Multi-objective Simulation-Optimization Using Simulated Annealing With Incomplete Preference Information

Harri Mäkelin

June 2, 2013

Contents

1 Introduction

Multi-objective simulation-optimization (MOSO) problems typically involve highly complex systems, uncertainty, and computational intensity, and are of high practical relevance judging from the large number of applications that have surfaced thus far. The feature that distinguishes MOSO from traditional multi-objective optimization is that the objective functions must be evaluated using a stochastic simulation model of the system under consideration. In general, MOSO problems are challenging because 1) there is typically no unique optimal solution with respect to all objectives, 2) uncertainty about the true values of the objective functions is present, and 3) evaluation of the objective functions is tedious when the simulation model is complex. MOSO problems can be formulated for both continuous and discrete decision variables, and techniques for such problems are surveyed in (Evans, Stuckman, and Mollaghasemi 1991, Rosen, Harmonosky, and Traband 2008). In this paper, MOSO problems with only continuous decision variables are considered.

The solution to a MOSO problem is represented by a set of non-dominated solutions that approximates the true Pareto front of the problem. A solution is dominated if another solution is at least as good with respect to all objectives and strictly better with respect to at least one objective. Determining the non-dominated solution set allows the decision maker's (DM) preferences to be incorporated a posteriori by selecting only the solutions that satisfy these preferences. While a number of techniques for determining the non-dominated solutions to deterministic multi-objective problems that are based on evolutionary algorithms have been developed, they treat the values of the objective functions, which are estimated during optimization, as deterministic and do not explicitly take into account uncertainty. However, techniques that are specifically designed for simulation-optimization have also appeared recently (e.g., Goh and Tan 2007, Lee et al. 2008, Syberfeldt et al. 2010, Mattila, Virtanen, and Hämäläinen 2012).

In cases where the DM is only interested in solutions in a subset of the entire Pareto front, a common approach for incorporating the DM's preferences prior to optimization is to aggregate the individual objective functions into a single objective function, transforming the problem into a single-objective one that can be solved with single-objective simulation-optimization techniques. Among the benefits of this approach is that only solutions that are of interest to the DM are obtained. In addition, the computational requirements of the problem are expected to be more manageable. The most common approach is to use a weighted aggregation of objectives (e.g., Persson et al. 2006). Other forms of preference statements include specifying goals or targets for the objectives (Baesler and Sepulveda 2001) or, similarly, a reference point (Siegmund et al. 2012). However, the difficulty in most of the existing approaches where preferences are incorporated into the optimization method itself is that very detailed preference information is required from the DM. The DM may, for instance, have to provide exact weights for several objectives. Furthermore, in a group decision-making context exact weights might be very difficult to determine. In short, providing complete preference information is often non-trivial (Chen and Lee 2010) or even impossible.

This paper presents a new algorithm, Simulated Annealing with Incomplete Preference Information (SA-IPI), for MOSO with continuous decision variables. SA-IPI is based on a multi-objective simulated annealing (SA) algorithm (Mattila, Virtanen, and Hämäläinen 2012) and incorporates a multi-attribute utility (MAU) function for the aggregation of objective functions under incomplete information. The multi-objective SA algorithm that SA-IPI is based on was designed for determining non-dominated solutions to MOSO problems and was chosen as the search method because it was proven to be efficient compared with competing EA's in extensive tests (Mattila, Virtanen, and Hämäläinen 2012). The algorithm assumes incomplete information is expressed as a set of feasible weights for the MAU function that reflect the relative importance of the objectives. Candidate solutions are then compared via a pairwise dominance relation (Weber 1987) which states that a solution dominates another one if its utility — as determined by the MAU function — is higher across all feasible weights. Consequently, SA-IPI focuses the search on the non-dominated solutions that are of interest to the DM, thereby resulting in a subset of the entire non-dominated set that is consistent with the DM's preferences. The benefits of using SA-IPI are 1) computational efficiency compared with approaches that focus on the entire non-dominated set even though the objective is to find a smaller subset, and 2) relaxed requirements regarding information required from the DM compared with MOSO approaches that require complete preference information.

Multi-objective optimization techniques that consider incomplete preference information are mostly designed for deterministic settings; Branke (2008) presents a survey of such techniques. For example, Quan et al. (2007) have used a MAU function where the weights are described through a feasible set in combination with an EA for a deterministic problem. Within MOSO, on the other hand, incomplete preference information has only been used in the context of ranking and selection (R&S), where a limited computing budget is allocated to a finite set of solutions so that the best solution is identified with high confidence. Both Branke and Gamer (2007) and Frazier and Kazachkov (2011) determine probability distributions for the weights of the objectives and calculate the expected utility of each solution over the distributions. To the best of the authors' knowledge, the existing literature on MOSO has not considered incomplete preference information in a way described in this paper. The benefits of the presented approach are illustrated through a series of test problems.

The paper is structured as follows. In section 2, the original SA algorithm for stochastic multi-objective problems is outlined and incomplete information is incorporated to arrive at the SA-IPI algorithm. In section 3, the experimental setup along with the performance indicators and test problems which are used for the evaluation and validation of SA-IPI against the original SA algorithm are presented. The results of the experiments are presented in section 4. Conclusions from the results of the experiment are drawn in section 5.

2 SA Algorithm With Incomplete Preference Information (SA-IPI)

The SA algorithm attempts to find the approximate Pareto front, i.e., a set of non-dominated solutions for the problem

$$\underset{\mathbf{x}\in\Theta}{min}\,(f_1(\mathbf{x}),\ldots,f_H(\mathbf{x})), \tag{1}$$

where $f_i(\mathbf{x})$ are the objective functions, $\mathbf{x} = [x_1,\ldots,x_n]$ is the decision variable vector, and Θ is the set of feasible decision variable vectors. The values of the objective functions $f_i(\mathbf{x})$ are obtained via simulation and can only be estimated with noise. In this paper, i.i.d. normal noise terms are assumed so that $f_i(\mathbf{x}) + \omega_i$ where $\omega_i \sim N(0,\sigma_i^2)$ is observed. The decision variables are assumed to be constrained by simple linear upper and lower bounds, i.e., $l_i < x_i < u_i$ so that Θ is a hyperrectangle. The goal of optimization is to find a set of non-dominated solutions to problem (1) that satisfy the DM's preferences which are expressed as intervals for the weights of the objective functions in the form $w_i \in [w_{i,l}, w_{i,u}]$.

2.1 Steps of SA-IPI

In this section, the steps of SA-IPI are outlined and a description of the performance measure it uses to evaluate candidate solutions is provided. Other features of the algorithm that are left unchanged such as perturbation policy are left outside consideration, and the reader is asked to refer to the original description of the algorithm (Mattila, Virtanen, and Hämäläinen 2012). The steps of the algorithm are as follows:

0. Generate initial solution $\tilde{\mathbf{x}}$ by sampling each element $\tilde{x}_i$ from a uniform distribution over the feasible region. Generate M samples $f_i(\tilde{\mathbf{x}}) + \omega_i$ of the objective functions $f_i(\tilde{\mathbf{x}})$. Initialize the set of non-dominated solutions S with maximum size N and add the current solution to S so that $S = \{\tilde{\mathbf{x}}\}$.

1. Perturb current solution so that $\acute{\mathbf{x}} = \text{perturb}(\tilde{\mathbf{x}})$ and generate M samples $f_i(\acute{\mathbf{x}}) + \omega_i$ of each objective function.

2. Calculate the performance measure of the current solution $G(\tilde{\mathbf{x}}|S,\acute{\mathbf{x}})$ and the candidate solution $G(\acute{\mathbf{x}}|S,\tilde{\mathbf{x}})$. If $G(\acute{\mathbf{x}}|S,\tilde{\mathbf{x}}) < G(\tilde{\mathbf{x}}|S,\acute{\mathbf{x}})$, accept the candidate

solution as the new current solution, otherwise accept the candidate solution as the current solution with probability

$$p = e^{-\frac{G(\acute{\mathbf{x}}|S,\tilde{\mathbf{x}})-G(\tilde{\mathbf{x}}|S,\acute{\mathbf{x}})}{T}},$$

where T is the temperature of the SA algorithm.

3. If the size of S is less than N, add $\acute{\mathbf{x}}$ into S. Otherwise if $G(\acute{\mathbf{x}}|S,\tilde{\mathbf{x}}) < \underset{\mathbf{x}\in S}{max}\, G(\mathbf{x}|S\backslash\mathbf{x},\acute{\mathbf{x}})$, replace the $\mathbf{x}$ in S with the highest performance measure with $\acute{\mathbf{x}}$.

4. Go back to step 1 unless maximum number of iterations is reached. Return S as the solution set.

The performance measure for the candidate solution in the algorithm outlined above is defined as the sum of probabilities that the candidate solution is dominated by the members of the non-dominated set S and the current solution $\tilde{\mathbf{x}}$, or more formally,

$$G(\acute{\mathbf{x}}|S,\tilde{\mathbf{x}}) = \sum_{\mathbf{x}\in S\cup\tilde{\mathbf{x}}} P(\mathbf{x} \succ \acute{\mathbf{x}}), \qquad (2)$$

and similarly for the current solution

$$G(\tilde{\mathbf{x}}|S,\acute{\mathbf{x}}) = \sum_{\mathbf{x}\in S\cup\acute{\mathbf{x}}\backslash\tilde{\mathbf{x}}} P(\mathbf{x} \succ \tilde{\mathbf{x}}). \qquad (3)$$

Details on how to calculate the pairwise probabilities of dominance $P(\mathbf{x} \succ \acute{\mathbf{x}})$ and $P(\mathbf{x} \succ \tilde{\mathbf{x}})$ with noisy objective functions and under incomplete information are provided in the next section.

2.2 Incorporating incomplete information

Incomplete preference information is incorporated to the algorithm through an additive multi-attribute utility function (Keeney and Raiffa 1976, von Winterfeldt and Edwards 1986) of the form

$$U(\mathbf{x},\mathbf{w}) = \sum_{i=1}^{H} w_i u_i(f_i(\mathbf{x})), \qquad (4)$$

where $\mathbf{x}$ is the decision variable vector, $f_i(\mathbf{x})$ are the objective functions, u_i are the individual utility functions the DM has assigned for the objective functions, and w_i are the weights that reflect the DM's preferences for each individual objective. With noisy objective functions the values $f_i(\mathbf{x})$ cannot be observed directly and expected values must be obtained by sampling. Incomplete information is allowed by assuming the values of the weights are uniformly distributed within an interval so that $w_i \in [w_{i,l}, w_{i,u}]$ instead of requiring exact values to be specified.

Pairwise dominance relations between solutions are then determined over the feasible weights; a solution $\mathbf{x}$ pairwise dominates solution $\mathbf{y}$ if

$$U(\mathbf{x},\mathbf{w}) \geq U(\mathbf{y},\mathbf{w})\,\forall\,\mathbf{w} \in W, \tag{5}$$

where W is the set of feasible weights. At least one of the inequalities has to be strict. Since the inequality constraints for $\mathbf{w}$ are linear (i.e., W is convex), and the MAU function is linear with respect to w_i, it is sufficient to evaluate the above condition only in the extreme points of W. Consequently, the transformed problem

$$\underset{\mathbf{x}\in S}{max}\,(U(\mathbf{x},\mathbf{w}_1),\ldots,U(\mathbf{x},\mathbf{w}_K)) \tag{6}$$

is obtained, where S is the set of solutions under consideration and $\mathbf{w}_1,\ldots,\mathbf{w}_K$ are the extreme points in W. Because the utilities $U(\mathbf{x},\mathbf{w}_i)$ in each of the extreme points of W are linear combinations of the utilities of the same objective function values $u_i(f_i(\mathbf{x}))$, it follows that the aggregated utilities $U(\mathbf{x},\mathbf{w}_i)$ are multivariate normally distributed. The estimated means $\hat{U}_{\mathbf{x},i}$ and the elements $\hat{\sigma}_{\mathbf{x},ij}$ of the covariance matrix $\hat{\Sigma}_{\mathbf{x}}/M$ are obtained from

$$\hat{U}_{\mathbf{x},i} = \frac{1}{M}\sum_{n=1}^{M} U_n(\mathbf{x},\mathbf{w}_i) \tag{7}$$

and

$$\hat{\sigma}_{\mathbf{x},ij} = \frac{1}{M-1}\sum_{n=1}^{M}(U_n(\mathbf{x},\mathbf{w}_i)-\hat{U}_i)(U_n(\mathbf{x},\mathbf{w}_j)-\hat{U}_j), \tag{8}$$

where M is the number of samples, $U_n(\mathbf{x},\mathbf{w}_i)$ are the observed realizations of $U(\mathbf{x},\mathbf{w}_i)$, and $i,j \in \{1,\ldots,K\}$. The differences $U(\mathbf{y},\mathbf{w}_i) - U(\mathbf{x},\mathbf{w}_i)$ follow a multivariate normal distribution with mean $\hat{\boldsymbol{U}}_{\mathbf{y}} - \hat{\boldsymbol{U}}_{\mathbf{x}} = [\hat{U}_{\mathbf{y},1} - \hat{U}_{\mathbf{x},1},\ldots,\hat{U}_{\mathbf{y},K} - \hat{U}_{\mathbf{x},K}]$ and covariance matrix $\hat{\Sigma} = \frac{\hat{\Sigma}_{\mathbf{x}}}{M} + \frac{\hat{\Sigma}_{\mathbf{y}}}{M}$. The probability that solution $\mathbf{x}$ dominates solution $\mathbf{y}$, i.e., the probability that the differences are less than zero, is calculated from the cumulative multivariate normal distribution function:

$$P(\mathbf{x} \succ \mathbf{y}) = F\left(\mathbf{0}, \hat{\boldsymbol{\mu}}_y - \hat{\boldsymbol{\mu}}_x, \frac{\hat{\Sigma}_{\mathbf{x}}}{M} + \frac{\hat{\Sigma}_{\mathbf{y}}}{M}\right). \tag{9}$$

In the above equation the point **0** at which the cumulative distribution is evaluated is a zero vector. The performance measure for the candidate and current solutions are now obtained from equations (2) and (3).

In practice, evaluating the cumulative values of the multivariate normal distribution such as in equation (10) is computationally expensive especially if more objective functions are introduced and the number of dimensions grows large. As the method scales badly with the number of objective functions, calculating the exact probability would only be feasible in cases where the cost of simulation (i.e., evaluating the objective function) is high relative to the cost of evaluating the cumulative distribution values. Costly computation is avoided, however, if lower or upper bounds for the probability of dominance are used instead of the exact probabilities.

One upper bound for the pairwise probability of dominance across all feasible weights W is the minimum of the pairwise probabilities of dominance in the extreme points $\mathbf{w}_i$ of W, or more formally

$$P(\mathbf{x} \succ \mathbf{y}) = \frac{\hat{U}(\mathbf{x}, \mathbf{w}_i) - \hat{U}(\mathbf{y}, \mathbf{w}_i)}{\sqrt{\frac{s_x^2 + s_y^2}{M}}}, \tag{10}$$

where $\hat{U}$ and s_i^2 are the estimated means and variances respectively. A lower bound, on the other hand, is found via the Bonferroni inequality. If E_i is the event of dominance in extreme point $\mathbf{w}_i$, then the lower bound for the intersection of E_i and thus for the probability of dominance across all extreme points is:

$$P\left(\bigcup_{i=1}^{n}(1 - E_i)\right) \leq \sum_{i=1}^{n}(1 - P(E_i))$$

$$\Leftrightarrow$$

$$P\left(\bigcap_{i=1}^{n} E_i\right) \geq 1 - \sum_{i=1}^{n}(1 - P(E_i)). \tag{11}$$

The decision then comes down to whether to use the upper or lower bound for calculations. In this experiment the upper bound is chosen as the estimator for the pairwise probability of dominance because the lower bound — being obtained through the Bonferroni inequality — might be too conservative for estimation purposes.

3 Experimental Setup

The performance of SA-IPI is validated by comparing it against the original SA algorithm (Mattila, Virtanen, and Hämäläinen 2012) over 16 test problems in 3 different test suites. This section describes the test problems and performance indicators that are used for evaluating the algorithms.

3.1 Test problems

The test cases that are chosen for validating SA-IPI include problems ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 from the ZDT test suite by (Zitzler, Deb, and Thiele 2000); problems DTLZ1 to DTLZ6 from the DTLZ test suite (Deb et al. 2002); and five problems as suggested by (van Veldhuizen 1999) which are renamed as MOP1, MOP2, MOP3, MOP4, and MOP6, and collectively as the MOP suite. The test problems cover a wide range of scenarios, such as non-uniform distributions of non-dominated solutions, local optima, and disconnected non-dominated sets, and as such are well suited for testing the capability of a multi-objective algorithm to handle difficulties.

Each test problem is of the same form (1) as presented in section 2. The values of the objective functions $f_i(\mathbf{x})$ are assumed to be affected by i.i.d. normal noise so that $f_i(\mathbf{x}) + \omega_i$ where $\omega_i \sim N(0, \sigma_i^2)$ is observed. In all test cases the noise level is set to 20% of the maximum absolute values of the objective functions in the actual non-dominated set. The number of objective functions is $H = 2$ for all problems in the ZDT and MOP suites and $H = 3$ for the DTLZ suite. The objective functions $f_1, \ldots, f_H$ as well as the number of decision variables and their lower bounds are presented for each problem in Table **??**. For the actual non-dominated sets of the test problems the reader may refer to the papers in which each test suite appears (van Veldhuizen 1999; Zitzler, Deb, and Thiele 2000; Deb et al. 2002).

The test problems are divided into three types: 1) convex problems (ZDT1, ZDT4, MOP1, MOP3), 2) concave problems (ZDT2, ZDT6, MOP2, DTLZ2, DTLZ3, DTLZ4), and 3) problems that are neither (ZDT3, MOP4, MOP6, DTLZ1, DTLZ5, DTLZ6). The differentiation between convex and concave problems is relevant in our experiment because the actual non-dominated set that is consistent with the DM's preferences degenerates into a single point in concave problems. This is due to the nature of the problem as well as due to the weights used in the MAU function, and while it does not cause problems with optimization, it makes the use of certain performance indicators such as hypervolume ratio non-informative.

3.2 Performance indicators for the algorithms

In the experiment, two performance indicators that measure the distance of solutions to the actual non-dominated set as well as the diversity and distribution of the solutions are used to measure the quality of obtained non-dominated solutions. The values of the performance indicators are calculated from the noiseless objective function values to ensure consistency of results. Also, only the solutions that are pairwise non-dominated will be extracted for calculation of performance indicators from the non-dominated solution sets that are maintained by the algorithms since the other solutions are not relevant and it would only seem reasonable they should not affect the assessment of the quality of the algorithm's output.

The first performance indicator to be used is *generational distance*, denoted with I_{GD}, which measures the distance of the non-dominated solutions to the actual non-dominated set (van Veldhuizen 1999). Generational distance is defined as

$$I_{GD} = \frac{\left(\sum_{\mathbf{x}\in S} d_{\mathbf{x}}^2\right)^{1/2}}{|S|}, \tag{12}$$

where $d_{\mathbf{x}}$ is the minimum Euclidean distance of a non-dominated solution $\mathbf{x} \in S$ to the actual non-dominated set. Smaller values indicate better performance; a zero value would indicate that all the non-dominated solutions generated by the algorithm are part of the actual non-dominated set.

The second performance indicator is *hypervolume ratio*, denoted with I_{HV}, which measures the general quality of the non-dominated solutions (van Veldhuizen 1999). Hypervolume ratio is the ratio of the volume in the objective space dominated by the generated set of non-dominated solutions S and the volume dominated by the actual non-dominated set S_T. For the calculation of the indicator, two reference points are chosen that limit the part of the objective function space under consideration and guarantee that the volumes remain finite. The reference points are determined from the boundary values of the objective functions in the actual non-dominated set, i.e., $(\underline{f}_1^T, \ldots, \underline{f}_H^T)$ and $(\overline{f}_1^T, \ldots, \overline{f}_H^T)$ where the under- and overlines represent minimum and maximum values respectively. The Cartesian product $Z = [\underline{f}_1^T, \overline{f}_1^T] \times \ldots \times [\underline{f}_H^T, \overline{f}_H^T]$ is used to denote the hyperrectangle that is constrained by these points and contains S_T. Formally, hypervolume ratio is then expressed as

$$I_{HV} = \frac{\int_Z \mathbf{1}(\exists \mathbf{y} \in S, \mathbf{y} \succ \mathbf{z}) d\mathbf{z}}{\int_Z \mathbf{1}(\exists \mathbf{y} \in S_T, \mathbf{y} \succ \mathbf{z}) d\mathbf{z}}, \tag{13}$$

where $\mathbf{1}(\cdot)$ is an indicator function. The numerator represents the volume dominated by S and the denominator the volume dominated by S_T. Higher values indicate better performance; a value of 1 would mean all the non-dominated solutions generated by the algorithm are part of the actual non-dominated set. Because the value of the integral in Equation (13) cannot be calculated analytically, it is approximated in the experiment with an evenly spaced grid of sample points. A dense grid, i.e., a large sample size is used to obtain an accurate approximation.

3.3 Reference method

The original SA algorithm (Mattila, Virtanen, and Hämäläinen 2012) is used as a reference method for SA-IPI. Because the original algorithm does not use any preference information to direct its search and instead attempts to find an approximation for the entire Pareto front, the solution set that is returned is spread over a wider area than that of SA-IPI, thereby making direct comparison difficult. Therefore, only the solutions that are pairwise non-dominated under the assumed preference information are selected from the solution sets of both algorithms for comparison. Although this might tilt the scales slightly in favor of SA-IPI which will have more solutions concentrated in the area of the partial actual non-dominated set that is under consideration, it will make comparisons more consistent. Finally, performance indicators are computed for both algorithms with respect to the partial non-dominated set.

3.4 Preference information

Essentially, there are two ways preference information can be included in the experiment. The first one is through specifying the intervals for the weights of the MAU function, and the second one is through the specification of the univariate utility functions in the MAU function. Because the scale on which the values of an objective function are expressed is also an implicit preference statement and thus affects optimization results, the univariate objective functions are all scaled so that $u_i(f_i(\mathbf{x})) \in [0,1]$. For the scaling parameter several alternatives present themselves. The maximum value of the objective function in the space of feasible decision vectors Θ is one such alternative, as is dynamic scaling by the maximum values of the objective functions in the solution set maintained by the algorithm. However, these alternatives should cause problems in convergence when one or more of the objective functions are unbounded since some of the initial solutions might be quite far from the actual non-dominated set in some of the objectives.

Therefore, the univariate utility functions are scaled by a multiple of the maximum value of the corresponding objective function in the actual non-dominated set, i.e., so that $u_i(f_i(\mathbf{x})) = -\frac{f_i(\mathbf{x})}{n\bar{f}_i^T}$ where $n \in \mathbb{Z}_+$ is sufficiently large. In the experiment the MAU function is defined as

$$U(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^{H} -\frac{w_i f_i(\mathbf{x})}{n\bar{f}_i^T}. \tag{14}$$

where $n = 100$.

The problems are solved with three different preference weight sets. The weight sets used for the two-objective test problems (ZDT and MOP suites) are chosen so that 1) one objective is defined as the primary objective, 2) tighter bounds are chosen for that objective to express how much more important it is, and 3) a narrow interval is given for that objective indicating high degree of confidence on its approximate value. More specifically, the following weight sets are chosen: $w_1 \in [0.5, 1.0]$, $w_1 \in [0.6, 0.9]$, and $w_1 \in [0.7, 0.8]$ with $w_2 = 1 - w_1$ in all cases. For the three objective problems, the following weight sets were chosen in an analogous fashion: $w_1 \in [0.5, 1.0]$ and $w_2 \in [0.25, 0.5]$, $w_1 \in [0.6, 0.9]$ and $w_2 \in [0.2, 0.4]$, and $w_1 \in [0.7, 0.8]$ and $w_2 \in [0.15, 0.3]$, with $w_3 = 1 - w_1 - w_2$ in all cases.

4 Results

In this section, the performance of SA-IPI and the original SA algorithm (Mattila, Virtanen, and Hämäläinen 2012) are compared over the three test problem suites described in the previous section (ZDT, MOP, and DTLZ) with two different performance indicators, (12) and (13). The results are discussed separately for the three problem types: 1) convex, 2) concave, and 3) other type of problems.

The following parameters for both the original SA algorithm and SA-IPI are used in the experiment. The maximum perturbations $\delta_1, \ldots, \delta_n$ are $\delta_i = 1$ for the ZDT test suite, and $1/5$ of the range of each decision variable for the DTLZ and MOP suites. As for the number of objective function samples, $m = 20$ is used for the ZDT and DTLZ suites and $m = 5$ for the MOP suite. Otherwise, $T = 1$, $N = 100$ and $M = 30$ are used in all cases. Fixed computing budgets are used so that the total number of objective function samples amount to 50000 for both algorithms. In all test problems the noise level is set to 20% of the maximum absolute values of the objective functions in the actual non-dominated set. In brief, all parameter values are defined as in (Mattila, Virtanen, and Hämäläinen 2012) excluding the weight sets and univariate utility functions which are specified in section 3.4.

4.1 Convex problems

The convex problem set consists of the four problems ZDT1, ZDT4, MOP1, and MOP3. These problems are the most straightforward or ”easy” in the sense that the entire actual non-dominated set and partial actual non-dominated set that is consistent with the DM’s preferences are both smooth and continuous. Consequently, generational distance and hypervolume ratio can both be used to assess the quality of solutions for all four test problems. The results of the tests are summarized in Table 1.

4.2 Concave problems

The concave problem set consists of the problems ZDT2, ZDT6, MOP2, DTLZ2, DTLZ3, and DTLZ4. The concavity of the actual non-dominated set means the partial non-dominated set that is consistent with the DM’s preferences degenerates into a single point. Therefore, only generational distance should be taken into account in assessing the performance of the algorithm in concave problems. The results of the tests are summarized in Table 2.

Table 1: The table depicts the medians of the performance indicators that are calculated from the pairwise non-dominated solutions. A greyed out value indicates that the algorithm is statistically significantly better, whereas a bold value indicates it is worse at a 5% confidence level.

	Preference weights and performance indicators					
Problem and	$w_1 \in [0.5, 1.0]$		$w_1 \in [0.6, 0.9]$		$w_1 \in [0.7, 0.8]$	
algorithm	I_{GD}	I_{HV}	I_{GD}	I_{HV}	I_{GD}	I_{HV}
ZDT1						
SA-IPI	0.032	0.689	0.014	0.436	0.013	0.000
SA	0.033	0.738	0.020	0.454	0.030	0.000
ZDT4						
SA-IPI	0.277	0.133	0.507	0.000	0.516	0.000
SA	0.523	0.021	0.365	0.000	0.167	0.000
MOP1						
SA-IPI	0.005	0.788	0.008	0.702	0.047	0.327
SA	0.003	0.853	0.005	0.690	0.014	0.439
MOP3						
SA-IPI	0.015	0.896	0.006	1.057	0.013	NaN
SA	0.067	0.889	0.013	**0.873**	**0.034**	NaN

4.3 Other problems

The final problem set consists of problems where the actual non-dominated set is either discontinuous and/or might locally be either convex or concave. Consequently, the values or the hypervolume ratio might not be indicative of algorithm performance in some problems. The results of the tests are summarized in Table 3.

4.4 Lessons

Based on the results in Tables 1-3, the median values of the performance indicators are not better for SA-IPI than for the reference algorithm at a 5% confidence level. It would seem that the test problems are not suitable for evaluating the performance of the algorithm. For example, in test problem ZDT1, the first objective function is solely determined by the first decision variable; however, the value of the second objective function depends on all of the remaining 29 decision variables. Therefore, preference information can only be used in guiding the selection of the first decision variable, whereas it offers no additional benefit when it comes to selecting the values of the remaining 29 variables. To demonstrate the benefits of the SA-IPI algorithm, a different set of test problems should be used where the above problem does not exist.

Table 2: The table depicts the medians of the performance indicators that are calculated from the pairwise non-dominated solutions. A greyed out value indicates that the algorithm is statistically significantly better, whereas a bold value indicates it is worse at a 5% confidence level.

	Preference weights and performance indicators					
Problem and	$w_1 \in [0.5, 1.0]$		$w_1 \in [0.6, 0.9]$		$w_1 \in [0.7, 0.8]$	
algorithm	I_{GD}	I_{HV}	I_{GD}	I_{HV}	I_{GD}	I_{HV}
ZDT2						
SA-IPI	0.108	0.000	0.094	0.000	0.083	0.000
SA	0.097	0.000	0.069	0.000	0.078	0.000
ZDT6						
SA-IPI	0.350	0.000	0.505	0.000	0.652	0.000
SA	0.334	0.000	0.377	0.000	0.531	0.000
MOP2						
SA-IPI	0.705	NaN	0.991	NaN	0.992	NaN
SA	0.538	NaN	0.981	NaN	0.994	NaN
DTLZ2						
SA-IPI	0.050	0.000	0.064	0.000	0.111	0.000
SA	**0.181**	0.000	0.264	0.000	0.184	0.000
DTLZ3						
SA-IPI	6.394	0.000	5.404	0.000	4.325	0.000
SA	4.489	0.000	3.346	0.000	3.398	0.000
DTLZ4						
SA-IPI	0.055	0.000	0.075	0.000	0.079	0.000
SA	**0.231**	0.000	0.199	0.000	0.068	0.000

Table 3: The table depicts the medians of the performance indicators that are calculated from the pairwise non-dominated solutions. A greyed out value indicates that the algorithm is statistically significantly better, whereas a bold value indicates it is worse at a 5% confidence level.

	Preference weights and performance indicators					
Problem and	$w_1 \in [0.5, 1.0]$		$w_1 \in [0.6, 0.9]$		$w_1 \in [0.7, 0.8]$	
algorithm	I_{GD}	I_{HV}	I_{GD}	I_{HV}	I_{GD}	I_{HV}
ZDT3						
SA-IPI	0.069	0.691	0.053	0.326	0.059	0.000
SA	0.060	0.729	0.051	0.598	0.075	0.000
MOP4						
SA-IPI	0.971	0.000	0.952	0.000	1.218	0.000
SA	0.037	**0.902**	0.249	0.000	0.621	0.000
MOP6						
SA-IPI	0.000	0.925	0.000	0.000	0.000	0.000
SA	0.000	**0.976**	0.000	0.000	**0.000**	0.000
DTLZ1						
SA-IPI	0.769	0.000	0.681	0.000	0.526	0.000
SA	0.368	0.000	0.156	0.000	0.234	0.000
DTLZ5						
SA-IPI	0.102	0.000	0.099	0.000	0.154	0.000
SA	0.022	0.000	0.020	0.000	0.030	0.000
DTLZ6						
SA-IPI	0.317	0.000	0.745	0.000	1.246	0.000
SA	0.046	0.000	0.057	0.000	0.070	0.000

5 Conclusions

This paper presented a new algorithm, SA-IPI, for MOSO problems with incomplete preference information. SA-IPI is implemented on a state-of-the-art SA algorithm for multi-objective noisy problems (Mattila, Virtanen, and Hämäläinen 2012). As a novel feature, SA-IPI incorporates incomplete preference information by requiring intervals for the weights of an an additive MAU function to be specified. The algorithm then ranks candidate solutions based on the probability of pairwise dominance across all feasible weights.

The algorithm was validated on a total of 16 different test problems in three different test suites with 3 different weight sets. Two performance indicators — generational distance and hypervolume ratio — were used in evaluating the performance of SA-IPI compared to the original SA algorithm that does not utilize preference information. SA-IPI was found not to perform better in the tests than the original algorithm in terms of both generational distance and hypervolume ratio when the partial actual non-dominated solution set that was consistent with given preferences was used as the reference. This was reasoned to be due to the nature of the test problems rather than due to a problem with the method itself — a different set of test problems would be required to bring out the benefits of the algorithm. The algorithm therefore still needs further validation.

As the method used for incorporating incomplete preference information to the algorithm is, in fact, an adapted multi-criteria decision making method and thus not specific to simulated annealing, it could very well be extended to be used in other alternative techniques such as evolutionary algorithms. Considering further lines of development, only continuous decision variables were considered in this experiment even though discrete decision variables are very common in real world applications.

References

Baesler, F.F., and J.A. Sepulveda. 2001. “Reference Point-Based Evolutionary MultiObjective Optimization for Industrial Systems Simulation.” Edited by B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, *Proceedings of the 2001 Winter Simulation Conference*.

Branke, J¸rgen. 2008. “Consideration of Partial User Preferences in Evolutionary Multiobjective Optimization.” In *Multiobjective Optimization*, edited by J¸rgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Roman Slowinski, Volume 5252 of *Lecture Notes in Computer Science*, 157–178. Berlin: Springer.

Branke, J¸rgen, and Jochen Gamer. 2007. “Efficient Sampling in Multi-Criteria Selection.” *Proceedings of the 2007 INFORMS Simulation Society Workshop*. INFORMS, 42–46. [Online]. Available: `http://www.informs-sim.org/2007informs-csworkshop/09.pdf`.

Chen, Chun-Hung, and Loo Hay Lee. 2010. *Stochastic simulation optimization - an optimal computing budget allocation*. Volume 1 of *System Engineering and Operations Research*. Singapore: World Scientific.

Deb, Kalyanmoy, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. 2002. “Scalable Multi-Objective Optimizatin Test Problems.” *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*. Honolulu, Hawaii, 825–830.

Evans, Gerald W., Bruce Stuckman, and Mansooreh Mollaghasemi. 1991. “Multicriteria Optimization of Simulation Models.” *Proceedings of the 1991 Winter Simulation Conference*. Piscataway, New Jersey: IEEE, 894–900.

Frazier, Peter I., and Aleksandr M. Kazachkov. 2011. “Guessing Preferences: A New Approach to Multi-Attribute Ranking and Selection.” *Proceedings of the 2011 Winter Simulation Conference*. Piscataway, NJ: Institute of Electrical and Electronics Engineers Inc., 4324–4336.

Goh, C. K., and K. C. Tan. 2007. “An Investigation on Noisy Environments in Evolutionary Multiobjective Optimization.” *IEEE Transactions on Evolutionary Computation* 11 (3): 354–381.

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

Lee, Loo Hay, Ek Peng Chew, Suyan Teng, and Yankai Chen. 2008. “Multi-objective simulation-based evolutionary algorithm for an aircraft spare parts allocation problem." *European Journal of Operational Research* 189 (2): 476–491 (Sep.).

Mattila, Ville, Kai Virtanen, and Raimo P. Hämäläinen. 2012. "A Simulated Annealing Algorithm for Noisy Multiobjective Optimization." *Journal of Multi-Criteria Decision Analysis*.

Persson, Anna, Henrik Grimm, Amos Ng, Thomas Lezama, Jonas Ekberg, Stephan Falk, and Peter Stablum. 2006. "Simulation-based multiple-objective optimization of a real-world scheduling problem." *Proceedings of the 2006 Winter Simulation Conference*. Piscataway, NJ: IEEE, 1757–1764.

Quan, Gang, Garrison W. Greenwood, Donglin Liu, and Sharon Hu. 2007. "Searching for Multiobjectice Preventive Maintenance Schedules: Combining Preferences with Evolutionary Algorithms." *European Journal of Operational Research* 177 (3): 1969–1984.

Rosen, Scott L., Cathrine M. Harmonosky, and Mark T. Traband. 2008. "Optimization of Systems with Multiple Performance Measures via Simulation: Survey and Recommendations." *Computers & Industrial Engineering* 54 (2): 327–339.

Siegmund, Florian, Jacob Bernedixen, Leif Persson, Amos H. C. Ng, and Kalyanmoy Deb. 2012. "Reference Point-Based Evolutionary MultiObjective Optimization for Industrial Systems Simulation." Edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, *Proceedings of the 2012 Winter Simulation Conference*. Institute of Electrical and Electronics Engineers, Inc., Piscataway, New Jersey, 130:1–130:11.

Syberfeldt, Anna, Amos Ng, Robert I. John, and Philip Moore. 2010. "Evolutionary optimisation of noisy multi-objective problems using confidence-based dynamic resampling." *European Journal of Operational Research* 204 (3): 533–544.

van Veldhuizen, David A. 1999, June. "Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations." Ph.D. diss., Graduate School of Engineering, Air Force Institute of Technology, Dayton, OH.

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

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

Zitzler, Eckart, Kalyanmoy Deb, and Lothar Thiele. 2000. “Comparison of multiobjective evolutionary algorithms: Empirical results.” *Evolutionary Computation* 8 (2): 173–195 (Jun.).