



# Directed Multiobjective Optimization Based on the Weighted Hypervolume Indicator

Dimo Brockhoff, Johannes Bader, Lothar Thiele, Eckart Zitzler

## ► To cite this version:

Dimo Brockhoff, Johannes Bader, Lothar Thiele, Eckart Zitzler. Directed Multiobjective Optimization Based on the Weighted Hypervolume Indicator. *Journal of Multi-Criteria Decision Analysis*, 2013, Special Issue: Evolutionary Multiobjective Optimization: Methodologies and Applications, 20 (5-6), pp.291-317. 10.1002/mcda.1502 . hal-00920178

**HAL Id: hal-00920178**

**<https://inria.hal.science/hal-00920178>**

Submitted on 5 Jan 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## Directed Multiobjective Optimization Based on the Weighted Hypervolume Indicator

Dimo Brockhoff\*, Johannes Bader<sup>^</sup>, Lothar Thiele<sup>♦</sup>, and Eckart Zitzler<sup>\*</sup>

<sup>•</sup>DOLPHIN team, INRIA Lille - Nord Europe  
Parc scientifique de la Haute Borne, 40 av. Halley - Bât A - Park Plaza, 59650 Villeneuve d'Ascq, France,  
e-mail: dimo.brockhoff@inria.fr

<sup>^</sup>Swiss Federal Archives  
Archivstrasse 24, 3003 Berne, Switzerland,  
e-mail: johannes.bader@bar.admin.ch

<sup>♦</sup>Department of Information Technology and Electrical Engineering,  
Swiss Federal Institute of Technology (ETH) Zurich, Zurich, Switzerland,  
e-mail: lothar.thiele@tik.ee.ethz.ch

<sup>\*</sup>Institute for Continuing Professional Education,  
University of Teacher Education PHBern, Bern, Switzerland,  
e-mail: eckart.zitzler@phbern.ch

### ABSTRACT

Recently, there has been a large interest in set-based evolutionary algorithms for multiobjective optimization. They are based on the definition of indicators that characterize the quality of the current population while being compliant with the concept of Pareto-optimality. It has been shown that the hypervolume indicator which measures the dominated volume in the objective space enables the design of efficient search algorithms and, at the same time, opens up opportunities to express user preferences in the search by means of weight functions. The present paper contains the necessary theoretical foundations and corresponding algorithms to (a) select appropriate weight functions, to (b) transform user preferences into weight functions, and to (c) efficiently evaluate the weighted hypervolume indicator through Monte Carlo sampling. The algorithm **W-HypE** which implements the above concepts is introduced and the effectiveness of the search, directed towards the user's preferred solutions, is shown using an extensive set of experiments including the necessary statistical performance assessment.

**KEY WORDS:** multiobjective optimization, evolutionary algorithm, hypervolume, preference-based search

This is the peer reviewed version of the following article: D. Brockhoff, J. Bader, L. Thiele, and E. Zitzler. "Directed Multiobjective Optimization Based on the Weighted Hypervolume Indicator". In *Journal of Multi-Criteria Decision Analysis* 20(5–6), pages 291–317 (2013), which has been published in final form at <https://dx.doi.org/10.1002/mcda.1502>. This article may be used for non-commercial purposes in accordance with Wiley Terms and Conditions for Use of Self-Archived Versions.

# 1 Introduction

When approximating the Pareto-optimal set in terms of a set of trade-off solutions, one solves a set problem where the search space consists of all possible finite Pareto set approximations. This is the classical scenario that has been studied extensively in the field of evolutionary multiobjective optimization (EMO), and much research has been devoted to the question of how to define the optimization goal for this type of set problem. One possibility is to make use of so-called quality indicators: they assign each Pareto set approximation a real value reflecting its quality and therefore can be used as objective function for the underlying set problem. The hypervolume indicator is one of the most popular quality indicators and in recent years several algorithms have been proposed that directly use the hypervolume values in the selection phases to guide the search. The reason for the popularity of this indicator is its property of being strictly monotonic and thus represents a refinement of Pareto-dominance (Zitzler et al., 2008, 2003). With this property, it is possible to show that a hypervolume-based multiobjective optimizer converges to the Pareto-optimal set in the limit (Zitzler et al., 2010), while many EMO algorithms suffer from cyclic behavior (Wagner et al., 2007), mainly because the refinement condition is not met (Berghammer et al., 2010). As a result, one can observe a growing interest in hypervolume-based multiobjective search, both from a theoretical and a practical perspective, see e.g., (Bader et al., 2010; Beume and Rudolph, 2006; Bradstreet et al., 2009, 2008; Bringmann and Friedrich, 2008, 2009a,b; Emmerich et al., 2005; Fleischer, 2003; Fonseca et al., 2006; Friedrich et al., 2009; Igel et al., 2007; While et al., 2006).

The major challenge in this context is the integration of user preferences to *direct* the search which has gained a recent research interest in order to improve EMO algorithms' search abilities for many-objective problems (Deb and Jain, 2012; Hughes, 2011; Ishibuchi et al., 2008). It has been recently shown that the hypervolume indicator has a natural bias that affects the outcome of the search process (Auger et al., 2009c). In certain situations this bias may be appropriate, while in other situations the decision maker may be more interested in specific re-

gions of the objective space that require a different bias. Therefore, it is desirable to adjust the optimization goal according to the preferences of the user, i.e., to provide flexibility with respect to the search direction that the hypervolume indicator formalizes. A first concept and proof-of-principle results for this issue have been presented in (Zitzler et al., 2007) where weight functions have been introduced to define preference-specific hypervolume indicators. That paper has shown the potential of the weighted hypervolume indicator, but did not contain a general and practically applicable methodology for preference articulation in hypervolume-based search; in particular, it did not address the question of how to deal with problems with more than two objectives. Generalizations of these articulation approaches have been sketched in a previous conference publication (Auger et al., 2009a), but the overall methodology described in the following has been missing so far.

The present paper presents a generalized methodology for preference-directed hypervolume-based multiobjective search. In contrast to previous results, we present a complete picture and in particular show how to concretely use the weighted hypervolume approach in practice. In particular, the paper contains the following new results:

- A general approach to change the bias of the hypervolume indicator is proposed. In particular, a comprehensive toolkit is described consisting of useful classes of elementary weight functions and methods to compose them (Sec. 5). A large set of examples demonstrates how this approach is capable of integrating different types of user preferences, ranging from preference points to stressing objectives.
- It is discussed how to use a preference-specific hypervolume indicator for search by introducing the new algorithm **W-HypE** that relies on Monte Carlo sampling and thereby allows to tackle problems with an arbitrary number of objectives (Sec. 7).
- It is shown that the presented toolkit together with efficient sampling as provided by **W-HypE** allows emulating most relevant classical scalariza-

tion function approaches in a single set-based optimization framework (Sec. 6). In other words, several classical methods to articulate the preferences of a decision maker are transferred to population-based multiobjective search by providing the corresponding weight functions.

- An extensive experimental section discusses the various new concepts by means of visual inspection and statistical comparisons (Sec. 8). Both continuous and discrete scenarios are investigated—showing the generality and effectiveness of the new approach in practice as well as its scalability to many-objective problems.

The power of the methodology is its generality: it not only provides novel ways of preference articulation, but even allows to model existing scalarizing techniques such as weighted sum aggregation and desirability functions and to transfer them to set-based multiobjective optimization. The latter aspect opens new perspectives in joining interactive approaches in the field of multiple criteria decision making with the set-based approach pursued in the EMO field.

## 2 From the Hypervolume to the Weighted Hypervolume

The purpose of this section is to provide the necessary foundations for the new results described in the forthcoming sections. In particular, we will describe a basic indicator-based search algorithm, review the basic requirements for a suitable quality indicator, define the weighted hypervolume indicator and review some of its properties that appear to be relevant for the rest of the paper.

### 2.1 Basic Terms

As usual, we consider the minimization of a vector-valued objective function  $f = (f_1, \dots, f_n) : X \rightarrow \mathbb{R}^n$  where  $X$  denotes the decision space, i.e., the feasible set of alternatives for the optimization problem. The image of the decision space  $X$  using the objective function  $f$  is denoted as the objective space  $Z \subseteq \mathbb{R}^n$  with

$Z = \{f(x) \mid x \in X\}$ . A single alternative  $x \in X$  is sometimes named ‘solution’ and the corresponding objective value  $z = f(x) \in Z$  is named ‘objective vector’.

As we are attempting to minimize simultaneously the components of a vector-valued objective function, we need a preference relation that defines how a solution compares to another one. In this paper, we restrict ourselves to the common notion of Pareto-dominance.

**Definition 1.** A solution  $a \in X$  weakly Pareto-dominates a solution  $b \in X$ , denoted as  $a \leq b$ , if it is as least as good in all objectives, i.e.,  $f_i(a) \leq f_i(b)$  for all  $1 \leq i \leq n$ . Solution  $a$  is better than  $b$  or Pareto-dominating  $b$ , denoted as  $a < b$ , iff  $(a \leq b) \wedge (b \not\leq a)$ .

Equivalently, we can also say that  $a$  is better than  $b$  iff  $f_i(a) \leq f_i(b)$  for all  $1 \leq i \leq n$  and there exists at least one objective  $k$  where  $f_k(a) < f_k(b)$ . A solution is named Pareto-optimal, if there is no other solution in  $X$  that is better. The set of all Pareto-optimal solutions is denoted as the Pareto-optimal set and its image in objective space as the Pareto-optimal front.

In the recently developed class of preference-based optimization algorithms, decisions are based on the fact whether one set of solutions is preferable to another one. Therefore, the above (weak) Pareto-dominance is extended towards populations, i.e., sets of solutions (Zitzler et al., 2003). Naturally, we define a set of solutions  $A$  to weakly dominate another set  $B$  iff every solution in  $B$  is weakly Pareto-dominated by at least one solution in  $A$ .

**Definition 2.** A set of solutions  $A \subseteq X$  weakly dominates a set of solutions  $B \subseteq X$ , denoted as  $A \leq B$ , iff  $(\forall b \in B : (\exists a \in A : a \leq b))$ . Set  $A$  is better than set  $B$ , denoted as  $A < B$ , iff  $(A \leq B) \wedge (B \not\leq A)$ .

Unary set indicators, such as the hypervolume indicator, can now be used to represent the quality of a whole set of solutions by a single scalar value. This way, decisions in search algorithms can be based on quality indicators, i.e., by comparing the quality indicators of sets we determine the most preferred one.

**Definition 3.** A quality indicator function  $I$  maps each set of solutions  $A \subseteq X$  to a real number  $I(A) \in \mathbb{R}$ . It refines the Pareto-dominance iff

$$A < B \Rightarrow (I(A) > I(B))$$

for all sets of solutions  $A, B \subseteq X$ .

The above refinement condition can be interpreted as follows: If a set of solutions  $A$  is better than another set  $B$  according to Def. 2, then the quality indicator should also say so, i.e., it should satisfy  $I(A) > I(B)$ . It has been shown formally in (Zitzler et al., 2010) that a unary quality indicator as defined in Def. 3 (a) defines a total preorder on the set of all solution sets and (b) guarantees that a set with the maximal indicator value is minimal with respect to the set Pareto-dominance relation according to Def. 2. In other words, an algorithm based on such a quality indicator optimizes the objective functions while respecting the weak Pareto-dominance relation on sets.

## 2.2 A Simple Indicator-Based Search Algorithm

Based on the above considerations, we can define a simple indicator-based search algorithm. It is modeled after SPAM (Set Preference Algorithm for Multiobjective Optimization) which has been described in (Zitzler et al., 2010). The purpose in the context of this paper is to start with a simple baseline algorithm that will be refined in Sec. 7 in terms of user preference and search efficiency.

---

### Algorithm 1 SPAM Main Loop

---

```

1: generate initial set of solutions  $P$  of size  $m$ , i.e.,
   randomly choose  $m$  solutions
2: while termination criterion not fulfilled do
3:    $P' \leftarrow \text{heuristicSetMutation}(P)$ 
4:   if  $I(P') \geq I(P)$  then
5:      $P \leftarrow P'$ 
6: return  $P$ 

```

---

Algorithm 1 can be regarded as a simple hill-climber that uses the indicator function  $I$  to decide whether a new population  $P'$  is preferable to the previous one. The *heuristicSetMutation*-operator as described in Algorithm 2 determines such a new population based on the current one  $P$ . Only one possible variant is shown here that starts from  $k$  new individual solutions that are added to the current population  $P$  and removes  $k$  solutions from  $P \cup \{r_1, \dots, r_k\}$  in order to

---

### Algorithm 2 Heuristic Set Mutation

---

```

1: procedure heuristicSetMutation( $P$ )
2:   generate  $k$  solutions  $r_1, \dots, r_k \in X$  based on  $P$ 
3:    $P' \leftarrow P \cup \{r_1, \dots, r_k\}$ 
4:   while  $|P'| > m$  do
5:     for all  $a \in P'$  do
6:        $\delta_a \leftarrow I(P') - I(P' \setminus \{a\})$ 
7:       choose  $p \in P'$  with  $\delta_p = \min_{a \in P'} \delta_a$ 
8:        $P' \leftarrow P' \setminus \{p\}$ 
9:   return  $P'$ 

```

---

achieve a constant population size of  $m$ . In particular, those solutions which lead to the smallest loss in the set-based quality indicator  $I$  are removed one-by-one. Other variants are possible, for example removing in a single step the optimal *set* of  $k$  solutions that leads to the smallest indicator loss, see, e.g., (Bringmann and Friedrich, 2009b), but these subset selection approaches come along with larger computational costs such that the step-by-step procedure is used more often in practice.

Various indicators have been defined to measure the quality of a solution set, see e.g., (Zitzler et al., 2003) for an overview, and the hypervolume indicator and its generalizations are examples of unary indicators that refine Pareto dominance (Zitzler et al., 2010). It has been used both for performance assessment in multi-objective optimization (Zitzler and Thiele, 1998a) as well as to guide the search in various hypervolume-based evolutionary optimizers (Beume et al., 2007; Igel et al., 2007; Knowles et al., 2006; Zitzler et al., 2007; Zitzler and Künzli, 2004). The following section reviews some of its basic properties.

## 2.3 The Weighted Hypervolume Indicator

In its simplest form, the standard hypervolume indicator is evaluating a solution set by assigning the “size of the objective value space which is covered by [the set]” to it (Zitzler and Thiele, 1998b) or in other words the Lebesgue measure of the objective space that is dominated by the set and bounded by a so-called reference point. The left-hand plot of Fig. 1 is illustrating this

for a 3-objective problem.

The weighted hypervolume indicator  $I_H^w(A, R)$  on the other hand is a generalization of this standard hypervolume indicator and represents the *weighted* volume of the objective space weakly dominated by a set of solutions  $A$  with respect to a given reference set  $R$  consisting of one or several reference objective vectors.

**Definition 4.** Let  $A \subseteq X$  be a set of solutions,  $R \subset \mathbb{R}^n$  a set of reference points, and  $w : \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$  a positive weight function. The weighted hypervolume indicator  $I_H^w(A, R)$  of  $A$  with respect to  $R$  is then defined as

$$I_H^w(A, R) = \int_{z \in H(A, R)} w(z) \cdot dz \quad (1)$$

where  $H(A, R)$  is the dominated space of  $A$  regarding  $R$ :

$$H(A, R) = \{z \in \mathbb{R}^n \mid \exists a \in A : \exists r \in R : (f(a) \leq z \leq r)\}.$$

The weight function is supposed to be integrable on any bounded set, i.e.,  $\int_{B(0, \gamma)} w(z) dz < \infty$  for any  $\gamma > 0$ , where  $B(0, \gamma)$  is the open ball centered in 0 and of radius  $\gamma$ .

In other words, we integrate the weight function  $w(z)$  for all points  $z \in \mathbb{R}^n$  that are enclosed between the image of the solutions in objective space  $f(A)$  and the reference set  $R$ , where ‘enclosed’ is interpreted in terms of weak Pareto-dominance. From another perspective, the weighted hypervolume indicator of  $A$  can be seen as the weighted Lebesgue measure  $\lambda_w(H(A, R))$  of the set  $H(A, R)$  where the function  $w(z)$  weights the importance of each point  $z \in H(A, R)$ .

The graphical representation in the right-hand plot of Fig. 1 illustrates the weighted hypervolume  $I_H^w$  for a bi-objective problem and a reference set consisting of one point only. The plot shows the objective values of nine points on the first two axes and the weight function  $w$  on the third axis. The weighted hypervolume indicator  $I_H^w(A, R)$  for the set  $A$  of nine points equals the integral of the weight function over the objective space that is weakly dominated by the set  $A$  and which weakly dominates the reference point  $r = (r_1, r_2)$ .

As has been pointed out in Def. 3, an indicator refines the Pareto-dominance if a better set leads to a larger indicator value. We will prove this property for the weighted hypervolume indicator as defined above and thereby establish its usefulness in the context of preference-based multiobjective optimization algorithms, see, e.g., Algorithm 1.

**Proposition 1.** The weighted hypervolume indicator  $I_H^w(A, R)$  for some set of solutions  $A \subseteq X$  with respect to a reference set  $R \subset \mathbb{R}^n$  and a weight function  $w$  as defined in Def. 4 refines the Pareto-dominance according to Def. 3 if the following conditions are satisfied:

- $\forall x \in X : \exists r \in R : (f(x) < r)$
- $\int_{B(c, \gamma)} w(z) dz > 0$  for any  $c \in H(X, R)$ ,  $\gamma > 0$ , where  $B(c, \gamma)$  is the open ball with radius  $\gamma$  and center  $c$ .

*Proof.* If  $A < B$ , then  $\forall b \in B : \exists a \in A : (f(a) \leq f(b))$ . Therefore, we find from the definition of the dominated space in Def. 4 that  $H(B, R) \subseteq H(A, R)$ . As a result, we can write

$$\begin{aligned} I_H^w(A, R) &= \int_{H(A, R)} w(z) \cdot dz = \\ &= \int_{H(B, R)} w(z) \cdot dz + \int_{D(A, B, R)} w(z) \cdot dz \\ &= I_H^w(B, R) + \int_{D(A, B, R)} w(z) \cdot dz \end{aligned}$$

where  $D(A, B, R) = H(A, R) \setminus H(B, R)$  denotes the difference between the dominance spaces of  $A$  and  $B$ . It remains to be shown that the last integral is strictly positive. Because of the restriction on the weight function (strictly positive integral in any finite volume), we just need to show that  $D(A, B, R)$  has a strictly positive volume, i.e.,  $\int_{D(A, B, R)} dz > 0$ .

As  $A < B$ , there exists  $a^* \in A : \nexists b \in B : f(b) \leq f(a)$ . Now, we can write

$$\begin{aligned} D(A, B, R) &= \{z \mid (\exists a \in A : f(a) \leq z) \wedge \\ &\quad (\nexists b \in B : f(b) \leq z) \\ &\quad \wedge (\exists r \in R : z \leq r)\} \\ &\geq \{z \mid (f(a^*) \leq z) \wedge (\forall b \in B : f(b) \not\leq z) \\ &\quad \wedge (\exists r \in R : z \leq r)\} \end{aligned}$$

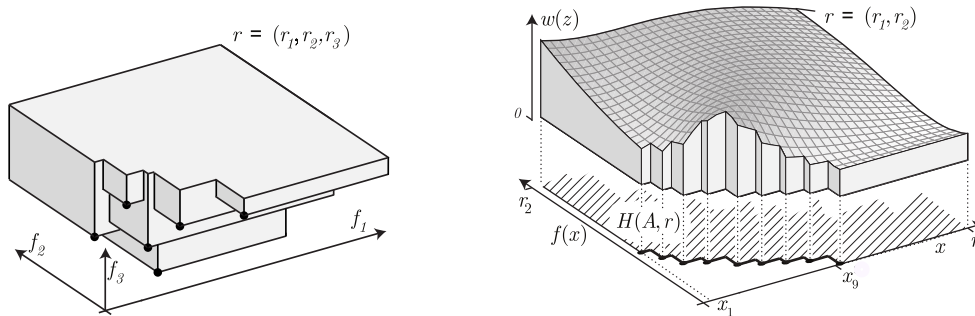


Figure 1: Illustration of the standard hypervolume indicator for a set of 3-objective objective vectors (left) and the weighted hypervolume indicator  $I_H^w(A, \{r\})$  (volume of the gray shape in the right-hand plot) for a set  $A$  of nine solutions (black dots) of a bi-objective problem. The plot shows an example of a weight function  $w(z)$ , where for all objective vectors  $z$  that are not dominated by  $A$  or not enclosed by  $r$  the function  $w$  is not plotted. The plot is taken from Fig. 1 in (Auger et al., 2012) and updated.

In order to satisfy the term  $\forall b \in B : f(b) \not\leq z$  we first select for each  $b$  an index  $k$  such that  $f_k(b) > f_k(a^*)$ . Such an index exists since for  $a^* \in A : \forall b \in B : f(b) \not\leq f(a^*)$ . Then we make sure that  $z_k < f_k(b)$  holds (and therefore  $f(b) \not\leq z$ ) by adding the following constraint for  $z$ :  $z_k < f_k(a^*) + \delta_k$  where  $\delta_k \leq f_k(b) - f_k(a^*)$ . In other words, we determine a vector  $\delta = (\delta_1, \dots, \delta_n) > 0$  by iteratively considering all  $b \in B$ , and for all indices where  $f_k(b) > f_k(a^*)$  holds, we update  $\delta_k$  as  $\delta_k := \min\{\delta_k, f_k(b) - f_k(a^*)\}$ , starting with all  $\delta_k = \infty$ . Then we can write

$$D(A, B, R) \geq \{z \mid (f(a^*) \leq z < f(a^*) + \delta) \wedge (\exists r \in R : z \leq r)\}$$

Because of the first condition in Proposition 1 there exists  $r^* \in R : f(a^*) < r^*$ . Therefore, we can replace the condition  $\exists r \in R : z \leq r$  by  $z < f(a^*) + (r^* - f(a^*))$  where  $(r^* - f(a^*)) > 0$ . If we now replace the previously defined  $\delta$  by  $\delta' := \min\{\delta, r^* - f(a^*)\} > 0$  we obtain

$$D(A, B, R) \geq \{z \mid f(a^*) \leq z < f(a^*) + \delta'\}$$

which is a strictly positive volume.  $\square$

The above property ensures that in preference-based algorithms like Algorithm 1, we are optimizing towards a final population  $P$  that contains Pareto-optimal solutions, i.e., solutions which are not dominated by

any other solution contained in  $X$ . On the other hand, as the size  $m = |P|$  is usually much smaller than the size of the Pareto-optimal set, only a subset of all non-dominated solutions can be in  $P$  at best. Therefore, any indicator that quantifies the quality of a population inevitably introduces some bias, see for example (Auger et al., 2009c) for a discussion about the search bias of the hypervolume indicator. In case of the *weighted* hypervolume indicator, this bias cannot only be controlled but also used to encode user preferences in the search (Auger et al., 2009b). To this end, we need to understand and quantify the relation between the weight function  $w$  and a subset of Pareto-optimal solutions  $P^*$  that has the maximal weighted hypervolume indicator value  $I_H^w(P^*, R)$ .

## 2.4 Weight Function and Preference Information

A thorough characterization of the distribution of Pareto-optimal solutions which—as a set—achieve the maximal weighted hypervolume indicator value has been presented for bi-objective scenarios in (Auger et al., 2009b,c). Main results and findings will be summarized in the following where we restrict ourselves to a bi-objective problem with  $f = (f_1, f_2) : X \rightarrow \mathbb{R}^2$ . Moreover, we suppose that there exists a continuous function  $g(z_1)$  such that any objective vec-

for  $z = (z_1, z_2) = (z_1, g(z_1))$  for  $z_1 \in [z_{\min}, z_{\max}]$  is Pareto-optimal. In other words, the function  $g(z_1)$  together with the interval  $[z_{\min}, z_{\max}]$  describes the Pareto-optimal front.

We are interested in the following question: Let us suppose that a population of Pareto-optimal solutions  $P^*$  has a fixed size of  $m$  and maximizes the hypervolume indicator  $I_H^w$ , i.e., it has the maximal indicator value of all subsets of solutions of size  $m$ . What is the distribution of points on the Pareto-optimal front? To get a closed-form solution, we suppose that the number of points in the subset  $P^*$  approaches infinity and we are interested in the density of points  $\delta_F(z_1)$  on the front, i.e., within a small piece of length  $h$  on the front curve at  $(z_1, g(z_1))$  we find  $m \cdot h \cdot \delta_F(z_1)$  solutions in  $P^*$ . It has been shown in (Auger et al., 2009b) that

$$\delta_F(z_1) \propto \frac{\sqrt{-g'(z_1) \cdot w(z_1, g(z_1))}}{\sqrt{1 + g'(z_1)^2}} \quad (2)$$

where we suppose that  $g(z_1)$  is continuous, differentiable, its derivative  $g'(z_1)$  is continuous as well and  $w(z_1, z_2)$  denotes the weight function of the weighted hypervolume indicator.

When looking at (2) for an unweighted hypervolume indicator, one can notice that the maximal density is obtained if  $g'(z_1) = -1$  and that the density approaches 0 if the slope of the Pareto front approaches 0 or  $-\infty$ . For illustration purposes, Fig. 2 shows the Pareto front shape  $g(z_1)$ , the approximate optimal distribution of 20 points (black dots) obtained by Algorithm 1, and the density  $\delta_F(z_1)$  (hatched area) for the unweighted hypervolume indicator on the continuous test problems DTLZ2 and DTLZ7 (Deb et al., 2005).

Equation (2) characterizes the density  $\delta_F(z_1)$  of points that maximize the weighted hypervolume indicator for a given weight function  $w(z_1, z_2)$  and front shape  $g(z_1)$ . The result can also be interpreted in the opposite direction: given user-defined preference, expressed by a density, the corresponding weight function can be derived. This allows to model user preference in a concise manner by optimizing the weighted hypervolume indicator. Let the desired density of the user be  $\delta_F(z_1)$ , then

$$w(z_1, g(z_1)) \propto \frac{1 + g'(z_1)^2}{-g'(z_1)} \cdot \delta_F(z_1)^2. \quad (3)$$

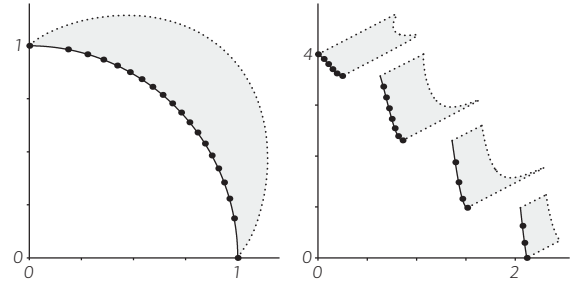


Figure 2: Pareto front shape  $g(z_1)$ , approximate optimal distribution of 20 points (black dots), and the density  $\delta_F(z_1)$  (dotted line, in polar coordinates) for the unweighted hypervolume indicator on the continuous test problems DTLZ2 and DTLZ7.

As an example, Fig. 3 shows the distribution of 50 points obtained using an algorithm similar to Algorithm 1 for two desired densities  $\delta_F(z_1)$ , expressed in polar coordinates (see (Auger et al., 2009b) for details). The resulting density of points comes very close to the desired density, demonstrating that (2) not only serves as a better theoretical understanding of the weighted hypervolume, but also is of practical relevance.

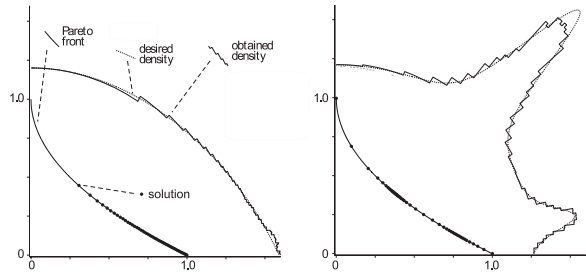


Figure 3: The figure shows 50 solutions (black dots) found by optimizing the weighted hypervolume indicator with weight functions corresponding to two types of desired densities  $\delta$ , according to (3). In addition to the obtained 50 solutions, the corresponding histogram (step-functions) as well as the desired densities (dotted lines) are shown in polar coordinates. The plots are revised versions from (Auger et al., 2009b).

Despite the favorable properties of the weighted hypervolume indicator in terms of preference-based mul-



tiobjective search, two main problems need to be resolved:

- The calculation of the weighted hypervolume indicator is computationally expensive, especially in high dimensions and for general weight functions.
- The step from search preferences towards weight functions that lead to efficiently computable indicators has not been investigated so far.

The following sections will be devoted to answers to both open questions.

### 3 General Considerations on the Choice of the Weight Function

Due to the #P-hardness of the standard hypervolume calculation (Bringmann and Friedrich, 2008), the computation of the exact hypervolume for high dimensions and for a large number of points is already computationally expensive in the non-weighted case, i.e., where  $w = 1$ . Moreover, the calculation for a general weight function  $w$  involves the additional difficulty of computing multi-dimensional integrals as in (1) for which often no closed analytical expressions are known. Although it is sufficient to compute the integral of the weight function in a rectangle for some of the exact algorithms, see (Knowles, 2002; While et al., 2006; Zitzler, 2001), only a few 2-objective weight functions have been proposed for which such integrals can be computed analytically, see (Zitzler et al., 2007). At the same time, it was argued that a generalization to three or more objectives is not straightforward. In addition, the usage in more involved algorithms, such as the asymptotically fastest known algorithm for exact hypervolume calculation (Beume, 2009, for  $n \geq 3$ ), is not straightforward as the integral has to be computed within a geometric shape called *trellis*.

To avoid the above described difficulties, the approximative calculation of the hypervolume by means of Monte Carlo sampling has been proposed, see (Bader et al., 2010; Bader and Zitzler, 2011; Bringmann and Friedrich, 2008; Everson et al., 2002). In its simplest form,  $N$  random objective vectors

$X_1, \dots, X_N$  are drawn uniformly in a sampling (hyper-)box and the sum of all samples which are dominated by a solution set  $A$ , multiplied by the weight and normalized by the overall number of samples, is used as the estimate of the weighted hypervolume indicator  $I_H^w(A, R)$

$$I_H^w(A, R) \approx \frac{1}{N} \sum_{\substack{1 \leq k \leq N \\ X_k \in H(A, R)}} w(X_k) \quad (4)$$

Fig. 4b illustrates the Monte Carlo sampling approach. In principle, any weight function  $w : \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$  can be sampled with (4), but the accuracy of the estimation heavily depends on its particular choice. For example, if the weight function has steep peaks and is low for a large portion of the objective space, most of the uniformly drawn samples have almost no influence on the resulting sum. With Hoeffding's inequality for bounded random variables (Hoeffding, 1963, Eq. 2.6), one can show that the size of a confidence interval for the right-hand side of Eq. 4, given a fixed confidence level  $\alpha$ , is proportional to the supremum of  $w$ .<sup>1</sup>

A different sampling method, that has been first proposed in (Auger et al., 2009a) for the hypervolume computation, leads to an accuracy which is independent of the weight function.<sup>2</sup> The appropriately normalized weight function  $w$  is thereby interpreted as a density function of a probability distribution (weight density function) and sampling is done according to it, see Fig. 4. This will result in a larger number of samples in regions with high weight and fewer samples in regions with a small influence on the weighted hypervolume.

If we denote by  $X^w$  the random variable with probability density function  $w$  and by  $X_1^w, \dots, X_N^w$  its  $N$  independent instantiations, the weighted hypervolume

<sup>1</sup>Since the integrands in Eq. 4 are bounded by  $a_i = 0$  and the supremum  $b_i = w^{\sup}$  of the weight function, the right-hand side of Eq. 2.6 in (Hoeffding, 1963) results in  $\alpha = e^{-2Nt^2/w^{\sup}}$  and thus an interval size of  $2t = w^{\sup} \cdot \sqrt{\frac{2}{N} \ln 1/\alpha}$ .

<sup>2</sup>Where the confidence interval size is  $2t = \sqrt{\frac{2}{N} \ln(1/\alpha)}$  for a given confidence level  $\alpha$  according to Hoeffding's bound.

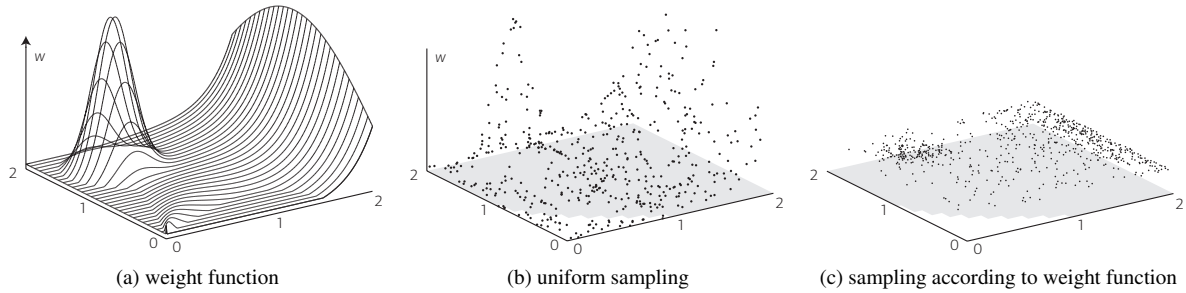


Figure 4: Illustrates the two sampling procedures presented in Section 3 when applied to the weight function shown in (a). In (b), 500 samples are drawn uniformly within  $[0, 0] \times [2, 2]$  and are thereafter multiplied by the corresponding weight. In (c), samples are generated according to the weight function, and do not need to be multiplied by the weight.

$I_H^w(A, R)$  can be approximated by

$$I_H^w(A, R) \approx \frac{1}{N} |\{X_k^w : 1 \leq k \leq N, X_k^w \in H(A, R)\}| \quad (5)$$

Also here, any density function can be used as  $w$  in principle, but the approach highly relies on the fact that it is possible to draw samples according to  $w$  efficiently, like for example if  $w$  is a multivariate normal distribution. Also if the weight function  $w$  is separable and the corresponding cumulative density functions for each objective are invertible,  $w$  can be sampled efficiently. We refer to Devroye (1986) for details as well as for an overview of other distributions that can be sampled efficiently.

For example, if

$$w(x_1, \dots, x_n) = \begin{cases} \frac{10}{3} e^{-\frac{10}{3} x_1} & x \in [0, 2] \times [0, 1.3]^{n-1} \\ 0 & x \notin [0, 2] \times [0, 1.3]^{n-1} \end{cases}$$

as shown in Fig. 5 for the bi-objective case, we can sample  $w$  by independently drawing  $X = (x_1, \dots, x_n)$  uniformly at random within  $[0, 1]^n$  and use the variable transformation  $X^w = (x_1^w, \dots, x_n^w)$  with  $x_1^w = -\frac{3}{10} \ln x_1$  and  $x_k^w = 1.3 \cdot x_k$  for  $2 \leq k \leq n$ , see for example (Devroye, 1986, page 29).

## 4 Simple Weight Functions

In this section, we present several simple weight functions which can be sampled easily and which allow the incorporation of basic user preferences into hypervolume-based search. We will see later on in Sec. 5 how those simple weight functions build the basis of a more general weight function toolkit which makes it possible to formalize even more preference types with the hypervolume indicator, see Sec. 6.

### 4.1 Stressing Objectives with Exponential Weights

Often, a user might want to optimize preferably a single objective  $f_s$  in order to see the possible ranges of this specific objective while other objectives are less important. In other words, the search algorithm should “stress” the importance of good  $f_s$  values in the population. A weight function for such a scenario is therefore supposed to increase for decreasing values of  $f_s$  and have a constant value in direction of the other objectives in order to not introduce additional preferences.

In (Zitzler et al., 2007), an exponential weight function was proposed for this purpose whose marginal distribution for objective  $f_s$  is an exponential distribution with rate parameter  $\lambda$  and whose marginal distributions of the remaining objectives are uniform dis-

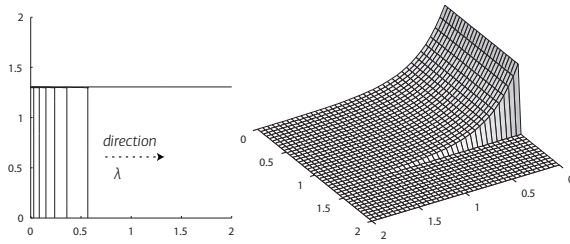


Figure 5: Simple weight functions: objective stressing (exponential).

tributions

$$w(z_1, \dots, z_n) = \begin{cases} \left( \prod_{i \neq s} (b_i^u - b_i^l) \right)^{-1} \lambda e^{-\lambda(z_s - b_s^l)} & z \in B \\ 0 & z \notin B \end{cases}$$

where  $B = [b_1^l, b_1^u] \times \dots \times [b_n^l, b_n^u]$  denotes the space with non-zero probability density. The spread of the distribution is thereby inversely proportional to the parameter  $\lambda$ , i.e., the smaller  $\lambda$ , the steeper the weight function increases at the border of the objective space.

Fig. 5 shows such a weight function for a bi-objective problem when stressing  $f_1$  with an exponential distribution ( $\lambda = 10/3$ ) while using a uniform distribution in the interval  $[0, 1.3]$  in the second objective ( $B = [b_s^l, b_s^u] \times [b_2^l, b_2^u] = [0, \infty] \times [0, 1.3]$ ).

## 4.2 Guiding the Search Towards Preference Points with Normal Distributions

Another way of specifying preferences is to set preference points, see (Wierzbicki, 1999). In brief, a preference point is a user-selected point in objective space that would be *sufficient* for a decision maker, i.e., once a (Pareto-optimal) point dominating the preference point is obtained, the search can be stopped. If the preference point is infeasible, points as close as possible to the preference point should be obtained.

In terms of the weighted hypervolume, the weight  $w(z)$  at a certain point  $z$  in objective space should increase if  $z$  gets closer to the preference point. A multivariate normal distribution with the preference point as its mean is one possibility to articulate preferences towards a preference point and has been presented in

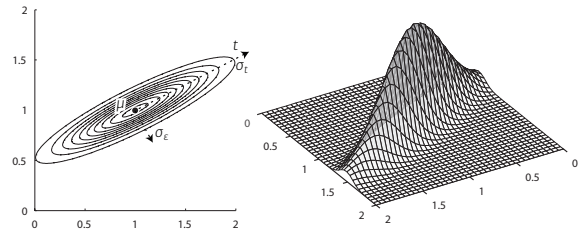


Figure 6: Simple weight functions: preference point (normal distribution).

(Auger et al., 2009a). The multivariate normal distribution has the advantage that we can easily sample points according to this weight function for problems with a high number of objectives.

We denote the preference point as  $m \in \mathbb{R}^n$  and, in addition, define a direction  $t \in \mathbb{R}^n$  as well as two standard deviations  $\sigma_\varepsilon, \sigma_t \in \mathbb{R}$  to articulate preferences towards  $m$  with the weight

$$w_{\text{pref}}(z) = \frac{1}{(2\pi)^{n/2} |C|^{1/2}} e^{-\frac{1}{2}(z-m)^T C^{-1}(z-m)} \quad (6)$$

Here,  $C := \sigma_\varepsilon^2 \mathbf{I} + \sigma_t^2 t t^T / \|t\|^2$  is the covariance matrix of the normal distribution and  $|C|$  its determinant. The equi-density contour lines of such a weight function are ellipsoids whose principal axis are  $t$  or orthogonal to  $t$ , see Fig. 6. The lengths of these axes are determined by the two given standard deviations  $\sigma_t$  and  $\sigma_\varepsilon$ . The variance  $\sigma_t$  influences the range of objective vectors in direction of  $t$  that are affected by the weight function while the variance  $\sigma_\varepsilon$  influences the range of the weight function in direction of the remaining  $n-1$  axes that are perpendicular to  $t$ .

## 4.3 Preference Regions with Uniform Weights

If the search should be concentrated on certain regions of the objective space, it makes sense to use a piecewise constant weight function. A higher (constant) weight is assigned to the preferred region than to the rest of the search space. To be able to sample easily the corresponding distribution, it is useful to restrict the usage of such a uniform distribution to preference

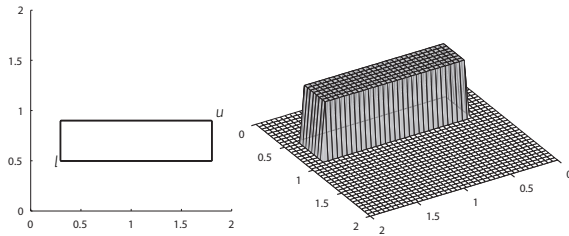


Figure 7: Simple weight functions: uniform.

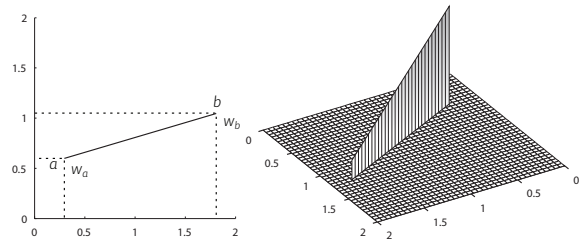


Figure 8: Simple weight functions: ridge function.

regions of rectangular (in bi-objective problems) or (hyper-)cuboidal shapes (in higher dimensions). Fig. 7 shows an example of such a weight function.

#### 4.4 Guiding Single Solutions with Dirac-Type Weights

As a limit case, one can use dirac-type weight functions which have the value 0 almost everywhere but whose integral has a non-zero value, see, e.g. (Zitzler et al., 2007). If such a weight function is a sum of one-dimensional dirac-type functions, then only solutions close to the objective vectors with non-zero weight have a non-zero hypervolume contribution. For example, in case of a one-dimensional dirac-type function, see Fig. 8, only a single solution has a positive hypervolume contribution<sup>3</sup> and therefore, a multiobjective optimization method that uses the corresponding indicator exclusively will lose diversity among the solutions and tends to find a single solution. In order to guide the search towards the preferred *region* in the objective space and to allow for efficient sampling, dirac-type functions should be used together with a smoothing operator as described in Sec. 5.

### 5 A Weight Function Toolkit Allowing Efficient Sampling

In the following, we propose a general weight function toolkit that allows the formalization of user preferences with the weighted hypervolume indicator in an

<sup>3</sup>The ridge with positive weight function can only intersect with one of the pairwise non-dominated sets of objective vectors solely dominated by a single solution.

easy way. Section 6 provides several examples how the weight function toolkit can be employed to articulate classical preference relations with the weighted hypervolume indicator approach.

To illustrate the main components of the weight function toolkit—leading from a user preference to a weight function—we use a simple artificial example of optimizing the design of a noise protection system: the bi-objective problem consists in minimizing the sound pressure  $p$  which can be lowered to 0 with additional costs, see Fig. 9a.

#### 5.1 Transforming the Objective Functions

In the artificial example, the scaling of the pressure may not reflect the intuition of the decision maker. Usually, a logarithmic unit of measurement that expresses the magnitude of sound intensity relative to a reference level is used, namely decibel (dB). This way, the interest of a decision maker will not be focused on a particular fraction of the decision space. Fig. 9b visualizes the rescaling of the first objective. In the context of the desirability function, see Section 6.3, we will discuss the relation between weight functions for the hypervolume indicator and objective space scaling.

#### 5.2 Choosing a Weight Function

The main step when formalizing user preferences in terms of the weighted hypervolume is to choose the underlying weight function. In principle, any weight function can be used here, but according to the discussion in Section 3, we recommend to use a weight function that can be sampled efficiently such as the uniform

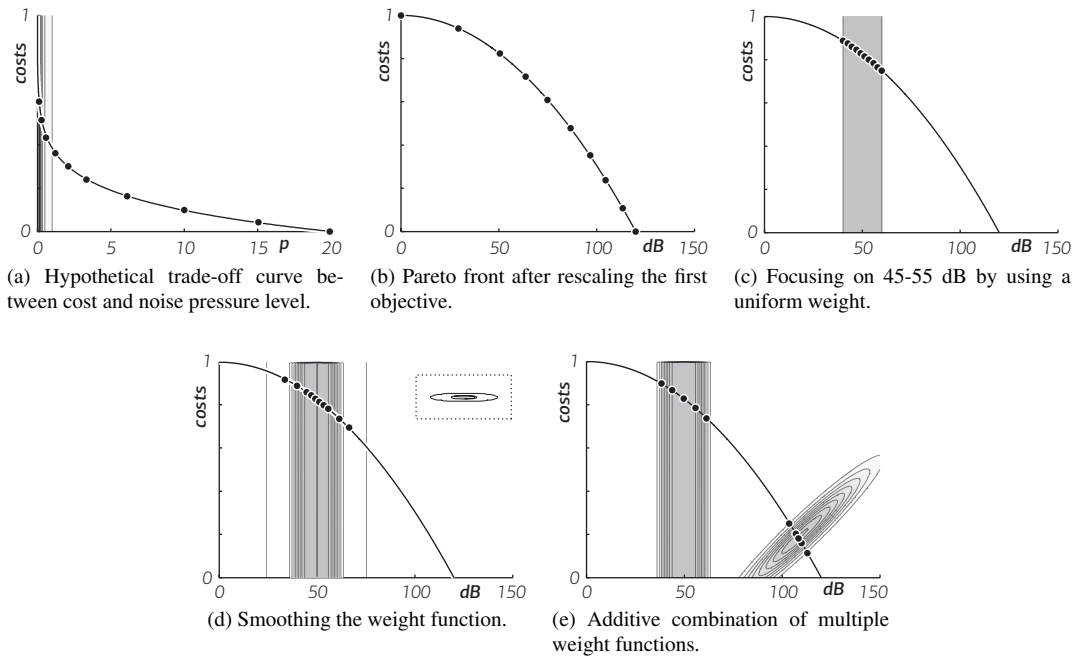


Figure 9: Toolkit example of an artificial noise protection design.

weight in a rectangle in Fig. 9c. As choosing weight functions can be difficult to an unexperienced DM, Sec. 6 will later on propose several example weight functions which simulate classical preference articulation methods with the weighted hypervolume indicator approach.

### 5.3 Smoothing

As has been mentioned in Sections 4.3 and 4.4, uniform or dirac-type weight functions will usually be smoothened in order to (a) guide the optimization towards the preferred region in objective space and (b) in order to get a reasonable number of distinct solutions in the optimized population. This smoothing can be achieved by convolving suitable weight density functions

$$w_{conv}(z) := w_1(z) * w_2(z) \dots * w_q(z) \quad (7)$$

where  $q$  denotes the number of weight functions, and  $w_i(z)$  represents the  $i$ th weight density function. The

convolution operator is defined as

$$(w_i * w_{i+1})(z) = \int_{\mathbb{R}^n} w_i(y) w_{i+1}(z - y) dy \quad (8)$$

Since the convolution operator is associative, (7) can be calculated iteratively.

Although any number of weight functions can be combined by convolution, we focus on the convolution of two weight density functions  $w_{conv} := w_o(z) * w_s(z)$ , where we view the result  $w_{conv}$  as a modified version of  $w_o$ . In other words,  $w_s(z)$  is tailored to “smooth” the original weight function  $w_o(z)$ . This is particularly useful, when  $w_o(z)$  is zero almost everywhere (i.e., the set of objective vectors  $z$  with positive weight is a null set), as it is the case for ridges or the dirac delta. Such dirac-type functions often arise when translating classical methods to hypervolume-based search, such as weighted sum, Tchebycheff,  $\varepsilon$ -constraint, weighted metrics or goal programming, see Section 6. Also within our noise protection system example, the convolution of the uniform weight with a normal distribution smoothenes the distribution

of points found, see Fig. 9d.

Sampling a convoluted weight density function  $w_{conv}$  can be easily performed, see also (Devroye, 1986): if the weight functions  $w_i$  are interpreted as probability densities, then the convolution according to (7) corresponds to the probability density of the sum  $X_1 + \dots + X_q$  of independent random samples  $X_i$  whose respective density is  $w_i$ . In other words, in order to get one sample, one first draws a sample from each of the convoluted densities, and then computes their sum. Note, that any  $w_i$  in (7) can be a linear combination according to (9), and any convoluted weight function  $w_{conv}$  can be used in a linear combination.

## 5.4 Combining Multiple Weight Functions

A wide range of different user preferences can be represented by combining (convolved) weight functions. We here present only one possibility, namely to combine  $q$  weight density functions  $w_1(z), \dots, w_q(z)$  by a linear combination

$$w_{lc}(z) = p_1 w_1(z) + \dots + p_q w_q(z) \quad (9)$$

where the  $p_i$  are positive real numbers that sum up to one, i.e.,  $p_1 + \dots + p_q = 1$ .

In order to sample the weight density function  $w_{lc}(z)$  constructed according to (9), random samples can be generated using the following steps: At first, select a weight function  $i$  by generating a random integer with probability vector  $(p_1, \dots, p_q)$ . Then generate a sample with density  $w_i(z)$ . In other words, we sample each of the densities  $w_i$  independently with probabilities  $p_i$  and take the union of all generated samples.

Figure 9e exemplarily shows the combination of the smoothed uniform weight of Fig. 9d with a normal distribution to additionally obtain solutions close to a preference point.

## 6 Formulating Classical Preference Articulation Approaches with the Weighted Hypervolume Indicator Toolkit

Several classical approaches to formalize user preferences exist. For three examples, namely the Tchebycheff approach,  $\varepsilon$ -constraints, and desirability functions, we show here how those preference models can be integrated within one and the same *set-based* approach in the context of the weighted hypervolume indicator.

### 6.1 Tchebycheff Approach

The *weighted Tchebycheff approach*, see (Miettinen, 1999) for details, consists of specifying a weight  $W_i$  for each objective (with  $\sum_i W_i = 1$ ) and minimizing

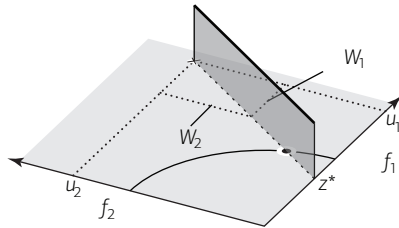
$$\max_{i=1, \dots, n} \{W_i \cdot |f_i(x) - z_i^*|\} \quad (10)$$

where  $z^* = (z_1^*, \dots, z_n^*)$  is denoted as the ideal point and  $x \in X$ .

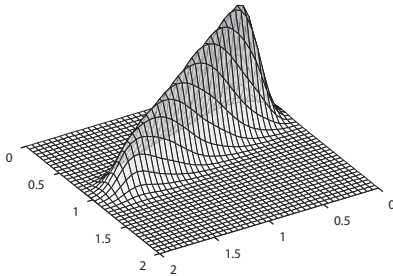
We can articulate the weighted Tchebycheff problem in the weighted hypervolume scenario by using a ridge-type weight function which is non-zero only along the line  $z^* + t \cdot W$  with  $W = (W_1, \dots, W_n)$  and  $t \in \mathbb{R}$  and zero elsewhere, see Section 4.4. Using this weight function in a multiobjective optimizer with hypervolume-based selection directly corresponds to assign to the solution  $z$  with the smallest value in (10) a positive fitness and to all other solutions a fitness of zero. Typically, this approach results in a very low diversity and yields only one non-dominated solution in the final population. In order to obtain a solution set instead of single solutions, we therefore recommend to smooth the above weight function with a normal distribution as described in Section 5 such that all objective vectors in a population have a non-zero influence on the weighted hypervolume indicator, see Fig. 10.

### 6.2 $\varepsilon$ -Constraints

Another classical way of incorporating preferences is to minimize a certain objective  $f_l$  while the solutions



(a) Tchebycheff ridge-type weight function.



(b) Convolved ridge-type weight function.

Figure 10: Illustration of Tchebycheff approach.

are constrained by upper bounds  $\varepsilon_i \in \mathbb{R}$  in all other objectives, see again (Miettinen, 1999) for details. In the weighted hypervolume scenario, such an  $\varepsilon$ -constraint problem can be articulated by a weight function of the form

$$w(z) = \prod_{\substack{1 \leq i \leq n \\ f_i \notin L}} \sigma(\varepsilon_i - z_i) \quad (11)$$

where  $\sigma(x) = 0$  if  $x < 0$  and  $\sigma(x) = 1$  otherwise. The above construction of a weight function yields positive weighted hypervolume values only for solutions that are feasible. In particular, we allow here for optimizing several objectives in a set  $L \subset \{f_1, \dots, f_n\}$  simultaneously while all other objectives  $f_i \notin L$  are constrained to values  $\leq \varepsilon_i$ .

Again, we recommend the smoothing with a normal distribution as described in Section 5 to obtain sufficiently diverse sets of solutions. To disregard infeasible solutions, we recommend to keep the weight function zero if the constraint is not fulfilled by using a *negative* normal distribution as smoothing function.

Although some parts of the objective space will be then assigned a weight function that is zero, the non-dominated sorting in the fitness assignment scheme proposed in Sec. 7 allows that the search can be driven towards the feasible region.

### 6.3 Desirability Functions

Specifying user preferences in terms of *desirability functions* (Harrington, 1965) is usually done by mapping the objective vectors  $z = (z_1, \dots, z_n) \in Z$  via one so-called desirability function  $\varphi_i(z_i)$  per objective to

$$\varphi(z) = (\varphi_1(z_1), \varphi_2(z_2), \dots, \varphi_n(z_n)) \quad (12)$$

and maximizing the scalarization  $s(z) := \prod_{i=1}^n \varphi_i(z)$ . Here, we restrict ourselves to strictly monotonic functions  $\varphi_i : \mathbb{R} \rightarrow [0, 1]$ ,  $f_i(x) \mapsto \varphi_i(f_i(x))$ . Figure 13 gives an example that is later on used in the experiments, but any strictly monotonic function such as proposed in the original work of (Harrington, 1965) can be used.

With respect to this preference model, it is worth to mention the work by Wagner and Trautmann (2010) where an approach is presented in which the objective functions are transformed by means of desirability functions as in Eq. 12 and the algorithm SMS-EMOA is used to optimize the transformed objectives. Since the SMS-EMOA aims at maximizing the (standard) hypervolume indicator, Wagner and Trautmann (2010) argue qualitatively how the transformation of the objectives changes “the shape of the Pareto front in desirability space” and, as a result, how the final distribution of points on the front is affected in terms of the density result in Eq. 2 (Auger et al., 2009c, 2012). In the following, we will see that the transformation of objectives via strictly monotonic desirability functions can also be seen in the context of the weighted hypervolume which allows us to characterize the influence of desirabilities on optimal  $\mu$ -distributions *quantitatively*.

In fact, Theorem 1 presented in Appendix A proves that transforming the objective functions  $f_i(x)$  to  $\varphi_i(f_i(x))$  with strictly monotonically increasing  $\varphi_i$  and using the unweighted hypervolume indicator is equivalent to using the weighted hypervolume indica-



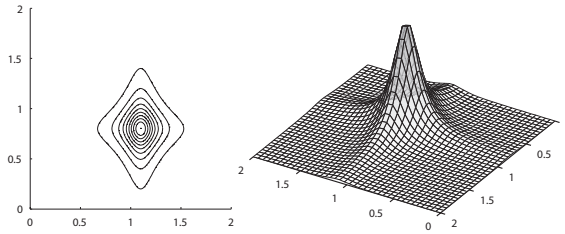


Figure 11: Weight function  $w(z) = \prod_{i=1}^n \frac{b_i}{\pi(1+b_i^2(z_i-a_i)^2)}$  for simulating the desirability function approach with strictly monotone desirability functions as given by Eq. 21 where  $a_1 = 1.1$ ,  $a_2 = 0.8$ ,  $b_1 = 7$ , and  $b_2 = 5$ .

tor without transforming the objectives but with

$$w(z) = \prod_{i=1}^n \varphi'_i(z_i) \quad (13)$$

as weight function. Since the desirability functions are, by definition, to be maximized and we assume minimization of the original objectives, monotonous desirability functions are *decreasing* and the corresponding weight function  $w_d(z)$  for a desirability function  $\varphi(z)$  is therefore  $w_d(z) = \prod_{i=1}^n -\varphi'_i(z_i)$ . Fig. 11 gives an example of such a weight function resulting from a strictly monotonous desirability function defined later on in Eq. 21. The main difference to the traditional approach is here that the weighted hypervolume allows to find a set of solutions instead of only the one solution which maximizes  $s(z)$ .

The new result together with Eq. 2 also allows to better understand desirability functions in the context of hypervolume-based search: not only can the influence of the desirability functions on the optimal distributions of  $\mu$  points be given explicitly but it also follows that, for example, linear scalings of the objectives do not change the optimal density.

## 7 Optimizing the Weighted Hypervolume Indicator

Now, given the weighted hypervolume indicator  $I_H^w(A, R)$  with a specific weight function  $w$  and a spe-

cific reference set  $R$ , the question is how to employ it within a search algorithm. As long as the weight function can be integrated analytically, there is little difference to the original hypervolume indicator: the indicator values can be computed according to the ‘hypervolume by slicing objectives’ principle (Emmerich and Fonseca, 2011; While et al., 2006) where the dominated portion of the objective space is split into hyperrectangles the volumes of which are summed up, see (Bader and Zitzler, 2011); for each hyperrectangle, now, the integral of the weight function over this hyperrectangle replaces the original plain volume. In this case, the hypervolume calculations are not expensive if the number of objectives is low (Emmerich and Fonseca, 2011; Fonseca et al., 2006) and the simple indicator-based search algorithm presented in Sec. 2.2 can be used for searching for a Pareto set approximation with maximum  $I_H^w(A, R)$  value—or any other regular hypervolume-based search technique, e.g., (Emmerich et al., 2005; Igel et al., 2007).

From a practical point of view, though, a more flexible scheme is desirable as the discussions in Sec. 3 indicate; weight functions that are useful for preference articulation can often not be integrated analytically. Furthermore, the exact computation of the indicator values restricts the applicability of the search engine to problems with few objectives only, cf. (Bringmann and Friedrich, 2008). What we present in the following is an approach that addresses both issues simultaneously. The idea is to estimate the weighted hypervolumes by means of Monte Carlo sampling instead of computing them exactly; thereby high dimensional objective spaces as well as arbitrary weight functions become feasible. The algorithm, **W-HypE**, which is presented in the following, is an extension of the **Hypervolume Estimation Algorithm for Multiobjective Optimization (HypE)** described in (Bader and Zitzler, 2011) for the weighted hypervolume indicator. The main loop of **W-HypE** corresponds to Alg. 1. However, the heuristic set mutation procedure of Alg. 2 is replaced by the one outlined in Alg. 3 which differ in two respects:

1. **W-HypE** employs a non-dominated sorting (Goldberg, 1989; Srinivas and Deb, 1994) first such that the hypervolume-based selection only needs to be carried out for the last front that



**Algorithm 3** Improved Heuristic Set Mutation

---

```

1: procedure improvedHeuristicSetMutation(P)
2:   generate  $k$  solutions  $r_1, \dots, r_k \in X$  based on P
3:    $P' \leftarrow P \cup \{r_1, \dots, r_k\}$ 
4:    $P'' \leftarrow \emptyset$ 
5:    $P''' \leftarrow \emptyset$ 
6:   /* perform non-dominated sorting first */
7:   repeat
8:      $P'' \leftarrow P'' \cup P'$ 
9:      $P''' \leftarrow \{a \in P'; \nexists b \in P' : b < a\}$ 
10:     $P' \leftarrow P' \setminus P'''$ 
11:   until  $|P''| + |P'''| \geq m$ 
12:   /* shrink last front using hypervolume indicator */
13:   while  $|P''| + |P'''| > m$  do
14:     compute fitness estimate  $\hat{\delta}_p$  for each  $p \in P'''$ 
15:     choose  $p \in P'''$  with  $\hat{\delta}_p = \min_{a \in P'''} \hat{\delta}_a$ 
16:      $P''' \leftarrow P''' \setminus \{p\}$ 
17:   return  $P'' \cup P'''$ 

```

---

not completely fits into the new population—this scheme is used by most hypervolume-based search algorithms and ensures a finer grained ranking than the simple scheme in Alg. 2;

2. the fitness  $\delta_p$  of an individual  $p$  in **W-HypE** is (i) determined slightly differently and (ii) estimated and not calculated exactly.

We first describe the fitness assignment scheme of **W-HypE**, before we discuss how the fitness values can be estimated using Monte Carlo sampling. As mentioned in the introduction, the focus lies here on presenting the complete picture of **W-HypE** which combines the same fitness assignment scheme of **HypE** (Bader and Zitzler, 2011) with the hypervolume sampling ideas of (Bader et al., 2010) and (Auger et al., 2009a).

## 7.1 Fitness Assignment Scheme

Most hypervolume-based multiobjective optimizers use the loss of hypervolume as a fitness measure to assess the importance of an individual  $p$  in the popu-

lation  $P$ . More precisely, the fitness  $\delta_p$  of  $p$  is computed as  $I_H^w(P, R) - I_H^w(P \setminus \{p\}, R)$  which graphically can be interpreted as removing the hyperrectangle  $H(\{p\}, P, R)$  from the dominated area, see Fig. 12 for an example; here the hyperrectangle

$$H(A, P, R) := H(A, R) \setminus H(P \setminus A, R) \quad (14)$$

represents the portion of the objective space that is jointly weakly dominated by the solutions in a solution set  $A \subseteq P$  and not weakly dominated by any other solution in  $P$ .

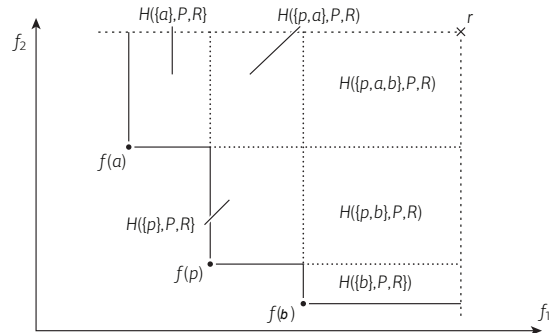


Figure 12: Illustration of how the dominated space is partitioned into hyperrectangles. The population  $P$  contains three individuals  $p$ ,  $a$ , and  $b$ , and the reference set  $R$  consists of a single reference point  $r$ .

As long as only a single solution needs to be removed from the current front in Steps 13 to 16 in Alg. 3, this fitness scheme reflects the optimal choice. However, if several solutions are to be selected for removal (one by one), then the importance of an individual also depends on the other individuals that are deleted. Consider for instance the case that  $p$  and afterwards  $a$  are removed from the population; this means the hyperrectangles  $H(\{p\}, P, R)$ ,  $H(\{a\}, P, R)$ , and  $H(\{p, a\}, P, R)$  vanish. Overall, the hypervolume is reduced by the sum of the volumes of these three hyperrectangles where  $\lambda_w(H(\{p\}, P, R))$  can be attributed to  $p$ ,  $\lambda_w(H(\{a\}, P, R))$  to  $a$ , and  $\lambda_w(H(\{p, a\}, P, R))$  half to  $p$  and half to  $a$  (because if either is kept the hyperrectangle will still be part of the dominated space).

These considerations lead to the idea of computing the *expected* loss in hypervolume that can be attributed

to an individual  $p$ , if  $p$  together with  $o - 1$  other solutions in  $P$  is removed (Bader and Zitzler, 2011). First, generalizing the above concept for a given subset  $A$  with  $|A| = o$  and  $p \in A$ , the hypervolume loss  $\delta_p^A$  attributed to  $p$  when  $A$  is removed amounts to:

$$\delta_p^A = \sum_{B \subseteq A, p \in B} \frac{1}{|B|} \lambda_w(H(B, P, R)) . \quad (15)$$

Certainly at Step 16 in Alg. 3 it is not clear which further  $o - 1$  solutions are chosen for removal in the subsequent steps, that means  $A$  is unknown. Therefore, we only can approximate the true  $\delta_p^A$  by assuming that the  $o - 1$  other solutions are chosen uniformly at random and take the average over all possible sets  $A$ :

$$\hat{\delta}_p = \frac{1}{\binom{|P|-1}{o-1}} \sum_{A \subseteq P, |A|=o, p \in A} \delta_p^A . \quad (16)$$

The value  $\hat{\delta}_p$  is considered to be the fitness of  $p$  and gives the expected hypervolume loss that can be attributed to  $p$  when  $p$  and  $o - 1$  uniformly randomly chosen solutions from  $P$  are removed from  $P$ . The formula can be simplified, cf. (Bader and Zitzler, 2011), such that the actual fitness calculation can be carried out along with a regular 'hypervolume by slicing objectives' computation:

$$\hat{\delta}_p = \sum_{i=1}^o \frac{\alpha_i}{i} \sum_{A \subseteq P, |A|=i, p \in A} \lambda_w(H(A, P, R)) \quad (17)$$

where  $\alpha_i := \prod_{j=1}^{i-1} \frac{o-j}{|P|-j}$ . As demonstrated in (Bader and Zitzler, 2011), this fitness scheme has not only advantages regarding the regular hypervolume-based fitness (i.e.,  $\delta_p = \lambda_w(H(\{p\}, P, R))$ ), but is useful in particular in the context of sampling.

## 7.2 Fitness Estimation By Hypervolume Sampling

Although the fitness values defined above can be computed exactly, see (Bader and Zitzler, 2011), a sampling approach, in which  $\hat{\delta}_p$  is only approximated, allows to circumvent the running time complexity of the exact hypervolume computation which grows exponentially with the number of objectives (Bringmann

and Friedrich, 2008) and to tackle an arbitrary number of objectives efficiently. To this end, first a sampling box  $S \subset Z$  needs to be defined such that it (i) contains both the image of the population in the objective space and the reference set and (ii) is as small as possible. We here use the following definition:

$$S := \{(z_1, \dots, z_n) \in Z \mid \forall 1 \leq i \leq n : l_i \leq z_i \leq u_i\} \quad (18)$$

where

$$l_i := \min_{a \in P} f_i(a) \quad u_i := \max_{(r_1, \dots, r_n) \in R} r_i \quad (19)$$

for  $1 \leq i \leq n$ ; hence, the volume  $V$  of the sampling space  $S$  is given by  $V = \prod_{i=1}^n \max\{0, u_i - l_i\}$ .

As discussed in Section 3, the idea now is to randomly draw samples from  $S$  and count, roughly speaking, for each hyperrectangle  $H(A, P, R)$  how many samples are hits, i.e., inside  $H(A, P, R)$ , and how many are misses, i.e., outside. Thereby, the number of hits divided by the number of samples provides an estimate  $\hat{\lambda}(H(A, P, R))$  for the ratio of the volume of  $H(A, P, R)$  and  $V$  in the unweighted case, i.e., if  $w(z) = 1$  for all  $z \in Z$ . In the general case, we propose to use sampling according to the weight function in order to determine  $\hat{\lambda}_w(H(A, P, R))$ , see Section 3.

For fitness estimation, though, it is not necessary to explicitly determine the  $\hat{\lambda}_w(H(A, P, R))$  values for all hyperrectangles  $H(A, P, R)$ . Instead, for each sampling point  $Z^j$  the fitness estimates of all individuals can be updated directly. First, the set  $A$  of all solutions weakly dominating  $Z^j$  is determined, implying that  $Z^j$  is a hit regarding  $H(A, P, R)$  (and only regarding  $H(A, P, R)$ ). Then, for each individual  $p \in A$  the fitness estimate  $\hat{\delta}_p$  is updated as follows:

$$\hat{\delta}_p = \hat{\delta}_p + \frac{\alpha|A|}{|A|} \cdot \frac{V}{M} \quad (20)$$

provided that  $H(A, P, R)$  is a relevant partition, i.e.,  $A$  lies not beyond the reference set and does not contain more elements than the number of solutions to be removed from  $P'''$ . The full fitness estimation procedure, which details Step 14 in Alg. 3, is given by Alg. 4.

**Algorithm 4** Fitness Estimation

---

```

1: /* determine sampling box  $S$  */
2: for  $i \leftarrow 1, n$  do
3:    $l_i \leftarrow \min_{p \in P'''} f_i(p)$ 
4:    $u_i \leftarrow \max_{(r_1, \dots, r_n) \in R} r_i$ 
5:  $S \leftarrow [l_1, u_1] \times \dots \times [l_n, u_n]$ 
6:  $V \leftarrow \prod_{i=1}^n \max\{0, (u_i - l_i)\}$ 
7: /* initialize fitness estimates */
8: for  $p \in P'''$  do
9:    $\hat{\delta}_p \leftarrow 0$ 
10: /* calc. number of solutions to be removed from  $P'''$  */
11:  $o \leftarrow |P''| + |P'''| - m$ 
12: /* perform sampling */
13: for  $j \leftarrow 1, M$  do
14:   choose  $Z^j \in S$  at random according to  $w$ 
15:   if  $\exists r \in R: Z_i^j \leq r_i$  for all  $1 \leq i \leq n$  then
16:      $A \leftarrow \{p \in P'''; f_i(p) \leq X_i^j \text{ for all } 1 \leq i \leq n\}$ 
17:     if  $|A| \leq o$  then
18:        $\alpha \leftarrow \prod_{l=1}^{|A|-1} \frac{o-l}{|P'''|-l}$ 
19:       for  $p \in A$  do
20:          $\hat{\delta}_p \leftarrow \hat{\delta}_p + \frac{\alpha}{|A|} \cdot \frac{V}{M}$ 

```

---

## 8 Experimental Validation

In the following, instances of the different methods to articulate user preference by a weighted hypervolume presented in Sections 4, 5 and 6 are investigated. Thereby, the algorithmic framework presented in Section 7 is used. All algorithms are applied to test problems with different numbers of objectives to explore the crucial questions, whether (a) optimizing the respective weight function leads to the desired distribution of solutions, and (b) whether using the weight-specific algorithm **W-HypE** is advantageous compared to using a general EMO algorithm such as NSGA-II, or compared to using **W-HypE** with a different weight function. To this end, we apply the following two techniques:

1. The obtained Pareto front approximations are shown visually for one representative run. This mainly serves to illustrate the preference the

weight function is expressing.

2. For a large number of runs, the hypervolume of all Pareto front approximations is calculated with respect to all weight functions used in this study. By statistical analyses it is then tested, whether the differences in hypervolume are statistically significant.

### 8.1 Experimental Setup

#### 8.1.1 Compared Weight Functions and Reference Algorithms

In the following, the algorithm **W-HypE** with the following weight functions is investigated (the parameters used are listed in Table 1):

- $w_1$ : An exponential distribution to stress one of the objectives, according to Sec. 4.1.
- $w_2$ : A multivariate normal distribution according to Sec. 4.2 to stress one preference point  $m$ . For the bi-objective example, the preference point can be overachieved, while for the higher dimensions  $m$  is infeasible.
- $w_3$ : A distribution that mimics a Tchebycheff scalarization. The resulting Dirac ridge is set to linearly decrease to 0 towards the reference point. As smoothing function, a symmetric normal distribution is used.
- $w_4$ : The weight function that corresponds to the desirability function, see Fig. 13:

$$\varphi(z) = \frac{1}{2} - \frac{\arctan(b(z-a))}{\pi} \quad (21)$$

This desirability function mimics a preference to achieve the objective at  $a$ , where  $b$  determines the specificity of the preference.

- $w_5$ : A uniform distribution overlapping with a small portion of the Pareto front.
- $w_6$ : A Dirac ridge (for 2d), and uniform distribution (for 3d, 7d) respectively to mimic the  $\varepsilon$ -constraint. A negative-only normal distribution smoothens the uniform by convolution.

NSGA-II (Deb et al., 2000), SPEA2 (Zitzler et al., 2001), and IBEA (Zitzler and Künzli, 2004) serve as reference algorithms; for the latter, the  $\varepsilon$ -indicator has been used since preliminary experiments showed

Table 1: Parameters of the weight functions used in the ZDT and DTLZ experiments.

wdf	purpose	type	section	parameters
$w_1$	objective stressing	simple	4.1	$\lambda = 0.1, b^l = \mathbf{0}, b^u = \mathbf{5}, s_{2d} = s_{3d} = 1, s_{7d} = 3$
$w_2$	preference point	simple	4.2	$m_{2d} = (.7, .3), m_{3d} = (.2, .4, .8), m_{7d} = (.5, .4, .2, .1, .3, 0, .3), t = \mathbf{1}, \sigma_\varepsilon = 0.05, \sigma_t = 0.5$
$w_3$	Tchebycheff	classical	6.1	Dirac Ridge: $z_{2d}^* = (.4, 0), W_{2d} = (.423, .577); z_{3d}^* = (.2, .4, .8), W_{3d} = (.419, .355, .226); z_{7d}^* = (.5, .4, .2, .1, .3, 0, .3), W_{7d} = (.116, .126, .149, .161, .138, .172, .138);$ and $u = \mathbf{1.5}, w_s = 1, w_e = 0$ . Smoothing with normal distribution: $\sigma = 0.05$
$w_4$	desirability	classical	6.3	$a_{2d} = (.7, .3), b_{2d} = (10, 5); a_{3d} = (.2, .4, .8), b_{3d} = (20, 20, 20); a_{7d} = (.6, .5, .3, .2, .4, .1, .4), b_{7d} = (20, \dots, 20)$
$w_5$	preference region	simple	4.3	$l_{2d} = (.5, .1), l_{3d} = (.2, .4, .8), l_{7d} = (.5, .4, .2, .1, .3, 0, .3), u = \mathbf{1}$
$w_6$	$\varepsilon$ -constraint	classical	6.2	uniform weight: $l_{2d} = (.7, 0), l_{3d} = (.7, 0, 0), l_{7d}^1 = (.6, 0, 0, 0, 0, 0, 0), l_{7d}^2 = (0, 0, .6, 0, 0, 0, 0), l_{7d}^3 = (0, 0, 0, 0, .6, 0, 0), l_{7d}^4 = (0, 0, 0, 0, 0, 0, .6)$ ( $\alpha = 0.25$ each), $u$ corresponds to $l$ , except that 0 are replaced by 2. Smoothing with negative normal distribution: $\sigma_{2d} = 0.1, \sigma_{3d}, \sigma_{7d} = 0.02$

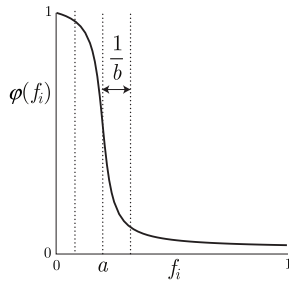


Figure 13: The desirability function according to (21) used in this study.

this variant to be superior to the one using the hypervolume indicator. The main purpose of comparing **W-HypE** against these standard algorithms is investigating the specificity of **W-HypE**, not showing a general superiority: if our concept of preference integration is reasonable, then none of the reference algorithms should provide better Pareto set approximations than **W-HypE** with respect to the preference considered.

The parameters of IBEA are set as  $\kappa = 0.05$  and

$\rho = 1.1$ . All algorithms are run for 100 generations. New individuals are generated by the SBX crossover operator with  $\eta_c = 15$  and by normally distributed mutation with standard deviation  $\sigma = 1/20$  (Deb, 2001). The crossover and mutation probabilities are set to 1 and 0.2 respectively.

### 8.1.2 Details on W-HypE and Test Problems

To optimize according to the weight functions listed in Sec. 8.1.1, **W-HypE** as presented in Sec. 7 is used. Mating selection is performed randomly, while environmental selection uses 10 000 samples to estimate the fitness values  $\delta_a$  according to (20). To substantially speed-up the algorithms, the removal of solutions occurs in a single operation on problems with more than two objectives, that is, without reestimating the fitness values after each removal step as in the greedy variant of **W-HypE** used on bi-objective problems. Samples are generated using MATLAB<sup>®</sup> partly built-in functions, but also user-defined functions.

For the bi-objective test problems, the population size  $\alpha$  and the number of offspring  $\lambda$  is set to 25. As test problem, ZDT1 (Zitzler et al., 2000) is used. For

three and more objectives,  $\alpha$  and  $\lambda$  are doubled to 50 and as test problem, DTLZ2 (Deb et al., 2005) is employed. For both test problems, the number of decision variables is 20.

### 8.1.3 Statistical Method

For a concise comparison of the methods with respect to the different weight functions involved, the following experiment is carried out: for a given number of objectives, 2, 3 or 7,  $L = 50$  runs are performed for all  $k = 9$  algorithms listed in Sec. 8.1.1. The  $k \cdot L = 450$  Pareto front approximations are then evaluated with respect to all  $b = 6$  weight functions by the following procedure: first, 100 000 samples are generated according to the examined weight function. Using these samples, the hypervolume of all runs is approximated, leading to  $b \cdot k \cdot L = 2700$  hypervolume values  $h_{ijl}$ .

For each pair of algorithm  $A_j$ ,  $1 \leq j \leq k$  and weight function  $w_i$ ,  $1 \leq i \leq b$ , the mean of all  $L$  hypervolume values is listed in a table. To simplify the presentation, the hypervolume values are normalized for each weight function  $w_i$ , such that for algorithm  $A_j$  and weight  $w_i$

$$\bar{h}_{ij} := \frac{\left(\frac{1}{L} \sum_{l=1}^L h_{ijl}\right) - \min_{jl} h_{ijl}}{\max_{jl} h_{ijl} - \min_{jl} h_{ijl}} \quad (22)$$

is reported. To test, whether a significant influence of the weight and the algorithm used exists on the hypervolume values  $h_{ijl}$ , the Scheirer-Ray-Hare (SRH) test (Scheirer et al., 1976) is used as a non-parametric version of two-way ANOVA. This test is based on ranks, extending Kruskal-Wallis to multiple factors. For all SRH tests, both the influence of the weight function and the algorithm, as well as the interaction thereof, is highly significant. The latter means that the reported hypervolume means  $\bar{h}_{ij}$  must be examined for each weight function—this is where the nature and direction of the interaction can be found. Therefore, a post-hoc multiple comparison is performed to see which differences in performance are significant for a fixed weight function. To this end, the Conover-Inman post hoc test with a significance level of 1% according to (Conover, 1999) is carried out. To display the effect size of the difference too, the mean rank of the algorithms is reported as well, normalized to 0 (achieving

the best possible ranks 1 to 50) to 1 (reaching the worst ranks  $8 \cdot 50 + 1$  to  $9 \cdot 50$ ).

## 8.2 Results

### 8.2.1 Visual Inspection

The resulting populations after 100 generations are shown in Fig. 14 for the bi-objective ZDT1 problem, in Fig. 15 for the 3-objective DTLZ2 problem, and in Fig. 16 for the 7-objective DTLZ2 problem exemplary for one run. The weight functions are indicated by contour lines at the intervals of 10% of the maximum value that arises. The contour lines do not reflect the actual weight but only the relative distribution thereof. Additionally, a gray shading indicates the weight (darker colors meaning larger weight). As we tested multiple runs for each test case that led to similar results, we display only one run to illustrate the influence of the weight on the distribution of points.

As expected, we can see that **W-HypE** focused the search on regions where the weight is large. In particular, **W-HypE** allows to minimize certain objectives (a), focus the search towards preference points (b), along the direction given by the weights of the Tchebycheff approach (c), and towards points with higher values of the desirability functions (d). It also allows to focus on preference regions (e) and the resulting solutions can meet the desired constraints in the  $\varepsilon$ -constraint approach (f). In contrast, the reference algorithms IBEA, NSGA-II, and SPEA2 show more or less diverse sets of points which cannot be influenced directly by the preferences of the user. It is important to point out that, in comparison to the classical preference approaches, **W-HypE** always offers a *set* of solutions—allowing a decision maker to gain additional information about the local shape of the Pareto front close to the points of interest within one run of the algorithm.

### 8.2.2 Test Statistics

The mean hypervolume values as well as the mean ranks from the Conover-Inman tests for each combination of weight  $w_i$  and algorithm  $A_j$  are shown in Table 2a for  $n = 2$  objectives, Table 2b for  $n = 3$ , and Table 2c for  $n = 7$  respectively.

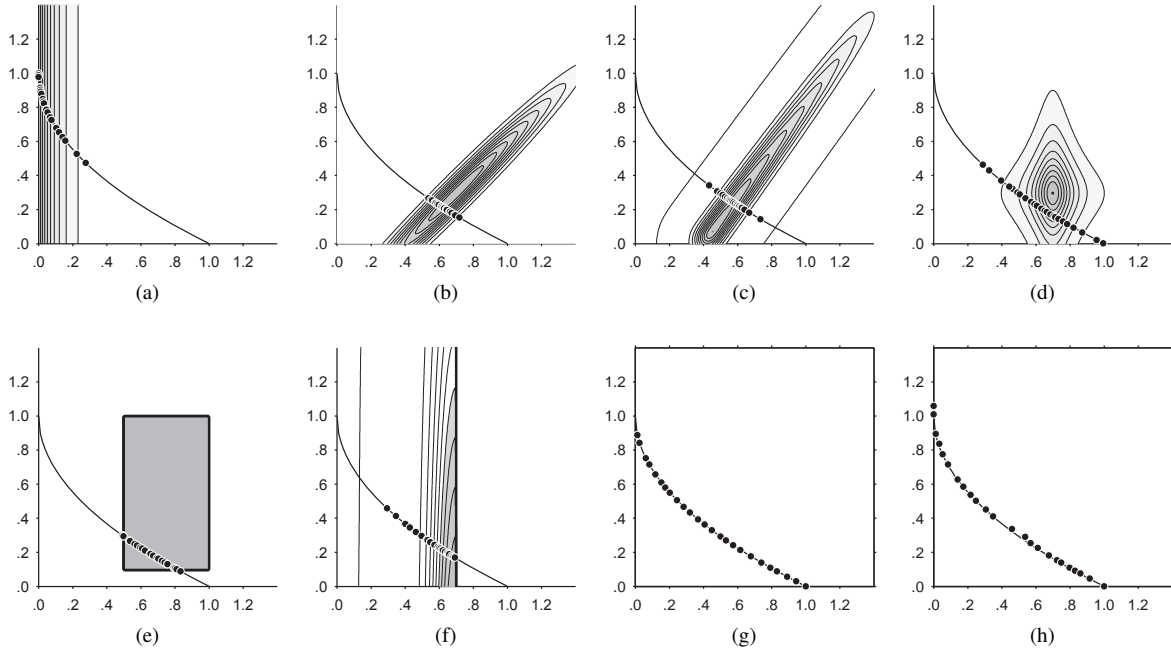


Figure 14: Pareto front approximations on bi-objective ZDT1 (front shown as solid line) for the following weight functions: (a)  $w_1$  (exponential), (b)  $w_2$  (preference point), (c)  $w_3$  (Tchebycheff), (d)  $w_4$  (desirability function), (e)  $w_5$  (uniform region), (f)  $w_6$  ( $\varepsilon$ -constraint), (g) IBEA, (h) NSGA-II. The employed weight functions are shown as contour lines and as gray shading (larger weight in darker regions).

We can observe that, for a given weight function, **W-HypE** optimizing this weight function is consistently resulting in the highest hypervolume values. The only exception where **W-HypE** optimizing the desired weight function is not statistically significantly outperforming all other algorithms is for the 7-objective DTLZ2 problem and the weight function  $w_1$  stressing the third objective. However, the only algorithm reaching higher hypervolume values in this case is IBEA which is known to accumulate solutions close to the boundaries of the Pareto front (Li et al., 2011), resulting in high weighted hypervolume values when the exponential weight  $w_1$  is employed.

### 8.2.3 Runtimes

It has to be remarked that, though the actual runtime of calculating the hypervolume indicator exactly is ex-

pected to increase exponentially with the number of objectives (Beume, 2009; Bringmann and Friedrich, 2008), the Monte Carlo sampling within **W-HypE** makes it feasible to solve problems with a reasonable number of objectives. With the current implementation<sup>4</sup>, the 7-objective runs presented here take on average between 1,67 seconds (for  $w_2$ ) and 4,22 seconds (for  $w_6$ ) per generation on an Intel Core 2 Duo laptop with 2.8GHz, 4GB of RAM, and Windows Vista. Other studies on integrating Monte Carlo sampling into steady-state algorithms such as the SMS-EMOA or the  $(\mu + 1)$ -MO-CMA-ES, which employ the standard hypervolume indicator, report comparable runtimes per function evaluation for the same population size of 50 (Voß et al., 2010). When **W-HypE** is, for example, run with a Gaussian weight function

<sup>4</sup>Combined Java/MATLAB code, available for download at <http://hypervolume.gforge.inria.fr/>.

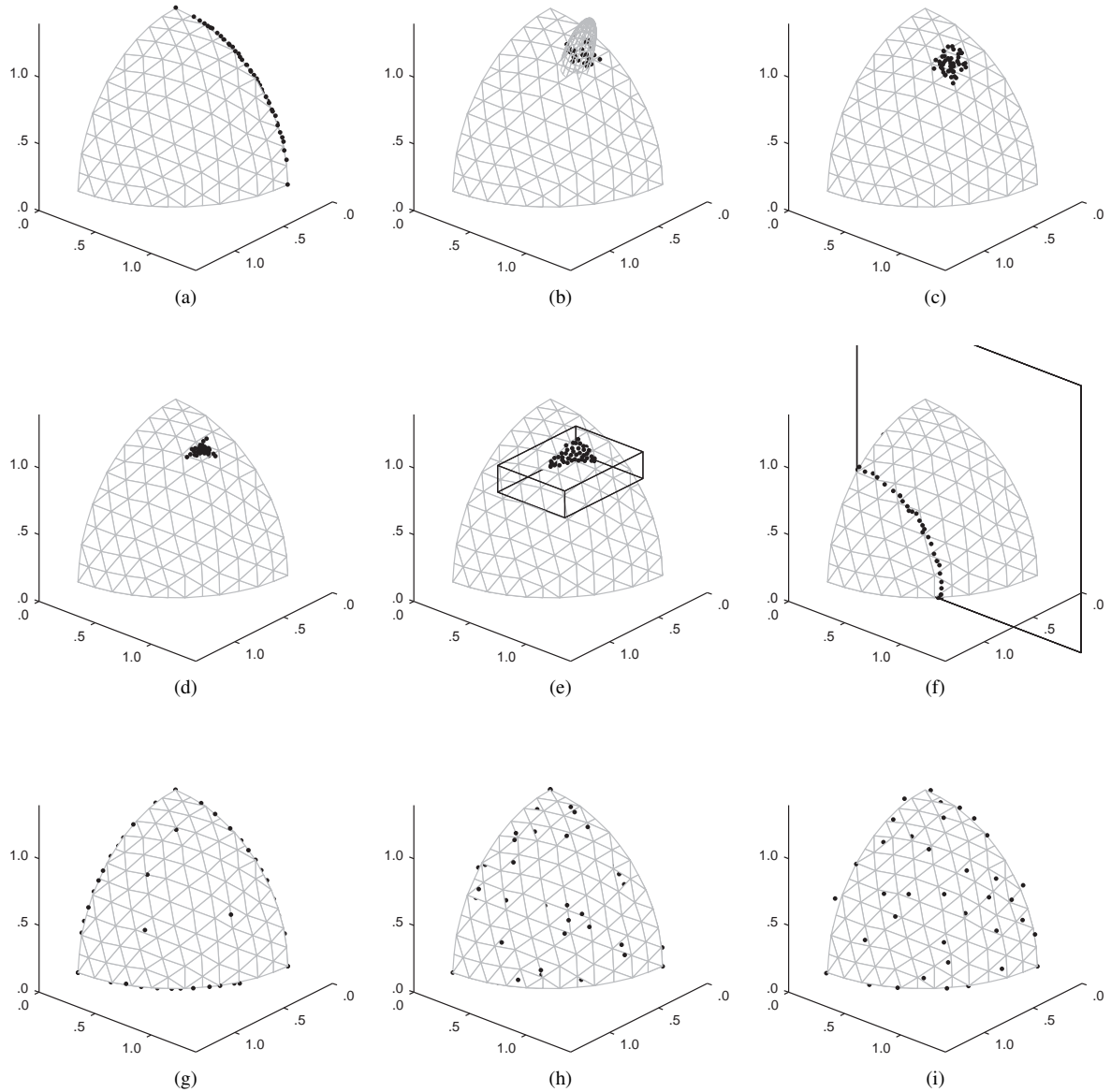


Figure 15: Pareto front approximations on DTLZ2 (sphere-shaped front) with three objectives for the following weight functions: (a)  $w_1$  (exponential), (b)  $w_2$  (preference point), (c)  $w_3$  (Tchebycheff), (d)  $w_4$  (desirability function), (e)  $w_5$  (uniform region), (f)  $w_6$  ( $\varepsilon$ -constraint), (g) IBEA, (h) NSGA-II, (i) SPEA2

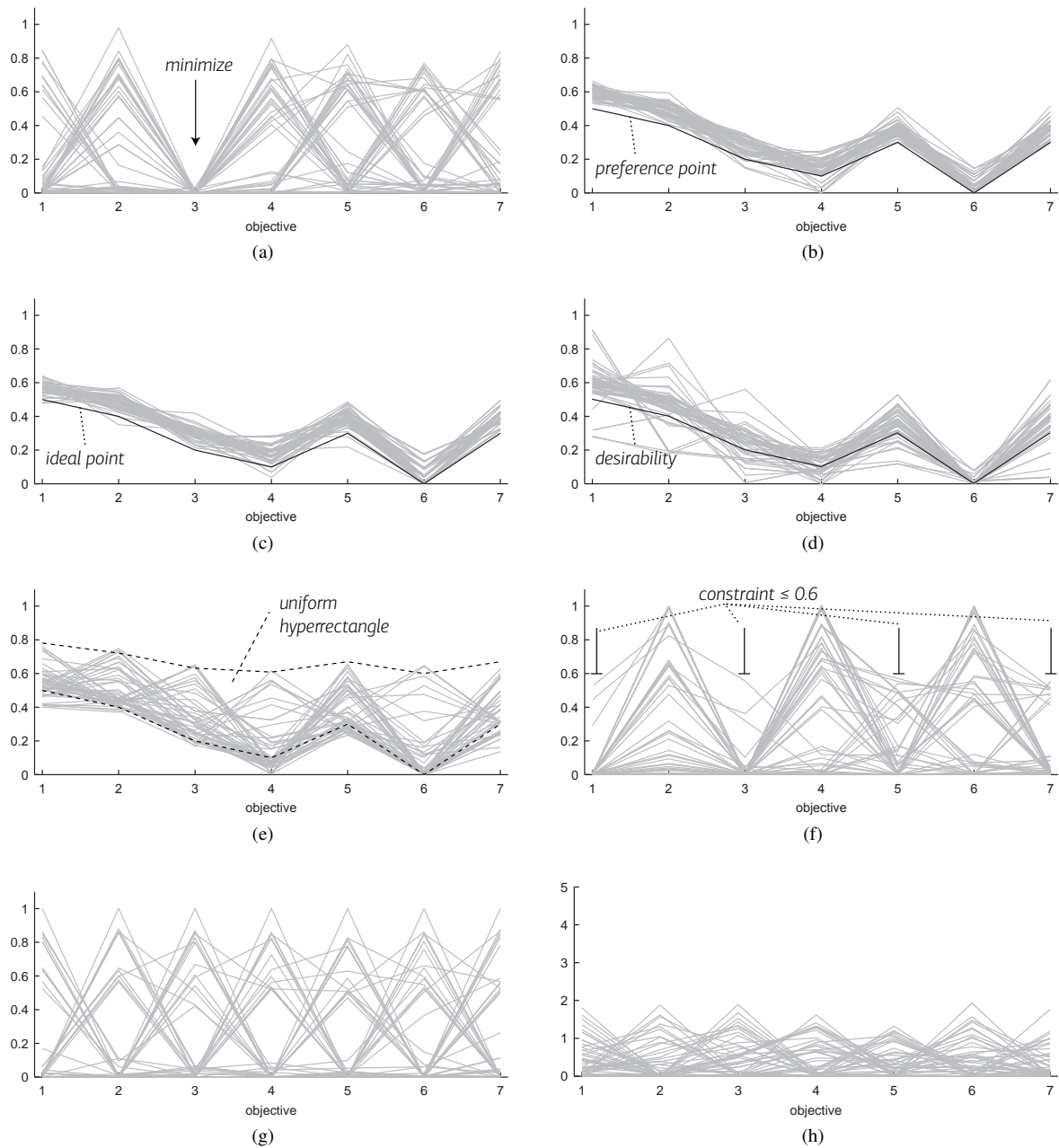


Figure 16: Parallel coordinates plots of the Pareto front approximations on DTLZ2 with 7 objectives for the following weight functions: (a)  $w_1$  (exponential), (b)  $w_2$  (preference point), (c)  $w_3$  (Tchebycheff), (d)  $w_4$  (desirability function), (e)  $w_5$  (uniform region), (f)  $w_6$  ( $\varepsilon$ -constraint), (g) IBEA, and (h) NSGA-II



Table 2: Normalized hypervolume values  $\bar{h}_{ij}$  with respect to the weight functions  $w_1$  to  $w_6$  of the nine algorithms (see Sec. 8.1.1) on ZDT1 with two objectives (a) and DTLZ2 with three (b) and seven objectives (c). In brackets, the normalized mean rank of the Kruskal-Wallis ranking is reported. The significantly largest values at  $\alpha = 0.01$  are highlighted in bold face.

(a) ZDT1 with two objectives									
	W-HyPE $w_1$	W-HyPE $w_2$	W-HyPE $w_3$	W-HyPE $w_4$	W-HyPE $w_5$	W-HyPE $w_6$	IBEA	NSGA-II	SPEA2
$w_1$	<b>1.00*</b> (0.00)	0.00 (1.00)	0.00 (0.88)	0.69 (0.58)	0.84 (0.47)	0.05 (0.75)	0.86 (0.46)	0.99 (0.24)	1.00 (0.13)
$w_2$	0.26 (1.00)	<b>1.00*</b> (0.00)	0.99 (0.35)	0.99 (0.45)	0.99 (0.31)	1.00 (0.14)	0.97 (0.65)	0.96 (0.81)	0.96 (0.79)
$w_3$	0.29 (1.00)	0.99 (0.30)	<b>1.00*</b> (0.00)	0.98 (0.48)	0.99 (0.32)	0.99 (0.16)	0.96 (0.65)	0.94 (0.81)	0.95 (0.79)
$w_4$	0.18 (1.00)	0.84 (0.79)	0.83 (0.83)	<b>1.00*</b> (0.00)	0.95 (0.48)	0.91 (0.63)	0.99 (0.13)	0.97 (0.36)	0.98 (0.28)
$w_5$	0.23 (1.00)	0.83 (0.84)	0.86 (0.77)	0.99 (0.12)	<b>1.00*</b> (0.00)	0.89 (0.64)	0.98 (0.25)	0.97 (0.47)	0.97 (0.40)
$w_6$	0.27 (1.00)	0.69 (0.87)	0.88 (0.73)	0.99 (0.13)	0.94 (0.49)	<b>1.00*</b> (0.00)	0.96 (0.30)	0.94 (0.52)	0.94 (0.46)

(b) DTLZ2 with three objectives									
	W-HyPE $w_1$	W-HyPE $w_2$	W-HyPE $w_3$	W-HyPE $w_4$	W-HyPE $w_5$	W-HyPE $w_6$	IBEA	NSGA-II	SPEA2
$w_1$	<b>0.99*</b> (0.00)	0.26 (0.50)	0.13 (0.74)	0.73 (0.25)	0.13 (0.75)	0.00 (1.00)	0.91 (0.13)	0.13 (0.75)	0.37 (0.40)
$w_2$	0.95 (0.42)	<b>1.00*</b> (0.01)	0.97 (0.24)	0.98 (0.14)	0.95 (0.46)	0.01 (1.00)	0.93 (0.60)	0.73 (0.84)	0.79 (0.78)
$w_3$	0.85 (0.53)	0.94 (0.32)	<b>1.00*</b> (0.00)	0.95 (0.31)	0.98 (0.13)	0.15 (1.00)	0.83 (0.59)	0.58 (0.82)	0.60 (0.80)
$w_4$	0.73 (0.25)	0.53 (0.49)	0.43 (0.77)	<b>0.97*</b> (0.00)	0.45 (0.71)	0.01 (1.00)	0.86 (0.13)	0.44 (0.71)	0.55 (0.45)
$w_5$	0.77 (0.54)	0.69 (0.70)	0.95 (0.25)	0.98 (0.13)	<b>1.00*</b> (0.00)	0.12 (1.00)	0.92 (0.37)	0.59 (0.82)	0.70 (0.69)
$w_6$	0.57 (0.53)	0.03 (0.94)	0.06 (0.83)	0.51 (0.59)	0.06 (0.86)	<b>1.00*</b> (0.00)	0.96 (0.13)	0.79 (0.37)	0.85 (0.26)

(c) DTLZ2 with seven objectives									
	W-HyPE $w_1$	W-HyPE $w_2$	W-HyPE $w_3$	W-HyPE $w_4$	W-HyPE $w_5$	W-HyPE $w_6$	IBEA	NSGA-II	SPEA2
$w_1$	1.00 (0.13)	0.08 (0.92)	0.07 (0.95)	0.67 (0.55)	0.36 (0.74)	0.90 (0.31)	<b>1.00*</b> (0.00)	0.88 (0.33)	0.66 (0.58)
$w_2$	0.06 (0.71)	<b>0.97*</b> (0.02)	0.98 (0.12)	0.91 (0.26)	0.67 (0.37)	0.09 (0.65)	0.23 (0.51)	0.00 (0.93)	0.00 (0.94)
$w_3$	0.44 (0.73)	0.99 (0.11)	<b>1.00*</b> (0.02)	0.96 (0.26)	0.89 (0.37)	0.60 (0.62)	0.68 (0.51)	0.02 (0.90)	0.00 (0.97)
$w_4$	0.34 (0.61)	0.62 (0.30)	0.58 (0.48)	<b>0.93*</b> (0.02)	0.71 (0.14)	0.06 (0.87)	0.64 (0.34)	0.11 (0.76)	0.02 (0.99)
$w_5$	0.26 (0.75)	0.83 (0.43)	0.85 (0.42)	0.95 (0.14)	<b>0.99*</b> (0.02)	0.50 (0.63)	0.89 (0.25)	0.00 (0.93)	0.00 (0.93)
$w_6$	0.79 (0.25)	0.02 (0.75)	0.02 (0.66)	0.13 (0.44)	0.11 (0.46)	<b>0.99*</b> (0.01)	0.97 (0.11)	0.02 (0.87)	0.00 (0.95)

Table 3: Used parameters of the biobjective 0/1 knapsack problem in PISA.

number of items	250
mutation type	independent bitflip
recombination	none
bitflip probability	2/250
population size	25
number of generations	15,000

with  $m_{5d} = (.5, .4, .2, .1, .3)$ ,  $t = 1$ ,  $\sigma_\varepsilon = 0.05$ , and  $\sigma_t = 0.5$  using 10,000 samples on the same 5-objective DTLZ2 problem as in (Voß et al., 2010), **W-HypE** is about twice as fast per hypervolume computation than the implementations of SMS-EMOA and MO-CMA-ES reported in (Voß et al., 2010). Note that the choice of the weight function in **W-HypE** by itself has an influence on the algorithm's runtime: while using  $w_2$  results in the smallest and using  $w_6$  results in the highest runtime in the above 7-objective example, the runtime per hypervolume estimation was only for  $w_6$  higher than the reported times in (Voß et al., 2010).

### 8.3 Applications in Discrete Domain

The previous results showed how the weighted hypervolume indicator approach of **W-HypE** changes the search bias according to a specified weight function. The used ZDT and DTLZ problems, however, are simple test functions with some known defects such as separability (Huband et al., 2006). Since the concepts presented in this paper are independent of the variation operators they should be therefore applicable to other problems as well. In order to show that **W-HypE** can also be applied to problems in discrete domain, we here use the presented algorithm to optimize a biobjective 0/1 knapsack problem, taken from the PISA test suite (Bleuler et al., 2003), and used with the settings of Table 3. Overall, four different weight functions have been tested: stressing the first objective ( $w_1^{KP}$ ), preference point ( $w_2^{KP}$ ), preference region ( $w_5^{KP}$ ), and  $\varepsilon$ -constraint ( $w_6^{KP}$ ), the definition of which can be found in Table 4.

#### 8.3.1 Visual Inspection

Figure 17 shows the final population after 375 000 function evaluations of an exemplary run for each of the four weight functions. In addition, the empirical 10% attainment surface from 10 independent runs of the standard **HypE** algorithm which employs the standard (unweighted) hypervolume indicator is shown for comparison. Note that the exact Pareto front for the used instance is not known.

It turns out that also in the discrete case of the knapsack problem, **W-HypE** is able to steer the search towards the user's preferences—although the distribution of solutions, in particular for  $w_1^{KP}$  and  $w_2^{KP}$ , do not show a nice convergence to the Pareto front in regions where the weight function is low but nevertheless points are generated. This might be caused by a generally lower rate of creating dominated solutions than for the DTLZ and ZDT functions<sup>5</sup>. What can also be observed is the fact that when focusing on certain regions of the objective space with **W-HypE**, the algorithm can find solutions in these specific regions which outperform the solutions found by **HypE** with the same number of function evaluations. This is especially evident for the populations shown in Fig. 17a) and c). However, note that this is not true for all runs and in particular not for all problems<sup>6</sup>.

#### 8.3.2 Test Statistics

Similar to Table 2, the mean hypervolume values and the mean ranks from a Conover-Inman test for each combination of weight and algorithm is shown in Table 5—here for 10 independent runs. As for the continuous test cases, it turns out that also for the discrete knapsack problem, **W-HypE** optimizing a specific weight function yields statistically significantly better hypervolume values regarding this weight function than when employing other weight functions.

<sup>5</sup>As a consequence, the number of function evaluations had to be chosen larger in the knapsack example than for the continuous test cases.

<sup>6</sup>When for example applied to the network processor design problem EXPO from the PISA test function suite, **W-HypE** is typically finding only solutions dominated by **HypE** when the user is aiming at regions of the Pareto front where few points are located or where there is a hole in the discontinuous Pareto front (results not

Table 4: Parameters of the weight functions used in the knapsack experiments.

wdf	purpose	type	section	parameters
$w_1^{\text{KP}}$	objective stressing	simple	4.1	$\lambda = 100$ , $b^l = (3\,600, 0)$ , $b^u = (10\,000, 10\,000)$ , $s = 1$
$w_2^{\text{KP}}$	preference point	simple	4.2	$m_{2d} = (4\,800, 4\,200)$ , $t = (1, 2)$ , $\sigma_\varepsilon = 70$ , $\sigma_t = 700$
$w_5^{\text{KP}}$	preference region	simple	4.3	$l = (4\,200, 4\,200)$ , $u = (4\,600, 4\,800)$
$w_6^{\text{KP}}$	$\varepsilon$ -constraint	classical	6.2	uniform weight: $l = (4\,200, 0)$ , $u = (4\,200, 10\,000)$ . Smoothing with negative normal distribution and $\sigma = 100$

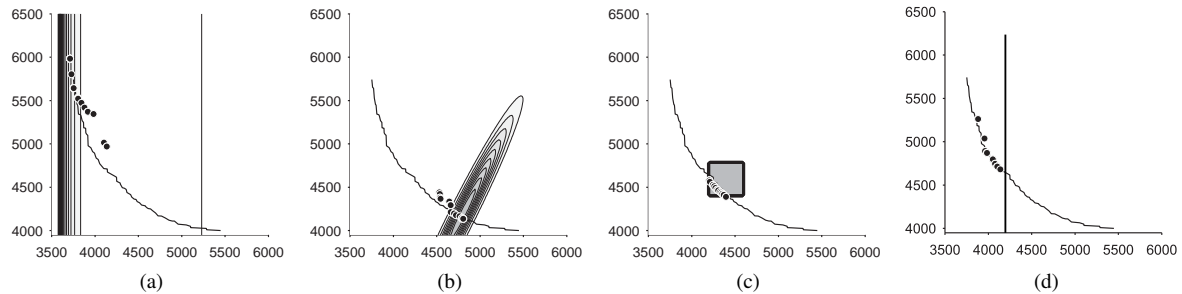


Figure 17: Final Population of an exemplary run of **W-HypE** (after 15 000 generations) on the bi-objective knapsack problem:  $w_1^{\text{KP}}$  (a),  $w_2^{\text{KP}}$  (b),  $w_5^{\text{KP}}$  (c), and  $w_6^{\text{KP}}$  (d). In addition, the empirical 10% attainment surface for 10 independent runs of **HypE** in generation 15 000 is shown.

Table 5: Normalized hypervolume values  $\bar{h}_{ij}$  with respect to the weight functions  $w_1^{\text{KP}}$ ,  $w_2^{\text{KP}}$ ,  $w_5^{\text{KP}}$ , and  $w_6^{\text{KP}}$  on the bi-objective knapsack problem for the four **W-HypE** versions employing those weight functions as well as for the standard **HypE** algorithm. In brackets, the normalized mean rank of the Kruskal-Wallis ranking is reported. The significantly largest values at  $\alpha = 0.01$  are highlighted in bold face.

	<b>W-HypE</b> $w_1^{\text{KP}}$	<b>W-HypE</b> $w_2^{\text{KP}}$	<b>W-HypE</b> $w_5^{\text{KP}}$	<b>W-HypE</b> $w_6^{\text{KP}}$	<b>HypE</b>
$w_1^{\text{KP}}$	<b>0.60*</b> (0.07)	0.00 (1.00)	0.01 (0.75)	0.18 (0.48)	<b>0.39*</b> (0.20)
$w_2^{\text{KP}}$	0.17 (1.00)	<b>0.99*</b> (0.00)	0.67 (0.50)	0.40 (0.76)	0.95 (0.25)
$w_5^{\text{KP}}$	0.06 (0.92)	0.10 (0.83)	<b>0.98*</b> (0.00)	0.47 (0.50)	0.80 (0.25)
$w_6^{\text{KP}}$	0.91 (0.50)	0.00 (0.96)	0.04 (0.79)	<b>0.99*</b> (0.01)	0.98 (0.24)

## 9 Conclusions

We have presented an approach that allows to include preferences of a decision maker into multiobjective evolutionary search by an appropriate weighting of the hypervolume indicator. Based on its proven refinement of Pareto-dominance, efficient indicator-based optimization algorithms can be developed. The paper elaborates the above approach by (a) presenting an efficient method based on Monte Carlo sampling to compute the weighted hypervolume indicator for large population sizes and high dimensions, by (b) developing a toolkit of useful weighting functions as well as composition methods, and by (c) showing how various classical preference articulation methods can be transferred to evolutionary search methods. An extensive experimental section validates the efficiency and practicality of the new approach by visual as well as statistically verified results. Open issues are related to the embedding of the above optimization kernel into an adaptive optimization approach that allows a close interaction of a decision maker. For example, it may be useful to continuously steer and refine the search in some form of interaction.

## A Weight Function Equivalent to Transforming the Objectives Via Monotonic Desirability Functions

**Theorem 1.** *Given the desirability function  $\varphi(z) = (\varphi_1(z), \dots, \varphi_n(z))$  where  $\varphi_i : \mathbb{R} \rightarrow [0, 1]$  are strictly monotonously increasing for all  $1 \leq i \leq n$ . Transforming the objective functions  $f_i$  into  $\varphi_i \circ f_i$  and using the unweighted hypervolume indicator is equivalent to using the weighted hypervolume indicator with weight function  $w(z) = \prod_{i=1}^n \varphi'_i(z_i)$ .*

*Proof.* After transforming the objective values to  $z^t = \varphi(z)$ ,  $z^t \in \varphi(Z)$ , the (unweighted) hypervolume of (1)

shown).

with  $w(z) = 1$  reads as

$$I_H^t(A, R) = \int_{z^t \in \varphi(H(A, R))} 1 dz^t \quad (23)$$

where

$$\begin{aligned} \varphi(H(A, R)) &= \{z \in Z \mid \exists a \in A: \exists r \in R: \varphi(f(a)) \leq z \leq \varphi(r)\} \\ &= \{z^t \in \varphi(Z) \mid \exists a \in A: \exists r \in R: f(a) \leq \varphi^{-1}(z^t) \leq r\}. \end{aligned}$$

When changing the variable  $z^t$  back to the original  $z$  according to  $z = \varphi^{-1}(z^t)$ , the differentials  $dz$  and  $dz^t$  relate as  $dz/dz^t = |J_{\varphi^{-1}}(z^t)|$  where  $|J_{\varphi^{-1}}(z^t)|$  denotes the determinant of the Jacobian matrix  $J_{\varphi^{-1}}(z^t)$  of the inverse of the desirability function  $\varphi$ . Since  $\varphi_i : z_i \mapsto \varphi_i(z)$ , we have

$$\begin{aligned} |J_{\varphi^{-1}}(z^t)| &= \prod_{i=1}^n [\varphi_i^{-1}]'(z_i^t) = \prod_{i=1}^n \frac{1}{\varphi'_i(\varphi_i^{-1}(z_i^t))} \\ &= \prod_{i=1}^n \frac{1}{\varphi'_i(z_i)} \end{aligned}$$

and (23) becomes

$$\begin{aligned} I_H^w(A, R) &= \int_{z \in H(A, R)} \prod_{i=1}^n \varphi'_i(z_i) dz \\ &= \int_{z \in H(A, R)} w^t(z) dz \end{aligned}$$

where  $w^t = \prod_{i=1}^n \varphi'_i(z_i)$  can be seen as the new weight function.  $\square$

## References

- Auger, A., Bader, J., Brockhoff, D., and Zitzler, E. (2009a). Articulating User Preferences in Many-Objective Problems by Sampling the Weighted Hypervolume. In *Genetic and Evolutionary Computation Conference (GECCO 2009)*, pages 555–562. ACM.
- Auger, A., Bader, J., Brockhoff, D., and Zitzler, E. (2009b). Investigating and Exploiting the Bias of the Weighted Hypervolume to Articulate User Preferences. In *Genetic and Evolutionary Computation Conference (GECCO 2009)*, pages 563–570. ACM.

- Auger, A., Bader, J., Brockhoff, D., and Zitzler, E. (2009c). Theory of the Hypervolume Indicator: Optimal  $\mu$ -Distributions and the Choice of the Reference Point. In *Foundations of Genetic Algorithms (FOGA 2009)*, pages 87–102. ACM.
- Auger, A., Bader, J., Brockhoff, D., and Zitzler, E. (2012). Hypervolume-based Multiobjective Optimization: Theoretical Foundations and Practical Implications. *Theoretical Computer Science*, 425:75–103.
- Bader, J., Deb, K., and Zitzler, E. (2010). Faster Hypervolume-based Search using Monte Carlo Sampling. In *Conference on Multiple Criteria Decision Making (MCDM 2008)*, pages 313–326. Springer.
- Bader, J. and Zitzler, E. (2011). HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. *Evolutionary Computation*, 19(1):45–76.
- Berghammer, R., Friedrich, T., and Neumann, F. (2010). Set-based Multi-objective Optimization, Indicators, and Deteriorative Cycles. In *Genetic and Evolutionary Computation Conference (GECCO 2010)*, pages 495–502.
- Beume, N. (2009). S-Metric Calculation by Considering Dominated Hypervolume as Klee's Measure Problem. *Evolutionary Computation*, 17(4):477–492.
- Beume, N., Naujoks, B., and Emmerich, M. (2007). SMS-EMOA: Multiobjective Selection Based on Dominated Hypervolume. *European Journal on Operational Research*, 181:1653–1669.
- Beume, N. and Rudolph, G. (2006). Faster S-Metric Calculation by Considering Dominated Hypervolume as Klee's Measure Problem. Technical Report CI-216/06, SFB 531 Computational Intelligence, Universität Dortmund.
- Bleuler, S., Laumanns, M., Thiele, L., and Zitzler, E. (2003). PISA—A Platform and Programming Language Independent Interface for Search Algorithms. In *Conference on Evolutionary Multi-Criterion Optimization (EMO 2003)*, pages 494–508. Springer.
- Bradstreet, L., Barone, L., and While, L. (2009). Updating Exclusive Hypervolume Contributions Cheaply. In *Congress on Evolutionary Computation (CEC'2009)*, pages 538–544. IEEE Press.
- Bradstreet, L., While, L., and Barone, L. (2008). A Fast Incremental Hypervolume Algorithm. *IEEE Transactions on Evolutionary Computation*, 12(6):714–723.
- Bringmann, K. and Friedrich, T. (2008). Approximating the Volume of Unions and Intersections of High-Dimensional Geometric Objects. In *International Symposium on Algorithms and Computation (ISAAC 2008)*, pages 436–447. Springer.
- Bringmann, K. and Friedrich, T. (2009a). Approximating the Least Hypervolume Contributor: NP-hard in General, But Fast in Practice. In *Conference on Evolutionary Multi-Criterion Optimization (EMO 2009)*, pages 6–20. Springer.
- Bringmann, K. and Friedrich, T. (2009b). Don't Be Greedy When Calculating Hypervolume Contributions. In *Foundations of Genetic Algorithms (FOGA 2009)*, pages 103–112. ACM.
- Conover, W. J. (1999). *Practical Nonparametric Statistics*. John Wiley, 3rd edition.
- Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, Chichester, UK.
- Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. (2000). A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In *Conference on Parallel Problem Solving from Nature (PPSN VI)*, pages 849–858. Springer.
- Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2005). Scalable Test Problems for Evolutionary Multi-Objective Optimization. In Abraham, A., Jain, R., and Goldberg, R., editors, *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, pages 105–145. Springer.
- Deb, K. and Jain, H. (2012). Handling Many-Objective Problems Using an Improved NSGA-II

- Procedure. In *IEEE Congress on Evolutionary Computation (CEC 2012)*, pages 1–8. IEEE Press.
- Devroye, L. (1986). *Non-Uniform Random Variate Generation*. Springer.
- Emmerich, M., Beume, N., and Naujoks, B. (2005). An EMO Algorithm Using the Hypervolume Measure as Selection Criterion. In *Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, pages 62–76. Springer.
- Emmerich, M. and Fonseca, C. (2011). Computing Hypervolume Contributions in Low Dimensions: Asymptotically Optimal Algorithm and Complexity Results. In *Conference on Evolutionary Multi-Criterion Optimization (EMO 2011)*, pages 121–135. Springer.
- Everson, R., Fieldsend, J., and Singh, S. (2002). Full Elite-Sets for Multiobjective Optimisation. In *Conference on adaptive computing in design and manufacture (ADCM 2002)*, pages 343–354. Springer.
- Fleischer, M. (2003). The measure of Pareto optima. Applications to multi-objective metaheuristics. In *Conference on Evolutionary Multi-Criterion Optimization (EMO 2003)*, pages 519–533. Springer.
- Fonseca, C. M., Paquete, L., and López-Ibáñez, M. (2006). An Improved Dimension-Sweep Algorithm for the Hypervolume Indicator. In *Congress on Evolutionary Computation (CEC 2006)*, pages 1157–1163. IEEE Press.
- Friedrich, T., Horoba, C., and Neumann, F. (2009). Multiplicative Approximations and the Hypervolume Indicator. In *Genetic and Evolutionary Computation Conference (GECCO 2009)*, pages 571–578. ACM.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, Massachusetts.
- Harrington, J. (1965). The Desirability Function. *Industrial Quality Control*, 21(10):494–498.
- Hoeffding, W. (1963). Probability Inequalities for Sums of Bounded Random Variables. *J Am Stat Assoc*, 58(301):13–30.
- Huband, S., Hingston, P., Barone, L., and While, L. (2006). A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506.
- Hughes, E. J. (2011). Many-Objective Directed Evolutionary Line Search. In *Genetic and Evolutionary Computation Conference (GECCO 2011)*, pages 761–768. ACM.
- Igel, C., Hansen, N., and Roth, S. (2007). Covariance Matrix Adaptation for Multi-objective Optimization. *Evol Comput*, 15(1):1–28.
- Ishibuchi, H., Tsukamoto, N., and Nojima, Y. (2008). Evolutionary Many-Objective Optimization: A Short Review. In *Congress on Evolutionary Computation (CEC 2008)*, pages 2424–2431. IEEE Press.
- Knowles, J. (2002). *Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization*. PhD thesis, University of Reading.
- Knowles, J., Corne, D., and Fleischer, M. (2006). Bounded Archiving using the Lebesgue Measure. In *Congress on Evolutionary Computation (CEC 2003)*, pages 2490–2497, Canberra, Australia. IEEE Press.
- Li, M., Liu, L., and Lin, D. (2011). A Fast Steady-State  $\varepsilon$ -Dominance Multi-Objective Evolutionary Algorithm. *Computational Optimization and Applications*, 48(1):109–138.
- Miettinen, K. (1999). *Nonlinear Multiobjective Optimization*. Kluwer, Boston, MA, USA.
- Scheirer, C. J., Ray, W. S., and Hare, N. (1976). The Analysis of Ranked Data Derived from Completely Randomized Factorial Designs. *Biometrics*, 32(2):429–434.
- Srinivas, N. and Deb, K. (1994). Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248.

- Voß, T., Friedrich, T., Bringmann, K., and Igel, C. (2010). Scaling Up Indicator-based MOEAs by Approximating the Least Hypervolume Contributor: A Preliminary Study. In *GECCO workshop on Theoretical Aspects of Evolutionary Multiobjective Optimization*, pages 1975–1978. ACM.
- Wagner, T., Beume, N., and Naujoks, B. (2007). Pareto-, Aggregation-, and Indicator-based Methods in Many-objective Optimization. In *Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*, pages 742–756. Springer.
- Wagner, T. and Trautmann, H. (2010). Integration of Preferences in Hypervolume-Based Multi-Objective Evolutionary Algorithms by Means of Desirability Functions. *IEEE Transactions on Evolutionary Computation*, 14(5):688–701.
- While, L., Hingston, P., Barone, L., and Huband, S. (2006). A Faster Algorithm for Calculating Hypervolume. *IEEE Transactions on Evolutionary Computation*, 10(1):29–38.
- Wierzbicki, A. (1999). Reference Point Approaches. In Gal, T., Stewart, T., and Hanne, T., editors, *Multicriteria Decision Making: Advances in MCDM Models, Algorithms, Theory, and Applications*, pages 9–1–9–39. Kluwer.
- Zitzler, E. (2001). Hypervolume metric calculation. <ftp://ftp.tik.ee.ethz.ch/pub/people/zitzler/hypervol.c>.
- Zitzler, E., Brockhoff, D., and Thiele, L. (2007). The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration. In *Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*, pages 862–876. Springer.
- Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195.
- Zitzler, E., Knowles, J., and Thiele, L. (2008). Quality Assessment of Pareto Set Approximations. In Branke, J., Deb, K., Miettinen, K., and Slowinski, R., editors, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, pages 373–404. Springer.
- Zitzler, E. and Künzli, S. (2004). Indicator-Based Selection in Multiobjective Search. In *Conference on Parallel Problem Solving from Nature (PPSN VIII)*, pages 832–842. Springer.
- Zitzler, E., Laumanns, M., and Thiele, L. (2001). SPEA2: Improving the Strength Pareto Evolutionary Algorithm. TIK Report 103, Computer Engineering and Networks Laboratory (TIK), ETH Zurich.
- Zitzler, E. and Thiele, L. (1998a). An Evolutionary Approach for Multiobjective Optimization: The Strength Pareto Approach. TIK Report 43, Computer Engineering and Networks Laboratory (TIK), ETH Zurich.
- Zitzler, E. and Thiele, L. (1998b). Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. In *Conference on Parallel Problem Solving from Nature (PPSN V)*, pages 292–301.
- Zitzler, E., Thiele, L., and Bader, J. (2010). On Set-Based Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, 14(1):58–79.
- Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., and Grunert da Fonseca, V. (2003). Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132.