates whether l is compatible with p in terms of blood type. Following the current allocation guideline, we specify the compatibility into three categories: *identical*, *compatible but not identical*, and *non-compatible*. We assign $B(l,p) = 1$ if l and p's blood types are identical; $B(l,p) = 0$ if they are incompatible; and $B(l,p) = 0.5$ if they are compatible but not identical. Also in the equation, $S(p)$ indicates whether patient p has acute condition, i.e., $S(p) = 1$, or chronic condition, i.e., $S(p) = 0$; and $M(p)$ quantifies the disease severity of each chronic patient p. More specifically, $M(p)$ is a linear transform of the MELD (i.e., Model for End-Stage Liver Disease) score and MELD score is commonly used to assess end-stage liver disease patients' severity. In the equation, $L(l,p)$ indicates the geographic proximity between l and p. Following the current allocation guideline, we specify the proximity into three levels: *local*, *regional*, and *national*. In the numerical studies, we assign $L(l,p) = 1$ if l and p are from an area serviced by the same organ procurement organization (OPO); $L(l,p) = 0$ if they are from different allocation regions; and $L(l,p) = 0.5$ if they are from the same allocation region but not from the same OPO service area. For more information on current OPO service areas and regional composition, we refer to Feng et al. [1]. Finally, $T(p)$ standardizes the waiting time for each waiting list patient p, i.e., $T(p)$

is the ratio of patient p's waiting time to the longest waiting time among all waiting list patients nationwide when l is procured.

In the proposed single-score ranking formula, the weight vector $\omega := (\omega_S, \omega_M, \omega_L, \omega_T)$ specifies the weights assigned to various decision criteria used in allocation prioritization. Without loss of generality, we restrict the sum of the weights to be 1. When a liver is procured, a priority score is computed for each waiting list patient based on the proposed formula. The patients with higher scores are given higher priorities for the liver offer. A patient can be offered with the liver only after the offer has been rejected by all patients with higher scores.

With the introduction of the single-score ranking formula, we are now ready to propose the multi-objective optimal allocation prioritization scheme design problem, which is presented as:

$$\min_{\omega \in \mathcal{W}} \mathbb{E}_{\Theta}[\mathbf{F}(\omega, \theta)], \tag{2}$$

where $\omega \in \mathcal{W}$ is the vector of the weights and $\mathcal{W} = \{\omega \in [0,1]^m \mid \sum_{i=1}^{m} \omega_i = 1\}$. We use θ to denote a scenario to the system uncertainty, which specifies a stream of procured livers and the waiting list at the time of each procurement, and use Θ to denote the set that contains all the scenarios. The function vector $\mathbf{F}(\cdot,\cdot) : \mathcal{W} \times \Theta \to \mathbb{R}^n$, represents a vector of n system performance measures, i.e., $\mathbf{F}(\cdot,\cdot) := \{F_1(\cdot,\cdot), \ldots, F_n(\cdot,\cdot)\}$. Based on the ranking formula, we set $m = 4$ for all four weights considered in the formula. In the numerical studies, we consider two system metrics, i.e., $n = 2$, which are intuitively in disagreement. The two metrics are average pre-transplant patient mortality rate and average post-transplant graft survival rate.

In this paper, we assume that a closed-form expression does not exist for $\mathbf{F}(\cdot,\cdot)$ and one must evaluate it through simulation. Hence, we develop a simulation-based multi-objective genetic algorithm (GA). In the proposed algorithm, we iteratively update the set of weight vectors (i.e., the population maintained in the GA solution procedure). The focus of our algorithmic design is on parent selection. For which, we adapt the fitness assignment method developed by Zitzler and Thiele [2], which utilizes the concept of Pareto dominance. Furthermore, we adapt the two-phase selection procedure developed by Chen and Lee [3] to identify non-dominant solutions, which incorporates Pareto dominance into the ranking-and-selection scheme. Additionally, the proposed algorithm consists of GA operations provably effective in the literature. In our numerical studies,

we first specify the values of several algorithmic parameters in order to achieve promising computational performance. We then use the identified parameters to generate a Pareto front for the multi-objective optimal allocation prioritization scheme design problem.

In this paper, we make contribution at two fronts. We are the first that conduct comprehensive investigation on live allocation guideline optimization with the use of detailed allocation system modeling and simulation. To facilitate the simulation-based multi-objective optimization, we offer computational insights into integration of Pareto ranking-and-selection and genetic algorithms.

The remainder of the paper is organized as follows. In Section 2, we present the literature review for previous work on organ allocation policy design and simulation based multi-objective genetic algorithms. In Section 3, we briefly describe the two models used for the proposed stochastic optimization problem. In Section 4, we propose a simulation-based multi-objective genetic algorithm for the optimization problem. In Section 5, we describe preliminary experiments for tuning the algorithmic parameters with a metamodel of $\mathbf{F}(\cdot, \cdot)$, developed by Feng et al. [1]. In Section 6, we report numerical studies that apply the proposed algorithm directly to $\mathbf{F}(\cdot, \cdot)$ with a well calibrated liver allocation simulation model by Feng et al. [1]. In Section 7, we draw conclusions and outline future research.

2. Literature Review

In this section, we first describe the important work on organ allocation policy design in the area of operations research. Then we briefly review algorithm development in the areas of multi-objective optimization, genetic algorithms, and simulation optimization, with emphasis given to those methods that integrate ideas from two or more areas.

2.1. Operations Research on Organ Allocation Policy Design

Organ transplantation and allocation has drawn great attention of operations researchers, with the majority of the research focusing on liver and kidney. For a recent survey on organ allocation optimization and acceptance decision analysis, we refer to Alagoz et al. [4]. The relevant work can be categorized into two classes: those that consider decision problems at the operational level, mainly on transplantation timing and eligibility, and those that consider problems related to policy development. In the formal class, this line of fruitful research was initiated by David and Yechiali

[5, 6, 7], Ahn and Hornberger [8], and Hornberger and Ahn [9]. In the past decade, we have witnessed several accomplishments with increased incorporation of practical realism. Howard [10] developed a model to investigate the tradeoff faced by a potential recipient between accepting the present donor organ offer and rejecting it in hope to receive one with better quality in the future. Alagoz et al. [11] considered the problem that decides the optimal time of accepting a living-donor liver for maximizing a patient's total life expectancy. Alagoz et al. [12] further considered the optimal timing problem for cadaveric livers and investigated the effect of each waiting list patient not knowing his ranking on the list. Sandıkçı et al. [13] modeled the privacy concerns of waiting list patients and provided numerical studies to suggest a more transparent waiting list.

In the aspect of organ matching policy design, Zenios et al. [14] developed a deterministic fluid model to capture the watching list patient population dynamics with such components as waiting list additions, as well as waiting list removals due to receiving transplant, death, and other reasons. The authors studied the effect of modifying prioritization criteria. Su and Zenios [15] developed a stylistic sequential assignment model to seek optimal matching strategies for kidney allocation. Kong et al. [16] developed a static allocation model to analyze the effect of geographic proximity given the currently implemented allocation hierarchy. Meanwhile, A number of simulation models have been developed for quantitatively assessing the current allocation policy and its alternatives. For liver allocation, these models include Pritsker et al. [17], Ratcliffe et al. [18], Thompson et al. [19], Shechter et al. [20], and Feng et al. [1]. Similar simulation models exist for kidney [21] and heart [22].

2.2. Simulation-Based Multi-objective Genetic Algorithm

When dealing with multi-objective optimization, the notion of Pareto optimality is often applied, which refers to a set of solutions that are optimal in the sense that no other feasible solutions are superior to any of them with respect to all the objectives. These solutions are termed Pareto solutions or non-dominated solutions. Several classical methods to find Pareto optimal solutions have been developed, including the weighted sum approach, goal programming, ϵ-constraint method, among others. For a review of these methods, we refer to Cohon [23]. These methods convert a multi-objective optimization problem into a sequence of single-objective problems, in each of which a Pareto solution can be obtained. Although many of these methods guarantee the convergence to

the Pareto-optimal set, they are associated with several shortcomings as follows: (1) they require a closed-form objective function; (2) they require some prior knowledge on the problem; (3) many of them require the objective function to be convex; and (4) only one Pareto-optimal solution can be obtained in an iteration.

To overcome the above shortcomings, meta-heuristic methods have been developed. For example, Nam and Park [24], Czyzzak and Jaszkiewicz [25], and Suppapitnarm et al. [26] applied simulated annealing to multi-objective optimization. Genetic algorithms have been recognized to be well suited for multi-objective optimization since a diverse set of solutions is maintained and different solution regions may be searched with these algorithms. In addition, these algorithms require little prior knowledge on the problem as opposed to the aforementioned classic methods. Schaffer [27] developed the first multi-objective genetic algorithm. Though the algorithm does not consider the notion of Pareto optimality. Other multi-objective genetic algorithms include Fonseca et al. [28], Horn et al. [29], Srinivas et al. [30] and Zitzler et al. [2]. In general, these algorithms differ in three aspects [31, 32, 33]: (1) procedure of fitness assignment; (2) approach of maintaining solution diversity; and (3) whether Pareto solutions are stored in a set different from the one storing the population.

Simulation optimization problems refer to optimization problems that have a stochastic objective function and constraints, but have no analytical expression. Thus they can only be evaluated with computer simulation. For comprehensive reviews, we refer to Fu [34, 35]. Two well known approaches for simulation optimization are response surface methods [36] and ranking-and-selection procedures [37]. For multi-objective simulation optimization problems, one common approach is to transform a multi-objective problem into a single-objective problem [38, 39, 40].

Relatively little research has utilized the notion of Pareto optimality. Ryu et al. [41] proposed a response surface method with an adaptive weighted sum approach to find the Pareto front for deterministic problems. Chen and Lee [3] extended the ranking-and-selection procedure and proposed a two-phase Pareto set selection procedure. The first phase of the procedure is to insert to the Pareto set those solutions that are best in at least one objective. Then it performs pairwise comparisons between any pair of solutions that are temporarily identified as non-dominated, and discards those that are dominated. Our work borrows ideas from the two-phase Pareto set selection procedure. However, we further incorporate the procedure into a genetic algorithm framework.

3. A Brief Review of Self-developed Liver Allocation System Models

We first review a simulation model of the U.S. liver allocation system, previously developed by some of the same authors; see Feng et al. [1] for more details. We use this model as the black-box stochastic simulation model in this paper. This simulation model has five modules: patient generation, donor generation, donor-recipient matching, pre-transplant medical status update, and post-transplant survival prediction. For each patient entity, the patient generation module generates the following attributes: whether a re-listed patient, medical urgency, waiting time, as well as blood type and registered OPO (used to determine her geographic proximity). New patient arrivals follow a non-stationary Poisson process; re-listed patient arrivals depend on the predicted graft failure times. When a re-listed patient is generated, her registered OPO and blood type remain the same as before, but her waiting time is reset to 0. The donor generation module creates a stream of donor liver arrivals following a non-stationary Poisson process. If the donor is procured but determined not to be wasted, the donor-recipient matching module prioritizes the waiting list patients by various prioritization schemes, including the single-score ranking formula described earlier. The pre-transplant medical status update module updates the medical urgency for each patient that will remain on the waiting list for an additional day. If a patient remains on the waiting list for an additional day, we update her medical urgency with individual-level transitions whose transition probabilities were determined by a retrospective clustering based time-series analysis interpolation study. We quantify chronic patients medical urgency using MELD scores, but do not differentiate acute patients medical urgency (in other words, acute is only one state on the transitions). The post-transplant survival prediction module predicts how long an organ recipient and the transplanted graft can be functioning after the transplant occurs. If the graft is predicted to function longer, then the patient will die and be removed from the system; if the patient is predicted to function longer, then the patient will get re-listed with a probability at the time the graft fails.

Next we review a set of metamodels that estimate system outcomes of the aforementioned liver allocation simulation. These metamodels were also developed by Feng et al. [1]. In this paper, we use these metamodels to tune the GA parameters because the full-fledged simulation model is computationally expensive. To develop the metamodels, a mixture design was used to determine

a set of weight vectors to be the design points. For each design point, the simulation model was run for 10 replications. Then the results from various design points were fitted into full factorial linear, quadratic, cubic models. Finally, the best model for each system outcome was identified. With each metamodel, the weight vector that optimizes the corresponding system outcome was also obtained. In this paper, we include these previously identified optimal weight vectors in the initial population of the GA.

4. A Simulation-based Genetic Algorithm Framework

In this section, we first provide an overview of the GA framework and an illustration of the framework in *Figure 1*. Now that we present a GA method in this paper, we use *chromosome* and *solution* interchangeably in the remainder of the paper.

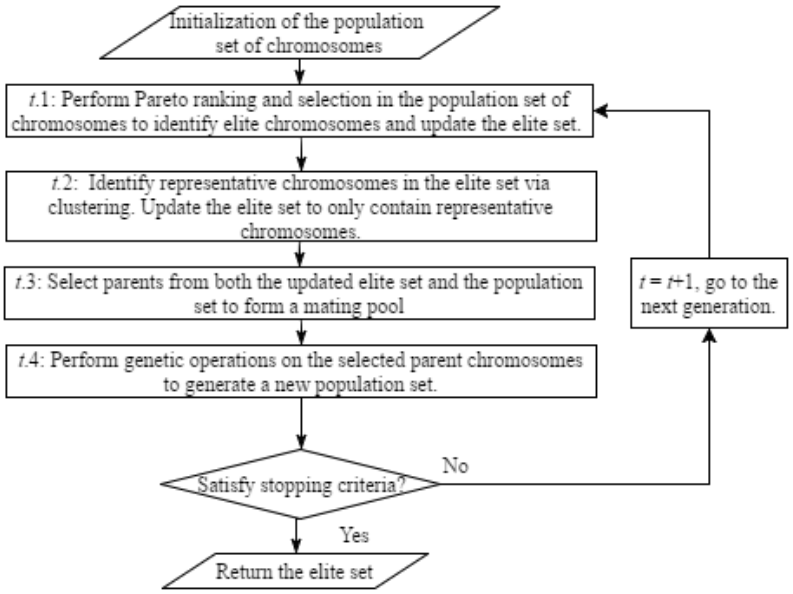


Figure 1: Flow Chart of the Genetic Algorithmic (GA) Framework

We provide a more detailed description of the overall framework in the following. The algorithm uses real numbered coding for the chromosomes. At each generation t, the algorithm maintains a population set of chromosomes, denoted by $\mathcal{P}_t$, and a separate elite set of chromosomes, denoted by $\mathcal{E}_t$. In the elite set, each chromosome is required not to be dominated by any chromosome in $\mathcal{S}_t = \mathcal{P}_t \cup \mathcal{E}_t$. As the algorithm starts, i.e., $t = 1$, the initial population set can be randomly generated or constructed with prior knowledge. Both initial population and elite sets are set empty

Algorithm 1: Simulation-based Genetic Algorithm Framework

- **Initialization.**
  Set the initial population set $\mathcal{P}_1$ with $|\mathcal{P}_1| \leq n_{\mathcal{P}}$. Set $t = 1$.
- **Step t.1: Pareto Ranking and Selection.**
  Set $\mathcal{E}_t = \{\omega^i \in \mathcal{P}_t | \nexists \omega^k \in \mathcal{P}_t, F_j(\omega^k) \leq F_j(\omega^i), \forall j = 1, \ldots, n; F_{\hat{j}}(\omega^k) < F_{\hat{j}}(\omega^i)$ and for some $\hat{j}\}$
- **Step t.2: Clustering.**
  Reduce the elite set $\mathcal{E}_t$ via clustering such that $|\mathcal{E}_t| \leq n_{\mathcal{E}}$. Each cluster is then represented by a centroid chromosome, which has the smallest aggregate distance to all other chromosomes in the same cluster. Then update $\mathcal{S}_t = \mathcal{P}_t \cup \mathcal{E}_t$.
- **Step t.3: Parent Selection for reproduction.**
  Select parents from $\mathcal{S}_t$ to form a mating pool $\mathcal{M}_t$.
- **Step t.4: Genetic Operations.**
  Generate the new population set $\mathcal{P}_t$ from $\mathcal{M}_t$.
- **Termination.**
  If t reaches $t = T$, where T is the desired generation limit, output $\mathcal{E}_t$ and stop; otherwise, set $t = t + 1$ and go to Step t.**1**.

and capped with constants $n_{\mathcal{P}}$ and $n_{\mathcal{E}}$, respectively.

At each generation t, first in step t.1, a Pareto ranking-and-selection procedure is used to compare chromosomes in $\mathcal{P}_t$ and $\mathcal{E}_t$ with respect to all objectives. Then non-dominated chromosomes are inserted to $\mathcal{E}_t$. Since significant computational effort will be required to evaluate the objectives for all chromosomes in $\mathcal{S}_t$ and there may be little separation among different chromosomes in $\mathcal{E}_t$, we thus cap the size of $\mathcal{E}_t$ with a pre-specified upper bound, i.e., $|\mathcal{E}_t| \leq n_{\mathcal{E}}$, via clustering in step t.2. Thus, $\mathcal{E}_t$ is updated in this step. Then in step t.3, the fitness value is determined for each chromosome in $\mathcal{S}_t$, and a subset of $\mathcal{S}_t$ is selected to be parents based on their fitness values. Finally, in step t.4, genetic operations are applied to selected parent chromosomes to generate offsprings. Since generated offsprings may be infeasible, we implement a chromosome repair procedure to convert those infeasible chromosomes to feasible ones in this step as well. Those feasible chromosomes are included in $\mathcal{P}_t$. If necessary, their parents are excluded from $\mathcal{P}_t$ to cap its size below $n_{\mathcal{P}}$. If the number of GA generations does not reach the prespecified generation limit, the algorithm advances to the next generation once the above four steps are completed; Otherwise, the algorithm terminates and outputs the final elite set. In the remainder of this section, we provide more details on steps t.1-t.4.

4.1. Stochastic Pareto Ranking and Selection

We adapt the two-phase Pareto ranking-and-selection procedure developed by Chen and Lee [3]. We apply the procedure at each generation of the proposed GA, namely Step $t.1$ in *Algorithm 1.* The key of the procedure is to incorporate Pareto optimality into the ranking-and-selection procedure. In the first phase, the procedure identifies the best chromosome with respect to each performance measure, and inserts these chromosomes into the elite set. In the second phase, the procedure augments the elite set by inserting the chromosomes originally belonging to the population set if they are not dominated by any chromosome in the elite set. This procedure intelligently allocates computationally expensive simulations for chromosome evaluation and provides a probabilistic guarantee on the correct selection of an individual chromosome to be elite (or say non-dominated). *Figure 2* shows the flowchart of the two-phase Pareto ranking-and-selection procedure.

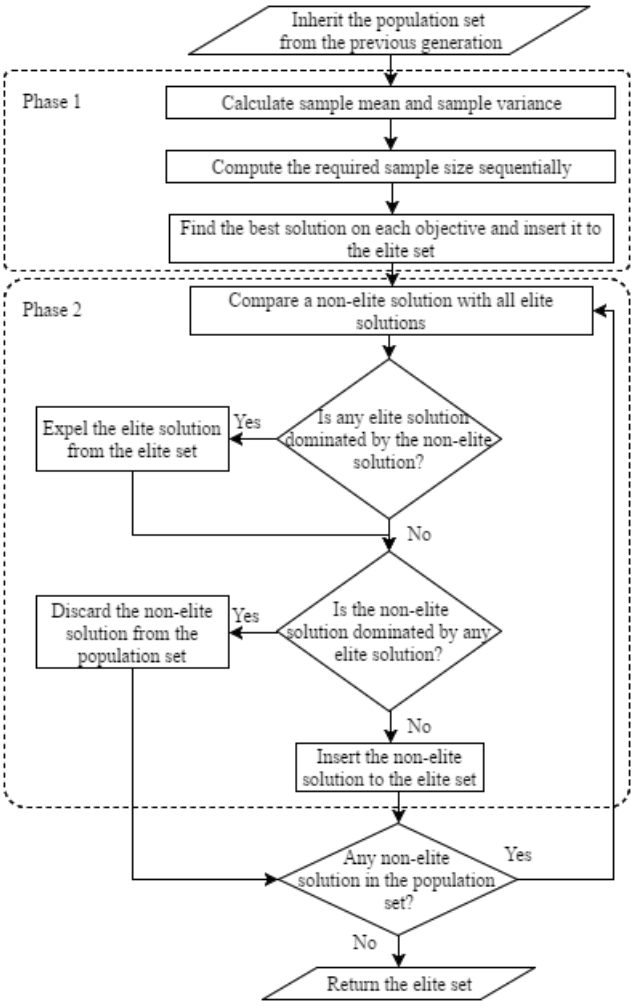


Figure 2: Flow Chart of the Two-phase Stochastic Pareto Ranking and Selection

Next we describe the two phases in detail. For presentation convenience, we drop the generation index t. In the first phase, we apply the sequential ranking-and-selection procedure, based on the indifference-zone approach [3], to select the best chromosome with respect to each individual performance measure. Like several other ranking-and-selection procedures in stochastic simulation [37, 42], the procedure assumes for each chromosome, the sample observations are independent and identically distributed with normal distribution, the variance of sample observations can be unknown and unequal among various chromosomes. We describe the first phase in the following and provide an algorithmic statement in *Algorithm 2*. At any generation, for each performance measure indexed by $j = 1, \ldots, n$, the procedure takes initial sampling on $F_j(\omega)$ for any given chromosome $\omega \in \mathcal{S}$, and calculates the sample means, denoted by $\bar{F}_j(\omega)$. The procedure then selects the two chromosomes with the largest sample means. We use $b_1^{(j)}$ and $b_2^{(j)}$ to denote their indices among chromosomes in $\mathcal{S}$, and thus the two largest sample means are $\bar{F}_j(\omega^{b_1^{(j)}})$ and $\bar{F}_j(\omega^{b_2^{(j)}})$. A t-test is then conducted for each performance measure to quantify the statistical significance of the difference between the two largest sample means. If such difference is statistically significant for some performance measure, we designate the better chromosome to be an elite chromosome. Otherwise, additional samples are allocated to realize the statistical significance. We determine the number of additional samples based on the initial sample variance, the critical Rinotts constant, and a quantity measuring the difference between the pair of chromosomes. The critical Rinotts constant is dependent on some pre-specified precision quantity, the number of compared chromosomes, and the initial sample size. The difference measure is computed as follows. For each $\omega^i \in \mathcal{S}$ and $j = 1, \ldots, n$, we have

$$d_{ij} = \begin{cases} \bar{F}_j(\omega^{b_2^{(j)}}) - \bar{F}_j(\omega^i), & \text{if } i = b_1^{(j)}; \\ \bar{F}_j(\omega^i) - \bar{F}_j(\omega^{b_1^{(j)}}), & \text{otherwise.} \end{cases} \tag{3}$$

In our numerical studies, it is suggested that the number of additional samples required can be large at any iteration when the difference between the sample means of the two best chromosomes is relatively small. Hence, we impose an upper bound, denoted by $\bar{\delta}$, on the increment of the sample size from the current iteration to the next iteration to ensure computational tractability. This approach is also intended to reduce the possibility of over-sampling. In the following, we present a formal algorithmic statement of the first-phase procedure.

Input Parameters: $\mathcal{S}, n_0, \bar{\sigma}$ and p^*, the population set from the previous generation, the pre-specified initial sample size, the maximum size of additional samples to be taken, and the upper bound on the probability that any selected chromosome from the elite set is non-dominated, respectively. Denote $n_{i,r}$ to be the number of samples allocated for chromosome $\omega^i \in \mathcal{S}$ at iteration r.

Algorithm 2: The First Phase of the Stochastic Pareto Selection

1. Set $r = 0, D_1 = D_2 = \emptyset$. Set $n_{i,0} = n_0, \forall i = 1, \ldots, |\mathcal{S}|$.
2. For each $\omega^i \in \mathcal{S}$ and $j \in J := \{1, \ldots, n\}$, calculate the sample means $\bar{F}_j(\omega^{b_1^{(j)}})$ and $\bar{F}_j(\omega^{b_2^{(j)}})$ for each $j \in J$, such that, $\bar{F}_j(\omega^{b_1^{(j)}}) \leq \bar{F}_j(\omega^{b_2^{(j)}}) \leq \cdots$.
3. Compute the t score for each $j \in J$ as
$$T_j = \frac{\bar{F}_j(\omega^{b_2^{(j)}}) - \bar{F}_j(\omega^{b_1^{(j)}})}{\sqrt{s^2_{b_2^{(j)}}/n_{b_2^{(j)},0} + s^2_{b_1^{(j)}}/n_{b_1^{(j)},0}}}$$
4. Set $D_1 = D_1 \cup \{\omega^{b_1^{(j)}} | T_j > t_{p,n_b^{(j)}-1}\}$ and $J = J \backslash \{j | T_j > t_{p,n_b^{(j)}}\}$, where $n_b^{(j)} = \min(n_{b_1}^{(j)}, n_{b_2}^{(j)}), p = (p^*)^{\frac{1}{|\mathcal{S}|-1}}$, and $t_{p,f}$ is the p^{th} percentile of the t distribution with f degrees of freedom.
5. If $J = \emptyset$, STOP and output D_1, and set $D_2 = \mathcal{S} \backslash D_1$, and let $n_i = n_{i,r}$, $\forall i = 1, \ldots, |\mathcal{S}|$.
6. Select $\tilde{j}$:= $\text{argmax}_{j \in J}\{T_j - t_{p,n_b^{(j)}-1} \Big| T_j < t_{p,n_b^{(j)}-1}\}$. Calculate the sample size at the next iteration for each i as $n_{i,r+1} = \max(n_0 + 1, \lceil (hs_{i\tilde{j}}/d_{i\tilde{j}})^2 \rceil)$, where $d_{i\tilde{j}}$ is computed according to Equation (3).
7. If $n_{i,r+1} \leq n_{i,r}, \forall \omega^i \in \mathcal{S}$, set $D_1 = D_1 \cup \{\omega^{b_1^{(\tilde{j})}} | T_{\tilde{j}} > t_{p,n_b^{(\tilde{j})}-1}\}, J = J \backslash \{j | T_j > t_{p,n_b^{(j)}}\}$, and go to Step 5; otherwise, take additional samples of size $\delta_{i,r+1} = \min(\bar{\delta}, n_{i,r+1} - n_{i,r}), \forall i = 1, \ldots, |\mathcal{S}|$.
8. Set $n_{i,r+1} = n_{i,r} + \delta_{i,r+1}$ and $r = r + 1$. Go to Step 2.

After identifying the best chromosome for each performance measure in the first phase, we insert all the best chromosomes in D_1, which later becomes the initial elite chromosome set for the second phase. We identify other elite chromosomes in the second phase by conducting pairwise comparisons between each non-elite chromosome in D_2 with each elite chromosome already identified in D_1. For each pairwise comparison, we examine the confidence interval of the difference on the sample means. Since the estimation of each sample mean may require a distinct sample size, which results in a different variance, we apply the Welch confidence interval [43] in the comparison. If a non-elite chromosome is not dominated by any elite chromosome, it becomes an elite chromosome and is

included in the elite chromosome set; if an elite solution is dominated by a non-elite chromosome, it is excluded from the elite set. This algorithm continues until all pairwise comparisons are made. Finally, the algorithm returns the elite set that contains all non-dominated chromosomes. In the following, we present a formal algorithmic statement of the second-phase procedure.

Input Parameters: D_1, D_2, n_i, and p^*, the initial elite chromosome set for the second phase, the initial non-elite chromosome set for the second phase, the sample size for each $\omega^i \in \mathcal{S}$ after the first phase, and the upper bound on the probability that any selected chromosome from the elite set is non-dominated, respectively.

Algorithm 3: The Second Phase of the Stochastic Pareto Selection

---

1. Select a chromosome $\omega^i \in D_2$, for each performance measure $j = 1, \ldots, n$, we calculate the confidence interval of the difference between the selected $\omega^i \in D_2$ and any $\omega^k \in D_1$, denoted by $W_{ik}^{(j)}$ as:

$$W_{ik}^{(j)} = (\bar{F}_j(\omega^i) - \bar{F}_j(\omega^k) - v_{ik}^{(j)}, \bar{F}_j(\omega^i) - \bar{F}_j(\omega^k) + v_{ik}^{(j)}) := (L(W_{ik}^{(j)}), U(W_{ik}^{(j)})),$$

where $v_{ik}^{(j)} = t_{p, f_{ik}^{(j)}} \sqrt{s_{ij}^2/n_i + s_{kj}^2/n_k}, p = (p^*)^{\frac{1}{|D_1 \cup D_2| - 1}}$, and $f_{ik}^{(j)}$, the degree of freedom is calculated as:

$$f_{ik}^{(j)} = \frac{(s_{ij}^2/n_i + s_{kj}^2/n_k)^2}{(s_{ij}^2/n_i)^2/(n_i - 1) + (s_{kj}^2/n_k)^2/(n_k - 1)}$$

2. If $L(W_{ik}^{(j)}) \leq 0$, for all j, then set $D_1 = D_1 \backslash \{\omega^k\}$, i.e., ω^k is dominated by ω^i. If for some k, there exists some j such that $U(W_{ik}^{(j)}) < 0$, then set $D_1 = D_1 \cup \{\omega^i\}$, i.e., ω^i is not entirely dominated by the elite chromosomes.
3. Set $D_2 = D_2 \backslash \{\omega^i\}$.
4. If $D_2 \neq \emptyset$, go to Step 1; otherwise, STOP and output $\mathcal{E} = D_1$.

4.2. Clustering

In many cases, non-dominated chromosomes should be clustered for the following reasons. One, it may be unnecessary to a decision maker to be presented with a Pareto front that contains an extremely large number of or even infinitely many chromosomes. However, such a Pareto front is likely to be derived if one keeps all identified non-dominated chromosomes in the GA search process. In addition, when dealing with too many non-dominated solutions, it is likely to increase computational burden in the GA [44]. Finally, non-dominated solutions are preferred to be distributed uniformly to avoid the GA convergence towards certain small regions, which is known

as *genetic drift* [45]. Thus, it is suggested or even mandatory to prune non-dominated solutions in order to maintain the solution diversity in the GA. In our GA implementation, we apply clustering to select representative chromosomes among those that are similar, e.g., only a short Euclidean distance exists between the estimates of their objective function values.

At each generation of the GA, after Pareto ranking-and-selection, we construct an elite chromosome set $\mathcal{E}$. We then apply clustering to reduce the size of the elite set while maintaining its representativeness. In our GA, we adapt the average linkage cluster method, which has been proven to perform well in GA [46]. At the beginning of the clustering, each chromosome constitutes a distinct cluster. The similarity between any two clusters is measured by the averaging Euclidean distance between the chromosomes from one cluster and those from the other. The two clusters with the smallest average distance are merged. The procedure continues until the number of clusters does not exceed $n_{\mathcal{E}}$, a pre-specified size limit. Each cluster is then represented by a centroid chromosome, which has the smallest aggregate distance to all other chromosomes in the same cluster. In the following, we present a formal algorithmic statement of the clustering procedure.

Algorithm 4: Clustering Procedure

---

1. Construct $\mathcal{C} := \bigcup_i \left\{\{\omega^i\}\right\}, \forall \omega^i \in \mathcal{E}$.
2. If $|\mathcal{C}| \leq n_{\mathcal{E}}$, go to Step 5.
3. Calculate the average distance between any two disjoint subsets C_m, C_n of $\mathcal{E}$, as:

$$d_{ij} := \frac{1}{|C_m||C_n|} \sum_{\omega^m \in C_m; \omega^n \in C_n} ||\omega^m - \omega^n||, \forall C_i, C_j \in \mathcal{C}, C_i \cap C_j = \emptyset,$$

   where $||\cdot||$ denotes the Euclidean distance on the space induced by ω's.
4. Set $\mathcal{C} \leftarrow \mathcal{C} \backslash \{C_{m^*}, C_{n^*}\} \cup \{C_{m^*} \cup C_{n^*}\}$, where $(m^*, n^*) = \arg\min_{m,n} d_{mn}$. Go to Step 2.
5. Find $\omega^{\tilde{i}} \in C_i$ for each $C_i \in \mathcal{C}$ such that $\tilde{i} = \arg\min_i d(\omega^i) := \arg\min_i \sum_{\omega^k \in C_i} ||\omega^i - \omega^k||$. Output the updated elite chromosome set $\mathcal{E}$ that contains all $\omega^{\tilde{i}}$, i.e., $\mathcal{E} = \bigcup_i \{\omega^{\tilde{i}}\}$ for each $C_i \in \mathcal{C}$.

4.3. Parent Selection for Reproduction

At each generation of the GA, after updating $\mathcal{E}$ with clustering, we use it together with $\mathcal{P}$ to select parents to form the mating pool $\mathcal{M}$. Such selection is based on the fitness of each chromosome in $\mathcal{E} \cup \mathcal{P}$. In general, fitness assignment is based on counting the number of dominance among the chromosomes. Unlike in the problems with deterministic objectives, the dominance between any

pair of chromosomes is certain and thus the fitness of each chromosome can be uniquely determined, the fitness can only be determined probabilistically in the problems with stochastic objectives. Hence, we use the dominance probability to determine the fitness values.

With the sample mean and variance of each chromosome obtained in the stochastic Pareto ranking-and-selection procedure, we can compute the dominance probability with respect to each performance measure. Then we compute the joint dominance probability as follows, under the assumption that the performance measures are independent. Given an elite chromosome $\omega^i \in \mathcal{E}$ and a non-elite solution $x^k \in \mathcal{P}$, the probability that ω^i dominates ω^k, denoted by ρ_{ik}, is as:

$$\rho_{ik} = \prod_j P(F_j(\omega^i) \leq F_j(\omega^k)) = \prod_j \Phi(-\frac{\bar{F}_j(\omega^i) - \bar{F}_j(\omega^k)}{\sqrt{s_{ij}^2 + s_{kj}^2}}) \tag{4}$$

where $\Phi(\cdot)$ is the cumulative probability function for standard normal distribution.

With the joint dominance probabilities, we then assign a fitness value to each chromosome. Several fitness assignment methods for multi-objective optimization problems are given in Konak and Smith [32]. In our algorithm, we adapt a ranking procedure, which is based on Pareto dominance and developed by Zitzler and Thile [2]. The fitness of an elite chromosome is defined to be proportional to the number of chromosomes it dominates. The fitness of a dominated chromosome is defined as one plus the sum of fitness of non-dominated chromosomes by which it is dominated. Note that smaller fitness values imply more fitting among the chromosomes. After assigning the fitness values, two chromosomes are randomly selected with binary tournament selection [47]. The one with better fitness is selected as a parent and inserted into the mating pool for reproducing offsprings. The parent selection procedure continues until the size of the mating pool reaches the population size. In the following, we present a formal algorithmic statement of the procedure.

Input Parameters: $\mathcal{P}$ and $\mathcal{E}$.

Algorithm 5: Parent Selection Operation

1. **Fitness Assignment.**
   (a) Calculate the dominance between each pair of $x^i \in \mathcal{E}$ and $x^k \in \mathcal{P}$ as in (4).
   (b) Calculate the fitness value of each $\omega^i \in \mathcal{E}$ as $f_i := 1 + \sum_{\omega^k \in \mathcal{E}} \frac{\rho_{ik}}{1 + n_{\mathcal{S}}}$, where $n_{\mathcal{S}} = |\mathcal{P} \cup \mathcal{E}|$.
   (c) Calculate the fitness value of $\omega^k \in \mathcal{P}$ as $f_k := 1 + \sum_{\omega^i \in \mathcal{P}} \rho_{ik} f_i$.
2. **Binary Tournament Selection**. Randomly select two solutions $\omega^m, \omega^n \in \mathcal{P} \cup \mathcal{E}$. Set $\mathcal{M} \leftarrow \mathcal{M} \cup \{\omega^n\}$ if $f_n \leq f_m$. Otherwise, set $\mathcal{M} \leftarrow \mathcal{M} \cup \{w^m\}$.
3. If $|\mathcal{M}| \leq |\mathcal{P}|$, go to Step 3. Otherwise, STOP and ouput $\mathcal{M}$.

4.4. Genetic Operations

Since the chromosomes are not encoded with binary strings, we develop genetic operations for real-number representation, namely convex combination for crossover operations and non-uniformity for mutation operations. Herrera et al. [48] provide an overview of genetic operations for real-number representation.

At each generation of the GA, parents in the mating pool obtained from the parent selection procedure are used for crossover with a pre-specified *probability of crossover*. For each pair of parents, we generate two children by applying convex combination of the parents with a random coefficient. We replace the parents with the children. After crossover, we apply nonuniform mutation for every element in a solution with a pre-specified *probability of mutation*. With the mutation operation, each selected element moves towards the two limits on the element. The direction along which the element moves and the distance with which it moves are determined randomly. In addition, the distance is degraded in later generations to ensure the search space to be more global in early generations and more local in later generations. To repair a likely infeasible solution after the mutation, we standardize the solution by the sum of its elements. At the end of the genetic operations, the chromosomes in the mating pool form the population at the next generation. In the following, we present a formal algorithmic statement for the genetic operations at each generation. We denote ω_i to be the i^{th} element in chromosome ω.

Input Parameters: $\mathcal{M}, p_c, p_m,$ and b, the mating pool, the crossover probability, the mutation probability, and the degree of non-uniformity, respectively.

Algorithm 6: Genetic Operations

1. For any pair of chromosomes $(\omega^i, \omega^j) \in \mathcal{M}$, generate random numbers μ_i, and μ_j. If $\mu_i, \mu_j \leq p_c$, apply a crossover operation as follows.
   (a) Generate a random number $\lambda \in [0, 1]$, set $\bar{\omega}^i = \lambda\omega^i + (1-\lambda)\omega^j$ and $\bar{\omega}^j = \lambda\omega^j + (1-\lambda)\omega^i$.
   (b) Set $\mathcal{P} \leftarrow \mathcal{P} \backslash \{\omega^i, \omega^j\} \cup \{\bar{\omega}^i, \bar{\omega}^j\}$.
   (c) Set $\omega^i \leftarrow \bar{\omega}^i$ and $\omega^j \leftarrow \bar{\omega}^j$
2. For each solution ω^i, apply a mutation operation as follows.
   (a) Generate a random number $\mu_l^i \in [0, 1]$ for its l^{th} element.
   (b) If $\mu_l^i < p_m$, mutate ω^i with a random number $r \in [-1, 1]$ as:

$$\bar{\omega}_l^i = \begin{cases} \omega_l^i + r(1 - \frac{t}{T})^b [U_l^i(\omega_l^i) - \omega_l^i], \text{ if } r \geq 0; \\ \omega_l^i + r(1 - \frac{t}{T})^b [\omega_l^i - L_l^i(\omega_l^i)], \text{ if } r < 0. \end{cases}$$

   where T is the upper limit on the total number of generations, b is the degree of non-uniformity, and $U(\cdot)$ and $L(\cdot)$ are the upper and lower bounds on the element, respectively.
   (c) Set $\omega_j^i \leftarrow \omega_j^i / (\sum_j \bar{\omega}_j^i)$.
3. STOP and output $\mathcal{P}$.

5. Preliminary Experiments for Parameter Tuning

Before performing the GA algorithm on the self-developed liver allocation simulation model, we assessed the effect of algorithm parameters' values and identified promising values for these parameters by performing the GA algorithm on the self-developed metamodel. For more information on the two models, we refer the readers back to *Section 3*. The parameters include three categories: (1) two parameters on chromosome sets, (2) three parameters on genetic operations, and (3) two parameters on Pareto selection. The resulting Pareto fronts would be compared with varying values of the parameters in one category and fix those in other categories. The parameters on chromosome sets are the constant number of chromosomes in the population set, $n_{\mathcal{P}}$, and the maximum number of elite chromosomes, $n_{\mathcal{E}}$. The parameters on genetic operations are the probability of crossover, p_c,

the probability of mutation, p_m, and the degree of non-uniformity in mutation, b. One parameter on Pareto selection, denoted by p^*, specifies the desired minimal probability of correct selection. The other parameter a is used for checking whether additional samples are needed. If the ratio of the sample standard error to the sample mean is no larger than a, the sample size is then regarded sufficient to estimate the expectation of the system performance for the candidate solution. In other words, the number of additional samples, $\sigma_{i,r+1}$ in Step 8 at the first phase of stochastic Pareto selection (*Algorithm* 2) is set to be 0, if $\frac{S_{\tilde{j}i}/\sqrt{n_{i,r}}}{\bar{F}_{\tilde{j}i}} \leq a$.

To identify promising parameter settings, we set the initial sample size to be 20 throughout our experiments and run the GA algorithm on the metalmodels to compare the Pareto fronts. In the following figures, we negate the sign for graft survival rate for appearance of Pareto fronts. Intuitively, promising parameter settings should yield those Pareto fronts at the left bottom corner, indicating lower pre-transplant patient mortality rate and higher post-transplant graft survival rate. Alternatively, to make the comparison more precisely, we applied the notion of *relative dominance* as introduced in *Appendix A*.

We took an ad-hoc way to examine parameter settings under the requirement that the experiment needs to be completed within a reasonable time window. We started the examination on the setting of $(n_{\mathcal{E}}, n_{\mathcal{P}})$. Figure 3 reports the Pareto fronts for different $(n_{\mathcal{E}}, n_{\mathcal{P}})$ values. The Pareto fronts for different settings are distinguished by color and marker. By comparing the relative dominance, it appeared that $(n_{\mathcal{E}}, n_{\mathcal{P}}) = (5, 5)$ had the most superior Pareto front. On the other hand, when the size of the elite size is small, it may not provide sufficiently many Pareto solutions since there would only be a small number of elite solutions on the Pareto front. As a compromise, we chose $(\hat{n}_{\mathcal{E}}, \hat{n}_{\mathcal{P}}) = (10, 5)$, where the hat means the selected setting.

Next we examined different settings of (p^*, a) while the values of $(\hat{n}_{\mathcal{E}}, \hat{n}_{\mathcal{P}})$ were used in the experiments. The results are reported in Figure 4. The figure shows that higher probabilities of correct selection p^* provided inferior Pareto fronts. This can be explained as follows. For increased p^*, more samples would be needed, which leads to less computational budget for searching new solutions, and consequently, results in inferior Pareto fronts. By comparing the relative dominance, we chose the setting of $(\hat{p}^*, \hat{a}) = (0.60, 0.05)$.

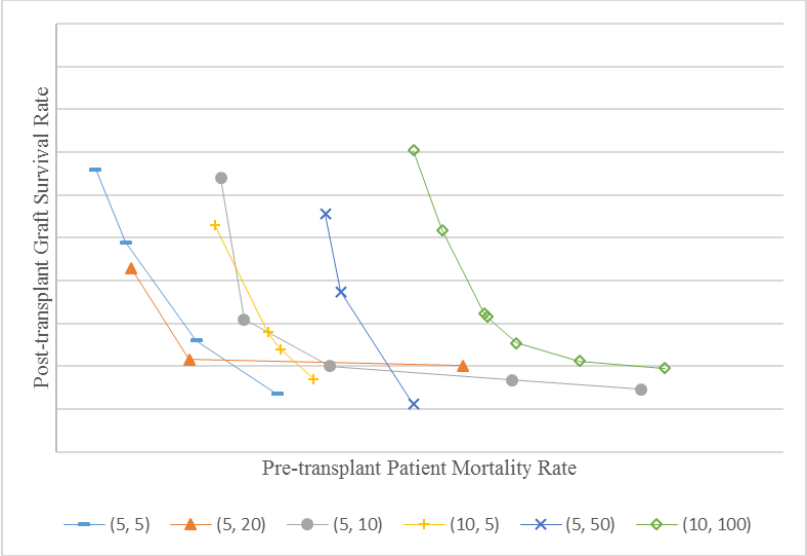


Figure 3: Pareto Fronts for Various Settings of $(n_{\mathcal{E}}, n_{\mathcal{P}})$

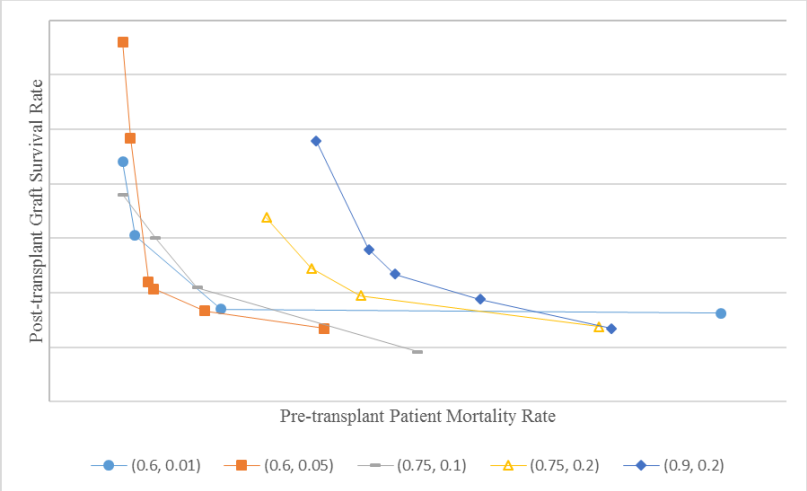
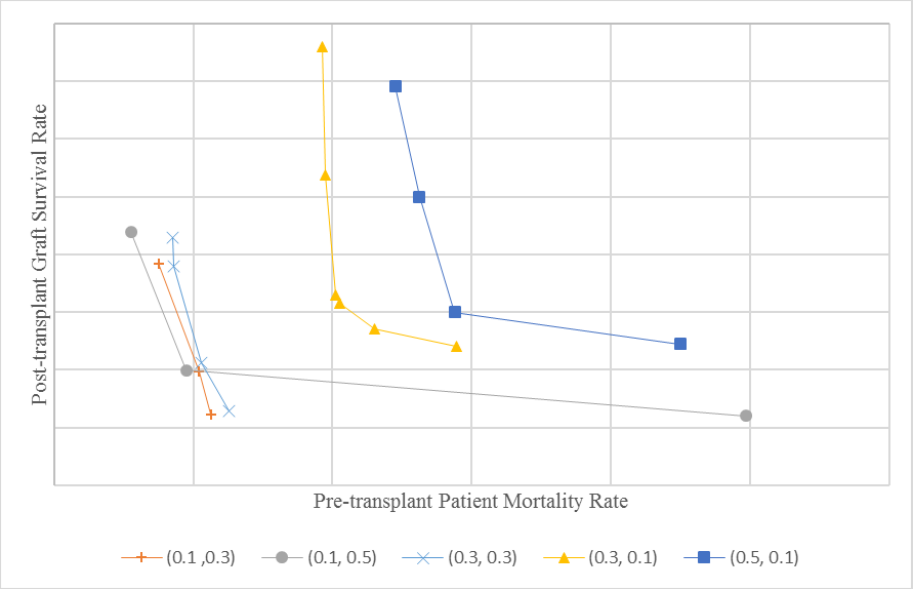
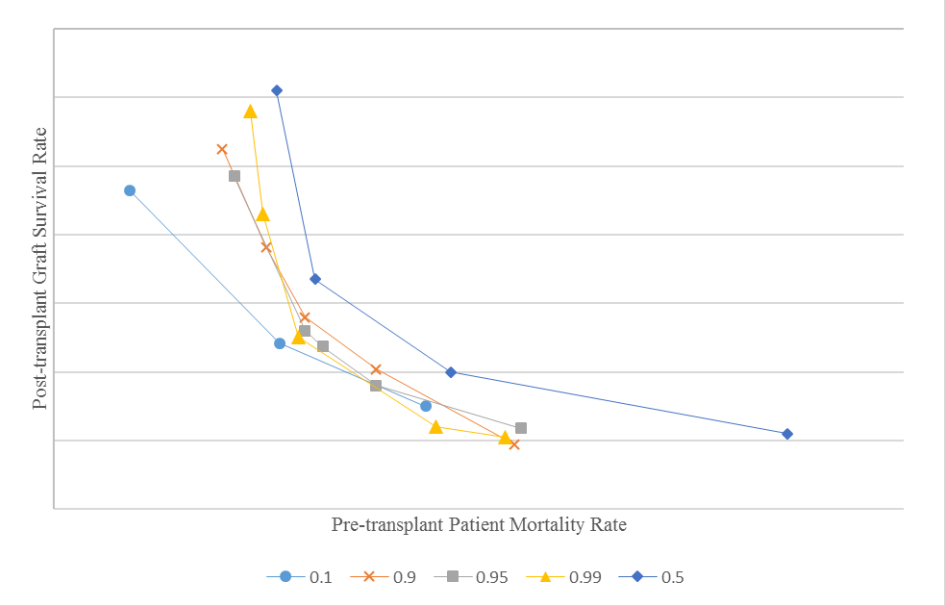


Figure 4: Pareto Fronts for Various Settings of (p^*, a)

Finally, we examined different settings of (p_c, p_m). The results are reported in Figure 5. Figure 5a suggests that p_m pushes the resultant Pareto fronts leftward if it increases. Figure 5b implies that there is no clear correlation between the Pareto front and the value of b. By comparing the relative dominance, we chose the setting of $(\hat{p}_c, \hat{p}_m, \hat{b}) = (0.10, 0.30, 0.90)$.



(a) Parameters (p_c, p_m)



(b) Parameter b

Figure 5: Pareto Fronts for Various Settings of (p_c, p_m, b)

6. Numerical Studies with the Full-Fledged Simulation

We specified the promising parameter settings identified from Section 5 and ran the multi-objective genetic algorithm over the full-fledged simulation. The algorithm was terminated at the 500^{th} generation. Figure 6 shows the Pareto fronts after selected numbers of generations. Note that the negative scales shown along the y-axis were resulted from minimizing the negation of the objective pre-transplant graft survival rate. The display here is similar to Figures 3, 4, and 5, in that regard. For display purpose, these Pareto fronts were modified by removing the elite solutions that are dominated by the convex combinations of other elite solutions. Note that in our genetic algorithm, only pairwise comparisons were conducted between elite solutions. Thus, connecting the successive elite solutions may result in non-convex line segments. It is clearly shown in Figure 6 that as the genetic algorithm progresses, the Pareto front is moved towards left and bottom of the figure. Further note that there could be intersections among the Pareto fronts (e.g., between the Pareto front after 250^{th} generation and the one after 300^{th} generation). This occurs because we performed clustering to select representative elite solutions and to include only those in the elite set for future generations. Hence, it is possible that an elite solution from a previous generation might have remained dominant in later generations if it was kept in the elite set. However, it was deleted with the clustering.

As shown in Figure 6, our proposed algorithm is a viable solution approach. The relative dominance of the Pareto front after the 500^{th} generation over the initial Pareto front is 2.11, indicating significant dominance. Comparing the elite solutions at the terminal generation with the initial generation, we found that the weights on all criteria at the terminal generation are between the weights in the two rather dichotomously selected solutions at the initial generation. Not surprisingly, as a result, the post-transplant graft survival rate and the pre-transplant wait-list patient mortality rate fall in the two extreme objective values from the initial generation. Generally speaking, the weights in the elite solutions obtained in later generations tend to have larger values on w_T and lower values on w_S, which implies more weight to be placed on the criterion of waiting time and less weight to be placed on criterion of medical urgency. Intuitively, larger value on w_T leads to increased graft survival rate whereas larger value on w_S leads to decreased patient mortality rate. So the results suggest the genetic algorithm tend to find those elite Pareto solutions

that favor increased post-transplant graft survival than reduced pre-transplant patient mortality.

Further, observing the progression of the Pareto front, we found that the quality of the Pareto front improved noticeably from the initialization phase to the 50^{th} generation into the genetic algorithm, while the improvement was of a much smaller scale from the 50^{th} to the 500^{th} generation. This may imply that with the current parameter setting, the algorithm has presented some form of convergence after a relatively small amount of time. In principle, the time spent at each generation does not differ much.

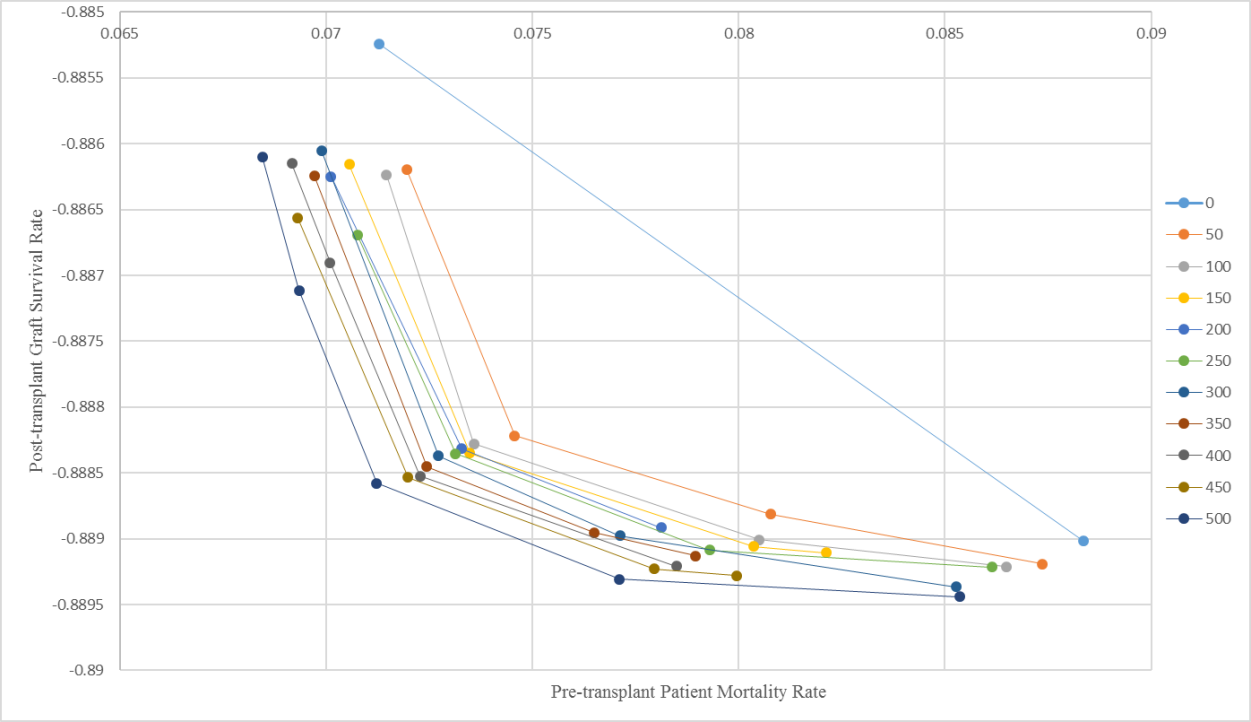


Figure 6: Progression of the Pareto Fronts

7. Conclusions and Future Research

We developed a simulation-based multi-objective genetic algorithm for the cadaveric liver allocation prioritization scheme optimization problem, which is based on a ranking formula consisting of four allocation weighting coefficients and two previously developed liver allocation system models. Our proposed algorithm constructs a set of Pareto solutions for diverse system performance measures, which cannot be directly evaluated with closed-form expressions of the weighting coefficients. The algorithm incorporates Pareto dominance into a two-phase ranking-and-selection procedure to

identify non-dominated solutions. It also employs a genetic algorithm to select parent solutions and generate offsprings for the next generation. The Pareto solutions obtained will allow policy makers to select their desired weighting coefficients based on their preferences on the performance measures. To efficiently ensure the viability of the algorithm, we based a response-surface driven surrogate model to select promising algorithm parameter settings.

The main challenge in our algorithm design was how to effectively adapt a genetic algorithm framework to deal with the multi-objective stochastic optimization problem. The uncertainty arising in the objectives requires two methodological innovations in the genetic algorithm: (1) fitness assignment to each solution for later selection operation, and (2) finding non-dominated solutions by comparing alternatives. The first innovation was computing the stochastic dominance between any two solutions to determine the fitness of a solution. The second innovation was adapting a two-phase Pareto selection procedure based on a sequential ranking-and-selection to determine the non-dominated solutions in a population.

We present a few future research items in the following. A crucial assumption in the algorithm design is the independence among the system outcomes, which ensure the validity of the formula used in the algorithm to compute the joint dominance probability, i.e., equation (4). To relax this assumption, we will update the formula, perhaps considering the use of Bonferroni dominance, and perhaps modify the fitness assignment as well. We also noticed non-convexity exists on the Pareto fronts generated by the algorithm. Hence, there is a need to consider stochastic dominance due to linear combination for elite solution selection in the algorithm. Overall, future research shall be intended to further improve the efficiency of the algorithm, especially under limited computational budget. Such research requires in-depth understanding of the tradeoff between system outcome estimate improvement for a candidate solution and Pareto front improvement along the genetic algorithm. If higher accuracy is desired in solution comparison and selection, more samples on each solution are required, which leads to relatively little computational power allocated to explore better Pareto fronts. On the other hands, if more computational power is devoted to search the entire feasible solution space, then each generational Pareto front may be obtained with higher uncertainty. One possible approach is to devote more computational power to searching elite solutions at early generations and then increase the accuracy of Pareto selection at later generations.

References

[1] W.-H. Feng, N. Kong, and H. Wan, "A simulation study of cadaveric liver allocation with a single-score patient prioritization formula," *Journal of simulation*, vol. 7, no. 2, pp. 109–125, 2013.

[2] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength of pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[3] E. J. Chen and L. H. Lee, "A multi-objective selection procedure of determining a pareto set," *Computers and Operations Research*, vol. 36, no. 6, pp. 1872–1879, 2009.

[4] O. Alagoz, A. J. Schaefer, and M. S. Roberts, "Optimizing organ allocation and acceptance," in *Handbook of Optimization in Medicine* (P. M. Padalos and H. E. Romeijn, eds.), pp. 1–21, New York, NY: Springer, 2009.

[5] I. David and U. Yechiali, "A time-dependent stopping problem with application to live organ transplants," *Operations Research*, vol. 33, no. 3, pp. 491–504, 1985.

[6] I. David and U. Yechiali, "Sequential assignment match processes with arrivals of candidates and offers," *Probability in the Engineering and Information Sciences*, vol. 4, no. 4, pp. 413–430, 1990.

[7] I. David and U. Yechiali, "One-attribute sequential assignment match processes in discrete time," *Operations Research*, vol. 43, no. 5, pp. 413–430, 1995.

[8] J. C. H. Jae-Hyeon Ahn, "Involving patients in the cadaveric kidney transplant allocation process: A decision-theoretic perspective," *Management Science*, vol. 42, no. 5, pp. 629–641, 1996.

[9] J. C. Hornberger and J. H. Ahn, "Deciding eligibility for transplantation when a donor kidney becomes available," *Medical Decision Making*, vol. 17, no. 2, pp. 160–170, 1997.

[10] D. H. Howard, "Why do transplant surgeons turn down organs?: A model of the accept/reject decision," *Journal of Health Economics*, vol. 21, no. 6, pp. 957–969, 2002.

[11] O. Alagoz, L. M. Maillart, A. J. Schaefer, and M. S. Roberts, "The optimal timing of living-donor liver transplantation," *Management Science*, vol. 50, no. 10, pp. 1420–1430, 2004.

[12] O. Alagoz, L. M. Maillart, A. J. Schaefer, and M. S. Roberts, "Determining the acceptance of cadaveric livers using an implicit model of the waiting list," *Oper. Res.*, vol. 55, pp. 24–36, Jan. 2007.

[13] B. Sandıkçı, L. M. Maillart, A. J. Schaefer, O. Alagoz, and M. S. Roberts, "Estimating the patient's price of privacy in liver transplantation," *Operations Research*, vol. 56, no. 6, pp. 1393–1410, 2008.

[14] S. A. Zenios, G. M. Chertow, and L. M. Wein, "Dynamic allocation of kidneys to candidates on the transplant waiting list," *Operations Research*, vol. 48, no. 4, pp. 549–569, 2000.

[15] X. Su and S. A. Zenios, "Patient choice in kidney allocation: A sequential stochastic assignment model," *Operations Research*, vol. 53, no. 3, pp. 433–455, 2005.

[16] N. Kong, A. J. Schaefer, B. Hunsaker, and M. S. Roberts, "Maximizing the efficiency of the u.s. liver allocation system through region design," *Management Science*, vol. 56, no. 12, pp. 2111–2122, 2010.

[17] A. A. B. Pritsker, D. L. Martin, J. S. Reust, M. A. Wagner, O. P. O. P. Daily, A. M. Harper, E. B. Edwards, L. E. Bennett, J. R. Wilson, M. E. Kuhl, J. P. Roberts, M. D. Allen, and J. F. Burdick, "Organ transplantation policy evaluation," in *WSC '95: Proceedings of the 27th conference on Winter simulation*, (Arlington, VA), pp. 1323–1341, December 1995.

[18] J. Ratcliffe, T. Young, M. Buxton, T. Eldabi, R. Paul, A. Burroughs, G. Papatheodoridis, and K. Rolles, "A simulation modeling approach to evaluating alternative policies for the management of the waiting list for liver transplantation," *Health Care Management Science*, vol. 4, no. 2, pp. 117–124, 2001.

[19] T. D., L. Waisanen, R. Wolfe, R. M. Merion, K. McCullough, and A. Rodgers, "Simulating the allocation of organs for transplantation," *Health Care Management Science*, vol. 7, no. 4, pp. 331–338, 2004.

[20] S. M. Shechter, C. L. Bryce, O. Alagoz, J. E. Kreke, J. E. Stahl, A. J. Schaefer, D. C. Angus, and M. S. Roberts, "A clinically based discrete-event simulation of end-stage liver disease and the organ allocation process," *Medical Decision Making*, vol. 25, no. 2, pp. 199–209, 2005.

[21] S. E. Taranto, A. M. Harper, E. B. Edwards, J. D. Rosendale, M. A. McBride, O. P. Daily, D. Murphy, B. Poos, J. Reust, and B. Schmeiser, "Developing a national allocation model for cadaveric kidneys," in *WSC '00: Proceedings of the 32nd Winter Simulation Conference*, (Orlando, FL), pp. 1971–1977, December 2000.

[22] W. B. van den Hout, J. M. A. Smits, M. C. Deng, M. Hummel, F. Schoendube, H. H. Scheld, G. G. Persijn, and G. Laufer, "The heart-allocation simulation model: A tool for comparison of transplantation allocation policies: A tool for comparison of transplantation allocation policies," *Transplantation*, vol. 76, no. 10, pp. 1492–1497, 2003.

[23] R. M. Anderson and R. M. May, *Multiobjective Programming and Planning.* New York, NY: Dover Publications, 2004.

[24] D. Nam and C. H. Park, "Multiobjective simulated annealing: A comparative study to evolutionary algorithms," *Journal of Simulation*, vol. 2, no. 2, pp. 87–97, 2002.

[25] P. Czyżżak and A. Jaszkiewicz, "Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization," *Journal of Multi-Criteria Decision Analysis*, vol. 7, no. 1, pp. 34–47, 1998.

[26] K. A. Suppapitnarm, K. A. Seffen, G. T. Parks, and P. J. Clarkson, "A simulated annealing algorithm for multiobjective optimization," *Engineering Optimization*, vol. 33, no. 1, pp. 59–85, 2000.

[27] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proceedings of the 1st International Conference on Genetic Algorithms*, (Hillsdale, NJ, USA), pp. 93–100, L. Erlbaum Associates Inc., 1985.

[28] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization," in *Genetic Algorithm: Proceedings of the 5th International Conference on Genetic Algorithms*, (San Mateo, CA), pp. 416–423, July 1993.

[29] J. Horn, N. Nafpliotis, and D. E. Goldberg, "A niched pareto genetic algorithm for multiobjective optimization," in *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, volume 1*, (Piscataway, NJ), June 1994.

[30] N. Srinivas and D. Kalyanmoy, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.

[31] C. A. Coello, "A comprehensive survey of evolutionary-based multiobjective optimization techniques," *Knowledge and Information Systems*, vol. 1, no. 3, pp. 129–156, 1999.

[32] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Reliability Engineering and System Safety*, vol. 91, no. 9, pp. 992–1007, 2006.

[33] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and Multidisciplinary Optimization*, vol. 26, no. 6, pp. 369–395, 2004.

[34] M. C. Fu, "Optimization via simulation: A review," *Annals of Operations Research*, vol. 53, no. 1, pp. 199–247, 1994.

[35] M. C. Fu, "Optimization for simulation: Theory and practice," *INFORMS Journal on Computing*, vol. 14, no. 3, pp. 192–215, 2002.

[36] K.-H. Chang, L. J. Hong, and H. Wan, "Stochastic trust region gradient-free method (strong): A new response-surface-based algorithm in simulation optimization," in *WSC '00: Proceedings of the 39th conference on Winter simulation*, (Washington, DC), pp. 346–354, December 2007.

[37] J. Swisher, S. Jacobson, and E. Yücesan, "Discrete-event simulation optimization using ranking, selection, and multiple comparison procedures: A survey," *ACM Transaction on Modeling and Computer Simulation*, vol. 13, no. 2, pp. 129–192, 2003.

[38] F. F. Baesler and J. A. Sep/'ulveda, "Simulation optimization: Multi-response simulation optimization using stochastic genetic search within a goal programming framework," in *WSC '00: Proceedings of the 32nd conference on Winter simulation*, (Washington, DC), pp. 788–794, December 2000.

[39] D. Morrice, J. Butler, and P. Mullarkey, "An approach to ranking and selection for multiple performance measures," in *WSC '98: Proceedings of the 30th conference on Winter simulation*, (Washington, DC), pp. 719–726, December 1998.

[40] R. Teleb and F. Azadivar, "A methodology for solving multi-objective simulation-optimization problems," *European Journal of Operational Research*, vol. 72, no. 1, pp. 135–145, 1994.

[41] J.-H. Ryu, S. Kim, and H. Wan, "Pareto front approximation with adaptive weighted sum method in mult-objective simulation optimization," in *WSC '09: Proceedings of the 41st conference on Winter simulation*, (Washington, DC), pp. 719–726, December 2009.

[42] D. Goldsman and B. L. Nelson, "Statistical screening, selection, and multiple comparison procedures in computer simulation," in *Proceedings of the 30th Conference on Winter Simulation*, WSC '98, (Los Alamitos, CA, USA), pp. 159–166, IEEE Computer Society Press, 1998.

[43] W. D. K. Averill M. Law, *Simulation Modeling and Analysis*. McGraw Hill Higher Education, 2000.

[44] A. Gaspar-Cunha, P. Oliveria, and J. A. Covas, "Use of genetic algorithms in multicriteria optimization to solve industry problems," in *Proceedings of the Seventh International Conference on Genetic Algorithms* (T. Bäck, ed.), 1997.

[45] D. E. Goldberg and P. Segrest, "Finite Markov chain analysis of genetic algorithms," in *Proceedings of the Second International Conference on Genetic Algorithms* (J. J. Grefenstette, ed.), Lawrence Erlbaum Associates, 1987.

[46] J. N. Morse, "Reducing the size of the nondominated set: Pruning by clustering," *Computers and Operations Research*, vol. 7, no. 1-2, pp. 55–66, 1980.

[47] D. E. Goldberg, B. Korb, and K. Deb, "Messy genetic algorithms: Motivation, analysis, and first results," *Complex Systems*, vol. 3, no. 5, pp. 493–530, 1989.

[48] F. Herrera, L. M., and J. L. Verdegay, "Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis," *Artificial Intelligence Review*, vol. 12, no. 4, pp. 265–319, 1998.

Appendices

A. Relative Dominance for Parameters Tuning

For more precise comparison among the Pareto solutions, we introduce *relative dominance*, and the derivation is as follows. Considering solutions in two set, $w_i \in C_1$ and $w_j \in C_2$ for $i = 1, \ldots, n_1$, and $j = 1, \ldots, n_2$. Then recall the dominance of w_i to w_j, denoted as p_{ij} in *equation* 4. We define a function D mapping an ordered pair (C_1, C_2) into a non-negative value as: $D(C_1, C_2) = \dfrac{\sum_i \sum_j p_{ij}}{n_1}$, where the value of $D(C_1, C_2)$ means the average number of solutions in the set C_2 dominated by each solution in the set C_1. For example, if $D(C_1, C_2)$ is close to 0, then the solutions in C_2 are barely dominated by those in C_1. If $D(C_1, C_2) > D(C_2, C_1)$, each solution in the set C_1 dominates more solutions than those in the set C_2, then C_1 is a better Pareto front than C_2. Then we get the *relative dominance* as $R(C_1, C_2) = \dfrac{D(C_1, C_2)}{D(C_2, C_1)}$. If $R(C_1, C_2) < 1$, the set C_1 is inferior to C_2 in terms of Pareto optimality; If $R(C_1, C_2) > 1$, the set C_1 is superior to C_2. If $R(C_1, C_2) = 1$, then each of the sets C_1, C_2 is no better than the other. The computed *relative dominance* for the configurations we have tested is presented in the tables below.

Table A1: Relative Dominance Between Settings of Parameters $(n_{\mathcal{E}}, n_{\mathcal{P}})$

	(5, 5)	(5, 10)	(5, 20)	(10, 5)	(10, 100)
(5, 5)	1.000	6.770	1.139	7.452	1145
(5, 10)	0.148	1.000	0.236	0.806	5.105
(5, 20)	0.878	4.234	1.000	4.393	29.26
(10, 5)	0.134	1.241	0.228	1.000	103.9
(10, 100)	0.001	0.196	0.034	0.010	1.000

Table A2: Relative Dominance Between Settings of Parameters (p^*, a)

	(0.60, 0.01)	(0.60, 0.05)	(0.75, 0.10)	(0.75, 0.20)	(0.90, 0.20)
(0.60, 0.01)	1.000	0.598	0.671	8.045	25.49
(0.60, 0.05)	1.672	1.000	1.152	9.993	25.18
(0.75, 0.10)	1.489	0.868	1.000	7.914	15.10
(0.75, 0.20)	0.124	0.100	0.126	1.000	1.382
(0.90, 0.20)	0.039	0.040	0.066	0.724	1.000

Table A3: Relative Dominance Between Settings of Parameters (p_c, p_m)

	(0.1, 0.3)	(0.1, 0.5)	(0.3, 0.1)	(0.3, 0.3)	(0.5, 0.1)
(0.1, 0.3)	1.000	1.546	45.93	1.310	6690
(0.1, 0.5)	0.647	1.000	2.897	0.458	4.345
(0.3, 0.1)	0.022	0.345	1.000	0.022	9.643
(0.3, 0.3)	0.763	2.183	45.29	1.000	2913
(0.5, 0.1)	0.000	0.230	0.104	0.000	1.000

Table A4: Relative Dominance Between Settings of Parameter b

	0.10	0.50	0.90	0.95	0.99
0.10	1.000	1.429	0.748	0.991	1.008
0.50	0.700	1.000	0.519	0.681	0.681
0.90	1.337	1.928	1.000	1.321	1.337
0.95	1.009	1.468	0.757	1.000	1.012
0.99	0.992	1.469	0.748	0.988	1.000