An Interactive Simple Indicator-Based Evolutionary Algorithm (I-SIBEA) for Multiobjective Optimization Problems

Tinkle Chugh(✉), Karthik Sindhya, Jussi Hakanen, and Kaisa Miettinen

Department of Mathematical Information Technology, University of Jyvaskyla, P.O. Box 35, FI-40014 Jyvaskylan yliopisto, Finland
tinkle.t.chugh@student.jyu.fi,
{karthik.sindhya,jussi.hakanen,kaisa.miettinen}@jyu.fi

Abstract. This paper presents a new preference based interactive evolutionary algorithm (I-SIBEA) for solving multiobjective optimization problems using weighted hypervolume. Here the decision maker iteratively provides her/his preference information in the form of identifying preferred and/or non-preferred solutions from a set of nondominated solutions. This preference information provided by the decision maker is used to assign weights of the weighted hypervolume calculation to solutions in subsequent generations. In any generation, the weighted hypervolume is calculated and solutions are selected to the next generation based on their contribution to the weighted hypervolume. The algorithm is compared with a recently developed interactive evolutionary algorithm, W-Hype on some benchmark multiobjective optimization problems. The results show significant promise in the use of the I-SIBEA algorithm. In addition, the performance of the algorithm is demonstrated using a human decision maker to show its flexibility towards changes in the preference information. The I-SIBEA algorithm is found to flexibly exploit the preference information from the decision maker and generate solutions in the regions preferable to her/him.

1 Introduction

Industrial optimization problems often involve multiple conflicting objectives, which usually have multiple Pareto optimal solutions with different trade-offs. Different methods have been proposed in the literature (see e.g. [13]) and evolutionary multiobjective optimization (EMO) algorithms [5,6] have often been applied to solve multiobjective optimization problems and find an approximation of the Pareto front consisting of all the Pareto optimal solutions. However, finding an approximation of the Pareto front is not easy, especially when objective and constraint functions are computationally expensive.

When EMO algorithms are used to find an approximation of the Pareto front, a human decision maker (DM) who is an expert in the domain of the problem is supposed to choose one among several nondominated solutions for implementation or further evaluation. Such an approach is often termed as a posteriori

A. Gaspar-Cunha et al. (Eds.): EMO 2015, Part I, LNCS 9018, pp. 277–291, 2015.
DOI: 10.1007/978-3-319-15934-8_19

approach in multiobjective optimization [13]. Since finding a good approximation of the Pareto front is often difficult, especially when more than two computationally expensive objectives are involved, it is practical to approximate a region of the Pareto front that is of interest to the DM. At least two different approaches involving preference information have been considered in the literature:

1. *a priori approaches* where the DM initially expresses her/his preference information, which is subsequently used to find a set of solutions reflecting her/his preferences [7,9], and
2. *interactive methods* where the DM iteratively provides her/his preference information and drives the algorithm towards her/his preferred region(s) of the Pareto front [8,12,15].

We can easily incorporate DM's preference information in indicator based evolutionary algorithms. These algorithms have been proposed in the literature [3,4] to handle a large number of objectives and in this article we focus our attention on these algorithms. In them, a hypervolume of the dominated region of the objective space is used as the indicator of the quality of the approximation of the Pareto front due to the Pareto compliance of the indicator [16]. However, as the number of objectives increases, the calculation of the hypervolume gets extremely time consuming. Recently, a Monte-Carlo simulation based approach to calculate hypervolume has been proposed to speed up the calculation [2].

In this paper, we propose a new interactive preference based EMO algorithm called *interactive simple indicator-based evolutionary algorithm (I-SIBEA)* where different weights are associated with different regions of the Pareto front such that the importance given by the DM for different regions of the Pareto front can be altered. In the proposed algorithm, we extend the simple indicator-based evolutionary algorithm (SIBEA) [16] to take into account the preference information of the DM iteratively and direct the search towards the preferred regions of the DM. Specifically, the DM is iteratively shown a set of nondominated solutions and asked to provide her/his preferences by classifying this set into preferred and/or non-preferred solutions. The weights of the preferred solutions in the weighted hypervolume calculation are subsequently altered such that their selection pressure is increased. Using preference information for both the preferred and non-preferred solutions simultaneously is a novel approach in preference based EMO algorithms and provides more flexibility to the DM in guiding the search.

The rest of the paper is organized as follows. In Section 2, we introduce the main concepts and discuss how the preference information is incorporated into the method. Then I-SIBEA algorithm is presented in Section 3 with detailed description. In Section 4, we present preliminary numerical experiments used to test the method. Finally, the conclusions are drawn in Section 5.

2 Main Concepts and Principles of Utilizing Preference Information from the Decision Maker in I-SIBEA

2.1 Concepts and Notations

We consider multiobjective optimization problems of the form [13]:

$$\begin{aligned} &\text{minimize } \{f_1(x), \ldots, f_k(x)\} \\ &\text{subject to} \quad x \in S \end{aligned} \tag{1}$$

with $k(\geq 2)$ objective functions $f_i(x) : S \rightarrow \Re$. The vector of objective function values is denoted by $f(x) = (f_1(x), \ldots, f_k(x))^T$. For the simplicity of presentation, we assume that all the objective functions are to be minimized. If some objective function f_i is to be maximized, it is equivalent to minimize $-f_i$. The (nonempty) feasible region (set) S is a subset of the decision variable region $\Re^n$ and consists of decision variable vectors $x = (x_1, \ldots, x_n)^T$ that satisfy all the constraints. The image of the feasible region S in the objective region $\Re^k$ is called the feasible objective region (set) denoted by Z. The elements of Z are called feasible objective vectors denoted by $f(x)$ or $z = (z_1, \ldots, z_k)^T$, where $z_i = f_i(x), i = 1, \ldots, k$, are the objective function values. An ideal objective vector $z^* \in \Re^k$ is determined by minimizing each objective function individually, that is $z_i^* = \underset{x \in S}{\text{minimize}} f_i(x)$. We say that a vector $z^1 \in \Re^k$ is said to weakly dominate a vector $z^2 \in \Re^k$ and denoted by $z^1 \preceq z^2$ if and only if for all $1 \leq i \leq k$: $f_i(x^1) \leq f_i(x^2)$.

In this paper, we consider an interactive preference based EMO algorithm, wherein a DM iteratively provides her/his preference information as a set of preferred and/or non-preferred solutions. To emphasize solutions in the preferred region, the weighted hypervolume, $I_H^w(A)$ is used, where A is the set of nondominated solutions in the objective space. The weighted hypervolume is defined as the integral over the product of the weight distribution function $w(z)$ and the attainment function $\alpha(z)$ [16], that is,

$$I_H^w(A) = \int\int_Z w(z)\alpha_A(z)dz$$

where

$$\alpha_A(z) = \begin{cases} 1 & \text{if } A \preceq z \\ 0 & \text{else} \end{cases}$$

and $A \preceq z$ represent that at least one element of A weakly dominates $z \in Z$.

2.2 Incorporating Preference Information into the Algorithm

There are different ways to obtain preference information from the DM. In the preference based EMO algorithms literature where the hypervolume based selection criterion is used [3,4,16], as far as we know, only the preferred solutions are considered as the preference information from the DM. In the proposed I-SIBEA algorithm, we provide the flexibility to the DM to give her/his preferences by selecting preferred and/or non-preferred solutions among a set of nondominated solutions shown to her/him. For example, if the DM selects only preferred solutions, the rest of the solutions can be regarded as either non-preferred solutions or solutions with no preference information. However, in this study we consider them as non-preferred solutions. On the other hand, if the DM selects both preferred and non-preferred solutions, the rest of the solutions are regarded as solutions with no preference information.

It is often assumed that the DM has prior information about the preferred solutions before starting the solution process [3,16]. In the I-SIBEA algorithm, it is not assumed that the DM has some prior information about preferred and/or non-preferred solutions and that the DM is consistent during interaction. The DM iteratively gives her/his preference information, which is used by the I-SIBEA algorithm to focus its search towards solutions that lie in the preferred region. In what follows, we discuss how DM's preferences are incorporated in the I-SIBEA algorithm.

As mentioned in the introduction, the proposed algorithm extends SIBEA to consider the preference information of the DM. After a fixed number of generations of SIBEA, in the first interaction with the DM, a set $A \subset Z$ of non-dominated solutions (in the objective space) is shown to the DM. The number of solutions shown is a parameter that the DM can set. Next, we suppose that the DM selects preferred and non-preferred solutions from the set A. Therefore, the obtained preference information creates a partition of A into three non-overlapping subsets:

$$AA = \{z \in A \mid z \text{ is preferred by the DM}\}$$
$$RA = \{z \in A \mid z \text{ is non-preferred by the DM}\}$$
$$IA = \{z \in A \mid \text{no preference information is available from the DM for z}\}$$

and $A = AA \cup RA \cup IA$.

After the partitioning of A into the three subsets, Z is partitioned into three regions based on the preferences from the DM. The regions are called dominated (Do), preferred (Pr) and no preference information (In) and an example illustrating them is shown in Fig. 1 for a biobjective optimization problem. The shaded region in the Fig. 1 represents the infeasible region.

In what follows, three regions, Do, Pr and In are defined based on preference information from the DM. The weight distribution function is then derived using hypervolume of these three regions which is incorporated into the algorithm to calculate the weighted hypervolume.

The region Do is the part of Z which is weakly dominated by at least one element of RA:

$$Do = \{z \in Z \mid \text{there exists } \overline{z} \in RA, \overline{z} \preceq z\}.$$

The hypervolume $\mu(Do)$ and the weighted hypervolume $w(Do)$ for region Do are calculated as

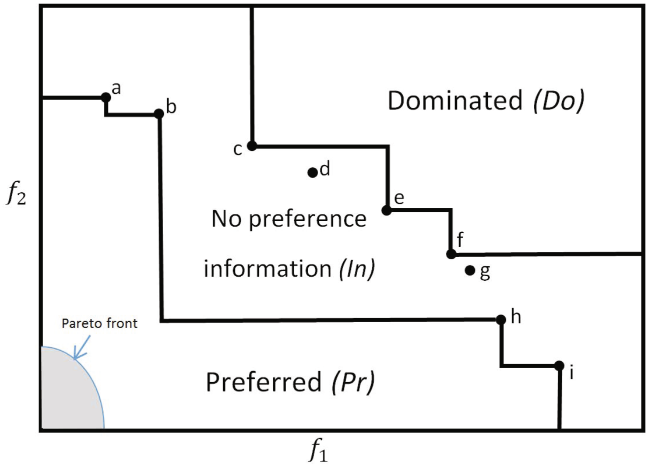

Fig. 1. $AA = \{a, b, h, i\}, RA = \{c, e, f\}, IA = \{d, g\}$. Regions: dominated, no preference information and preferred

$$\mu(Do) = \int \int_z \alpha_{RA}(z) dz$$

$$w(Do) = I_H^{(w)}(RA).$$

The region Pr is the part of Z which weakly dominates at least one element of AA:

$$Pr = \{z \in Z \mid \text{there exists } \overline{z} \in AA, z \preceq \overline{z}\}.$$

The hypervolume $\mu(Pr)$ and the weighted hypervolume $w(Pr)$ for region Pr are calculated as

$$\mu(Pr) = \int \int_z \alpha_{AA}(z) dz$$

$$w(Pr) = I_H^{(w)}(AA).$$

The region In is the remaining part of Z (Fig. 1):

$$In = Z \setminus \{Do \cup Pr\}$$

with the hypervolume $\mu(In) = 1 - \mu(Do) - \mu(Pr)$.

The reason to partition Z into these three regions is to emphasize the solutions that lie in the preferred region (Pr). There can exist several ways to implement this principle. In the literature [3,12,16], several weight distribution functions (e.g. stressing objectives with exponential weights, guiding single solutions with dirac-type weights etc.) are used to incorporate the DM's preference information in the solution process. We present here a uniform weight distribution as one of the possibilities. As the DM wants to avoid the non-preferred solutions, $w(z)$ remains zero for the region Do. Therefore, we define the weight distribution function as:

$$w(z) = \begin{cases} 0 & \text{for all } z \in Do \\ 1 & \text{for all } z \in In \\ 1 + \frac{\mu(Do)}{\mu(Pr)} & \text{for all } z \in Pr \end{cases}$$

(so that $\int\int_Z w(z)dz = 1$).

This weight distribution function is then used to calculate the weighted hypervolume in the subsequent generations and the solutions are selected based on their contribution to the weighted hypervolume. In this way, the preference information from the DM is incorporated into the algorithm. In what follows, the I-SIBEA algorithm is presented with detailed description.

3 Interactive Simple Indicator-Based Evolutionary Algorithm (I-SIBEA)

The main motivation of the proposed algorithm is to direct its search process towards solutions that lie in the preferred region defined by the DM's preferences. To do this, solutions having a large contribution to the weighted hypervolume are selected and solutions having the smallest contribution to the weighted hypervolume are removed from the population after every generation. This criterion of selecting solutions is common among hypervolume based search algorithms [16,17]. In the proposed I-SIBEA, in addition to hypervolume based selection criterion, different preference information from the DM is incorporated into the algorithm. In this algorithm, the DM gives her/his preference information by selecting preferred and/or non-preferred solutions. This preference information guides the algorithm to focus its search direction for solutions that lie in the preferred region. The algorithm is presented in the I-SIBEA algorithm and we discuss now the step by step procedure of the algorithm.

Initially, a population P of individuals of size NP is created randomly in step 1. Next in step 2, crossover and mutation operators are used to create an offspring population Q of the same size (NP). The parent and the offspring populations are combined $P := P + Q$ and then environmental selection is used to select individuals as mentioned in step 3 of the I-SIBEA algorithm. Nondominated sorting [14] is used to rank the individuals of the combined population and different fronts $F_i, i = 1, 2, \ldots$ are identified. These fronts are added to an empty set $P1$ as long as the size of the population of $P1$ becomes equal to or exceeds NP. If the size of $P1$ is NP, the population for next generation is set as $P := P1$. Otherwise, the set of individuals in the worst rank front in $P1$ is identified and denoted by P'. To remove solutions from the worst rank front so that the population size of $P1$ does not exceed NP, the usual hypervolume based selection is used. For each solution $z \in P'$, the loss in the hypervolume $d(z) = I(P') - I(P' \setminus z)$ is determined, where I is the hypervolume indicator and represented

Algorithm: An interactive simple indicator-based evolutionary algorithm (I-SIBEA)

Input to algorithm: NP = population size; NG = maximum number of generations
Input from DM: DA = maximum number of solutions to be shown to the DM (default is maximum 5); AA = preferred and RA = non-preferred solutions after each interaction; H = maximum number of interactions
Output: f^* = Pareto optimal solution obtained by projecting the most preferred solution to the Pareto front, where $f^* \subseteq A$ and A is the set of nondominated solutions in the last population
Step 1 (Initialization): Generate an initial set P of decision vectors of size NP; set the generation counter $m := 1$; set the interaction step $intr := 0$; $NA :=$ number of points in A; set number of generation before first interaction $NI := round(NG/H)$; set $N := NI$; set the hypervolume indicator $I := \mu(\cdot)$.
Step 2 (Mating): Create an offspring population Q using crossover and mutation operators. Set $P := P + Q$ (multi-set union).
Step 3 (Environmental Selection): Rank the population P using nondominated sorting and identify different fronts $F_i, i = 1, 2, \ldots$ and do the following four steps $(a-d)$.

a. Set a new population $P1 = \phi$. Set a count $i = 1$ and perform $P1 = P1 + F_i$ and as long as $|P1| \geq NP$ and set $i = i + 1$. Here, $|P1|$ denotes the cardinality of $P1$.
b. If $|P1| = NP$, set $P := P1$ and go to step 4 otherwise determine the set of individuals $P' \subseteq P1$ with the worst rank.
c. if $m \leq N$ and N = NI
   For each solution $z \in P'$ determine the loss $d(z)$ in the hypervolume I if it is removed from P', i.e., $d(z) := I(P') - I(P' \setminus z)$.
   else
   Identify P'_{Pr} i.e. the solutions $x \in P'$ belonging to region Pr and perform $P1 = P1 \setminus P' + P'_{Pr}$

   1. If $|P1| \geq NP$, for each solution in $z \in P'$ belonging to Pr determine the loss $d(z)$ in the hypervolume I if it is removed from P', i.e., $d(z) := I(P') - I(P' \setminus z)$.
   2. Else determine the loss in weighted hypervolume $d(z) := I(P') - I(P' \setminus z)$ for each solution $z \in P'$ belonging to the regions Do and In.

d. Remove the $|P1| - NP$ solutions from P' with the smallest loss $d(z)$ (ties are broken randomly) and include the remaining solutions of P' into $P1$. Set $P := P1$.

Step 4: If $m \geq NG$ or $m \geq N$ then go to step 5. Otherwise set $m := m + 1$ and go to step 2.
Step 5 (Identify A): Set A as the set of nondominated solutions in P. If $NA > DA$, remove additional solutions by using e.g. clustering.
Step 6 (Interaction with DM): Show DA solutions of A to the DM and set $intr := intr + 1$. If the DM wants to stop or $m \geq NG$, go to step 7 otherwise go to step 8.
Step 7 (Termination): Ask the DM to select the most preferred solution (f^*) from DA. Obtain the final solution by projecting f^* to the Pareto front and terminate the algorithm.
Step 8: Ask the DM to classify DA into AA and/or RA and derive the sets Do, In and Pr to get the updated weighted hypervolume $w(\cdot)$. Set $I := w(\cdot)$; $NI = \frac{NG-NI}{H-intr}$; $N := m + NI$ and $m := m + 1$. Go to Step 2.

as the hypervolume or the weighted hypervolume for a given set. The solution with the smallest loss is removed until the size of the population does no longer exceed NP and the population is set as $P := P1$ for the next generation. After a fixed number of generations, NI in step 4, the DM interacts with the algorithm. In step 5, a fixed number of nondominated solutions $DA \subseteq A$ (input from the DM) is identified and then shown to her/him in step 6, where A is the set of nondominated solutions. There exist different ways to select the fixed number of solutions from A and we use k-means clustering [11] in this study. Here, a solution is selected randomly from each cluster and shown to the DM. In step 7, if the DM wants to quit, s(he) selects the most preferred solution f^* from DA. The final solution is obtained by projecting f^* to the actual Pareto front by optimizing an achievement scalarizing function (ASF) [13], that is by solving the problem

$$\begin{gathered} \text{minimize} \max_{i=1,\ldots,k} [w_i(f_i(x) - f_i^*)] + \rho \sum_{i=1}^{k} w_i(f_i(x) - f_i^*) \\ \text{subject to } x \in S. \end{gathered} \tag{2}$$

where $\rho > 0$ is the augmentation coefficient which takes a small positive value e.g. 10^{-6}. The weight vector $w_i = \frac{1}{z_i^{max} - z_i^{min}}$ is assigned to each objective function. The maximum and minimum values of each objective function in the set A are represented by z_i^{max} and z_i^{min}, respectively. One of the advantages for using an ASF is that the optimal solution of an ASF is always Pareto optimal [13]. Therefore, optimizing an ASF ensures that final solution is locally Pareto optimal. Since we assume that less is preferred to more for the DM, the projected solution is at least as preferred to the DM as the solution s(he) selected. We can utilize an equivalent differentiable formulation of ASF when all the objective functions are differentiable by adding extra real valued variable, δ and k new constraints [13]

$$\begin{gathered} \text{minimize } \delta + \rho \sum_{i=1}^{k} w_i(f_i(x) - f_i^*) \\ \text{subject to } w_i(f_i(x) - f_i^*) \leq \delta \text{ for all } i = 1, \ldots, k \\ x \in S \;\; \delta \in \Re. \end{gathered} \tag{3}$$

In addition to termination by the DM, the solution process is ended if the maximum number of generations (NG) is reached. In that case, solutions $DA \subseteq A$ are shown to the DM and (s)he is asked to select the the most preferred solution (f^*). This solution is then projected to the Pareto front and the final solution is obtained by solving problem (2) or (3) with a single objective optimization method appropriate to the characteristics of the problem in question. A local search method can be used since the evolutionary algorithm is supposed to take care of the global search. If the termination criterion is not met, the DM is then asked in step 8 to select preferred (AA) and non-preferred solutions (RA) from DA to get the three non-overlapping subsets AA, RA and IA. The three regions

Do, Pr and In are then derived using this preference information to get the weight distribution function. This completes one interaction with the DM. The weight distribution function is then used to calculate the weighted hypervolume as the selection criterion in the subsequent generations.

In the next generation (after the first interaction), the offspring are created again in step 2 and other steps are then followed. If the population size exceeds NP in $P1$, solutions $z \in P'$ belonging to region Pr are identified and denoted by P'_{Pr}. The set $P1$ is then updated as $P1 := P1 \setminus P' + P'_{Pr}$. If the size of the population of $P1$ exceeds NP, the usual hypervolume based selection is used to remove the solutions from P' belonging to region Pr. Otherwise, the solutions $z \in P'$ belonging to regions Do and In are added to $P1$. If the population size exceeds NP, the weighted hypervolume based selection is used to remove solutions $z \in P'$ belonging to regions Do and In. This principle of selecting individuals emphasizes solutions in the preferred region. The regions Do, Pr and In are updated after every interaction after the DM has classified DA into AA and RA. In this way, the DM gives her/his preference and the weights of the solutions in the weighted hypervolume calculation are altered in such a way that solutions in Pr are emphasized and solutions in Do are avoided.

In the proposed algorithm, the DM has the freedom to choose the number of times (s)he wishes to interact with the algorithm. From this input, the maximum number of generations (NG) is uniformly divided by H to get the number of generations before each interaction. For example, the first interaction will take place after $NI = NG/H$ generations and the second interaction will take place after $N = NI + (NG - NI)/(H - 1)$ generations. Even though, the maximum number of interactions H is given by the DM in the beginning, the DM is free to change it during any interaction. However, this is not presented in the algorithm but if the DM gives her/his updated number of interactions, the remaining generations can be uniformly divided accordingly. In the next section, the algorithm is tested using some benchmark problems.

4 Numerical Experiments

The I-SIBEA algorithm was tested on standard benchmark problems [5] with 2-3 objectives and 7-11 decision variables. Firstly, we compare I-SIBEA against a recently proposed interactive weighted hypervolume based algorithm called W-Hype [4]. One of the main differences between W-Hype and I-SIBEA is that W-Hype considers information for only preferred solutions while I-SIBEA considers information for both preferred and non-preferred solutions. To enable easy comparison, we use the same set of test problems i.e. DTLZ2, ZDT4 and DTLZ1 and the same criterion of [12] to get the number of generations before each interaction as used in W-Hype. The parameter values used in I-SIBEA are provided in Table 1. In all three problems, polynomial mutation (distribution index is 20 and probability of mutation is 1/number of decision variables) and simulated binary crossover (distribution index is 20 and probability of crossover is 0.9) were used. While testing the algorithm for these test problems, we replaced the

DM by a weighted Chebyshev function $\max_{i=1,\ldots,k}[w_i(f_i(x) - z_i^*)]$ at each interaction step with z_i^* as the ideal objective vector. The weight vector $(w_1, \ldots, w_k)^T$ is assigned to each objective function and used to describe the DM's preferences. After each interaction step, the solution that minimized the weighted Chebyshev function was considered as the preferred solution (AA) and the rest of the solutions were considered as non-preferred solutions (RA) i.e. it was assumed that there were no solutions with no preference information (IA). This setting was used to be able to compare I-SIBEA with W-Hype (where this setting had been used).

Table 1. Parameters used in this study

	DTLZ2	ZDT4	DTLZ1
Number of decision variables/objectives	11/2	10/2	7/3
Ideal vector	(0,0)	(0,0)	(0,0,0)
Population size	50	100	200
Number of interactions	2,4,6,8	4,6	4,6
Weight vector	(0.2,0.8)	(0.5,0.5)	(0.7,0.2,0.1)
Number of independent runs	30	50	10
Total number of function evaluations	20,000	40,000	120,000

To measure the performance of the proposed algorithm, mean, standard deviation, absolute deviation and optimal Chebyshev function value were calculated after a maximum number of function evaluations. In the three tables reporting the results of I-SIBEA and W-Hype, the values of performance criteria are written in bold face if the difference was greater than 0.001. The algorithm was also tested by varying the maximum number of interactions. The comparison of the present algorithm with W-Hype for DTLZ2 is shown in Table 2.

Table 2. Results for DTLZ2: algorithm, number of interactions (H), mean, standard deviation (Std.), absolute deviation (Abs.), optimal Chebyshev function value (C^*) and number of function evaluations (nfun)

Algorithm	H	Mean	Std.	Abs.	C^*	nfun
I-SIBEA	2	0.21500	0.052800	0.03230	0.19400	**20,000**
	4	0.19430	0.000571	0.00030	0.19400	**20,000**
	6	0.19410	0.000121	0.00006	0.19400	**20,000**
	8	0.19410	0.000042	0.00003	0.19400	**20,000**
W-Hype	2	**0.19418**	**0.000114**	**0.00016**	0.19403	25,000
	4	0.19413	0.000064	0.00010	0.19403	25,000
	6	0.19411	0.000053	0.00009	0.19403	25,000
	8	0.19410	0.000049	0.00007	0.19403	25,000

The results show that W-Hype performed better than I-SIBEA for $H = 2$. Otherwise, equivalent results were obtained for $H = 4, 6, 8$ and I-SIBEA needed fewer function evaluations when compared with W-Hype. The total number of function evaluations used for this problem by W-Hype and I-SIBEA were $25,000$ and $20,000$, respectively. In addition, better results were obtained by both algorithms with increase in H. We also observed that, after a certain number of generations, the mean of the weighted Chebyshev function did not change for $H = 4, 6$ and 8 which indicates the convergence of the algorithm. Moreover, there was no considerable difference in the results for $H = 6$ and $H = 8$ in I-SIBEA and, therefore, we restricted ourselves to $H = 4$ and 6 when solving the following two problems.

In case of the ZDT4 problem, the weight vector $w = (0.5, 0.5)^T$ was used in the weighted Chebyshev function to identify the preferred solution (AA). The comparison of I-SIBEA with W-Hype is shown in Table 3. I-SIBEA performed better than W-Hype both in terms of results obtained and the number of function evaluations used. I-SIBEA used 40,000 function evaluations and converged in 50% fewer function evaluations as comparison with W-Hype.

Table 3. Results for ZDT4: algorithm, number of interactions (H), mean, standard deviation (Std.), absolute deviation (Abs.) , optimal Chebyshev function value (C^*) and number of function evaluations (nfun)

Algorithm	H	Mean	Std.	Abs.	C^*	nfun
I-SIBEA	4	**0.19180**	**0.001600**	**0.00100**	0.19100	**40,000**
	6	**0.19110**	**0.000262**	**0.00016**	0.19100	**40,000**
W-Hype	4	0.35591	0.203362	0.16493	0.19098	80,000
	6	0.36171	0.230273	0.17073	0.19098	80,000

Next, we tested I-SIBEA on a 3-objective DTLZ1 problem. The weight vector $w = (0.7, 0.2, 0.1)^T$ was used in weighted Chebyshev function to identify the preferred solution (AA). Table 4 shows the results of this study. For this problem as well, equivalent results were obtained in 62.5% fewer function evaluations. I-SIBEA used 120,000 function evaluations in contrast to W-Hype which used 320,000 function evaluations. In all three problems, better or equivalent results were obtained but I-SIBEA always consumed fewer function evaluations. The reason for fewer function evaluations using I-SIBEA can be attributed to the use of preference information of both preferred and non-preferred solutions when compared to W-Hype, where only preferred solutions were considered as the preference information from the DM. This extra information on non-preferred solutions can help the algorithm to avoid solutions in the corresponding regions in subsequent generations and converge faster to solutions in the preferred region. In addition, the DM has more options of how to express one's preferences.

To show the flexibility of the proposed algorithm, a fourth case study was performed on the ZDT4 problem (as it is easy to visualize a biobjective optimization problem), where a DM was involved. In the beginning of the solution process, the DM was asked to provide the maximum number of interactions i.e.

Table 4. Results for DTLZ1: algorithm, Number of interactions (H), mean, standard deviation (std.), absolute deviation (abs.), optimal Chebyshev function value (C^*) and number of function evaluations (nfun)

Algorithm	H	Mean	Std.	Abs.	C^*	nfun
I-SIBEA	4	0.03090	0.000397	0.00035	0.03050	**120,000**
	6	0.03080	0.000167	0.00014	0.03050	**120,000**
W-Hype	4	0.03048	0.000069	0.00005	0.03043	320,000
	6	0.03045	0.000026	0.00002	0.03043	320,000

how many times he wanted to interact with the algorithm. In addition, the flexibility is given to the DM to change the number of nondominated solutions (DA, default is maximum 5) he wanted to see during interaction. In this study, the maximum number of generations (NG) was uniformly divided into 6 times for the interaction with the DM as mentioned in the I-SIBEA algorithm. The other parameters used for this problem are shown in Table 5.

Table 5. Parameters used in the fourth case

	ZDT4
Number of decision variables/objectives	10/2
Population size	50
Number of interactions	6 (Input from the DM)
Maximum number of generation	400

The results of this study are shown in Fig. 2. The first scatter plot shows the nondominated solutions (A) before the first interaction. For the first interaction, the DM wanted to see 5 nondominated solutions and k-means clustering was used to get them. The DM then selected the preferred (AA) and non-preferred solutions (RA) which are shown in the second scatter plot. The solutions obtained before the second interaction are also plotted in the same plot to show the search direction of I-SIBEA. The solution process was then continued until the second interaction. In Fig. 2, the preferred and non-preferred solutions are shown for five interactions. In this study, the DM changed his preferences in the subsequent interactions or in other words, the DM was not consistent with his preferences as shown in Fig. 2. The I-SIBEA algorithm was found to exploit the preference information provided by the DM and generate solutions in the regions preferable to the DM. This shows that the algorithm is flexible to changes in the preferences and can find solutions in the preferred region. After completion of the maximum number of generations, the DM interacted again (6^{th} time) and selected the most preferred solution. This solution was then projected to the Pareto front by solving problem (3). We used here fmincon from MATLAB optimization toolbox to solve problem (3). In this study, $f_1 = 0.096327$ and $f_2 = 0.7074$ were the most preferred objective function values after the final interaction and $f_1 = 0.094345$ and $f_2 = 0.69284$ were the final objective function values.

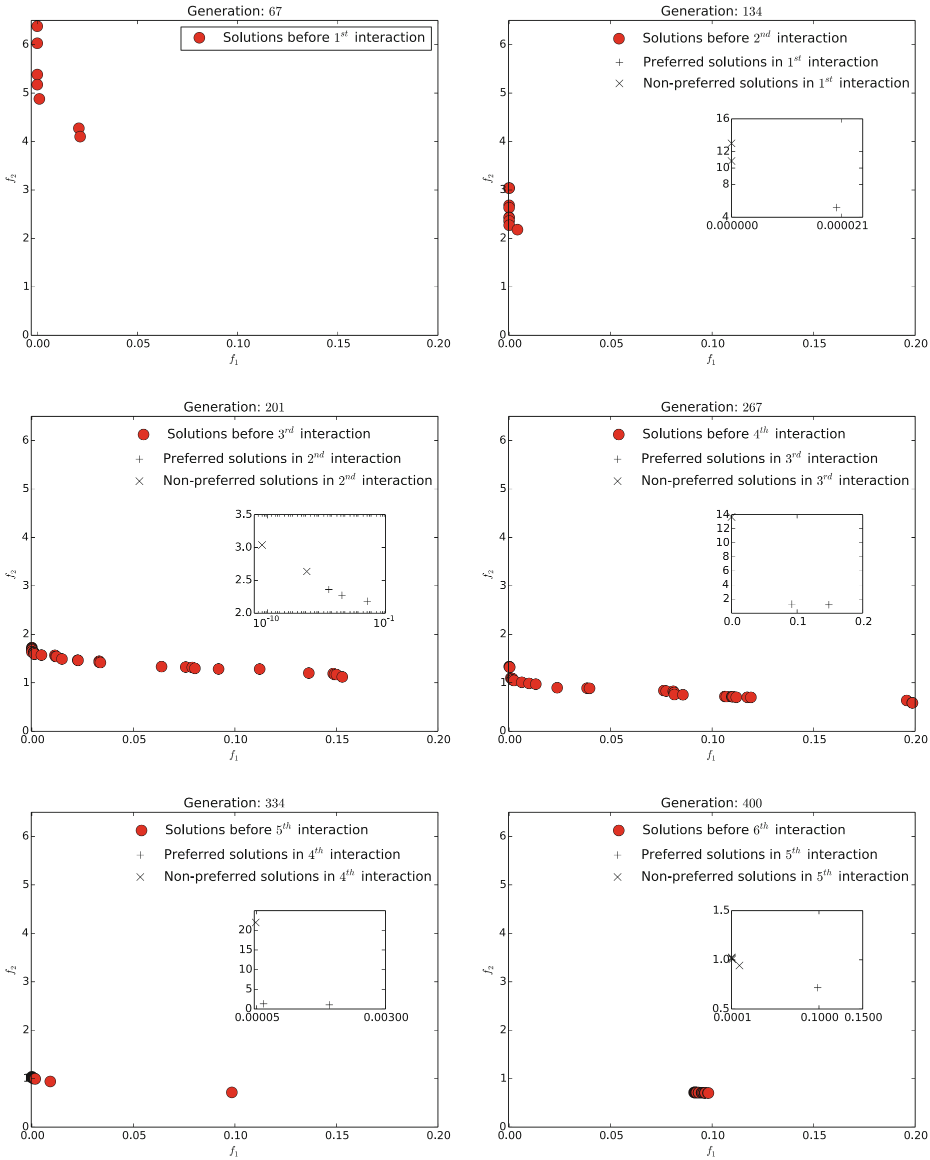

Fig. 2. Decision making process using I-SIBEA algorithm for the ZDT4 problem

The proposed algorithm directed its search towards the DM's preferences and also changed its search direction with changes in the preferences. Therefore, the algorithm emphasized solutions in the preferred region and was flexible to the DM's preferences. In addition, the optimality of the chosen preferred solution was guaranteed (at least locally).

5 Conclusions

In this paper, an interactive simple indicator-based evolutionary algorithm called I-SIBEA is proposed. In this algorithm, the DM's preferences are taken into account in terms of preferred and/or non-preferred solutions. The information for non-preferred solutions helps the algorithm to avoid such solutions in subsequent generations. In this algorithm, the DM can decide how many times s(he) wants to interact with the algorithm and how many solutions s(he) wants to compare while interacting. Therefore, the DM does not need to compare more solutions than (s)he is able to consider at a time. In addition, the algorithm is flexible towards changes in the preferences from the DM. Hence, the algorithm does not assume that the DM has some prior information about preferred and/or non-preferred solutions. Furthermore, unlike typical evolutionary algorithms that cannot guarantee optimality, at least local Pareto optimality of the final solution is guaranteed as it is projected to the Pareto front by optimizing an achievement scalarizing function.

We have compared the performance of I-SIBEA with the W-Hype algorithm. I-SIBEA performed equivalent or better in terms of results obtained but needed fewer function evaluations to get the final solution. In addition, the potential of the algorithm was demonstrated using a human DM to show its flexibility towards changes in the preferences. As indicator based algorithms can handle large numbers of objectives, therefore, next we plan to test the algorithm for more than three objectives and apply the DM's preferences in different ways. Additionally, we plan to develop a GUI which can be utilized with to solve real world multiobjective optimization problems.

Acknowledgement. The authors want to acknowledge a group discussion in the Dagstuhl seminar in 2009 that initiated the idea of the algorithm. In addition, Prof. A. Jaszkiewicz, Prof. T. Lust, Prof. J. Teghem, Prof. E. Zitzler, Prof. L. Thiele, Dr. J. Bader, Dr. T. Ulrich and Dr. B. Naujoks are acknowledged for their efforts in taking the idea further. This research of Dr. Karthik Sindhya was funded by Tekes - the Finnish Funding Agency for Innovation (the SIMPRO project).

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: 10th ACM SIGEVO Workshop on Foundations of Genetic Algorithms, pp. 87–102. ACM (2009)

2. Bader, J., Zitzler, E.: HypE: An algorithm for fast hypervolume-based many-objective optimization. Evolutionary Computation **19**, 45–76 (2011)
3. Brockhoff, D., Bader, J., Thiele, L., Zitzler, E.: Directed multiobjective optimization based on the weighted hypervolume indicator. Journal of Multi-Criteria Decision Analysis **20**, 291–317 (2013)
4. Brockhoff, D., Hamadi, Y., Kaci, S.: Using comparative preference statements in hypervolume-based interactive multiobjective optimization. In: Pardalos, P.M., Resende, M.G.C., Vogiatzis, C., Walteros, J.L. (eds.) LION 2014. LNCS, vol. 8426, pp. 121–136. Springer, Heidelberg (2014)
5. Coello, C.A.C., Lamont, G.B., Veldhuizen, D.A.V.: Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd edn. Springer, New York (2007)
6. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. Wiley, Chichester (2001)
7. Deb, K., Kumar, A.: Light beam search based multi-objective optimization using evolutionary algorithms. In: IEEE Congress on Evolutionary Computation (CEC 2007), pp. 2125–2132 (2007)
8. Deb, K., Sinha, A., Korhonen, P.J., Wallenius, J.: An interactive evolutionary multi-objective optimization method based on progressively approximated value functions. IEEE Transactions on Evolutionary Computation **14**, 723–739 (2010)
9. Deb, K., Sundar, J., Rao, U.B.N., Chaudhuri, S.: Reference point based multi-objective optimization using evolutionary algorithms. International Journal of Computational Intelligence Research **2**, 273–286 (2006)
10. Grunert da Fonseca, V., Fonseca, C.M., Hall, A.O.: Inferential Performance Assessment of Stochastic Optimisers and the Attainment Function. In: Zitzler, E., Deb, K., Thiele, L., Coello Coello, C.A., Corne, D.W. (eds.) EMO 2001. LNCS, vol. 1993, pp. 213–225. Springer, Heidelberg (2001)
11. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: A review. ACM Computing Surveys **31**, 264–323 (1999)
12. Köksalan, M., Karahan, I.: An interactive territory defining evolutionary algorithm: iTDEA. IEEE Transactions on Evolutionary Computation **14**, 702–722 (2010)
13. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer, Boston (1999)
14. Srinivas, N., Deb, K.: Multiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary Computation **2**, 221–248 (1994)
15. Thiele, L., Miettinen, K., Korhonen, P., Molina, J.: A preference-based evolutionary algorithm for multiobjective optimization. Evolutionary Computation **17**, 411–436 (2009)
16. 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)
17. Zitzler, E., Thiele, L., Bader, J.: On set-based multiobjective optimization. IEEE Transactions on Evolutionary Computation **14**, 58–79 (2010)