A Simple Indicator Based Evolutionary Algorithm for Set-Based Minmax Robustness

Yue Zhou-Kangas(✉) and Kaisa Miettinen

University of Jyvaskyla, Faculty of Information Technology,
P.O. BOX(35), Agora, 40014 University of Jyvaskyla, Finland
{yue.y.zhou-kangas,kaisa.miettinen}@jyu.fi

Abstract. For multiobjective optimization problems with uncertain parameters in the objective functions, different variants of minmax robustness concepts have been defined in the literature. The idea of minmax robustness is to optimize in the worst case such that the solutions have the best objective function values even when the worst case happens. However, the computation of the minmax robust Pareto optimal solutions remains challenging. This paper proposes a simple indicator based evolutionary algorithm for robustness (SIBEA-R) to address this challenge by computing a set of non-dominated set-based minmax robust solutions. In SIBEA-R, we consider the set of objective function values in the worst case of each solution. We propose a set-based non-dominated sorting to compare the objective function values using the definition of lower set less order for set-based dominance. We illustrate the usage of SIBEA-R with two example problems. In addition, utilization of the computed set of solutions with SIBEA-R for decision making is also demonstrated. The SIBEA-R method shows significant promise for finding non-dominated set-based minmax robust solutions.

Keywords: Minmax robust Pareto optimal solutions · Hypervolume
Set-based dominance · SIBEA · Uncertainty

1 Introduction and Background

The need to simultaneously consider multiple objectives and the existence of uncertainty from various sources complicate real-world optimization problems. Uncertainty due to for example imprecise data or uncertain future developments usually reflects as parameters in the objective functions. Traditional multiobjective optimization methods concentrate on optimizing multiple objectives simultaneously and finding a set of Pareto optimal or non-dominated solutions for deterministic formulations of problems. Different approaches can be used to find this set, for example with scalarization techniques (see e.g., [21]) or with evolutionary multiobjective optimization methods (see e.g., [8]). However, the involved uncertainty can affect deterministic Pareto optimal or non-dominated solutions

A. Auger et al. (Eds.): PPSN 2018, LNCS 11101, pp. 286–297, 2018.
https://doi.org/10.1007/978-3-319-99253-2_23

with undesired degradation in their objective function values. Thus, considering uncertainty in the optimization process is as important as optimizing multiple objectives simultaneously.

The goal of handling uncertainty and multiple objectives simultaneously is finding robust solutions that are sufficiently immune to the uncertainty and with trade-offs among the objectives. Different concepts of robustness and measures of robustness have been proposed in the literature. Typically, robustness measures are incorporated into evolutionary multiobjective optimization methods to quantify the effects of uncertainty on the objective function values (e.g., [4,9,12,17]). Different robustness concepts alter the definition of dominance. Based on the concepts, uncertain multiobjective optimization problems can be transformed to deterministic ones (as summarized in [14,25]). In addition, different possible values of uncertain parameters can be considered simultaneously during the optimization process (as e.g., in [22,24]).

Among the robustness concepts, the most widely used ones belong to the family of minmax robustness (e.g., [5,11,16]). Due to different possible values of the uncertain parameters, a solution in the decision space can correspond to a set of outcomes (i.e., objective function values). We refer to a set of outcomes corresponding to a solution as the outcome set of the solution. Minmax robustness compares the worst outcomes in the outcome sets and finds the best possible ones. The worst outcomes are referred to as the worst case outcome set.

Set-based minmax robustness [11] finds the solutions with the best worst case outcome sets by utilizing set-based dominance [23]. For feasible solutions considered, we need to identify their worst case outcome sets by maximizing the multiple objectives simultaneously in their outcome sets and compare them with set-based dominance. This series of tasks makes the computation of set-based minmax robust solutions challenging. Methods from robust optimization and mathematical optimization can only address the challenge partially.

Some solution methods via scalarizing and reformulating the scalarized subproblems have been proposed e.g., in [5,16]. However, typically the reformulations are based on some (strict) assumptions on the characteristics of the problem which cannot be always guaranteed in practical problems. If no assumptions on the characteristics can be made, using samples to replace the uncertainty set has been explored e.g., in [27]. The shortcoming is that the resulting solutions might not be or near to minmax robust. The needs of obtaining a more accurately approximated set of set-based minmax robust solutions have motivated us for further developments.

Different types of evolutionary multiobjective optimization methods have been able to approximate solutions for many challenging problems. For comparing worst case outcome sets, methods which combine non-dominated sorting and crowding distance are not suitable since defining the crowding distance between the worst case outcome sets is not possible. Decomposition based methods cannot be directly applied since we cannot directly associate worst case outcome sets to the weighting vectors. Set-based dominance has been utilized in the evolutionary multiobjective optimization community e.g., in [3,30]. The population is treated a whole set and set-based dominance is used to improve the population. Very recently, using set-based dominance to solve problems involving uncertainty has also attracted interest. In [15], a genetic algorithm has been proposed for solving combinatorial bi-objective optimization problems with a set of discrete values of the uncertain parameters. In [13], an evolutionary algorithm has been proposed for solving problems with interval uncertainty (i.e., the uncertain parameters stem from some intervals) with reformulated objective functions. A specific definition of set-based dominance has been used to compare the worst case outcomes in [2]. These earlier research demonstrates potential to address the challenges.

In this paper, we propose utilizing an evolutionary multiobjective optimization approach SIBEA-R to tackle the challenge of approximating set-based minmax robust Pareto optimal solutions. We extend SIBEA [28] for this purpose. We incorporate the definition of set-based minmax robustness into the SIBEA method and develop a non-dominated sorting procedure based on the lower set less order. We also utilize the hypervolume of the worst case outcome sets in the environmental selection process.

The rest of the paper is organized as follows: Sect. 2 presents some concepts we use in this paper. Section 3 presents SIBEA-R followed by some numerical examples of how it can be used in Sect. 4. Finally, Sect. 5 concludes the paper and identifies some future research directions.

2 Preliminaries

In this paper, we consider multiobjective optimization problems with uncertainty reflected in the parameters of the objective functions in the following form:

$$\left(\begin{array}{ll} \text{minimize} & (f(x,\xi) = f_1(x,\xi), \cdots, f_k(x,\xi))^T \\ \text{subject to} & x \in \mathfrak{X} \end{array} \right)_{\xi \in \mathcal{U}}, \tag{1}$$

where $x = (x_1, \cdots, x_n)^T$ is the decision vector from the feasible set $\mathfrak{X}$ in the decision space $\mathbb{R}^n$ whose components are called decision variables and ξ consists of the uncertain parameters which are assumed to stem from an uncertainty set $\mathcal{U}$. With ξ stemming from $\mathcal{U}$, a solution $x \in \mathfrak{X}$ is mapped in the objective space as a set-valued map [23] under the objective functions $f_1, \cdots, f_k$ to the objective space. We call this set-valued map the outcome set and denote by $f_{\mathcal{U}}(x) = \{f(x,\xi), \xi \in \mathcal{U}\}$. In the outcome set, a specific objective vector $f(x,\xi)$ is called an outcome.

The set-based minmax robust counterpart of (1) is presented in [11] as:

$$\underset{x \in \mathfrak{X}}{\text{minimize}} \ \underset{\xi \in \mathcal{U}}{\text{maximize}} \ f(x,\xi) = (f_1(x,\xi), \cdots, f_k(x,\xi))^T. \tag{2}$$

We say that a solution $x^* \in \mathfrak{X}$ is set-based minmax robust Pareto optimal for problem (1), if there does not exist another solution $x \in \mathfrak{X}$ such that $f_{\mathcal{U}}(x) \subseteq f_{\mathcal{U}}(x^*) - \mathbb{R}^k_{\geq}$, where $\mathbb{R}^k_{\geq} = \{x \in \mathbb{R}^k : x_i \geq 0, i = 1, \cdots, k\}$ [11]. This definition is based on the concept of lower set less order: let A and B be arbitrary closed sets, then $A \preceq^l B$ implies $A \subseteq B - \mathbb{R}^k_{\geq}$. Thus, when we compare two sets of vectors, we say $A \preceq^l B$ if for all $a \in A$ there exists $b \in B$ such that $a_i \leq b_i, i = 1, \cdots, k$.

Figure 1 illustrates an example of set-based minmax robustness with two objective functions to be minimized. In the example, we have a feasible set $\mathfrak{X} = \{x^1, x^2, x^3, x^4\}$ and an arbitrary uncertainty set $\mathcal{U}$. We plot the outcome set of the three solutions in the figure $f_{\mathcal{U}}(x^1)$ (bold solid curve), $f_{\mathcal{U}}(x^2)$ (bold dotted line), $f_{\mathcal{U}}(x^3)$ (bold dashed line), and $f_{\mathcal{U}}(x^4)$ (bold dash-dotted line). The gray thin lines help us to identify the borders of the outcome sets. Solution x^1 is a set-based minmax robust Pareto optimal solution, since $f_{\mathcal{U}}(x^1) - \mathbb{R}^2_{\geq}$ does not contain $f_{\mathcal{U}}(x^2)$ nor $f_{\mathcal{U}}(x^3)$. Similarly, we can see that x^2 and x^3 are also set-based minmax robust Pareto optimal solutions. However, x^4 is not set-based minmax robust Pareto optimal since $f_{\mathcal{U}}(x^4) - \mathbb{R}^2_{\geq}$ contains $f_{\mathcal{U}}(x^1)$ and $f_{\mathcal{U}}(x^3)$. The formulation (2) minimizes the worst case outcomes. As mentioned before, we need to first find the worst case outcomes and compare them as a whole. So, finding set-based minmax robust Pareto optimal solutions requires us to address these two challenges in a systematical way. Finding the worst case outcome set of a fixed solution $x \in \mathfrak{X}$ requires solving a multiobjective optimization problem with the objective functions to be maximized as follows:

$$\underset{\xi \in \mathcal{U}}{\text{maximize}} \quad (f_1(x,\xi), \cdots, f_k(x,\xi))^T. \tag{3}$$

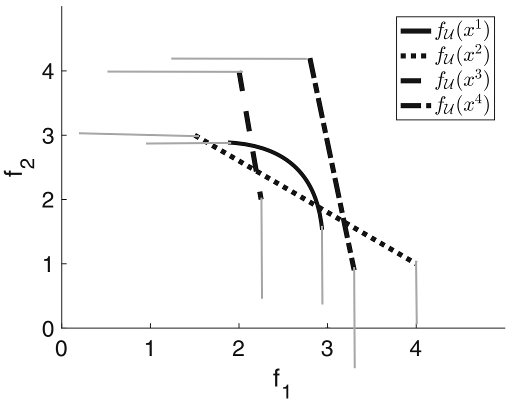


Fig. 1. Example of set-based minmax robustness

3 The SIBEA-R Method

In this section, we introduce SIBEA-R for approximating set-based minmax robust Pareto optimal solutions. We first introduce the steps of SIBEA-R. Then, we discuss details of the steps with a concentration on the further developments on SIBEA for set-based minmax robustness.

The SIBEA-R method takes the population size (NP) and the number of generations (NG) as the input and produces a set of non-dominated set-based minmax robust solutions A as the output. The basic steps are as follows:

Step 1. (Initialization) Generate an initial set of decision vectors P of size NP and find their worst case outcome sets by solving (3). Set the generation counter $m = 1$.
Step 2. (Mating) Create an offspring population Q using crossover and mutation operators and find their worst case outcome sets. Set $P = P \cup Q$.
Step 3. (Environmental selection) Rank the population P using lower set less order and sort the individuals into different fronts $F^i, i = 1, 2, \cdots$. and do the following:

- Set a new population $P^1 = \emptyset$. Set $i = 1$ and $P^1 = P^1 \cup F^1$. As long as $|P^1| < NP$, set $i = i+1$, $P^1 = P^1 \cup F^i$. The notation $|P^1|$ represents the cardinality of P^1.
- if $|P^1| = NP$, set $P = P^1$ and go to Step 4. Otherwise, do the following until $|P^1| = NP$: identify the solutions with the worst rank $P' \subset P^1$.
- For each solution $x \in P'$, determine the loss of the value of the hypervolume indicator $d(x)$ if it is removed from the set P'. Remove the solution with the smaller loss from P', i.e., set $P' = P' \setminus \{x\}$

Step 4. (Termination) If $m > NG$, set $A = P^1$ and stop. Otherwise, set $m = m + 1$ and go to Step 2.

In Steps 1 and 2, we consider the worst case outcome sets of the individuals and their offspring. We have mentioned earlier that for a fixed solution, finding its worst case outcomes is a multiobjective optimization problem with objectives to be maximized in the uncertainty set. We can solve the maximization problem with an evolutionary multiobjective optimization method to approximate a set of outcomes in the worst case. However, doing so requires a lot of computation resources. Thus, we should find a representative set of solutions of the maximization problem and use it to save the computation resource.

We propose to systematically solve a small number of scalarized subproblems to obtain the representative worst case outcome sets. For example, we can utilize the approach used in [6] to generate a set of evenly distributed points on a unit hyperplane in the objective space. Then, we use them as the reference points to optimize a series of the achievement scalarizing functions (see e.g., [26]). In what follows we denote the number of worst case outcomes in the representative worst case outcome set by W and the values of the uncertain parameters which the objective functions reach their worst case values by $\xi^w, w = 1, \cdots, W$. The number of function evaluations depends on the solver used to solve the scalarized subproblems. In case of discrete scenarios in the uncertainty set, the number of function evaluations is $k \times NP \times NG \times$ number of scenarios.

After we have found the representative worse case outcome sets of the individuals, we need to rank them and sort them into different fronts. We call this step set-based non-dominated sorting, where we define the dominance between two representative worst case outcome sets with lower set less order. The sorting procedure is inspired by that presented in [10]. The steps of the set-based non-dominated sorting are as follows:

Step 1. For each solution $p \in P$, set the domination count $n_p = 0$ and the set of solutions dominated by p as an empty set $S_p = \emptyset$. Set $P = P \setminus \{p\}$ and carry out the following steps:

(a) For each $q \in P$, do the following:
If for all $f(q, \xi^w), w = 1, \cdots, W$, there exists $f(p, \xi^w)$ such that $f(q, \xi^w) \leq f(p, \xi^w)$, set $n_p = n_p + 1$.
Otherwise if for all $f(p, \xi^w), w = 1, \cdots, W$, there exists $f(q, \xi^w)$ such that $f(p, \xi^w) \leq f(q, \xi^w)$, set $S_p = S_p \cup \{q\}$

(b) If $n_q = 0$, then $p^{rank} = 1$ and $F^1 = F^1 \cup \{p\}$.

Step 2. Set front counter i = 1

Step 3. Do the following steps until $F^i = \emptyset$
For each $p \in F^i$
for each $q \in S_p$
set $n_q = n_q - 1$
if $n_q = 0$, then $q^{rank} = i + 1$, and $F^{i+1} = F^{i+1} \cup \{q\}$, set $i = i + 1$ and continue with Step 3 to the next front.

In the set-based non-dominated sorting, Step 2(a) is for checking if $f_{\mathcal{U}}(p) \preceq^l f_{\mathcal{U}}(q)$ or $f_{\mathcal{U}}(q) \preceq^l f_{\mathcal{U}}(p)$. We pair-wise compare the solutions and go through the outcomes in the representative worst case outcome sets.

After we have sorted the solutions into different fronts, we start the environmental selection in Step 3. We fill the next generation population incrementally starting from solutions that are in F^1 until the number of solutions exceeds the population size NP. Then we delete the solutions from the last front based on the loss of the value of the hypervolume indicator (see e.g., [1,28]). We calculate the loss of the hypervolume when deleting a solution x' as $d(x') = H(S) - H(S')$, where $S = \{\tilde{f_{\mathcal{U}}}(x) : x \in P'\}$ and $S' = S \setminus \{\tilde{f_{\mathcal{U}}}(x')\}$. Here, we use $\tilde{f_{\mathcal{U}}}$ instead of $f_{\mathcal{U}}$ because we consider the representative worst case outcome sets.

After step 3, we have a new population. If the number of generations has been exceeded, we terminate the solution process and take the set-based non-dominated solutions of the last generation as the output set A. If the number of generations has not been exceeded, we continue by going to Step 2.

After obtaining the set A, a decision maker should choose a final solution. For example, [27] uses an interactive post-processing procedure to find the final solution based on preference information. In the interactive process, we present the outcome of a solution in the nominal case which is the undisturbed or usual case. Then, the decision maker can specify her or his preferences for a more desired solution until (s)he finds a satisfactory solution. The purpose is to help the decision maker to find the final solution based on the nominal value and at the same time the solution is the best possible when the worst case happens.

4 Numerical Results

In this section, we demonstrate the usage of the SIBEA-R method with two example problems. The examples help us to test our proposal of using set-based non-dominated sorting in an evolutionary algorithm. The first example problem is a simple linear problem based on one of the examples presented in [25]:

$$\left(\begin{array}{ll} \text{minimize} & \begin{pmatrix} 2\xi_1 x_1 - 3\xi_2 x_2 \\ 5\xi_1 x_1 + \xi_2 x_2 \end{pmatrix} \\ \text{subject to} & 0 \le x_1 \le 1.5 \\ & 0 \le x_2 \le 3 \end{array}\right)_{\xi \in \mathcal{U}}, \tag{4}$$

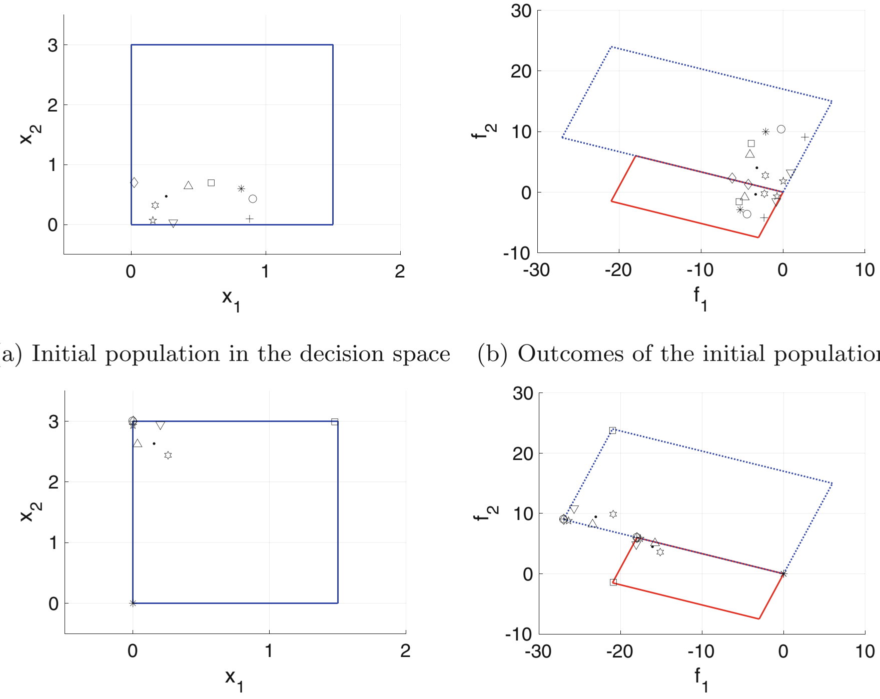
where $\mathcal{U} = \left\{ \left(\begin{smallmatrix} -1 \\ 2 \end{smallmatrix}\right), \left(\begin{smallmatrix} 2 \\ 3 \end{smallmatrix}\right) \right\}$.

In the experiments, we used the default setting of parameters as in the implementation of SIBEA in [7]. For (4), we can compute the outcomes in both possible sets of values for the uncertain parameters. We first illustrate the evolvement of the population, we visualize the initial generation in the decision space in Fig. 2a and in the objective space in Fig. 2b. In the figures, the solid lines are the borders of the feasible set and we visualize 10 individuals because of limited varieties of markers. In Fig. 2b, the same marker appears twice because of the two possible cases in $\mathcal{U}$. We use SIBEA-R to evolve the population by considering their outcome sets (each set consists of two outcomes with the same marker in the figure). After 100 generations, the last generation is shown in Fig. 2c in the decision space and in Fig. 2d in the objective space.

We then studied the final populations of 20 independent runs with $NP = 30$. It is not even possible to compute a complete set of set-based robust Pareto optimal solutions for linear problems like (4). To the best of our knowledge, methods with similar ideas in the literature (e.g., [2]) had a different definition of robust Pareto optimality. We cannot easily benchmark the example problems. Thus, we first visually compare the solutions computed by SIBEA-R with 30 solutions computed by the weighted-sum approach proposed in [11]. The purpose is to use the solutions computed by the weighted-sum approach as references.

Figures 3a and b illustrate the solutions computed by the weighted-sum approach and SIBEA-R. The solutions computed by the weighted-sum approach are marked as solid red circles in the figures and the solutions computed by SIBEA-R are marked by the gray plus sings. In the figures, the gray cloud consists of the solutions computed with 20 runs of the SIBEA-R method. We can see that SIBEA-R was able to find the solutions found by the weighted-sum approach. In addition, SIBEA-R also found other solutions in the interior of the feasible space. The existence of set-based minmax Pareto optimal solutions in the interior of the feasible space is proven in [20]. For example, the point $(0.5, 2.4)$ is set-based minmax robust Pareto optimal which can be checked by the definition. Based on the visualizations, we can observe that SIBEA-R has considered the outcomes concerning both sets of possible values of the uncertain parameters and found a set of non-dominated set-based minmax robust solutions.

The second example problem is based on a standard benchmark problem, ZDT2 (see, e.g., [8]). In this problem, we introduced two uncertain parameters which stem from a polyhedral uncertainty set. A polyhedral uncertainty set is given as the convex hull of a finite set of points. Even though modifying the problem can cause the loss of the characteristics of the carefully designed test problems, our purpose is to illustrate the solutions founds by SIBEA-R and the usage of them for decision making. For the ZDT2-based problem, we set



(a) Initial population in the decision space (b) Outcomes of the initial population

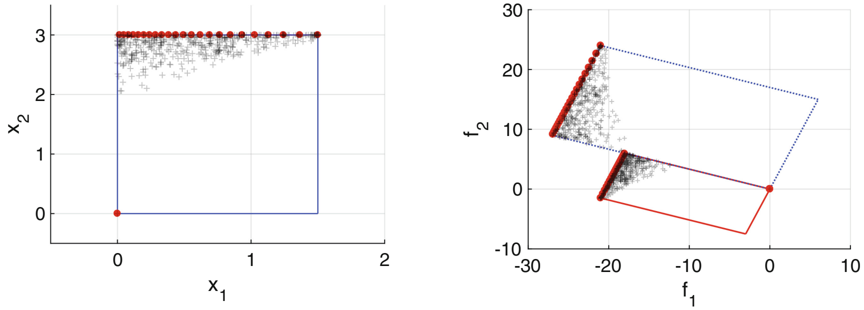
(c) Final population in the decision space (d) Outcomes of the final population

Fig. 2. The evolvement of the population by SIBEA-R

$NG = 100, NP = 30$ and found six worst case outcomes to represent the worst case outcome set. We run SIBEA-R 20 times to solve the problem.

We analyzed the results with the so-called average non-dominated objective space (i.e., the percentage of the volume of objective space between the ideal point and a reference vector which are not covered by the solutions) in each generation in all the runs to observe the convergence (see details in [29]). We also analyzed the attainment surface of the worst case outcome sets from multiple runs with the empirical attainment function graphical tools [18,19]. We visualized the $25\%, 50\%, 75\%$ attainment surfaces.

The average non-dominated objective space in each generation for the 20 runs of the ZDT2-based problem is illustrated in Fig. 4. The figure shows that the non-dominated objective space gradually reduced with generations and at the final generations, the average non-dominated space stayed stable. This means that the objective function values of solutions reduced along the generations. The attainment surfaces of the results from the 20 runs are shown in Fig. 5. The figure illustrates that the solutions tend to converge to the area bounded by the intervals $f_1 = [0.5, 0.8], f_2 = [0.2, 0.7]$. Based on the experiment results, we can observe that SIBEA-R was able to improve the populations with the generations and the final populations of different runs were similar.



(a) Solutions in the decision space (b) Solutions in the objective space

Fig. 3. Solutions computed by the weighted-sum approach and SIBEA-R

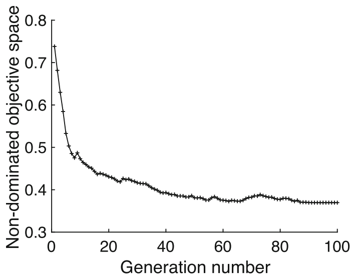


Fig. 4. Average non-dominated objective space, ZDT2-based problem

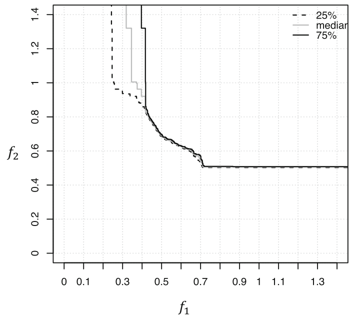


Fig. 5. Attainment surface, ZDT2-based problem

After SIBEA-R has found a set of non-dominated set-based minmax robust solutions, the set can be used for decision making. We illustrate the usage with a reference point-based interactive approach (see e.g., [21] for a detailed description). In a reference point-based approach, the decision maker specifies the desired objective function values as a reference point. We find a solution which satisfies the reference point as well as possible and present the solution to the decision maker. This kind of interactive process continues until the decision maker finds a most satisfactory solution. We used the final population of a run of the ZDT2-based problem and helped a decision maker to choose a final solution based on their outcomes in the nominal case. In the nominal case, the uncertain parameters behave normally without disturbance. So, we used the original ZDT2 problem as the nominal case. We carried out four iterations. The reference points and the solutions found are illustrated in Table 1. The solutions are also presented in Fig. 6 with different markers. The decision maker took the third solution as the final solution since it is the nearest to her desired values.

In the examples, we observed that SIBEA-R was able to find set-based minmax robust Pareto optimal solutions found by the weighted-sum approach. It was

Table 1. Interactive post-processing

Ref.	Solution	Marker
$(0.3, 0.7)^T$	$(0.43, 0.81)^T$	Square
$(0.3, 0.95)^T$	$(0.3, 0.91)^T$	Up triangle
$(0.5, 0.6)^T$	$(0.57, 0.67)^T$	Diamond
$(0.8, 0.6)^T$	$(0.61, 0.61)^T$	Down triable

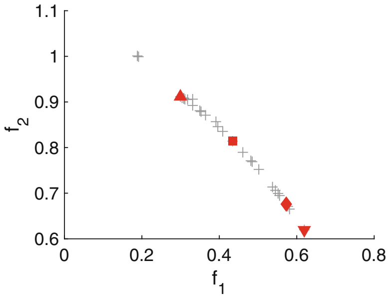


Fig. 6. Solutions found based on reference points

also able to find some solutions that the weighted-sum approach was not able to find. In the ZDT2-based problem, SIBEA-R was stable regarding finding similar final populations in different runs. These observations suggested that SIBEA-R has an appealing potential for approximating set-based minmax robust Pareto optimal solutions, which can be then used for decision making.

5 Conclusions

In this paper, we proposed SIBEA-R to compute an approximated set of set-based minmax robust Pareto optimal solutions. This is an initial study to explore opportunities evolutionary multiobjective optimization methods can provide in tackling challenges with robustness which are otherwise difficult. In SIBEA-R, instead of considering single outcomes, we considered the worst case outcome sets of solutions. We proposed a set-based non-dominated sorting procedure based on the lower set less order to rank the solutions for environmental selection. We illustrated the utilization of SIBEA-R with two example problems. The experiments on the example problems suggest that SIBEA-R can approximate set-based minmax robust Pareto optimal solutions. We also illustrated how the solutions found by SIBEA-R can be used in decision making.

Due to the set-based non-dominated sorting and the calculation of the hypervolume of outcome sets, SIBEA-R is computationally expensive and it tends to work with small population sizes. Thus, an immediate future research direction is to improve the computational efficiency and enable the calculation of a larger number of non-dominated set-based minmax robust solutions. In this paper, we only presented a limited amount of numerical experiments. It is necessary to extend the numerical experiments to a wider range of problems to further identify the strengths and limitations of SIBEA-R.

Acknowledgments. We thank Dr. Tinkle Chugh for useful discussions and providing an implementation of SIBEA. This research is related to Decision Analytics (DEMO).

References

1. Auger, A., Bader, J., Brockhoff, D., Zitzler, E.: Theory of the hypervolume indicator: Optimal μ-distributions and the choice of the reference point. In: Proceedings of the Tenth ACM SIGEVO Workshop on Foundations of Genetic Algorithms, pp. 87–102. ACM, New York (2009)
2. Avigad, G., Branke, J.: Embedded evolutionary multi-objective optimization for worst case robustness. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, GECCO 2008, pp. 617–624. ACM (2008)
3. Bader, J., Brockhoff, D., Welten, S., Zitzler, E.: On using populations of sets in multiobjective optimization. In: Ehrgott, M., Fonseca, C.M., Gandibleux, X., Hao, J.-K., Sevaux, M. (eds.) EMO 2009. LNCS, vol. 5467, pp. 140–154. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01020-0_15
4. Bader, J., Zitzler, E.: Robustness in hypervolume-based multiobjective search. Technical report, TIK Report 317 (2010)
5. Bokrantz, R., Fredriksson, A.: Necessary and sufficient conditions for Pareto efficiency in robust multiobjective optimization. Eur. J. Oper. Res. **262**(2), 682–692 (2017)
6. Cheng, R., Jin, Y., Olhofer, M., Sendhoff, B.: A reference vector guided evolutionary algorithm for many-objective optimization. IEEE Trans. Evol. Comput. **20**(5), 773–791 (2016)
7. Chugh, T., Sindhya, K., Hakanen, J., Miettinen, K.: An interactive simple indicator-based evolutionary algorithm (I-SIBEA) for multiobjective optimization problems. In: Gaspar-Cunha, A., Henggeler Antunes, C., Coello, C.C. (eds.) EMO 2015. LNCS, vol. 9018, pp. 277–291. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-15934-8_19
8. Coello, C.A.C., Lamont, G.B., Veldhuizen, D.A.V.: Evolutionary Algorithms for Solving Multi-Objective Problems. Springer, Heidelberg (2006). https://doi.org/10.1007/978-0-387-36797-2
9. Deb, K., Gupta, H.: Introducing robustness in multi-objective optimization. Evol. Comput. **14**(4), 463–494 (2006)
10. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
11. Ehrgott, M., Ide, J., Schöbel, A.: Minmax robustness for multi-objective optimization problems. Eur. J. Oper. Res. **239**(1), 17–31 (2014)
12. Gaspar-Cunha, A., Covas, J.A.: Robustness in multi-objective optimization using evolutionary algorithms. Comput. Optim. Appl. **39**(1), 75–96 (2007)
13. Gong, D., Sun, J., Miao, Z.: A set-based genetic algorithm for interval many-objective optimization problems. IEEE Trans. Evol. Comput. **22**(1), 47–60 (2018)
14. Ide, J., Schöbel, A.: Robustness for uncertain multi-objective optimization: a survey and analysis of different concepts. OR Spectr. **38**(1), 235–271 (2016)
15. Konur, D., Farhangi, H.: Set-based min-max and min-min robustness for multi-objective robust optimization. In: Coperich, K., Cudney, E., Hembhard, H. (eds.) Proceedings of the 2017 Industrial and Systems Engineering Research Conference, pp. 1–6. Institute of Industrial and Systems Engineers (2017)
16. Kuroiwa, D., Lee, G.M.: On robust multiobjective optimization. Vietnam J. Math. **40**(2, 3), 305–317 (2012)
17. Li, M., Azarm, S., Aute, V.: A multi-objective genetic algorithm for robust design optimization. In: Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation, GECCO 2005, pp. 771–778 (2005)

18. López-Ibáñez, M.: EAF graphical tools. http://lopez-ibanez.eu/eaftools
19. López-Ibáñez, M., Paquete, L., Stützle, T.: Exploratory analysis of stochastic local search algorithms in biobjective optimization. In: Bartz-Beielstein, T., Chiarandini, M., Paquete, L., Preuss, M. (eds.) Experimental Methods for the Analysis of Optimization Algorithms, pp. 209–222. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-02538-9_9
20. Majewski, D.: Robust bi-objective linear optimization. Master's thesis, University of Göttingen (2014)
21. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers, Boston (1999)
22. Miettinen, K., Mustajoki, J., Stewart, T.J.: Interactive multiobjective optimization with NIMBUS for decision making under uncertainty. OR Spectr. **36**(1), 39–56 (2014)
23. Rodrguez-Marn, L., Sama, M.: (λ, c)-contingent derivatives of set-valued maps. J. Math. Anal. Appl. **335**(2), 974–989 (2007)
24. Wiecek, M.M., Blouin, V.Y., Fadel, G.M., Engau, A., Hunt, B.J., Singh, V.: Multi-scenario multi-objective optimization with applications in engineering design. In: Barichard, V., Ehrgott, M., Gandibleux, X., T'Kindt, V. (eds.) Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications. LNE, vol. 618, pp. 283–298. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-540-85646-7_26
25. Wiecek, M.M., Dranichak, G.M.: Robust multiobjective optimization for decision making under uncertainty and conflict. In: Gupta, A., Capponi, A., Smith, J.C., Greenberg, H.J. (eds.) Optimization Challenges in Complex, Networked and Risky Systems, pp. 84–114 (2016)
26. Wierzbicki, A.P.: A mathematical basis for satisficing decision making. Math. Model. **3**(5), 391–405 (1982)
27. Zhou-Kangas, Y., Miettinen, K., Sindhya, K.: Interactive multiobjective robust optimization with NIMBUS. In: Proceedings of Clausthal-Goettingen International Workshop on Simulation Science 2017. Springer (2018, to appear)
28. Zitzler, E., Brockhoff, D., Thiele, L.: The hypervolume indicator revisited: on the design of pareto-compliant indicators via weighted integration. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 862–876. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-70928-2_64
29. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: improving the strength Pareto evolutionary algorithm for multiobjective optimization. Technical report, TIK Report 103 (2002)
30. Zitzler, E., Thiele, L., Bader, J.: On set-based multiobjective optimization. IEEE Trans. Evol. Comput. **14**(1), 58–79 (2010)