Using Comparative Preference Statements in Hypervolume-Based Interactive Multiobjective Optimization[⋆]

Dimo Brockhoff[1⋆⋆], Youssef Hamadi[2], and Souhila Kaci[3]

[1] INRIA Lille - Nord Europe, DOLPHIN team, 59650 Villeneuve d'Ascq, France
dimo.brockhoff@inria.fr *(corresponding author)*
[2] Microsoft Research, Cambridge, United Kingdom
[3] Université Montpellier 2, LIRMM, UMR 5506 - CC477, 161 rue Ada, 34095
Montpellier Cedex 5, France

Abstract. The objective functions in multiobjective optimization problems are often non-linear, noisy, or not available in a closed form and evolutionary multiobjective optimization (EMO) algorithms have been shown to be well applicable in this case. Here, our objective is to facilitate *interactive decision making* by saving function evaluations outside the "interesting" regions of the search space within a hypervolume-based EMO algorithm. We focus on a basic model where the Decision Maker (DM) is always asked to pick the most desirable solution among a set. In addition to the scenario where this solution is chosen directly, we present the alternative to specify preferences via a set of so-called comparative preference statements. Examples on standard test problems show the working principles, the competitiveness, and the drawbacks of the proposed algorithm in comparison with the recent iTDEA algorithm.

Keywords: Multiobjective optimization, interactive decision making, Evolutionary Multiobjective Optimization, preferences

1 Introduction

Multiobjective optimization problems with non-linear objectives which, in addition, can be noisy or not even given in closed form occur frequently in practical applications. Evolutionary Multiobjective Optimization (EMO) algorithms have been shown to be applicable in such cases and are typically used in an *a posteriori* scenario. Here, the EMO algorithm computes an approximation of the Pareto front that is then provided to a decision maker (DM) who is supposed to pick the most desired solution [9]. However, one often has to cope with many objectives and large search spaces where the current EMO algorithms need many function evaluations to converge to a good Pareto front approximation. On the

[⋆] This is an author version of the LION'2014 paper published by Springer Verlag. The final publication is available at www.springerlink.com.

[⋆⋆] All authors have also been participating in the CNRS-Microsoft chair "Optimization for Sustainable Development (OSD)" at LIX, École Polytechnique, France.

other hand, the DM is most of the time not even interested in finding solutions covering the entire Pareto front but only in finding solutions within certain *interesting* regions of it.

In such a scenario, it makes sense to interlace the search for a solution set with the articulation of preferences by a DM. Several such *interactive* EMO algorithms have been proposed in previous years in order to reduce the number of function evaluations by exploring only the regions of the search space, the DM is interested in, see for example [12, 16, 18, 20, 22]. Most of those interactive algorithms assume a single preference model and a change in the preference modeling would need a different algorithm [10, 20, 22]. An approach which is able to integrate several preference models is the weighted hypervolume indicator approach [25]. Its main idea is to define a weight function on the objective space and use the contribution to the weighted hypervolume indicator as the *fitness* of each solution within the EMO algorithm. By defining weight functions that induce lines of equal indicator values similar to the lines of equal utility for classical preference models and single solutions, it has been shown that the weighted hypervolume approach can “simulate” the optimization of several classical preference models [7]. However, the weighted hypervolume indicator has not been used yet in an interactive fashion. One goal of this paper is to show how this can be achieved.

To this end, we assume a very basic scenario: In each interaction step, the DM has to decide on the most preferred solutions within the EMO algorithm’s current population (or a subset thereof) and the weight function of the weighted hypervolume indicator within the algorithm W-HypE [7] is adapted accordingly. The first part of the paper is devoted to the simpler direct preference handling where the DM defines the most preferred solutions directly while in the last part of the paper, we show how the most preferred solutions can be specified indirectly with the help of comparative preference statements [17].

In the following, we briefly recapitulate the weighted hypervolume indicator and how it is employed in weighted hypervolume based algorithms (Sec. 3). We then present the proposed framework in which the preference towards a specific solution in the algorithm’s population is transformed into a weight function for the hypervolume indicator (Sec. 4). Experiments show how the DM’s interactive choices affect the search when used within the interactive W-HypE algorithm (Sec. 5). We also compare the proposed interactive W-HypE algorithm with the interactive EMO algorithm iTDEA from [18]. Finally, we present how comparative preference statements can be transformed into a preorder on the population’s solutions and further into a weight function for the indicator (Sec. 6).

2 Preference Articulation And Interactive Optimization In Evolutionary Multiobjective Optimization

Classical EMO approaches aim at finding an approximation of the Pareto front while the DM decides *a posteriori* which solution in the computed set is the most preferred one [9]. Recently proposed *interactive* EMO algorithms, on the other hand, involve the DM already during the search, typically with the need to present non-Pareto-optimal solutions to the DM [16]. At certain stages of the optimization that we call “interaction steps” the DM provides some kind of preference information which is then exploited by the EMO algorithm to find an approximation of the Pareto front which is biased towards the DM’s most preferred solutions. The known interactive EMO algorithms thereby differ mainly in the way the DM’s preference is modeled and used during the search.

According to [16], to which we refer for a broader overview of the topic, “probably the first interactive multiobjective metaheuristic” has been proposed as early as in 1993 [21]. In the meantime, several advanced algorithms have been proposed in the literature of which we briefly discuss the most important here. Thiele et al [22], for example, ask the DM to define desired solutions in the objective space (so-called reference points) and an achievement scalarizing function [19] towards the current reference point is integrated into a binary quality indicator within a state-of-the-art algorithm called IBEA. Deb and Kumar [10] incorporate reference *directions* into the NSGA-II algorithm while Deb et al [12] ask the DM to compare single solutions based on which a polynomial value function is created and optimized. Köksalan and Karahan [18] build their iTDEA algorithm around the idea of selecting only the best solution within a presented subset of the population. This specification of the most preferred solution among a set of solutions is also the scenario, we build upon in the following. The iTDEA algorithm is an interactive version of the original territory-defining evolutionary algorithm (TDEA) and employs in addition to the current population an archive of non-dominated solutions. Within the steady-state TDEA, the newly generated offspring solution is introduced into the archive if it is non-dominated with respect to the archive and at the same time does not fall into the so-called territory of the archive’s solution that is closest to the offspring. The territory of a solution is thereby a hyperbox around the solution’s objective vector with a given width. The main idea behind the interactive iTDEA is to adapt the sizes of the territories according to the DM’s preferences: if a solution lies within the region of the most-preferred solution, the territory size is decreased to allow for more solutions in this region and stays constant in less-preferred regions.

In [1] and [7], several ways to articulate the DM’s preferences within the class of *weighted hypervolume based EMO algorithms* have been presented. Here, we show that the weighted hypervolume approach can also be used in an interactive fashion. To this end, the information about the most preferred solutions specified by the DM is used to define a weight function that has larger values around the preferred solutions—resulting in populations which accumulate close to the solutions that were most-preferred in the previous interaction step.

3 The Weighted Hypervolume Indicator and Hypervolume-Based Selection

Throughout the paper, we assume minimization of k objective functions mapping a solution $x \in X$ from the search space X to its objective vector $f(x) = (f_1(x), \dots, f_k(x))$ in the so-called objective space $\mathbb{R}^k$. We call a solution x^*

Pareto-optimal if there is no other solution $x \in X$ such that x^* is *dominated* by x or, more formally, if there is no other $x \in X$ such that $\forall 1 \leq i \leq k : f(x) \leq f(x^*)$ and $\exists 1 \leq i \leq k : f(x) < f(x^*)$. The set of all Pareto-optimal solutions is called Pareto set and its image in objective space is called Pareto front. The weighted hypervolume indicator is then a set quality measure which assigns a (multi-)set of solutions $A \subseteq X$ the real number $I_{H,w}(A,r) = \int_{\mathbb{R}^k} w(z)\mathbf{1}_{H(A,r)}(z)dz$: the weighted Lebesgue measure of the objective space dominated by solutions in A, bounded by a reference point $r \in \mathbb{R}^k$, and weighted by $w : \mathbb{R}^k \to \mathbb{R}$ [25]. Thereby, $H(A,r) = \{z \in \mathbb{R}^k \,|\, \exists a \in A : f(a) \leq z \leq r\}$ and $\mathbf{1}_S$ is the indicator function of a set S, i.e., $\mathbf{1}_S(s) = 1$ if $s \in S$ and $\mathbf{1}_S(s) = 0$ otherwise. In case of $w(z) = 1$ for all $z \in \mathbb{R}^k$, we use the term (standard) hypervolume indicator.

The (weighted) hypervolume indicator is used frequently for performance assessment of multiobjective optimizers [24] but also in several recent EMO algorithms as optimization criterion within their selection step [2, 4, 15]. One of the main reasons for its popularity is the fact that the (weighted) hypervolume indicator is compliant with the dominance relation—implying that only Pareto-optimal solutions are found if the indicator is optimized [13]. However, optimizing the hypervolume indicator within the selection step of an EMO algorithm exactly is not always possible due to the complexity of the problem. Hence, state-of-the-art hypervolume-based EMO algorithms use two independent strategies to circumvent high computation times in practice: (i) greedy selection instead of finding the optimal subset of points and (ii) estimation of the integral in the indicator by means of Monte Carlo sampling. An algorithm that uses both ideas is HypE [2]. Moreover, HypE uses the idea of the *expected hypervolume loss* of a solution $a \in X$ as the quality of each solution a if a itself and i other randomly chosen solutions are deleted. The generalization of HypE to the weighted hypervolume indicator of [1, 7] is denoted W-HypE. Throughout the paper, we use 10,000 samples in each iteration of W-HypE and refer to [2] and [7] for further details of the algorithm.

4 Interactive Optimization With Weighted Hypervolume Based Selection

Basic Concept. The basic idea behind the proposed approach is to ask the DM to define the most preferred solutions among the current population of the EMO algorithm at certain iterations. These most preferred solutions can be either specified directly or indirectly. In the direct case, the DM picks the most preferred solution from a set of alternative solutions, typically within the EMO algorithm's current population. Once the most preferred solutions are known, they are used as the means of Gaussian weight functions within the W-HypE algorithm, while the directions of the distributions are determined by the extreme points of the current population. Together with the selection scheme of HypE, this will drive the population towards regions with higher weight function values, thus, towards the solutions preferred by the DM. The W-HypE algorithm follows the implementation of [7] and the parameters of the used interactive version are described in more detail in an accompanying technical report [8].

Weight Function. As Gaussian weight functions, which are efficient to sample, we use the ones proposed in [1]. Let $P = \{x_1, \dots, x_{|P|}\}$ be the current population and $b \in P$ the best solution picked by the DM. Then, $m = f(b)$ shall be the mean of the Gaussian distribution and t its direction such that the resulting weight function is $w(z) = \frac{1}{(2\pi)^{k/2}|C|^{1/2}} e^{-\frac{1}{2}(z-m)^T C^{-1}(z-m)}$. with $C := \sigma_\varepsilon^2 \mathbf{I} + \sigma_t^2 tt^T/||t||^2$ the covariance matrix, eigenvalues $\sigma_\varepsilon^2 + \sigma_t^2, \sigma_\varepsilon^2, \dots, \sigma_\varepsilon^2$ and eigenvectors $t_2, \dots, t_k$ taken from an orthogonal basis of the hyperplane orthogonal to t. The determinant of C is denoted as $|C|$. We propose to choose the direction vector t to be proportional to the population's spread as

$$t = \sqrt{k} \cdot \frac{(f_1(b) - f_{1,\max}, \dots, f_k(b) - f_{k,\max})}{\sqrt{(f_1(b) - f_{1,\max})^2 + \dots + (f_k(b) - f_{k,\max})^2}}$$

with k being the number of objectives, and $f_{i,\max}$ and $f_{i,\min}$ being the maximal and the minimal values in objective i found in the current population.

In the current implementation, and following preliminary experiments [8], two overlapping Gaussians with the same mean and direction but different eigenvalues are used: the first one is sampled with 80% and the second with 20% of the samples. The two variances of the first Gaussian are chosen as $\sigma_t = 0.5 \cdot \ell$ and $\sigma_\varepsilon = 0.01 \cdot \ell$ and as $\sigma_t = 0.5 \cdot \ell$ and $\sigma_\varepsilon = 0.1 \cdot \ell$ for the second where $\ell = ||(f_{1,\max} - f_{1,\min}, \dots, f_{k,\max} - f_{k,\min})||$ is the Euclidean distance between the current nadir and ideal point. If two or more solutions are preferred equally by the DM, two Gaussians are defined for each of them and the number of samples are distributed equally among the preferred points. For numerical stability, we choose a diagonal direction with a length of 0.01 in case that the population at the time of the interaction contains only copies of one and the same solution.

Until the first interaction step with the DM, the algorithm is using the standard hypervolume indicator in order to come up with a good spread of the solutions before the DM's decisions change the weight function as described above. In the current implementation, we use a reference point of 111^k, the weighted hypervolume indicator is sampled as above and for the standard hypervolume we sample within the box $[0, 111]^k$.

Interaction Steps. We follow the suggestion of Köksalan and Karahan [18] to specify when the DM is supposed to interact with the algorithm. Given the total number of iterations T of the algorithm and a number of times H, the DM is going to be asked about the most-desired solution, we perform the first interactive step after $T/3$ generations of W-HypE and each later interaction after additional $\frac{T}{2(H-1)}$ iterations. This results in a final optimization stage of (at least) $T/6$ generations in which no interaction is taking place [18]. The algorithm is therefore able to spend a considerable amount of function evaluations before the first and after the last interaction with the DM in order to allow the population to converge as far as possible. Non-integer values are rounded down to the nearest

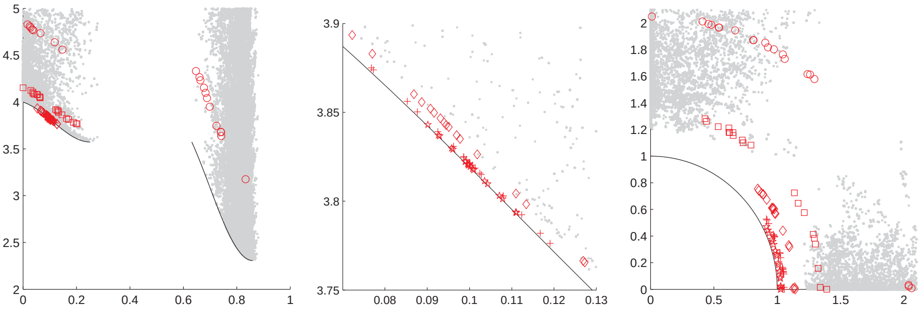

Fig. 1. Examples of interactive optimization runs on the DTLZ7 test function when the DM is preferring the solution closest to $f_1 = 0.1$ (left and middle) and on the DTLZ2 test function when the DM is choosing according to a weighted Chebyshev utility function with weights $(0.1, 0.9)$ (right). Black line: true Pareto front. Gray dots: all solutions visited in 10 independent HypE runs with the same amount of 20,000 function evaluations each. Markers depict the W-HypE populations at interaction steps 1 (○, iteration 333), 2 (□, iter. 499), 3 (◇, iter. 665), 4 (+, iter. 831), and after 1000 iterations (⋆). The middle plot zooms into the left one around the preferred $f_1 = 0.1$.

integer that is smaller than the computed iteration, giving interaction steps at iterations 166, 249, 332, and 415 for $T = 500$ and $H = 4$ for example.

Implementation Details for Two Example Runs. In order to show that the above approach is working, we implemented the interactive W-HypE algorithm within the algorithm package PISA [5][4]. Two example test runs are shown here in which the bi-objective DTLZ2 and DTLZ7 test problems [11] are optimized. The population size is set to $\mu = \lambda = 20$ and the number of decision variables is 100. We employ SBX crossover, polynomial mutation with no symmetric recombination and an individual mutation probability of $1/100$ together with 4 interaction steps in 1000 generations. All other parameters follow the standard PISA setting.

In the first example, shown in the left and middle plot of Fig. 1, the DM decided that the solution closest to $f_1 = 0.1$ is the most desired solution in each step. In the second example, shown in the right plot of Fig. 1, the DM decided according to a weighted Chebyshev utility function [19] with weights $(w_1, w_2) = (0.1, 0.9)$, i.e., chose the solution that minimizes $\max_{1\leq i\leq 2} w_i|f_i(x) - f_i^*|$ at each interaction step with $f^* = (0, 0)$ being the ideal point.

What can be seen from both examples of Fig. 1 is that the interactive W-HypE algorithm follows the directions specified by the DM, i.e., the population is moving towards the selected solution as well as towards better solutions in terms of Pareto dominance. Moreover, we can see that, when compared to the solutions of 10 independent HypE runs in which the standard hypervolume indicator is optimized, the interactive W-HypE algorithm finds solutions closer to the true Pareto front. Note that, however, the results shown in Fig. 1 stem from single algorithm runs which, due to the stochasticity of the algorithm, might not give

[4] The source code is available at `http://inrialix.gforge.inria.fr/interactive/`.

Table 1. Parameter values used in this study and in [18] where the DTLZ2 problem has not been used with two objectives.

	DTLZ2	ZDT4	DTLZ1
number of decision variables / objectives	11 / 2	10 / 2	7 / 3
weights	$(0.2, 0.8)$	$(0.5, 0.5)$	$(0.7, 0.2, 0.1)$
ideal vector	$(0, 0)$	$(0, 0)$	$(0, 0, 0)$
population size	50	200	400
number of interactions H	2, 4, 6, 8	4, 6	4, 6
number of independent runs	50	50	50
total number of funevals	25,000	80,000	320,000

an unbiased view on the real behavior of the algorithm. Hence, we investigate in the following in more detail how the interactive W-HypE algorithm works in terms of statistically sound results over independent algorithm runs.

5 Investigating and Comparing W-HypE in Depth

In the experimental validation of the interactive W-HypE algorithm to follow, we use scenarios from the study of Köksalan and Karahan [18] in order to be able to compare our algorithm with their iTDEA. As in [18], we assume that the DM is choosing the most preferred solution according to a weighted Chebyshev function $\max_{1\leq i\leq k} w_i|f_i(x)-f_i^*|$ with varying weights w_i and ideal point $f^* = \mathbf{0}$. The interactive steps of the W-HypE algorithm appear as in [18] at generations $\frac{T}{3}$, $\frac{T}{3}+\frac{T}{2(H-1)}$, $\frac{T}{3}+\frac{2T}{2(H-1)}$, $\ldots$, and $T-\frac{T}{6}$ if T is the total amount of iterations of the algorithm and H the number of interactions.

We use the above mentioned PISA [5] implementation of the interactive W-HypE algorithm as well as the PISA implementations of the DTLZ1, DTLZ2, and ZDT4 test functions. For the number of decision variables, we follow the recommendations in [11] (DTLZ1 and DTLZ2) and [9] (ZDT4) that have been also reported for the results in [18]. For each combination of problem, weight vector, and number H of interactions, we start 50 independent W-HypE runs with polynomial mutation ($\eta_m = 20$, probability of mutation 1/# decision variables), non-symmetric SBX crossover ($\eta_c = 20$), and standard PISA settings (again, as in [18]. Table 1 gives further parameter values chosen here and in [18].

In order to compare the algorithms, we report mean and standard deviation of the best Chebyshev utility function value U reached after T generations of the algorithms. Furthermore, we compare the mean values with respect to the optimal utility U^* of a Pareto-optimal solution. To this end, we report the absolute differences $U-U^*$ to the best value as well as the relative differences $(U-U^*)/(U^w-U^*)$ with U^w being the worst utility function value of a Pareto-optimal solution [18]. Values for U^* and U^w will be given for each test function and choice of weight vector. Values for iTDEA have been taken from [18] in the "no filter" variant as also here, no preprocessing of the data is performed before the solution sets are shown to the DM.

Table 2. Results for the 2-objective DTLZ2 problem when the DM is acting according to a weighted Chebyshev utility function with $w = (0.2, 0.8)$. Reported are algorithm name, number of interaction steps, mean and standard deviation (std) of the reached Chebyhev utility and the absolute (abs.dev.) and relative deviation (rel.dev.) from the optimal utility (U^*), given the worst utility (U^w) of a Pareto-optimal solution.

Algorithm	Interactions	Mean	std	abs.dev.	rel.dev.	U^*	U^w
WHypE	2	0.19418	0.000114	0.00016	0.0257%	0.19403	0.800
WHypE	4	0.19413	0.000064	0.00010	0.0162%	0.19403	0.800
WHypE	6	0.19411	0.000053	0.00009	0.0142%	0.19403	0.800
WHypE	8	0.19410	0.000049	0.00007	0.0113%	0.19403	0.800
HypE	0	0.19728	0.001531	0.00325	0.5365%	0.19403	0.800

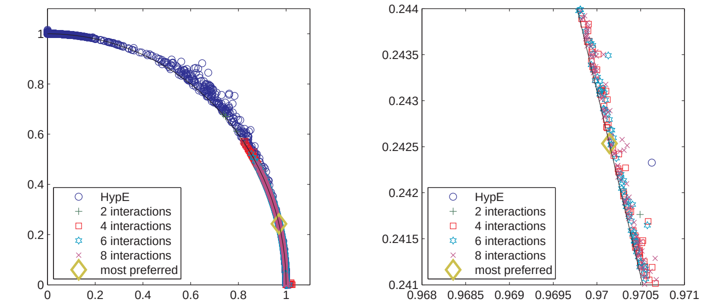

Fig. 2. Resulting final populations of all HypE and interactive W-HypE runs on the DTLZ2 test problem for the full true Pareto front (left) and a zoom (right) around the most desired point $(\sqrt{16/17}, \sqrt{1/17})$ (thick diamond) when the Chebyshev function with weight $(0.2, 0.8)$ is to be optimized. The true Pareto front is depicted in black.

After a first investigation about the influence of the number of interactions on the algorithm performance in the following section, we compare the W-HypE algorithm with the iTDEA of [18].

Varying the Number of Interaction Steps. Table 2 shows results on the 2-objective DTLZ2 function for a weight vector of $(0.2, 0.8)$. Altogether, four different choices for the number of interaction steps are compared with the original HypE that uses no interaction and the standard hypervolume indicator for selection.

It is evident from the results that asking the DM about the most preferred solution and incorporating this knowledge into the W-HypE algorithm is beneficial with respect to the final best solution obtained. The relative error drops from about 0.54% for HypE to less than 0.026% for the interactive W-HypE variants. Note that the value is that low because the W-HypE runs find solutions close to the true Pareto front and close to the most desired solution (see Fig. 2).

The main observation is that, in general, more interaction steps decrease the absolute and relative differences to the most desired point—until the short periods between the interaction steps do not allow for a sufficient optimization anymore. Although this does not happen here for up to 8 interaction steps, we will restrict ourselves to $H = 4$ and $H = 6$ as in [18] in the following.

Table 3. Results for the 2-objective ZDT4 problem when the DM is acting according to a weighted Chebyshev utility function with $w = (0.5, 0.5)$. Abbreviations as in Table 2.

Algorithm	Interactions	Mean	std	abs.dev.	rel.dev.	U^*	U^w
W-HypE	4	0.35591	0.203362	0.16493	53.3731%	0.19098	0.500
W-HypE	6	0.36171	0.230273	0.17073	55.2504%	0.19098	0.500
HypE	0	0.51604	0.216195	0.32506	105.1893%	0.19098	0.500
iTDEA, no filter	4	0.19115	0.000132	0.00017	0.0540%	0.19098	0.500
iTDEA, no filter	6	0.19111	0.000099	0.00013	0.0411%	0.19098	0.500
W-HypE, succ. runs	4	0.19100	0.000009	0.00002	0.0049%	0.19098	0.500
W-HypE, succ. runs	6	0.19098	0.000000	0.00000	0.0011%	0.19098	0.500

Comparison With iTDEA. The DTLZ2 function showed that the interactive W-HypE algorithm works. However, this was a quite simple test function and we investigate now what happens on the more complicated ZDT4 problem [9]. Table 3 shows the results.

First of all, we can see that the interactive W-HypE algorithm performs better than HypE (one-sided Wilcoxon rank-sum tests report statistical differences in favor of W-HypE in both cases of $H = 4$ and $H = 6$; p-value of ≤ 0.05 with Bonferroni correction). However, when compared to the results of [18], all hypervolume-based algorithms perform much worse. The reason for the bad Chebyshev values for the hypervolume-based algorithms is that the algorithm is most of the time stuck on a local Pareto front of which the ZDT4 problem has many. Interestingly enough, the W-HypE algorithm can find solutions on the true Pareto front within the given evaluation budget in 9 out of 50 runs for 4 interactions and in 10 runs for 6 interactions. Looking at the data of the successful runs only, one gets immediately performances comparable to or better than the iTDEA results in [18], cp. Table 3. This observation—and the fact that both algorithms are run for comparable numbers of function evaluations and similar variation operators—suggests that the bad results for W-HypE might come from either the different offspring population sizes ("steady-state" in the case of iTDEA and "$(\mu + \mu)$-selection" in the case of the W-HypE variants) or from the Monte Carlo sampling of the weighted hypervolume. Further investigations in this direction are left for future work.

Comparison for More Objectives. Next, we compare the W-HypE algorithm and HypE with the iTDEA of [18] on the 3-objective DTLZ1 problem. Table 4 shows the results when a Chebyshev utility function with weight vector of $\boldsymbol{w} = (0.7, 0.2, 0.1)$ is defining the DM's preferences. Here, the W-HypE as well as the HypE algorithm are able to reach solutions close to the true Pareto front and the resulting Chebyshev utility functions for the W-HypE algorithm are better than the ones reported for iTDEA [18]. As for DTLZ2, allowing for $H = 6$ interaction steps results in better Chebyshev function values than with $H = 4$ which, interestingly, does not hold for the iTDEA of [18] in this scenario.

Additional Remarks. To conclude, for the experiments with a direct interaction with the DM, we can say that if the operators and the test problem allow HypE

Table 4. Results for the 3-objective DTLZ1 problem when the DM is acting according to a weighted Chebyshev utility function with $w = (0.7, 0.2, 0.1)$. Abbreviations as in Table 2.

Algorithm	Interactions	Mean	std	abs.dev.	rel.dev.	U^*	U^w
WHypE	4	0.03048	0.000069	0.00005	0.0166%	0.03043	0.350
WHypE	6	0.03045	0.000026	0.00002	0.0057%	0.03043	0.350
HypE	0	0.03513	0.001753	0.00470	1.4716%	0.03043	0.350
iTDEA, no filter	4	0.03062	0.000080	0.00019	0.0592%	0.03043	0.350
iTDEA, no filter	6	0.03080	0.000324	0.00037	0.1155%	0.03043	0.350

to find solutions close to the true Pareto front, the corresponding interactive W-HypE algorithm is comparable if not even better than the iTDEA approach of [18]. However, using a steady-state selection as in the iTDEA might improve W-HypE especially for ZDT4 (ongoing work). As to the "real" computational effort, the Monte Carlo sampling of the interactive W-HypE algorithm with 10,000 samples in each generation is still reasonable if the population size of the algorithm is not too high: For example, on an Intel Core 2 Duo T9600, max. 0.05s are spent per function evaluation in the most expensive 3-objective example with population size 400—including the overhead of the PISA framework.

6 Defining the Most Preferred Solutions Via Comparative Preference Statements

Sometimes, a DM can define directly which of the solutions (for example within a sufficiently small set) is the most preferred one as we assumed in the above examples. However, this way to select preferred solutions may not be feasible in practice. This is because DMs are generally reluctant (or not able) to choose among "complete" solutions. In fact, objectives have not necessarily the same importance which may lead to a large number of incomparable solutions. On the other hand, DMs are generally keen to abstract their preferences and compare partial descriptions of solutions called compact preferences. More specifically, instead of providing preferences over solutions (by pairwise comparison or individual evaluation), they generally express preferences over partial descriptions of solutions, e.g., "I prefer solutions with low f_2 value over solutions with medium f_5 value". The task is then to derive a preference relation (a preorder) over solutions given a set of compact preferences. The order's minimal elements can be interpreted as the solutions, most preferred by the DM. Those minimal elements can then be used again as the means of Gaussian weight functions in W-HypE to steer the search towards the most preferred solutions. The problem of deriving a preference relation from a set of compact preferences is well studied in artificial intelligence (AI) [17]. Our aim in this section is to use insights from AI to reason about the DM's preferences. More specifically, we show how compact preference representation languages developed in AI that represent these partial descriptions of the DM's preferences can be transformed into a preference relation (which is a partial/complete (pre)order) on the solutions.

DMs may express compact preferences in different forms. Skipping the details of a formal presentation of these forms (we refer the reader to [17]), we stress that compact preferences implicitly or explicitly refer to comparative preference statements of the form "prefer α to β".

Comparative Preference Statements and Preference Semantics. Handling a comparative preference statement "prefer α to β" is easy when both α and β refer to a solution. However, this task becomes more complex when α and β refer to sets of solutions, in particular when they share some solutions. In order to prevent this situation, Hansson [14] interprets the statement "prefer α to β" as a choice problem between solutions satisfying $\alpha \wedge \neg\beta$ and solutions satisfying $\neg\alpha \wedge \beta$. Particular situations are those when $\alpha \wedge \neg\beta$ (resp. $\neg\alpha \wedge \beta$) is a contradiction or is not feasible in which case it is replaced with α (resp. β). We refer the reader to [14] for further details. For simplicity, we suppose that both $\alpha \wedge \neg\beta$ and $\neg\alpha \wedge \beta$ are consistent and feasible. Let us also mention that the translation of "prefer α to β" into a choice between $\alpha \wedge \neg\beta$-solutions and $\neg\alpha \wedge \beta$-solutions solves the problem of common solutions; however it does not give an indication on how solutions are compared. This problem calls for preference semantics.

Given $\alpha \lhd \beta$, as we will denote a comparative preference statement like "prefer α to β" for brevity, a *preference semantics* refers to the way $\alpha \wedge \neg\beta$-solutions and $\neg\alpha \wedge \beta$-solutions are rank-ordered. Different ways have been studied for the comparison of two sets of objects leading to different preference semantics. The most common ones are *strong* [6], *ceteris paribus* [14], *optimistic* [6], *pessimistic* [3], and *opportunistic* semantics [23]. Looking carefully at the definitions of the different semantics shows that they express more or less requirements on the way $\alpha \wedge \neg\beta$-solutions and $\neg\alpha \wedge \beta$-solutions are rank-ordered. As indicated by its name, strong semantics expresses the most requirements. It states that any $\alpha \wedge \neg\beta$-solution is preferred to any $\neg\alpha \wedge \beta$-solution. This semantics has been criticized in the literature since it generally leads to cyclic preferences when several preference statements are considered. Ceteris paribus semantics has been considered as a good alternative. It weakens strong semantics by comparing less solutions. Optimistic semantics is a left-hand weakening of strong semantics in the sense that instead of requiring that *any* $\alpha \wedge \neg\beta$-solution is preferred to any $\neg\alpha \wedge \beta$-solution, it states that *at least* one $\alpha \wedge \neg\beta$-solution is preferred to any $\neg\alpha \wedge \beta$-solution. Pessimistic semantics is a right-hand weakening of strong semantics and exhibits a dual behavior than the optimistic semantics. Lastly, opportunistic semantics is both left- and right-hand weakening of strong semantics and therefore the weakest among the semantics since it requires that at least one $\alpha \wedge \neg\beta$-solution should be preferred to at least one $\neg\alpha \wedge \beta$-solution. Besides having all their specific advantages and disadvantages, strong and ceteris paribus semantics are the most natural among the mentioned but they can also both return unjustified contradictory preferences, i.e., result in cyclic preferences on solutions [17]. This undesirable case occurs in the presence of defeasible preferences. Defeasible preferences mean that one has a preference and that preference is reversed in a particular context. For example we have "prefer α to β" and "prefer β to α when γ is true". These two preference statements should be consistently handled as they are not contradictory. They just require that the second preference overrides the first one when γ is true. As strong and ceteris paribus semantics are not suitable to reason about defeasible preferences, optimistic, pessimistic and opportunistic semantics have been defined. Without loss of generality, one can focus on these three semantics as they capture strong and ceteris paribus semantics (we skip the details due to space limitation). We will therefore show with the example of *optimistic semantics* how comparative preference statements can be employed in the interactive W-HypE algorithm.

Definition 1 (Optimistic semantics, [6]). *Let $\preceq \subseteq S \times S$ be a preference relation on a solution set $S \subseteq X$ and the corresponding strict preference relation $\prec$ defined by $\omega \prec \omega'$ iff $\omega \preceq \omega'$ holds but $\omega' \preceq \omega$ does not for all $\omega, \omega' \in S$. Furthermore, let $P = \alpha \lhd \beta$ be a comparative preference statement. Then, we say $\preceq$ satisfies $\alpha \lhd \beta$ under the optimistic semantics iff $\forall \omega \in \mathrm{nd}(\alpha \wedge \neg\beta, \preceq), \forall \omega' \in \mathrm{nd}(\neg\alpha \wedge \beta, \preceq) : \omega \prec \omega'$ where $\mathrm{nd}(P, \preceq)$ denotes the set of best solutions according to $\preceq$ that satisfy P. Formally, we write $\mathrm{nd}(P, \preceq) = \{\omega \in S \,|\, \omega$ satisfies P and $\nexists \omega' \in S : \omega' \prec \omega$ and ω' satisfies $P\}$.*

Example 1. Assume, we have five solutions a–e with objective vectors $f(a) = (1, 5)$, $f(b) = (2, 2)$, $f(c) = (3, 1)$, $f(d) = (3, 4)$, and $f(e) = (4, 2)$. When the DM states that "vectors with $f_1 < 3$ (statement 'α') are preferable over vectors with $f_2 < 3$ ('β')", only solution a satisfies $\alpha \wedge \neg\beta$ and solutions c and e satisfy $\neg\alpha \wedge \beta$. A transitive relation with $a \preceq b \preceq c \preceq d \preceq e$ (including the corresponding induced transitive relations) would be one of the possible preference relations that satisfies $\alpha \lhd \beta$ under the optimistic semantics, because $\mathrm{nd}(\alpha \wedge \neg\beta, \preceq) = \mathrm{nd}(\{a\}, \preceq) = \{a\}$, $\mathrm{nd}(\neg\alpha \wedge \beta, \preceq) = \mathrm{nd}(\{c, e\}, \preceq) = \{c\}$ and $a \prec c$.

The following section deals with the question of how such satisfying preference relations can be computed from a set of given comparative preference statements.

From Preference Sets to Preference Relations. The question that remains before using the comparative preference statements within the interactive W-HypE algorithm is how a preference relation $\preceq$ on the solutions can be computed which obeys a certain semantics. Several preference relations may satisfy a preference set $\mathcal{P}_\lhd$ but a unique preference relation can always be computed for each semantics given principles from non-monotonic reasoning called specificity principles. For details we refer the reader to [6, 17] and only present Algorithm 1 for the optimistic semantics and the minimal specificity principle here.

Algorithm 1 computes the final partial preference ordering $\preceq = (E_1, \ldots, E_l)$ on the solutions in a set A equivalence class by equivalence class—starting with the most preferred solutions in E_1. The sets $L(p)$ and $R(p)$ for all preference statements $p : (\alpha \lhd \beta) \in P$ are computed with $L(p)$ containing all solutions that satisfy $\alpha \wedge \neg\beta$ while $R(p)$ contains all solutions satisfying $\beta \wedge \neg\alpha$. An equivalence class contains all solutions for which no preference statement $\beta \wedge \neg\alpha$ is satisfied. The set of not assigned solutions is then updated as is the set of preference

Algorithm 1 Computing a Preference Relation from Preference Statements

```
Require: set $A \subset$ of solutions, comparative preference statements $P = \{p : \alpha \lhd \beta\}$
  Let $L(p) := \{t \,|\, t \in A$ s.t. $t$ satisfies $\alpha \wedge \neg\beta\}$ and $R(p) := \{t \,|\, t \in A$ s.t. $t$ satisfies $\beta \wedge \neg\alpha\}$ for all $p : \alpha \lhd \beta \in P$
  Let $\mathcal{L}(P) := \{(L(p), R(p)) \,|\, p \in P\}$
  $l = 0$
  while $A \neq \emptyset$ do
      $l = l + 1$
      $E_l = \{t \,|\, t \in A$ s.t. $\nexists (L(p), R(p)) \in \mathcal{L}(P) : t \in R(p)\}$
      if $E_l = \emptyset$ then
          stop (contradictory preferences); $l = l - 1$
      $A = A \setminus E_l$
      remove $(L(p), R(p))$ from $\mathcal{L}(P)$ if $L(p) \cap E_l \neq \emptyset$ (remove satisfied preferences)
  return $\preceq = (E_1, \ldots, E_l)$
```

statements to be satisfied. The algorithm stops if either an equivalence class is empty (and hence the preference statements are contradictory) or all solutions are assigned to their equivalence classes.

An Example. To show the usefulness of the above approach of specifying the DM's preference via comparative preference statements, we perform 10 independent W-HypE runs on the DTLZ2 problem with 5 objectives for 1000 generations (popsize 50). The DM is thereby asked 4 times to specify a set of preference statements (at generations 333, 499, 665, and 831). The same preference statements p_1 : prefer $f_2 < 0.05$ over $f_3 < 0.05$, p_2 : prefer $f_1 < 0.05$ over $f_4 < 0.1$, and p_3 : prefer $f_4 < 0.1$ over $f_5 < 0.5$ are used in all 4 interaction steps and interpreted according to the optimistic semantics. Then, a preorder on the current population of W-HypE is computed via Algorithm 1 and the minimal elements in the computed set E_1 are used as means for W-HypE's Gaussian weight functions.

Results. If the weight function of the interactive W-HypE is adapted according to the above comparative preference statements, the percentage of population members that fulfill the defined preference statements increases with each interaction. The lefthand side of Fig. 3 shows the corresponding boxplots. In addition to increasing the number of solutions which fulfill the specified preference statements, W-HypE also optimizes the objective functions which we can see when looking at all solutions of the 10 independent W-HypE runs at the first interaction step and at the end of the runs (Fig. 3, right).

7 Conclusions

Interactive Evolutionary Multiobjective Optimization (EMO) gained recent interest within the research community. In such interactive EMO algorithms, standard set-based EMO algorithms are combined with interactive decision maker (DM) sessions in which the DM articulates preferences towards solutions of interests. These solutions are, in turn, employed to steer the search algorithm towards

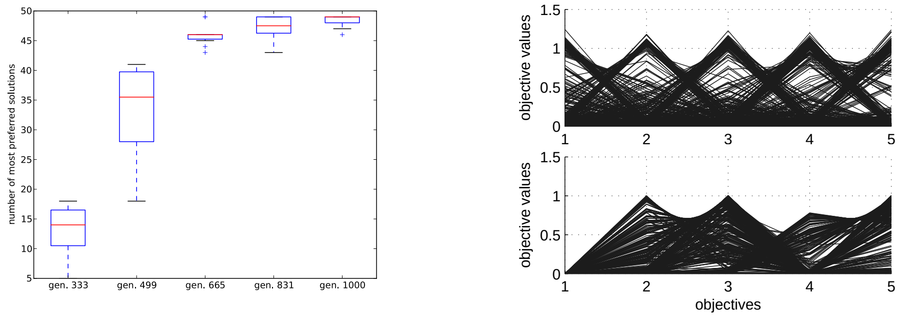

Fig. 3. Results for 10 W-HypE runs on DTLZ2 when the DM articulates comparative preference statements. Left: boxplots of the number of population members that fulfill the comparative preference statements at each interaction as well as after the run. Right: parallel coordinates plots of all solutions in the 10 runs at the first interaction step (generation 333, top) and in the end of the runs (at generation 1000, bottom).

preferred parts of the search space [16]. The weighted hypervolume indicator approach has been shown in [1, 7, 25] to be able to change the optimization goal for a hypervolume based EMO algorithm, which allows to steer the search. Hence, it is straightforward to use this approach also in an interactive manner. In this work, we presented a simple way to incorporate information about the DM's most preferred solutions into the weighted hypervolume based W-HypE algorithm. We also showed its working principles and the usefulness of the interactive approach with experiments on several standard test problems with respect to the proximity of the algorithm's population to the DM's most preferred solution. In comparison with the previously proposed interactive TDEA approach (iTDEA) of [18], the interactive W-HypE algorithm showed comparable or improving results if the algorithm allows to produce solutions close to the Pareto front. For the more complicated ZDT4 problem, the interactive W-HypE algorithm gained results comparable to iTDEA only in about 10% of the runs. Finally, we showed an example of how the most preferred solutions of the DM can be specified indirectly via a set of comparative preference statements—an approach borrowed from the field of artificial intelligence—within the same interactive W-HypE algorithm.

References

[1] Auger A, Bader J, Brockhoff D, Zitzler E (2009) Articulating User Preferences in Many-Objective Problems by Sampling the Weighted Hypervolume. In: Genetic and Evolutionary Computation Conference (GECCO 2009), ACM, pp 555–562

[2] Bader J, Zitzler E (2011) HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. Evol Comput 19(1):45–76

[3] Benferhat S, Dubois D, Kaci S, Prade H (2002) Bipolar representation and fusion of preferences in the possibilistic logic framework. In: KR'02, pp 421–432

[4] Beume N, Naujoks B, Emmerich M (2007) SMS-EMOA: Multiobjective Selection Based on Dominated Hypervolume. Eur J Oper Res 181(3):1653–1669

[5] Bleuler S, Laumanns M, Thiele L, Zitzler E (2003) PISA—A Platform and Programming Language Independent Interface for Search Algorithms. In: Evolutionary Multi-Criterion Optimization (EMO 2003), Springer, pp 494–508

[6] Boutilier C (1994) Toward a logic for qualitative decision theory. In: KR'94, pp 75–86

[7] Brockhoff D, Bader J, Thiele L, Zitzler E (2013) Directed Multiobjective Optimization Based on the Hypervolume Indicator. Journal of Multi-Criteria Decision Analysis 20(5–6):291–317. DOI: 10.1002/mcda.1502

[8] Brockhoff D, Hamadi Y, Kaci S (2012) Interactive Optimization With Weighted Hypervolume Based EMO Algorithms: Preliminary Experiments. Tech. rep., INRIA research report RR-8103

[9] Deb K (2001) Multi-Objective Optimization Using Evolutionary Algorithms. Wiley, Chichester, UK

[10] Deb K, Kumar A (2007) Interactive Evolutionary Multi-Objective Optimization and Decision-Making using Reference Direction Method. In: Genetic and Evolutionary Computation Conference (GECCO 2007), ACM, pp 781–788

[11] Deb K, Thiele L, Laumanns M, Zitzler E (2001) Scalable Test Problems for Evolutionary Multi-Objective Optimization. TIK Report 112, Computer Engineering and Networks Laboratory (TIK), ETH Zurich

[12] Deb K, Sinha A, Korhonen P, Wallenius J (2010) An Interactive Evolutionary Multi-Objective Optimization Method Based on Progressively Approximated Value Functions. IEEE T Evolut Comput 14(5):723–739

[13] Fleischer M (2003) The Measure of Pareto Optima. Applications to Multi-Objective Metaheuristics. In: Evolutionary Multi-Criterion Optimization (EMO 2003), Springer, pp 519–533

[14] Hansson S (2001) The structure of values and norms. In: Cambridge University Press

[15] Igel C, Hansen N, Roth S (2007) Covariance Matrix Adaptation for Multi-objective Optimization. Evol Comput 15(1):1–28

[16] Jaszkiewicz A, Branke J (2008) Interactive Multiobjective Evolutionary Algorithms. In: Multiobjective Optimization: Interactive and Evolutionary Approaches, Springer, pp 179–193

[17] Kaci S (2011) Working with Preferences: Less Is More. Springer,

[18] Köksalan M, Karahan I (2010) An Interactive Territory Defining Evolutionary Algorithm: iTDEA. IEEE T Evol Comput 14(5):702–722

[19] Miettinen K (1999) Nonlinear Multiobjective Optimization. Kluwer, Boston, MA, USA

[20] Phelps S, Köksalan M (2003) An Interactive Evolutionary Metaheuristic for Multiobjective Combinatorial Optimization. Manage Sci 49(12):1726–1738

[21] Tanino T, Tanaka M, Hojo C (1993) An Interactive Multicriteria Decision Making Method by Using a Genetic Algorithm. In: Conference on Systems Science and Systems Engineering, pp 381–386

[22] Thiele L, Miettinen K, Korhonen PK, Molina J (2009) A Preference-Based Interactive Evolutionary Algorithm for Multiobjective Optimization. Evol Comput 17(3):411–436

[23] van der Torre L, Weydert E (2001) Parameters for utilitarian desires in a qualitative decision theory. Applied Intel 14(3):285–301

[24] Zitzler E, Thiele L, Laumanns M, Fonseca CM, Grunert da Fonseca V (2003) Performance Assessment of Multiobjective Optimizers: An Analysis and Review. IEEE T Evol Comput 7(2):117–132

[25] Zitzler E, Brockhoff D, Thiele L (2007) The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration. In: Evolutionary Multi-Criterion Optimization (EMO 2007), Springer, pp 862–876