Robustness in Hypervolume-based Multiobjective Search

TIK Report 317

Johannes Bader johannes.bader@tik.ee.ethz.ch
Computer Engineering and Networks Laboratory, ETH Zurich, 8092 Zurich, Switzerland

Eckart Zitzler eckart.zitzler@tik.ee.ethz.ch
Computer Engineering and Networks Laboratory, ETH Zurich, 8092 Zurich, Switzerland

Abstract

The use of quality indicators within the search has become a popular approach in the field of evolutionary multiobjective optimization. It relies on the concept to transform the original multiobjective problem into a set problem that involves a single objective function only, namely a quality indicator, reflecting the quality of a Pareto set approximation. Especially the hypervolume indicator has gained a lot of attention in this context since it is the only set quality measure known that guarantees strict monotonicity. Accordingly, various hypervolume-based search algorithms for approximating the Pareto set have been proposed, including sampling-based methods that circumvent the problem that the hypervolume is in general hard to compute.

Despite these advances, there are several open research issues in indicator-based multiobjective search when considering real-world applications—the issue of robustness is one of them. For instance with mechanical manufacturing processes, there exist unavoidable inaccuracies that prevent a desired solution to be realized with perfect precision; therefore, a solution in terms of a concrete decision vector is not associated with just one one fixed vector of objective values, but rather with a range of objective values that reflect the variance when slightly changing the decision variables. As a consequence, the optimization model needs to account for such uncertainties and the search method is required to explicitly integrate robustness considerations.

While in single-objective optimization, there are various studies dealing with the robustness issue, there are considerably fewer in the multiobjective optimization literature and none in the context of hypervolume-based multiobjective search. This study is set in the latter context and addresses the question of how to incorporate robustness when using the hypervolume indicator within an evolutionary algorithm. To this end, three common robustness concepts are translated to and tested for hypervolume-based search on the one hand, and an extension of the hypervolume indicator is proposed on the other hand that not only unifies those three concepts, but also enables to realize much more general trade-offs between objective values and robustness of a solution. Finally, the approaches are compared on two test problem suites as well as on a newly proposed real-world bridge construction problem.

Keywords

robustness, hypervolume indicator, multiobjective optimization, multiobjective evolutionary algorithm, Monte Carlo sampling, truss bridge problem

1 Introduction

In multiobjective optimization, using the hypervolume indicator to find Pareto-optimal solutions has become popular in recent years. The reason for the popularity of this measure is its strict monotonicity with respect to Pareto dominance (Zitzler et al., 2003). As a consequence hypervolume-based algorithms can be designed to guarantee a theoretical convergence behavior (Brockhoff et al., 2008; Zitzler et al., 2009). Existing multiobjective techniques that combine Pareto dominance with a specific diversity measure, e.g., (Deb et al., 2000; Zitzler et al., 2001), often fail in this regard on problems involving a larger number (e.g., larger than four) of objectives due to cyclic behavior (Wagner et al., 2007; Zitzler et al., 2008). Although the basic hypervolume-based search approach has been investigated and extended in different directions, e.g., with regard to the bias of the indicator (Brockhoff et al., 2008; Auger et al., 2009b; Friedrich et al., 2009), to the incorporation of user preferences (Zitzler et al., 2007; Auger et al., 2009a), and to the hypervolume calculation resp. estimation (While et al., 2005; Beume and Rudolph, 2006; Fonseca et al., 2006; Bader and Zitzler, 2008), the issue of robustness has not been addressed so far to the best of our knowledge. A few studies have used the hypervolume as a measure of robustness though: Ge et al. (2005) have used the indicator to assess the sensitivity of design regions according to the robust design of Taguchi (1986); a similar concept by Beer and Liebscher (2008) uses the hypervolume indicator to measure the range of possible decision variables that lead to the desired range of objective values; a study by Hamann et al. (2007) applied the hypervolume indicator in the context of sensitivity analysis. However, none of these papers deals with integrating robustness into hypervolume-based multiobjective search.

Robustness becomes an important issue when tackling real-world applications. Solutions to engineering problems, for instance, can usually not be manufactured arbitrarily accurate such that the implemented solution and its objective values differ from the original specification, up to the point where they become infeasible. Designs which are seriously affected by perturbations of any kind might no longer be acceptable to a decision maker from a practical point of view—despite the promising theoretical result. The corresponding uncertainty due to production variations, which reflects the second category of uncertainty defined in Beyer and Sendhoff (2007), needs to be taken into account within both optimization model and algorithm in order to find robust solutions that are relatively insensitive to perturbations. Ideally, there exist Pareto-optimal designs whose characteristics fluctuate within an acceptable range. Yet, for the most part robustness and quality (objective values) are irreconcilable goals, and one has to make concessions to quality in order to achieve an acceptable robustness level.

Many studies have been devoted to robustness in the context of single-objective optimization, e.g., (Taguchi, 1986; Jin and Branke, 2005; Beyer and Sendhoff, 2007). However, most of these approaches are not applicable to multiobjective optimization. The first approaches (Kunjur and Krishnamurty, 1997; Tsui, 1999) to consider robustness in combination with multiple objectives are based on the design of experiment approach (DOE) by Taguchi (1986); however, they aggregate the individual objective functions such that the optimization itself is no longer of multiobjective nature. Only few studies genuinely tackle robustness in multiobjective optimization: one approach

by Teich (2001) is to define a probabilistic dominance relation that reflects the underlying noise; a similar concept by Hughes (2001) ranks individuals based on the objective values and the associated uncertainty; Deb and Gupta (2006, 2005) considered robustness by either adding an additional constraint or by optimizing according to a fitness averaged over perturbations. The following classification categorizes most existing robustness approaches in the evolutionary computing literature:

A — Replacing the objective value: Among the widest-spread approaches to account for noise is to replace the objective values by a measure or statistical value reflecting the uncertainty. Parkinson et al. (1993) for instance optimize the worst case. The same approach, referred to as "min max", is also employed in other studies, e.g., in (Kouvelis and Yu, 1997; Soares et al., 2009a,b) . Other studies apply an averaging approach where the mean of the objective function is used as the optimization criterion (Tsutsui and Ghosh, 1997; Jürgen Branke, 1998; Branke and Schmidt, 2003). In Mulvey et al. (1995) the objective values and a robustness measure are aggregated into a single value that servers as the optimization criterion.

B — Using one or more additional objectives: Many studies try to assess the robustness of solutions x by a measure $r(x)$, e.g., by taking the norm of the variance of the objective values (Jin and Sendhoff, 2003) or the maximum deviation from $f(x)$ (Deb and Gupta, 2006). This robustness measure is then treated as an additional objective (Jin and Sendhoff, 2003; Egorov et al., 2002; Li et al., 2005). A study by Burke et al. (2009) fixes a particular solution (a fleet assignment of an airline scheduling problem), and only optimizes the robustness of solutions (the schedule reliability and feasibility).

C — Using at least one additional constraint: A third possibility is to restrict the search to solutions fulfilling a predefined robustness constraint, again with respect to a robustness measures $r(x)$ (Gunawan and Azarm, 2004, 2005; Deb and Gupta, 2005, 2006).

Combinations of A and B are also used; Das (2000) for example considers the expected fitness along with the objective values $f(x)$, while Chen et al. (1996) optimize the mean and variance of $f(x)$.

Given these considerations, we investigate in the remainder of this paper how robustness can be integrated in hypervolume-based multiobjective search. First, we address the question of how the three existing approaches A, B, and C mentioned above can be translated to a concept for the hypervolume indicator. Second, we introduce a novel approach that represents a generalization of the hypervolume indicator unifying the three approaches, and third we propose different search algorithms implementing the various ideas. An empirical comparison on different test problems and a real-world problem provides valuable insights regarding advantages and disadvantages of the presented approaches.

2 Background

2.1 Hypervolume-based Multiobjective Search

In the following, we consider a multiobjective objective optimization problem $f : X \rightarrow Z$ where $X \subseteq \mathbb{R}^n$ denotes the decision space and n denotes the number of decision variables. The decision space is mapped to the objective space $Z \subseteq \mathbb{R}^d$ by d objective functions $(f_1(x), \ldots, f_d(x)) = f(x)$; without loss of generality, all objectives are to be

minimized. Optimization is performed according to a preference relation $\preceq$ on X. The minimal elements of the ordered set $(X, \preceq)$ constitute the optimal solutions forming the Pareto set, whose image under f is called Pareto front.

The weak Pareto dominance forms an important preference relation $\preceq_{\text{par}}$ on solutions, defined as $x \preceq_{\text{par}} y :\Leftrightarrow \forall i \in \{1, \ldots, n\} : f_i(a) \leq f_i(b)$. In the following, weak Pareto dominance $\preceq_{\text{par}}$ is used as the standard preference on solutions, however, other definitions, especially in the presence of uncertainty, are meaningful. In Sec. 3, we will propose different relations which rank solutions $a \in X$ not only based on the objective values $f(a)$, but also on a robustness measure $r(a)$. All relations introduced so far induce pre-orders and we use the usual definition for the corresponding strict orders (Harzheim, 2005); for example, Pareto dominance $\prec_{\text{par}}$ on two solutions $a, b \in X$ is given by $a \prec_{\text{par}} b \Leftrightarrow a \preceq_{\text{par}} b \wedge b \not\preceq_{\text{par}} a$.

The above definitions on single solutions can be transferred to an equivalent on sets (Zitzler et al., 2009) where the search space $\Psi = \mathcal{P}(X)$ consists of all possible sets of solutions $A \subseteq X$, and the objective space $\Omega = \mathcal{P}(Z)$ is formed by all sets $U \subseteq Z$ of objective vectors [1]. The set equivalent $F : \Psi \to \Omega$ of the objective function maps a set of solutions to their objective values, i.e., $X \mapsto \{y \mid \exists x \in X, f(x) = y\}$. The preference relation $\preceq$ on solutions is extended to a set preference $\preccurlyeq$ on sets A, B in the following canonical way:

$$A \preccurlyeq B :\Leftrightarrow \forall b \in B \; \exists a \in A : \; a \preceq b \tag{1}$$

where again the relation $\leqslant$ on the objective space Ω is given by the isomorphic mapping of $(\Psi, \preccurlyeq)$ to $(\Omega, \leqslant)$ given by the objective function f. Due to practical limitations, the search space is usually restricted to elements not exceeding a fixed number of elements α, i.e., $\Psi_{\leq\alpha} = \{A \in \Psi \mid |A| \leq \alpha\}$ where the relation $\preccurlyeq_{\leq\alpha}$ on $\Psi_{\leq\alpha}$ corresponds to the restriction of $\preccurlyeq$ to $\Psi_{\leq\alpha}$, i.e., $\preccurlyeq_{\leq\alpha} := \preccurlyeq \restriction \Psi_{\leq\alpha}$. Given such a relation on sets optimization algorithms strive to find (one of) the minimal elements of $\Psi_{\leq\alpha}$ for $\preccurlyeq \restriction \Psi_{\leq\alpha}$).

An important question in the context of optimizing sets is how to refine the Pareto dominance $\preccurlyeq_{\text{par}}$ for the large number of cases where neither $A \preccurlyeq_{\text{par}} B$ nor $B \preccurlyeq_{\text{par}} A$ holds, and therefore additional preference information is needed. Indicator functions represent one possibility to obtain a total relation on sets. They assign each set $A \in \Psi$ a value representing its quality. Given this value, the relation $\preceq_{\text{I}}$ on two sets $A, B \in \Psi$ is defined as $A \preceq_{\text{I}} B :\Leftrightarrow I(A) \geq I(B)$. The hypervolume indicator is one popular choice of indicator that has received more and more attention in recent years. It measures the (hyper-)volume of dominated portion of the objective space:

Definition 2.1. *Let $A \in \Psi$ denote a set of solutions and let r represents a reference point*[2]*, let $w : \mathbb{R}^k \to \mathbb{R}_{>0}$ denote a strictly positive weight function integrable on any bounded set. Then the hypervolume indicator for A is given as*

$$I_H^w(A, R) = \int_{(-\infty,\ldots,-\infty)}^{(\infty,\ldots,\infty)} \alpha_A(z) w(z) dz \quad \textit{with} \quad \alpha_A(z) = \begin{cases} 1 & A \preccurlyeq \{z\} \\ 0 & \textit{otherwise} \end{cases} \tag{2}$$

[1] In fact, A and U in their most general form are multisets where duplicates of the same solution and objective vector respectively are allowed. To avoid unwieldy notations, in this paper we stick to proper sets; the concepts, however, can be extended to multisets as well.

[2] In the most general definition of the hypervolume indicator, a reference set R is used instead of a single reference point r. Without loss of generality, throughout this study we consider only a single reference point r.

with $\alpha_A(z) = \mathbf{1}_{H(A,R)}(z)$ *where*

$$H(A,R) = \{z \mid \exists a \in A \; \exists r \in R \; : \; f(a) \leqslant z \leqslant r\} \tag{3}$$

and $\mathbf{1}_{H(A,r)}(z)$ *being the characteristic function of* $H(A,r)$ *that equals* 1 *iff* $z \in H(A,r)$ *and* 0 *otherwise.*

The reason for the popularity of the hypervolume indicator is that it is up to now the only indicator that is a refinement of the Pareto dominance, i.e., whenever for two sets $A, B \in \Psi$, $A \prec_{\text{par}} B$ holds, then $I_H^w(A,R) > I_H^w(B,R)$, see (Zitzler et al., 2009). As a consequence of this property, an improvement of any solution of set A will increase the indicator value, and the Pareto set A^* achieves the maximum possible hypervolume value. For this reasons, many modern algorithms use the hypervolume indicator as underlying (set-)preference for search (Igel et al., 2007; Emmerich et al., 2005; Fleischer, 2003; Knowles et al., 2006; Zitzler et al., 2007; Bader and Zitzler, 2009).

Although also useful for mating selection, the main field of application of the hypervolume indicator is environmental selection. The aim thereby is to select from a population of solutions a subset of predefined size, which will constitute the population of the next generation. Normally, hypervolume-based algorithms perform environmental selection by the following two consecutive steps:

1. At first, all solutions of the original population are divided into fronts by non-dominated sorting (Goldberg, 1989; Srinivas and Deb, 1994); while the aforementioned algorithms use Pareto-dominance as underlying dominance relation, in principle, however, also other relations can be used as will be demonstrated in Sec. 3. Following the non-dominated sorting the fronts are, starting with the best front, inserted into the new population as a whole as long as the number of solutions in the new population does not exceed the predefined population size.
2. The first front A which can no longer be inserted into the new population is thereafter truncated to the number of places to be filled. To this end, one after another the solution x_w is removed which causes the smallest loss in hypervolume $I_H^w(A,R) - I_H^w(A \backslash \{x_w\}, R)$. After reach removal, the hypervolume losses are recalculated. At the end, the truncated front is inserted in the new population which concludes environmental selection.

2.2 Robustness

Robustness of a solution informally means, that the objective values scatter only slightly under real conditions. These deviations, referred to as *uncertainty*, are often not considered in multiobjective optimization. This section shows one possibility to extend the optimization model proposed above by the consideration of uncertainty. As source of uncertainty, noise directly affecting the decision variable x is considered. This results in a random decision variable X^p, which is evaluated by the objective function instead of x. As distribution of X^p, this paper considers a uniform distribution:

$$B_\delta(x) := [x_1 - \delta, x_1 + \delta] \times \ldots \times [x_n - \delta, x_n + \delta] \; . \tag{4}$$

The distribution according to Eq. 4 stems from the common specification of fabrication tolerances. Of course, other probability distributions for X^p are conceivable as well; of particular importance is the Gaussian normal distribution, as it can be used to describe

many distributions observed in nature. Although not shown in this paper, the proposed algorithms work with other uncertainties just as well.

Given the uncertainty X^p, the following definition of Deb and Gupta (2005) can be used to measure the robustness of x:

$$r(x) = \frac{\|f^w(X^p) - f(x)\|}{f(x)} \tag{5}$$

where $f(x)$ denotes the objective values of the unperturbed solution, and $f^w(X^p)$ denotes the objective-wise worst case of all objective values of the perturbed decision variables X^p:

$$f^w(X^p) = \big(\max_{X^p} f_1(X^p), \ldots, \max_{X^p} f_d(X^p)\big) \tag{6}$$

From the multi-dimensional interval B_δ, the robustness measure $r(x)$ may be determined analytically (see Gunawan and Azarm (2005)). If this is not possible, for instance because knowledge of the objective function is unavailable, random samples are generated within $B_\delta(x)$ and evaluated to obtain an estimate of the robustness measure $r(x)$.

3 Concepts for Robustness Integration

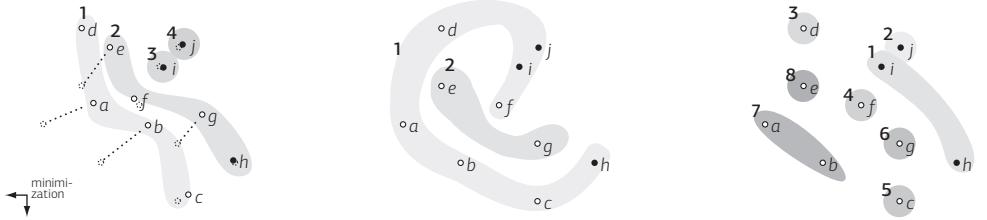
As already mentioned in the introduction, existing robustness integrating approaches can roughly be classified into three basic categories: (i) modifying the objective functions, (ii) using an additional objective, and (iii) using an additional constraint. To translate these approaches to hypervolume-based search, one or multiple of the three main components of hypervolume-based set preference need to be changed:

1. the preference relation is modified to consider robustness—this influences the nondominated sorting.
2. The objective values are modified before the hypervolume is calculated.
3. The definition of the hypervolume indicator itself is changed.

Depending on how the decision maker accounts for robustness, the preference relation changes to $\preceq_{\text{rob}}$. Note, that the relation $\preceq_{\text{rob}}$ does not need to be a subset of $\preceq$; in fact, the relation can even get reversed. For example, provided solution x is preferred over y given only the objectives $x \preceq y$, but considering robustness $y \preceq_{\text{rob}} x$ holds, for instance because y has a sufficient robustness level but x does not.

The most simple choice of dominance relation is $\preceq_{\text{rob}} \equiv \preceq_{\text{par}}$, that is to not consider robustness. This concept is used as reference in the experimental comparison in Sec. 5. Depending on the robustness of the Pareto set, optimal solutions according to $\preceq_{\text{par}}$ may or may not coincide with optimal solutions according to relations $\preceq_{\text{rob}}$ that consider robustness in some way.

In the following, other preference relations, corresponding to the approaches A,B, and C on page 3, are shown. All resulting relations $\preceq_{\text{rob}}$ thereby are not total. Therefore, to refine the relation, is is proposed to apply the general hypervolume-based procedure: first, solutions are ranked into fronts by nondominated sorting according to Sec. 2.1; after having partitioned the solutions, the normal hypervolume is applied on the objective values alone or in conjunction with the robustness measure (which case applies is mentioned when explaining the respective algorithm) to obtain a preference on the solutions.



(a) modifying the objectives (b) additional objective (c) additional constraint

Figure 1: Partitioning into fronts of ten solutions: a (robustness $r(a) = 2$), b (2), c (1.4), d (1.1), e (2), f (1.2), g (1.9), h (.5), i (.9), and j (.1) for the three approaches presented in Sec. 3. The solid dots represents robust solutions at the considered level of $\eta = 1$ while the unfilled dots represent non-robust solutions.

First, in Sec. 3.1, 3.2, and 3.3, we investigate how the existing concepts can be transformed to and used in hypervolume-based search. Then, in Sec. 3.4, these three concepts are unified into a novel generalized hypervolume indicator that also enables to realize other trade-offs between robustness and quality of solutions.

3.1 Modifying the Objective Functions

The first concept to incorporate robustness replaces the objective values $f(x) = (f_1(x), \ldots, f_d(x))$ by an evaluated version over all perturbations $f^p(X^p) = (f_1^p(X^p), \ldots, f_d^p(X^p))$, see Fig. 1(a). For example, the studies by Tsutsui and Ghosh (1997), Jürgen Branke (1998), and Branke and Schmidt (2003) all employ the mean over the perturbations, i.e.,

$$f_i^p(X^p) = \int_{x^p} f_i(x^p) p_{X^p}(x^p) dx \tag{7}$$

where $p_{X^p}(x^p)$ denotes the probability density function of the perturbed decision variable X^p given x. Taking the mean will smoothen the objective space, such that f^p is worse in regions where the objective values are heavily affected by perturbations; while, contrariwise, in regions where the objective values stay almost the same within the considered neighborhood, the value f^p differs only slightly. Aside from the altered objective value, the search problem stays the same. The regular hypervolume indicator in particular can be applied to optimize the problem. The dominance relation implicitly changes to $\preceq_{\text{rob}} = \preceq_{\text{repl}}$ with $x \preceq_{\text{repl}} y :\Leftrightarrow f^p(x) \leqslant f^p(y)$.

3.2 Additional Objective

Since the problems dealt with are already multiobjective by nature, a straightforward way to also account for the robustness $r(x)$ is to treat the measure as an additional objective (Jin and Sendhoff, 2003; Li et al., 2005; Guimaraes et al., 2006). As for the previous approach, this affects the preference relation and thereby non-dominated sorting, but also the calculating of the hypervolume. The objective function becomes $f^{\text{ao}} = (f_1, \ldots, f_d, r)$; the corresponding preference relation $\preceq_{\text{oa}}$ is accordingly

$$x \preceq_{\text{ao}} y :\Leftrightarrow x \preceq_{\text{par}} y \wedge r(x) \leq r(y) \ . \tag{8}$$

Considering robustness as an ordinary objective value has three advantages: first, apart from increasing the dimensionality by one, the problem does not change and existing multiobjective approaches can be used. Second, different degrees of robustness are promoted, and third, no robustness level has to be chosen in advance which would entail the risk of the chosen level being infeasible, or that the robustness level could be much improved with barely compromising the objective values of solutions. One disadvantage of this approach is to not focus on a specific robustness level and potentially finding many solutions whose robustness is too bad to be useful or whose objective values are strongly degraded to achieve an unnecessary large degree of robustness. A further complication is the increase in non-dominated solutions resulting from considering an additional objective, i.e., the expressiveness of the relation is smaller than the one of the previously stated relation $\preceq_{\text{repl}}$ and the relation proposed in the next section.

Fig. 1(b) shows the partitioning according to $\preceq_{\text{ao}}$. Due to the different robustness values, many solutions which are dominated according to objective values only—that is according to $\preceq_{\text{par}}$—become incomparable and only two solutions e and g remain dominated.

3.3 Additional Robustness Constraint

The third approach to embrace the robustness of a solution is to convert robustness into a constraint (Fonseca and Fleming, 1998; Gunawan and Azarm, 2005; Deb and Gupta, 2005), which is then considered by adjusting the preference relation affecting non-dominated sorting. We here use a slight modification of the definition of Deb and Gupta (2005) by adding the additional refinement of applying weak Pareto dominance if two non-robust solutions have the same robustness value. Given the objective function $f(x)$ and robustness measure $r(x)$, an optimal robust solution then is

Definition 3.1 (optimal solution under a robustness constraint)**.** *A solution $x^* \in X$ with $r(x^*)$ and $f(x^*)$ denoting its robustness and objective value respectively, both of which are to be minimized, is optimal with respect to the robustness constraint η, if it fulfills $x^* \in \{x \in X \mid \forall y \in X : x \preceq_{\text{con}} y\}$ where*

$$
\begin{aligned}
& r(x) \leq \eta \ \wedge \ r(x) > \eta \quad \vee \\
x \preceq_{\text{con}} y :\Leftrightarrow \ & x \preceq_{\text{par}} y \ \wedge \ (r(x) \leq \eta \ \wedge \ r(y) \leq \eta \ \vee \ r(x) = r(y)) \quad \vee \\
& r(x) < r(y) \ \wedge \ r(x) > \eta \ \wedge \ r(y) > \eta
\end{aligned}
\tag{9}
$$

denotes the preference relation for the constrained approach under the robustness constraint η.

This definition for single solutions can be extended to sets according to the following definition:

Definition 3.2 (optimal set under a robustness constraint)**.** *A set $A^* \in \Psi$ with $|A^*| \leq \alpha$ is optimal with respect to the robustness constraint η, if it fulfills*

$$
A^* \in \{A \in \Psi \mid \forall B \in \Psi \text{ with } |B| \leq \alpha : A \preccurlyeq_{con} B\} \tag{10}
$$

where $\preccurlyeq_{con}$ denotes the extension of the relation $\preceq_{con}$ (Eq. 9) to sets according to Eq. 1.

In the following, a solution x whose robustness $r(x)$ does not exceed the constraint, i.e., $r(x) \leq \eta$, is referred to as robust and to all other solutions as non-robust (Deb et al., 2002a).

Fig. 1(c) shows the allocation of solutions to fronts according to $\preceq_{\text{con}}$. The robustness constraint is set to $\eta = 1$, rendering all solutions with $r(x) \leq 1$ robust and with $r(x) > 1$ non robust, i.e., only *h*,*i*, and *j* are robust. In cases where solutions are considered robust or share the same robustness (*a*, *b*, and *e*), the partitioning corresponds to weak Pareto dominance on objective values. In all the remaining cases, partitioning is done according to the robustness value which leads to fronts independent of the objectives and containing only solutions of the same robustness $r(x)$.

3.4 Extension of the Hypervolume Indicator to Integrate Robustness Considerations

The three approaches presented above all allow to consider robustness in a way that is inherent to the algorithm. The first two approaches (Sec. 3.1 and 3.2) have a—predefined—way of trading off the robustness with the objective values. On the other hand, the constraint approach (Sec. 3.3) does not trade-off robustness, but rather optimizes with respect to a given robustness constraint. In this section a new approach is presented, which offers a larger degree of flexibility with respect to two important points: firstly, the concept allows to realize different trade-offs, which are not inherent to the concept, but rather can be defined by the decision maker, and secondly, even when trading-off robustness with objective values the optimization can be focused on a target robustness level.

The three approaches presented so far rely on modifying the dominance relation or the objective values to account for robustness. On solutions which are incomparable, the hypervolume indicator is then used to refine the respective dominance relation. That means, the robustness of solutions is not directly influencing the hypervolume calculation. In the following, a new concept is proposed: first, non-dominated sorting is carried out as for the regular hypervolume indicator, but according to the robustness integrating preference relation $\preceq_{\text{ao}}$ given by Eq. 8 on page 7. Then, an extension of the regular hypervolume indicator is calculated. The novel *robustness integrating hypervolume indicator* $I_H^{\varphi,w}(A, R)$ is based on the objective values of solutions in A, but also on the robustness values of the solutions. An additional desirability function thereby allows to trade-off robustness and quality of solutions in almost any way, including the three approaches presented in Sec. 3.1 to 3.3, as well as not considering robustness at all.

3.4.1 Methodology

The idea behind $I_H^{\varphi,w}$ is to modify the attainment function $\alpha_A(z)$ of the original hypervolume indicator definition, see Def. 2.1, in such a way that it reflects the robustness of solutions. In the original definition of the attainment function, $\alpha_A(z)$ is either 0 or 1; for any objective vector z not dominated by A, the attainment function is zero, while for a dominated vector z, $\alpha_A(z) = 1$ holds. Hence, a solution $x \in A$ always contributes 100% to the overall hypervolume, regardless of the robustness of the solution. To integrate robustness, the codomain of $\alpha_A(z)$ is extended to all values between 0 and 1. The new robustness integrating attainment function α_A^{φ} thereby is still zero for any objective vector z not dominated by A. In contrast to Def. 2.1, however, dominated objective vectors z are accounted based on the most robust solution dominating z. A desirability function of robustness φ determines the value of solutions, ranging from 0 (no contribution)

to 1 (maximum influence)[3].

Definition 3.3 (Desirability function of robustness)**.** *Given a solution $x \in A$ with robustness $r(x) \in \mathbb{R}_{\geq 0}$, the desirability function $\varphi : \mathbb{R}_{\geq 0} \to [0,1]$ assesses the desirability of a robustness level. A solution x with $\varphi(r(x)) = 0$ thereby represents a solution of no avail due to insufficient robustness. A solution y with $\varphi(r(y)) = 1$, on the other hand, is of maximum use, and further improving the robustness would not increase the value of the solution.*

Provided a function φ, the attainment function can be extended in the following way to integrate robustness:

Definition 3.4 (Robustness integrating attainment function α_A^φ)**.** *Given a set of solutions $A \in \Psi$, the robustness integrating attainment function $\alpha_A^\varphi : Z \to [0,1]$ for an objective vector $z \in Z$, and a desirability function $\varphi : r(x) \mapsto [0,1]$ is*

$$\alpha_A^\varphi(z) := \begin{cases} \varphi\big(\min_{x \in A, f(x) \leqslant z} r(x)\big) & \text{if } A \leqslant \{z\} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

Hence, the attainment function of z correspond to the desirability of the most robust solution dominating z; and is 0 if no solution dominates z.

Finally, the robustness integrating hypervolume indicator corresponds to the established definition except for the modified attainment function according to Def. 3.4:

Definition 3.5 (robustness integrating hypervolume indicator)**.** *The robustness integrating hypervolume indicator $I_H^{\varphi,w} : \Psi \to \mathbb{R}_{\geq 0}$ with reference set R, weight distribution function $w(z)$, and desirability function φ is given by*

$$I_H^{\varphi,w}(A) := \int_{\mathbb{R}^d} \alpha_A^\varphi(z) w(z) dz \tag{12}$$

where $A \in \Psi$ is a set of decision vectors.

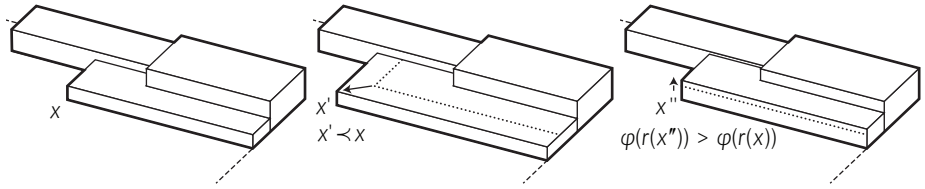
In the following, $I_H^{\varphi,w}$ is used to refer to the robustness integrating hypervolume indicator, not excluding an additional weight distribution function to also incorporate user preference. The desirability function φ not only serves to extend the hypervolume indicator, but implies a robustness integrating preference relation:

Definition 3.6. *Let $x, y \in X$ be two solutions with robustness $r(x)$ and $r(y)$ respectively. Furthermore, let φ be a desirability function $\varphi : r(x) \mapsto \varphi(r(x))$. Then x weakly dominates y with respect to φ, denoted $x \preceq_\varphi y$, iff $x \preceq_{\text{par}} y$ and $\varphi(r(x)) \geq \varphi(r(y))$ holds.*

Since a solution x can be in relation $\preceq_\varphi$ to y only if $x \preceq_{\text{par}} y$ holds, $\preceq_\varphi$ is a subrelation of $\preceq_{\text{par}}$, and generally increases the number of incomparable solutions. In order that $\preceq_\varphi$ is a reasonable relation with respect to Pareto dominance and robustness φ has to be monotonically decreasing as stated in the following Theorem:

Theorem 3.7. *As long as φ is a (not necessarily strictly) monotonically decreasing function, and smaller robustness values are considered better, the corresponding robustness integrating hypervolume indicator given in Def. 3.5 (a) induces a refinement of the extension of $\preceq_\varphi$ to sets, and (b) is sensitive to any improvement of non dominated solutions x with $\varphi(r(x)) > 0$ in terms of objective value or the desirability of its robustness.*

[3]The definition of desirabilty function used in this study is compliant with the definition known from statistical theory, cf. Abraham (1998).



(a) original set A (b) objective value of x improved (c) robustness of x improved

Figure 2: The robustness integrating hypervolume indicator is sensitive to improvements of objective values (b) as well as to increased robustness desirability (c).

Proof. Part 1: the robustness integrating hypervolume is compliant with the extension of $\preceq_\varphi$ to sets. Let $A, B \in \Psi$ denote two sets with $A \preceq_{\text{rob}} B$. More specifically this means, for all $y \in B$ $\exists x \in A$ such that $x \preceq_{\text{par}} y$ and $r(x) \leq r(y)$. Now let $y'_B(z) := \arg\min_{y \in B, f(y) \leqslant z} r(y)$. Then $\exists x'_A(z) \in A$ with $x'_A(z) \preceq_{\text{rob}} y'_B(z)$. This leads to $f(x'_A(z)) \leqslant f(y'_B(z)) \leqslant z$ and $r(x'_A) \leq r(y'_B)$. The latter boils down to $\varphi(r(x'_A)) \geq \varphi(r(y'_B))$, hence $\alpha^\varphi_A(z) \geq \alpha^\varphi_B(z)$ for all $z \in Z$, and $I_H^{\varphi,w}(A) \geq I_H^{\varphi,w}(B)$.

Part 2: the Def. 3.5 is sensitive to improvements of objective value and desirability. Let $x \in A$ denote the solution which is improved, see Fig. 2(a). First, consider the case where in a second set A', x is replaced by x' with $r(x') = r(x)$ and $x' \prec_{\text{par}} x$. Then there exists a set of objective vectors W which is dominated by $f(x')$ but not by $f(x)$. Because of $\varphi(r(x)) > 0$, the gained space W increases the overall hypervolume, see Fig. 2(b). Second, if x is replaced by x'' with the same objective value but a higher desirability of robustness, $\varphi(r(x'')) > \varphi(r(x))$, the space solely dominated by x'' has a larger contribution due to the larger attainment value in this area, and again the hypervolume indicator increases, see Fig. 2(c). □

Note that choices of φ are not excluded for which the attainment function $\alpha^\varphi_A(z)$ can become 0 even if a solution $x \in A$ dominates the respective objective vector z—namely if all solution dominating z are considered infeasible due to their bad robustness. Provided that φ is chosen monotonically decreasing, many different choices of desirability are possible. Here, the following class of functions is proposed, tailored to the task of realizing the approaches presented above. Besides the robustness value, the function takes the constraint η introduced in Sec. 3.3 as an additional argument. A parameter θ defines the shape of the function and its properties:

$$\varphi_\theta(r(x), \eta) = \begin{cases} \left(\frac{r(x)}{r_{max}} - 1\right)\theta + (1+\theta)H_1(\eta - r(x)) & \theta \leq 0 \\ \exp\left(3 \cdot \frac{r(x)-\eta}{\eta \log(1-\theta)}\right) & 0 < \theta < 1, r(x) > \eta \\ 1 & \text{otherwise} \end{cases} \tag{13}$$

where $H_1(x)$ denotes the Heaviside function[4], and r_{max} denotes an upper bound of the robustness measure. The factor 3 in the exponent is chosen arbitrarily, and only serves the purpose of producing a nicely shaped function. By changing the shape parameter

[4] $H_1(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$

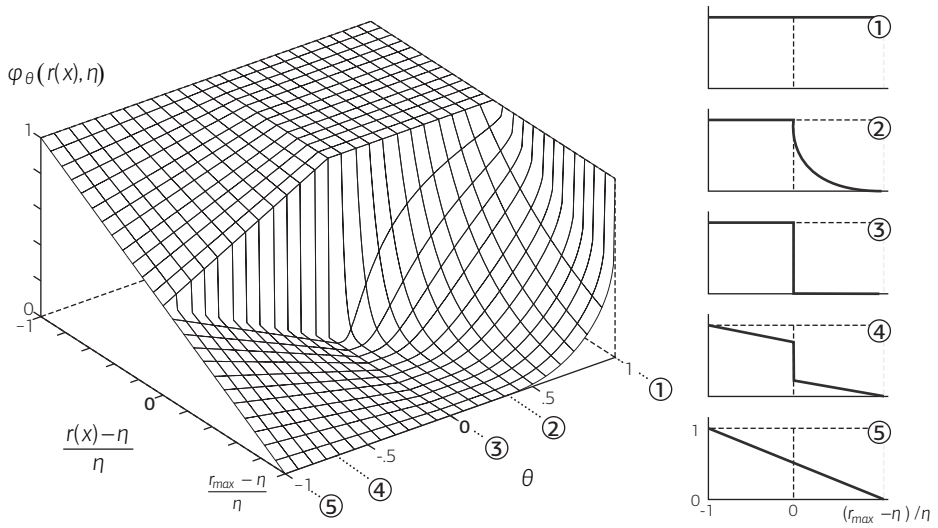


Figure 3: Desirability function $\varphi_\theta(r(x),\eta)$ according to Eq. 13. By changing the shape parameter θ, different shapes of φ can be realized as shown in the cross-sectional plots ① to ⑤. The robustness measure $r(x)$ has been normalized to η, such that only solutions with $r(x) \leq 0$ are classified robust.

θ, different characteristics of φ can be realized that lead to different ways of trading off robustness versus objective value, see Fig. 3:

① $\theta = 1$: For this choice, $\varphi_1(r(x),\eta) \equiv 1$. This means the robustness of solutions is not considered at all.

② $0 < \theta < 1$**:** All solutions with $r(x) \leq \eta$ are maximally desirable in terms of robustness. For non robust solutions, the desirability decreases exponentially with exceedance of $r(x)$ over η, where smaller values of θ lead to a faster decay. This setting is similar to the simulated annealing approach that will be presented in Sec 4.3.2 with two major differences: first, the robustness level is factored in deterministically, and secondly, the robustness level is traded-off with the objective value, meaning a better quality of the latter can compensate for a bad robustness level.

③ $\theta = 0$**:** In contrast to case ②, all solutions exceeding the robustness constraint are mapped to zero desirability, and therefore do not influence the hypervolume calculation. This corresponds to the original constraint approach from Sec. 3.3.

④ $-1 < \theta < 0$**:** Negative choices of θ result in robust solutions getting different degrees of desirability, meaning only perfectly robust solutions ($r(x) = 0$) get the maximum value of 1. The value linearly decreases with $r(x)$ and drops passing over the constraint η, where the closer θ is to zero the larger the reduction. The value then further decreases linearly until getting zero for r_{max}.

⑤ $\theta = -1$**:** In contrast to ④, the desirability φ continuously decreases linearly from $\varphi_{-1}(0,\cdot) = 1$ to $\varphi_{-1}(r_{max},\cdot) = 0$ without drop at $r(x) = \eta$. This corresponds to considering robustness as an additional objective, see Sec. 3.2.

Calculating the generalized hypervolume indicator in Eq. 12 can be done in a similar fashion as for the regular hypervolume indicator by using the 'hypervolume by

slicing objectives' approach (Zitzler, 2001; Knowles, 2002; While et al., 2006), where for each box the desirability function needs to be determined. Faster algorithms, for instance Beume and Rudolph (2006), can be extended to Def. 12 as well; however, it is not clear how the necessary adjustments affect the runtime.

3.5 Discussion of the Approaches

All three existing approaches mentioned in the introduction can be translated to hypervolume-based search without major modifications necessary. Modifying the objective functions offers a way to account for uncertainty without changing the optimization problem, such that any multiobjective optimization algorithm can be still applied. In particular, the hypervolume indicator is directly applicable. However, no explicit robustness measure can be considered, nor can the search be restricted to certain robustness levels. The latter also holds when treating robustness as an additional objective. The advantage of this approach lies in the diversity of robustness levels that are obtained. On the flip side of the coin, many solutions might be unusable because they are either not robust enough or have a very bad objective value to achieve an unessential high robustness. Furthermore, the number of nondominated solutions increases, which can complicate search and decision making. Realizing robustness as an additional constraint allows to focus on one very specific level of robustness, thereby searching more targeted which potentially leads to a more efficient search. However, focusing can be problematic if the required level can not be fulfilled.

All approaches pursue different optimization goals, such that depending on the decision maker's preference, one or another approach might be appropriate. The extended hypervolume indicator constitutes the most flexible concept, as it allows to realize arbitrary desirability functions the decision maker has with respect to robustness of a solution. All three conventional approaches are thereby special realizations of desirability functions, and can be realized by the robustness integrating hypervolume indicator.

4 Search Algorithm Design

Next, algorithms are presented that implement the concepts presented in Sec. 3. First, the three conventional concepts are considered, where for the constraint approach two modifications are proposed. Secondly, the generalized hypervolume indicator is tackled, and an extension of the Hypervolume Estimation Algorithm for Multiobjective Optimization (HypE), see Bader and Zitzler (2009), is derived such that the indicator is applicable to many objective problems.

4.1 Modifying the Objective Functions

As discussed in Sec. 3.1, when modifying the objective functions to consider robustness, any multiobjective algorithm—hypervolume-based algorithms in particular—can be applied without any adjustments necessary. Hence, for instance SIBEA (Zitzler et al., 2007) can be employed as it is.

4.2 Additional Objectives

Only minor adjustments are necessary to consider robustness as an additional objective: since the number of objectives increases by one, the reference point or the reference set of the hypervolume indicator need to be changed. In detail, each element of the reference set needs an extra coordinate resulting in $d+1$ dimensional vectors. Due to the additional objective, the computational time increases, and one might have to switch to approximations schemes, e.g., use HypE (Bader and Zitzler, 2009) instead of the exact hypervolume calculation (Beume and Rudolph, 2006; While et al., 2005).

4.3 Additional Robustness Constraints

In the following, different approaches to consider robustness as an additional constraints are discussed. First, in Sec. 4.3.1, a baseline algorithm is presented that optimizes according to Def. 3.2. This approach directly employs the dominance relation presented in Sec. 3.3. As will be discussed, this approach presents the risk of premature convergence, which is addressed by three different advanced approaches in Sec. 4.3.2

4.3.1 Baseline Approach

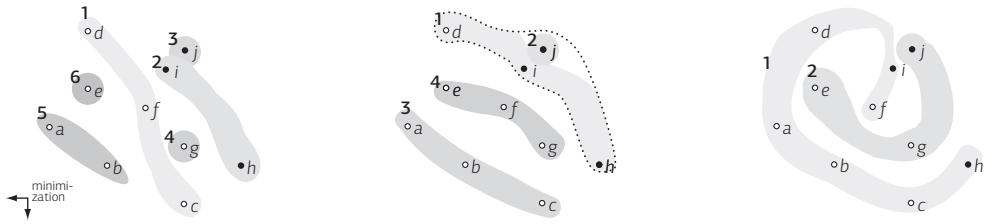
In order to realize the plain constraint approach, as illustrated in Sec. 3.3, in hypervolume-based search, the only change to be made concerns the dominance ranking, where the relation shown in Eq. 9 is employed instead of $\preceq_{\text{par}}$, see Fig. 1(c). In the constraint approach as presented in Sec. 3.3, a robust solution thereby is always preferred over a non-robust solution regardless of their respective objective value. This in turn means, that the algorithm will never accept a non robust solution in favor of a more robust solution. Especially for very rigid robustness constraints $\eta \ll 1$ this carries a certain risk of getting stuck early on in a region with locally minimal robustness, which does not even need to fulfill the constraint η. To attenuate this problem, next three modifications of the baseline algorithm are proposed that loosen up the focus on a robustness constraint.

4.3.2 Advanced Methods

The first modification of the baseline approach is based on relaxing the robustness constraint at the beginning of search; the second algorithm does not introduce robustness into some parts of the set which is thus allowed to converge freely even if its elements exceed the robustness constraint. Finally, a generalization of the constraint method is proposed that allows to focus on multiple robustness constraints at the same time.

Approach 1 — Simulated Annealing The first algorithm uses the principle of simulated annealing when considering robustness with respect to a constraint η. In contrast to the baseline approach, also solutions exceeding the robustness constraint can be marked robust. The probability in this case thereby depends on the difference of the robustness $r(x)$ to the constraint level η, and on a temperature T:

$$P(x \text{ robust}) = \begin{cases} 1 & r(x) \leq \eta \\ u \leq e^{-(r(x)-\eta)/T} & \text{otherwise} \end{cases} \tag{14}$$



(a) simulated annealing (b) reserve algorithm (c) generalized hypervolume

Figure 4: Partitioning into fronts of same ten solutions from Fig. 1 for the two advanced constraint methods (a), (b), and for the generalized hypervolume indicator. The solid dots represents robust solutions at the considered level of $\eta = 1$ while the unfilled dots represent non-robust solutions. For (a), solutions d, f, and c are classified robust too.

where $u \sim U(0,1)$ is uniformly distributed within 0 and 1. The temperature T is exponentially decreased every generation, i.e., $T = T_0 \cdot \gamma^g$ where g denotes the generation counter, $\gamma \in]0,1[$ denotes the cooling rate, and T_0 the initial temperature. Hence, the probability of non robust solutions being marked robust decreases towards the end of the evolutionary algorithm. In the example shown in Fig. 4, the solutions d, f, and c are classified as robust—although exceeding the constraint $\eta = 1$. Since these solutions Pareto-dominate all (truly) robust solutions, they are preferred over these solutions unlike in the baseline algorithm, see Sec. 3.3.

Approach 2 — Reserve Approach The second idea to overcome locally robust regions is to divide the population into two sets: on the first one no robustness considerations are imposed, while for the second set (referred to as the reserve) the individuals are selected according to the baseline constraint concept. This enables some individuals, namely those in the first set, to optimize their objective value efficiently. Although these individuals are very likely not robust, they can improve the solutions from the second set in two ways: (i) a high quality solution from the first set gets robust through mutation or crossover and thereby improves the reserve, (ii) the objective values of a robust solution are improved by crossover with an individual from the first set. However, since at the end only the reserve is expected to contain individuals fulfilling the constraint, one should choose the size of the reserve β to contain a large portion of the population, and only assign few solutions to the first set where robustness does not matter.

In detail, the reserve algorithm proceeds as follows. First, the membership of a solution to the reserve is determined; a solution x is included in the reserve, denoted by the indicator function $\chi_{\text{rsv}}(x)$, if either it is robust and there are less than $\beta - 1$ other solutions that are also robust and dominate x; or if x is not robust but still is among the β most robust solutions. Hence

$$\begin{aligned} \chi_{\text{rsv}}(x) = 1 :\Leftrightarrow & r(x) \leq \eta \wedge \left|\{y \preceq_{\text{par}} x \,|\, y \in X, r(y) \leq \eta\right| \leq \beta \quad \vee \\ & r(x) > \eta \wedge |\{y \,|\, y \in X, r(y) \leq r(x)| \leq \beta \end{aligned} \tag{15}$$

Given the membership to the reserve, the preference relation is:

$$x \preceq_{\text{rsv}} y :\Leftrightarrow \begin{cases} 1 & \chi_{\text{rsv}}(x) = 1 \wedge \chi_{\text{rsv}}(y) = 0 \\ 0 & \chi_{\text{rsv}}(x) = 0 \wedge \chi_{\text{rsv}}(y) = 1 \\ x \preceq_{\text{par}} y & \text{otherwise} \end{cases} \tag{16}$$

For the example in Fig. 4(b) let the reserve size be $\beta = 4$, leaving one additional place not subject to robustness. Because there are fewer solutions which fulfill the robustness constraint than there are places in the reserve, all three robust solutions are included in the reserve, see dashed border. In addition to them, the next most robust solution (d) is included to complete the reserve. Within the reserve, the solutions are partitioned according to their objective value. After having determined the reserve, all remaining solutions are partitioned based on their objective value.

Approach 3 — Multi-Constraint Approach So far, robustness has been considered with respect to one robustness constraint η only. However, another scenario could include the desire of the decision maker to optimize multiple robustness constraint at the same time. This can make sense for different reasons: (i) the decision maker wants to learn about the problem landscape, i.e., he likes to know for different degrees of robustness the objective values that can be achieved; (ii) the decision maker needs different degrees of robustness, for instance because the solution are implemented for several application areas that have different robustness requirements, and (iii) premature convergence should be avoided.

The optimize according to multiple robustness constraints, the idea is to divide the population into several groups, which are subject to a given constraint. In the following the baseline algorithm from Sec. 4.3.1 is used as a basis. The proposed concept not only allows to optimize different degrees of robustness at the same time, but also to put a different emphasis on the individual classes by predefining the number of solutions that should have a certain robustness level. Specifically, let $C = \{(\eta_1, s_1), \ldots, (\eta_k, s_l)\}$ denote a set of l constraints $\eta_1, \ldots, \eta_l$ where for each constraint the user defines the number of individuals $s_i \in \mathbb{N}_{>0}$ that should fulfill the respective constraint η_i (excluding those individuals already belonging to a more restrictive constraint). Hence, $c_1 + \cdots + c_l = |P|$, and without loss of generality let assume $\eta_1 < \eta_2 < \cdots < \eta_l$. The task of an algorithm is then to solve the following problem:

Definition 4.1 (optimal set under multiple robustness constraint)**.** *Consider $C = \{(\eta_1, s_1), \ldots, (\eta_k, s_l)\}$, a set of l robustness constraints η_i with corresponding size s_i. Then a set $A^* \in \Psi_\alpha$, i.e., $|A^*| \leq \alpha$, is optimal with respect to C if it fulfills $A^* \in \{A \in \Psi_\alpha \mid \forall B \in \Psi_\alpha : A \preccurlyeq_C B\}$ where $\preccurlyeq_C$ is given by*

$$A \preccurlyeq_C B :\Leftrightarrow \forall (\eta_i, s_i) \in C : \forall B' \in B_{s_i}\, \exists A' \in A_{s_i} \text{ s.t. } A' \preccurlyeq_{\eta_i} B' \tag{17}$$

where $\preccurlyeq_{\eta_i}$ denotes the extension of any relation proposed in Sec. 3 to sets, as stated in Eq. 9.

In order to optimize according to Def. 4.1, Algorithm 1 is proposed: beginning with the most restrictive robustness level η_1, one after another s_i individuals are added to the new population. Thereby, the individual increasing the hypervolume at the current robustness level η_i the most is selected. If no individual increases the hypervolume, the most robust is chosen instead.

Algorithm 1 Classes algorithm based on the greedy hypervolume improvement principle. Beginning with the most robust class, solutions are added to the final population P' that increase the hypervolume at the respective level the most, given the individuals already in P'.

Require: Population P, list of constraint classes $C = \{(\eta_1, s_1), \ldots, (\eta_l, s_l)\}$, with $\eta_1 \leq \cdots \leq \eta_l$.

```
P' = {}
for i = 1 to l do                                   ▷ iterate over all classes (η_i, s_i) ∈ C
    for j = 1 to s_i do                             ▷ fill current class
        x' ← arg max_{x∈P\P'} I_H^{φ(·,η_i),w}(x ∪ P', R)
        if I_H^{φ(·,η_i),w}(x' ∪ P', R) = I_H^{φ(·,η_i),w}(P', R) then   ▷ x' has no contribution
            x' ← arg min_{x∈P\P'} r(x)              ▷ get the most robust instead
        P' ← P' ∪ x'
return P'
```

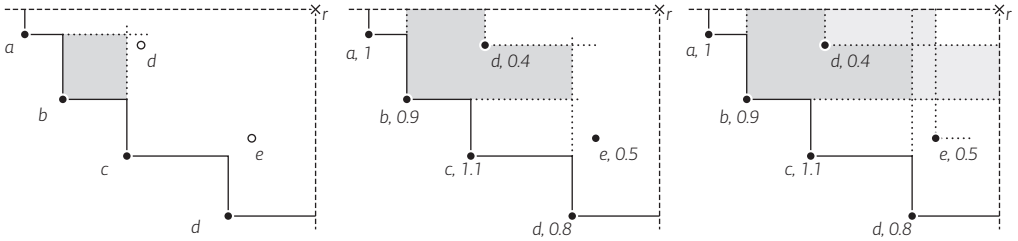


Figure 5: In (a) the affected hypervolume region when removing b is shown, if robustness is not considered (dark gray). Adding the consideration of robustness (values next to solution labels), the affected region increases (b). Foreseeing the removal of two other solutions apart from b, other regions dominated by b might also be lost (light gray areas).

4.4 Optimizing the Generalized Hypervolume Indicator

To optimize according to the generalized hypervolume indicator, the same greedy procedure as presented in Sec. 2.1 can be used. Thereby, two differences arise:

1. first off, non-dominated sorting is done according to $\preceq_\varphi$ (Def. 3.6) and not with respect to $\preceq_{\text{par}}$. In Fig. 5, for instance, the solutions d and e are in different fronts than a for $\preceq_{\text{par}}$ (Fig. 5(a)), but belong to the same front for $\preceq_\varphi$ (Fig. 5(a));
2. secondly, the hypervolume loss is calculated according to the new indicator, i.e., the loss is $I_H^{\varphi,w}(A, R) - I_H^{\varphi,w}(A \backslash x, R)$, see gray shaded areas in Figures 5(a) and 5(b).

In this study, however, we use an advanced selection procedure introduced by HypE in Bader and Zitzler (2008, 2009). It performs regular non-dominated sorting, but uses an advanced fitness calculation scheme; rather than considering the loss when removing the respective solution, this scheme tries to estimates the expected loss taking into account the removal of additional solutions, see Fig. 5(c).

Although the exact calculation of this fitness is possible, in this study the focus is on its approximation by Monte Carlo sampling, as also implemented in HypE. The basic idea is to first determine a sampling space S. From this sampling space, M samples then are drawn to estimate the expected hypervolume loss. The derivation of the necessary adjustments to consider the robustness integrating hypervolume indicator are rather complicated and are therefore moved to the Appendix A.1. Altogether, the HypE routine to consider robustness corresponds to the regular HypE algorithm as listed in Bader and Zitzler (2008, 2009, Algorithm 3) except for the modified fitness calculation. Algorithm 2 shows the new code, where only Lines 15 to 29 differ from the original definition (Lines 15 to 27).

The advantage of using HypE is that the algorithms allows to also optimize many objective function in reasonable time. For this reason, in the experimental validation HypE will be used to optimize according to $I_H^{\varphi,w}$, where the desirability function φ given in Eq. 13 is used.. The parameter θ of this function is thereby either fixed, or geometrically decreased in each generation from 1 to $\theta_{\text{end}} \in]0,1]$, i.e., in generation g, θ corresponds to $\theta_g = \gamma^g$ with $\gamma = \sqrt[g_{\max}]{\theta_{\text{end}}}$.

5 Experimental Validation

In the following experiments, the algorithms from Sec. 4 are compared on two test problem suites and on a real world bridge truss problem presented in Appendix A.2. The different optimization goals of the approaches rule out a fair comparison as no single performance assessment measure can do justice to all optimization goals. Nonetheless, the approaches are compared on the optimality goal shown in Def. 3.2, which will favor the constraint approach. Yet, the approach presented in Sec. 3.1 is excluded from the experimental comparison, since the approach is not based on a robustness measure $r(x)$.

The following goals will be pursued by visual and quantitative comparisons:

1. the differences between the three existing approaches (see page 3) are shown;
2. it is investigated, how the extended hypervolume approach performs, and how it competes with the other approaches, in particular, the influence of the desirability function is investigated;
3. it is is examined, whether the multi-constraint approach from Sec. 4.3.2 has advantages over doing independent runs or considering robustness as an additional objective.

5.1 Experimental Setup

The performance of the algorithms is investigated with respect to optimizing a robustness constraint η. The following algorithms are compared:

- as a baseline algorithm, HypE without robustness consideration, denoted $\text{HypE}_{no.\,rob.}$;
- Alg_{ao} using an additional objective;
- the constraint approaches from Sec. 4.3, i.e., baseline Alg_{con}, simulated annealing $Alg_{sim.\,ann.}$, reserve approach Alg_{rsv}, and multiple classes $Alg_{classes}$;
- HypE using the generalized hypervolume indicator, see Sec. 4.4.

Algorithm 2 Hypervolume-Based Fitness Value Estimation for $I_H^{\varphi,w}$

Require: population $P \in \Psi$, reference set $R \subseteq Z$, fitness parameter $k \in \mathbb{N}$, number of sampling points $M \in \mathbb{N}$. Returns an estimate of the (robustness integrating) hypervolume contributions $\mathcal{F}$.

procedure *estimateHypervolume*(P, R, k, M)
  for $i \leftarrow 1, d$ **do** ▷ determine sampling box S
    $l_i = \min_{a \in P} f_i(a)$
    $u_i = \max_{(r_1,\ldots,r_d) \in R} r_i$
  $S \leftarrow [l_1, u_1] \times \cdots \times [l_d, u_d]$
  $V \leftarrow \prod_{i=1}^{d} \max\{0, (u_i - l_i)\}$ ▷ volume of sampling box
  $\mathcal{F} \leftarrow \bigcup_{a \in P} \{(a, 0)\}$ ▷ reset fitness assignment
  for $j \leftarrow 1, M$ **do** ▷ perform sampling
    choose $s \in S$ uniformly at random
    if $\exists r \in R : \ s \leq r$ **then**
      $p \leftarrow |P|$ ▷ population size
      $UP \leftarrow \bigcup_{a \in P, \ f(a) \leq s} \{f(a)\}$ ▷ solutions dominating sample s
      $e \leftarrow$ elements of UP sorted such that $r(e_1) \leq \cdots \leq r(e_n)$
      $n \leftarrow |UP|$ ▷ number of solutions dominating the sample
      for $v = 0$ to $n - 1$ **do** ▷ check all cutoff levels
        $q \leftarrow n - v - 1$
        $\alpha \leftarrow P_v(p, q, k)$ ▷ probability of loss according to Eq. 21
        for $f = 1$ to $n - v$ **do** ▷ update fitness of all contributing solutions
          if f equals $n - v$ **then** ▷ least robust solution
            $inc \leftarrow \alpha \cdot (\varphi(r(e_f)))$
          else ▷ slice to less robust solution $f + 1$
            $inc \leftarrow \alpha \cdot (\varphi(r(e_f) - \varphi(r(e_{f+1}))))$
          for $j = 1$ to f **do** ▷ update fitness
            $(a, v) \leftarrow (a, v) \in \mathcal{F}$ where $a \equiv e_f$
            $\mathcal{F}' \leftarrow \left(\mathcal{F}' \setminus (a, v)\right) \cup (a, v + inc/f)$
      $\mathcal{F} \leftarrow \mathcal{F}'$
  return $\mathcal{F}$

So far, the focus was on environmental selection only, i.e., the task of selecting the most promising population P' of size α from the union of the parent and offspring population. To generate the offspring population, random mating selection is used, although the principles proposed for environmental selection could also be applied to mating selection. From the mating pool SBX and Polynomial Mutation, see Deb (2001), generate new individuals.

The first test problem suite used is WFG (Huband et al., 2006), and consists of nine well-designed test problems featuring different properties that make the problems hard to solve—like non-separability, bias, many-to-one mappings and multimodality. However, these problems are not created to have specific robustness properties and the robustness landscape is not known. For that reason, six novel test problems are proposed called BZ that have different, known robustness characteristics, see Appendix A.2.2. These novel problems allow to investigate the influence of different robustness landscapes on the performance of the algorithms. In addition to the two test problems

Table 1: Parameter setting used for the experimental validation. The number of generations was set to 1000 for the test problems, and to 10000 for the bridge problem.

parameter	value	*continued*	
η_{mutation}	20	population size α	25
$\eta_{\text{crossover}}$	15	number of offspring μ	25
individual mutation prob.	1	number of generations g	1000/10000
individual recombination prob.	0.5	perturbation δ	0.01
variable mutation prob.	$1/n$	nr. of neighboring points H	25
variable recombination prob.	1	neighborhood size δ	0.01

suites, the algorithms are compared on a real world truss building problem stated in Appendix A.2.2, where also additional results on this problem are presented. For the robustness integrating HypE, see Sec. 4.4, the variant using a fixed θ is considered (denoted by $\text{HypE}_{\theta f}$), as well as the variant with θ decreasing in each generation to θ_{end}. This latter variant is referred to as $\text{HypE}_{\theta_{\text{end}}a}$.

5.1.1 Experimental Settings

The parameters η_{mutation} and $\eta_{\text{crossover}}$ of the Polynomial Mutation, and SBX operator respectively, as well as the corresponding mutation and crossover probabilities, are listed in Tab. 1. Unless noted otherwise, for each test problem 100 runs of 1000 generations are carried out. The population size α and offspring size μ are both set to 25. For the BZ robustness test problems, see Appendix A.2.2, the number of decision variables n is set to 10, while for the WFG test problems the recommendations of the authors are used, i.e., the number of distance related parameters is set to $l = 20$ and the number of position related parameters k is set to 4 in the biobjective case, and to $k = 2 \cdot (d - 1)$ otherwise. Except for Fig. 8, two objective are optimized.

In all experiments on the two test problem suites, the extend of the neighborhood $\mathcal{B}_\delta$ is set to $\delta = 0.01$. To estimate $f^w(x, \delta)$, for every solution 25 samples are generated in the neighborhood of x and $\hat{f}^w(x, \delta)$ is determined according to Eq. 6. After each generation, all solutions are resampled, even those that did not undergo mutation. This prevents that a solution which, only by chance, reaches a good robustness estimate, persists in the population. For the real world bridge problem, on the other hand, a problem specific type of noise is used that allows to analytically determine the worst case, see Appendix A.2.1.

For the $Alg_{sim.ann.}$ approach the cooling rate γ is set to 0.99. The reference set of the hypervolume indicator is set to $R = \{r\}$ with $r = (3, 5)$ on WFG, with $r = (6, 6)$ on WFG, and with $r = (0, 2000)$ on the bridge problem[5]. The $Alg_{classes}$-approach proposed in Sec. 4.3.2 uses the following constraints: for BZ $(\eta_1, \ldots, \eta_5) = (.01, .03, .1, .3, \infty)$. For WFG, due to generally higher robustness levels on these test problems, the classes were set to $(\eta_1, \ldots, \eta_5) = (.001, .003, .01, .03, \infty)$. In both cases, the class sizes were $(s_1, \ldots, s_5) = (4, 4, 6, 4, 6)$ which gives a populations size of 24. For the bridge problem, the classes are set to $(.001, .01, .02, 0.1, \infty)$ with 6 individuals in each class. The size of

[5]For this problem, the first objective is to be maximized.

the bridge is set to 8 decks, i.e., spanning a width of 40 m. For comparison with a single constraint it is set to $\eta = 0.02$. For each comparison, 100 runs of 10000 generations have been performed.

In this paper, two types of uncertainty are used. Firstly, for the test problems, where $x \in \mathbb{R}^n$ holds, X^p is assumed to be uniformly distributed within $B_\delta(x)$ according to Eq. 4. Random samples are generated within $B_\delta(x)$, and evaluated to obtain an estimate of the robustness measure $r(x)$. Secondly, for the real world application, a problem specific type of noise is considered as outlined in Appendix A.2.1. For this second type of noise, along with the structure of the problem, the worst case can be determined analytically.

5.1.2 Performance Assessment

For all comparisons, the robustness of a solutions has to be assessed. To this end, 10000 samples are generated within B_δ. For each objective separately, the 5% largest values are then selected. By these 500 values, the tail of a Generalized Pareto Distribution is fitted, see S. Kotz and S. Nadarajah (2001). The method of moments is thereby used to obtain a first guess, which is then optimized maximizing the log-likelihood with respect to the shape parameter k and the logarithm of the scale parameter, $\log(\sigma)$[6]. Given an estimate for the parameters $\hat{k}$ and $\hat{\sigma}$ of the Generalized Pareto Distribution, the worst case estimate $\hat{f}_i^w(x)$ is then given by

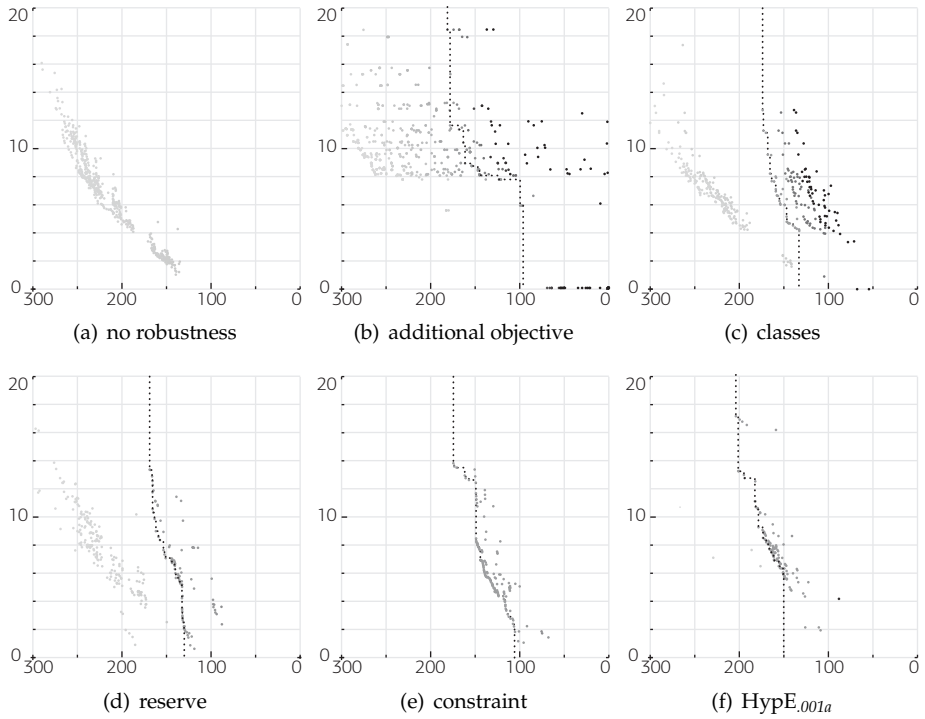
$$\hat{f}_i^w(x) = \begin{cases} \hat{\theta} - \hat{\sigma}/\hat{k} & \hat{k} < 0 \\ \infty & \textit{otherwise} \end{cases} \tag{18}$$

where $\hat{\theta}$ denotes the estimate of the location of the distribution given by the smallest value of the 5% percentile.

The performance of algorithms is assessed in the following manner: at first, a visual comparison takes place by plotting the objective values and robustness on the truss bridge problem Appendix A.2.1. The influence of θ of the robustness integrating HypE is then further investigated on BZ1. Secondly, all algorithms are compared with respect to the hypervolume indicator at the optimized robustness level η. To this end, the hypervolume of all robust solutions is calculated for each run. Next, the hypervolume values of the different algorithms are compared using the Kruskal-Wallis test and post hoc applying the Conover-Inman procedure to detect the pairs of algorithms being significantly different. The performance $P(\mathcal{A}_i)$ of an algorithm i then corresponds to the number of other algorithms, that are significantly better. See Bader and Zitzler (2008, 2009) for a detailed description of the significance ranking. The performance P is calculated for all algorithms, on all test problems of a given suite.

In addition to the significance rank at the respective level η, the mean rank of an algorithm when ranking all algorithms together is reported as well. The reason for plotting the mean rank instead of the significance is to also get an idea of the effect size of the differences—due to the large number of runs (100), differences might show up as significant although the difference is only marginal. The mean rank is reported not

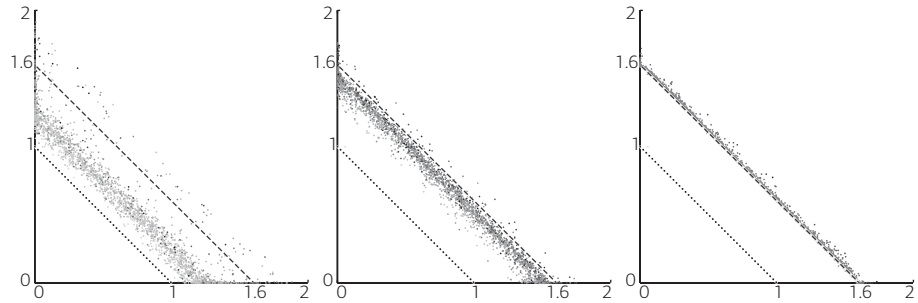
[6]Note that the maximum likelihood approximation is only efficient for $k \geq -1/2$ (S. Kotz and S. Nadarajah, 2001). Preliminary studies, however, not only showed $k \geq -1/2$ for all test problems considered, but also revealed that k is the same for all solutions of a given test problem.



(a) no robustness (b) additional objective (c) classes

(d) reserve (e) constraint (f) HypE$_{.001a}$

Figure 6: Pareto front approximations on the bridge problem for different algorithms. Since the first objective of the bridge problem, the structural efficiency, has to be maximized, the x-axis is reversed such that the figure agrees with the minimization problem. The robustness of a solution is color coded, lighter shades of gray stand for more robust solution. The dotted line represents the Pareto front of robust solutions (for (a), no robust solutions exists).



(a) HypE with fixed $\theta = 0.8$ (b) HypE with fixed $\theta = 0.1$ (c) HypE with fixed $\theta = 0.001$

Figure 7: Solutions of 100 runs each on BZ1. The dotted line indicates the Pareto front, the dashed line the best robust front.

only for the optimized level η, but a continuous range of other robustness levels as well to get an idea of the robustness distribution of the different algorithms.

Table 2: Comparison of HypE$_{.001a}$ and HypE$_{.1f}$ to different other algorithms for the hypervolume indicator. The number represent the performance score $P(A_i)$, which stands for the number of contenders significantly dominating the corresponding algorithm A_i, i.e., smaller values correspond to better algorithms. Zeros have been replaced by "·".

	Alg_{ao}	Alg_{con}	$Alg_{classes}$	HypE$_{.001a}$	HypE$_{.1f}$	HypE$_{no.\,rob.}$	Alg_{rsv}	$Alg_{sim.\,ann.}$
BZ1	6	·	5	·	·	7	3	4
BZ2	4	7	2	2	4	·	·	5
BZ3	·	1	1	2	2	7	3	1
BZ4	2	5	1	3	4	7	·	6
BZ5	4	·	4	·	·	7	1	6
BZ6	·	4	2	4	4	5	1	2
WFG1	4	1	·	1	4	7	1	4
WFG2	5	7	·	1	1	4	1	1
WFG3	1	2	·	2	3	3	2	2
WFG4	6	3	·	1	3	6	2	2
WFG5	6	1	·	1	1	7	2	3
WFG6	6	1	3	·	·	·	·	·
WFG7	7	·	5	·	·	4	3	·
WFG8	6	3	6	·	·	1	·	2
WFG9	6	1	·	1	1	6	4	5
Bridge 4	1	4	1	·	3	7	4	6
Bridge 6	3	4	2	·	·	7	4	6
Bridge 8	3	5	2	1	·	7	4	5
Bridge 10	3	4	2	·	·	7	3	5
Bridge 12	4	4	2	1	·	7	2	4
Total	**77**	**57**	**38**	**20**	**30**	**106**	**40**	**69**

5.2 Results

5.2.1 Visual Comparison of Pareto fronts

First, the algorithms are compared visually on the bridge problem (see Appendix A.2.1 with 8 decks. Fig. 6 shows the undominated solutions—according to relation $x \preceq_{\text{ao}} y$ in Eq. 8 on page 7—of 100 runs for algorithms presented in Sec. 4.

As the Pareto-set approximations in Fig. 6 show, a comparison of the different approaches is difficult: depending on how robustness is considered, the solutions exhibit different qualities in terms of objective values and robustness. It is up to decision maker to chose the appropriate method for the desired degree of robustness. The existing three approaches thereby constitute rather extreme characteristics. As the name implies, the HypE$_{no.\,rob.}$ approach only finds non-robust solutions, but in exchange converges further to the Pareto optimal front. However, the approaches Alg_{ao}, $Alg_{classes}$, and Alg_{rsv} considering robustness (at least for some solutions), outperform HypE$_{no.\,rob.}$ in some

regions even with respect to non-robust solutions: the Alg_{rsv} finds better solutions for low values of f_2, while the other two approaches outperform the $\text{HypE}_{no.\,rob.}$ algorithm on bridges around $f_2 = 10\,\text{m}$. Considering robustness as an additional objective leads to a large diversity of robustness degrees, however, misses solutions with small values of f_2. This might be due to the choice of the reference point of the hypervolume indicator, and the fact that only 25 solutions are used. The $Alg_{classes}$ algorithms optimizes five robustness degrees, two of which are classified as non-robust—these two are lying close together because the robustness level of the second last class with $\eta_4 = 0.1$ does barely restrict the objective values. Despite—or perhaps because of—the Alg_{rsv} algorithm also keeping non-robust solutions, the robust front is superior to the one of the Alg_{con} algorithm. The HypE approach with θ decreasing to .001 advances even further. The result of the $Alg_{sim.\,ann.}$ algorithm does not differ much from the Alg_{con} approach and is therefore not shown.

5.2.2 Influence of the Desirability Function φ

As the previous section showed, the robustness integrating HypE seems to, at least for small θ, have advantages over the Alg_{con}. Another advantage is its ability to adjust the trade-off between robustness and objective value quality. Fig. 7 shows the influence of different θ on the robustness and quality of the found solutions on BZ1. For this test problem, the robustness of solutions increases with distance to the (linear) Pareto front, see Appendix A.2.2. The new indicator allows to realize arbitrary trade-offs between robustness and objective values, ranging from solutions with good objective value to very robust solutions. In the present example, only when choosing $\theta < 0.1$, solutions robust at the constraint level are obtained. In the following a version with θ fixed to 0.1 is used (referred to as $\text{HypE}_{.1f}$), and a version with θ decreasing to 0.001 (referred to as $\text{HypE}_{.001a}$).

5.2.3 Performance Score over all Test Problems

To obtain a more reliable view of the potential of the different algorithms, the comparison is extended to all test problems. To this end, the performance score $P(\mathcal{A}_i)$ of an algorithm i is calculated as outline in Sec. 5.1.2. Tab. 2 shows the performance on the six BZ, the nine WFG, and five instances of the bridge problem. Overall, $\text{HypE}_{.001a}$ reaches the best performance, followed by $\text{HypE}_{.1f}$, $Alg_{classes}$, and Alg_{rsv}. four algorithms show a better performance than Alg_{con}.

Hence, not only are two modifications of the constraint approach ($Alg_{classes}$, Alg_{rsv}) able to outperform the existing constraint approach, but the robustness integrating hypervolume indicator as well is overall significantly better than Alg_{con}.

On the bridge problem, $\text{HypE}_{.001a}$ performs best and is significantly better than the other three algorithms. In case of Alg_{rsv}, and Alg_{con}, it is better in nearly 100% of all runs. On WFG, $Alg_{classes}$ overall performs best. An exception is WFG6-8, where the other three algorithms are all better. On BZ1, 3 and 5, $\text{HypE}_{.001a}$ and Alg_{con} perform best, on the remaing BZ problems Alg_{rsv} works best.

5.2.4 Application to Higher Dimensions

All algorithms proposed are not restricted to biobjective problems, but can also be used on problems involving more objectives. Since the runtime increases exponentially with

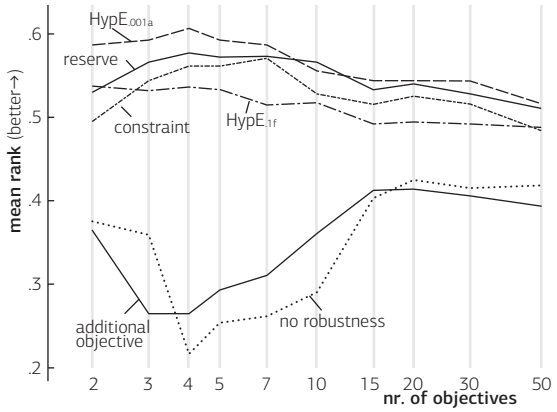


Figure 8: Average Kruskal-Wallis ranks over all WFG test problems at the robustness level $\eta = 0.01$ for different number of objectives.

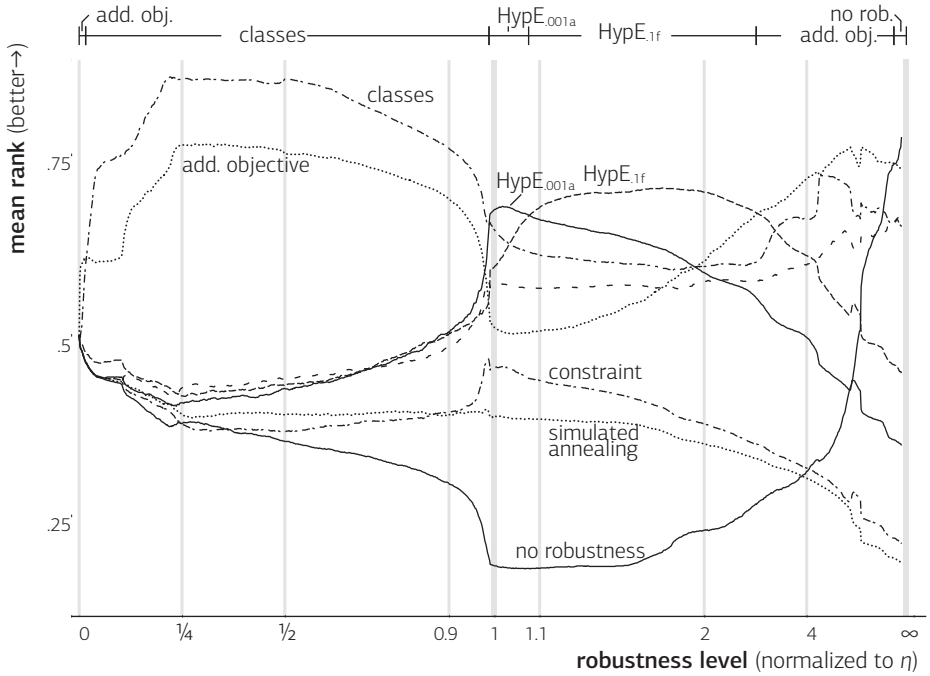


Figure 9: Average Kruskal-Wallis ranks over all BZ, WFG, and bridge test problems combined for different degrees of robustness. The best algorithm for a particular robustness level are indicated on top.

the number of objectives, HypE is applied to optimize the hypervolume when facing more than three objectives. Fig. 8 shows the mean Kruskal-Wallis rank for a selected subset of algorithms at different number of objectives. The algorithm $\text{HypE}_{.001a}$ shows the best performance except for 10 objectives, where the mean rank of the Alg_{rsv} approach is larger (although not significantly). Except for 7 and 20 objectives, $\text{HypE}_{.001a}$ is significantly better than Alg_{con}. On the other hand, $\text{HypE}_{.1f}$ performs worse than the constraint approach for all considered number of objectives except the biobjective case. This might indicate, that the parameter θ in Eq. 3 needs to be decreased with the number of objectives, because the trade-off between objective value and robustness is shifted towards objective value for larger number of objectives. However, further investigations need to be carried out to show the influence of θ when increasing the number of objectives.

5.2.5 Performance over Different Robustness Levels

In the previous comparisons, solutions robust at the predefined level have been considered. What happens if loosening or tightening up this constraint? Fig. 9 illustrates the mean hypervolume rank, normalized such that 0 corresponds to the worst, and 1 to the best quality. The mean ranks are shown for different levels of robustness ρ, normalized such that the center corresponds to the level optimized. For levels of robustness stricter than η, the $Alg_{classes}$ approach reaches the best hypervolume values. Around η, $\text{HypE}_{.001a}$ performs best and further decreasing the robustness level, $\text{HypE}_{.1f}$ overtakes. Further decreasing the robustness, Alg_{ao}, and finally $\text{HypE}_{no.\,rob.}$ are the best choices.

5.2.6 Optimizing Multiple Robustness Classes

Although the $Alg_{classes}$ approach proved useful even when only one of its optimized classes are considered afterwards, the main strength of this approach shows when actually rating the hypervolume of the different classes optimized. Tab. 3 list the significance rankings for the different classes averaged over all test problems of BZ, WFG, and truss bridge. Doing $Alg_{ind.\,runs}$ is significantly worse than the remaining approaches considered. This indicates, that optimizing multiple robustness levels concurrently is beneficial regardless of the robustness integration method used. Overall, the $Alg_{classes}$ approach reaches the best total significance score (69), the algorithms scores best on all classes except the one without robustness ($\eta = \infty$), where the $\text{HypE}_{no.\,rob.}$ outperforms the other algorithms.

6 Conclusions

This study has shown different ways of translating existing robustness concepts to hypervolume-based search, including (i) the modification of objective values, (ii) the integration of robustness as an additional objective, and (iii) the incorporation of robustness as an additional constraint. For the latter, three algorithmic modifications are suggested to overcome premature convergence. Secondly, an extended definition of the hypervolume indicator has been proposed that allows to realize the three approaches, but can also be applied to more general cases, thereby flexibly adjusting the trade-off between robustness and objective values while still being able to focus on a particular robustness level. To make this new indicator applicable to problems involving a large number of objectives, an extension of HypE (Hypervolume Estimation Algorithm for

Table 3: Comparison of the algorithms: $Alg_{ind.\,runs}$, Alg_{ao}, $\text{HypE}_{no.\,rob.}$, and $Alg_{classes}$. For each optimized class the sum of the performance score is reported for each of the three considered problem suites BZ, WFG, and the bridge problem.

		$Alg_{ind.\,runs}$	Alg_{ao}	$\text{HypE}_{no.\,rob.}$	$Alg_{classes}$
.01	BZ	0	3	3	3
	Bridge	8	10	3	3
.001	WFG	3	5	8	2
.03	BZ	10	4	17	2
	Bridge	8	10	1	4
.003	WFG	8	11	13	6
.1	BZ	13	4	17	1
	Bridge	8	10	1	4
.01	WFG	13	15	12	5
.3	BZ	10	4	17	2
	Bridge	10	8	0	5
.03	WFG	18	12	12	6
∞	BZ	17	9	3	5
	Bridge	12	6	1	5
	WFG	27	11	1	14
Total		**177**	**135**	**112**	**69**

Multiobjective Search) (Bader and Zitzler, 2008, 2009) has been presented.

The main results of this study can be summarized as follows:

- The three above mentioned approaches each have their advantages: modifying the objective values allows to use existing approaches without increasing the complexity of the problem; using an additional objective yields to a large diversity of different degrees of robustness, while using an additional constraint allows to focus on one desired robustness level.
- The existing constraint approach can be improved by the proposed algorithmic modifications, in particular by the reserve approach and by optimizing multiple classes; the latter furthermore illustrates that if all desired classes of robustness are known, it is better to optimize those concurrently instead of doing multiple independent runs or to generally consider robustness as an additional objective.
- The novel robustness integrating hypervolume indicator seems to offers many advantages: first, the concept allows to realize not only the three existing approaches, but also other arbitrary trade-offs the decision maker expresses. Second, this new approach—if concentrating on a single robustness constraint—is able to produce better solutions than the baseline constraint approach.

The investigations presented in this paper are not only valuable in the robustness context; in fact, they represent general ways to incorporate additional criteria and constraints in hypervolume-based search. A promising direction for future research is in particular how multiple (and not only a single) constraints can be integrated in a general manner.

A Appendix

A.1 HypE for the Generalized Hypervolume Indicator

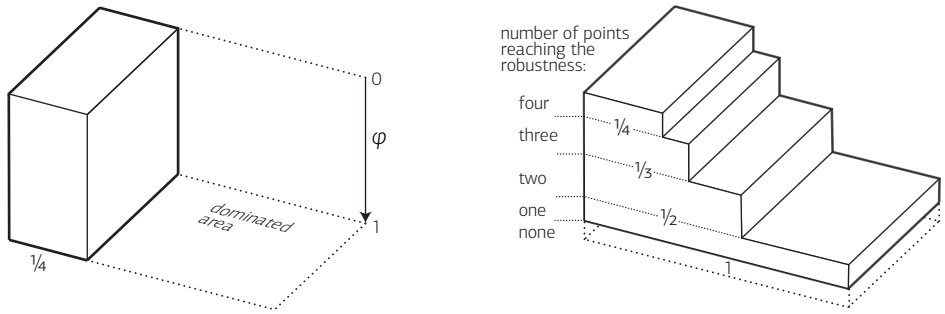
The complexity of calculating the regular hypervolume is $\sharp P$ complete as proven by Bringmann and Friedrich (2008). The complexity for the generalized hypervolume indicator, see Eq. 12, is even increased.

For this reason, an approximation scheme has to be used to exploit the potential of the robustness integrating hypervolume indicator for larger number of objectives. To this end, in the following section HypE, introduced in Bader and Zitzler (2009), is extended.

Introducing Robustness to HypE HypE needs to be modified in order to be applicable to the robustness integrating hypervolume indicator (Def. 3.5) due to the following observations. In case of the regular hypervolume indicator, a dominated region is accounted 100% as long as at least one point dominates it. So the only case HypE has to consider is removing all points dominating the portion altogether. For different points having different degrees of robustness, the situation changes: even though a partition dominated by multiple points would stay dominated if one removes not all dominating points, the robustness integrating hypervolume might nevertheless decrease due to the non bivariate attainment function. For example, if the most desirable point in terms of robustness is removed, then the attainment function is decreased and thereby also the hypervolume indicator value, see Theorem 3.7.

The second difference to the original HypE algorithm concerns the way the contribution is shared among the dominating point. For the regular hypervolume indicator, all solutions dominating the considered region are equally valuable since they entail the attainment function being 1. Consequently, the region is split equally among all solutions. For example, if four solutions dominate a region, the fitness of each is increased by $1/4$ of the volume of the region, see Fig. 10(a). For the robustness integrating hypervolume indicator, one has to distinguish different layers of robustness, which are achieved by subsets of all individuals dominating the region. Consequently, the hypervolume of these layers should only be split among those solutions actually reaching that robustness level, see Fig. 10(b). In the following, first the question is tackled of how to distribute the layers of robustness of a dominated portion among solutions. Then the probability is derived, that each one of these layers is lost.

Distributing Hypervolume among Solutions Let A denote the set of points and $UP \subseteq A$ those solutions, that dominate the region U under consideration. To illustrate the extended calculation of the robustness integrating HypE, consider ten points $A = \{x_1, \ldots, x_{10}\}$. The first four points $UP = \{x_1, \ldots, x_4\}$ dominate the region $U = H(\{x_1, \ldots, x_4\}, \{x_1, \ldots, x_{10}\}, R)$. Additionally, let $r(x_1) \leq r(x_2) \leq r(x_3) \leq r(x_4)$. First, a few simple cases are considered before presenting the final, and rather intriguing, formula to calculate the fitness of a point. First of all, it is investigated how much the robustness integrating hypervolume I_A^{φ} decreases when removing points from the set UP and how to attribute this loss to individuals. Assume x_2 or any other point which is less robust than x_1 is removed. In this case, the robustness integrating hypervolume does not decrease at all, since the attainment function depends only on the most robust point dominating the partition, in our case on x_1. Hence, a removal only



(a) regular HypE (b) robustness integrating HypE

Figure 10: Illustrates how a portion of the objective space U is attributed to the most robust among four points dominating the region. In case of no robustness considerations, the solutions all get $1/4$ of $\lambda(U)$ as every other solution, see (a). When considering robustness, different layers of robustness have to be distinguished, see (b): the first, and most robust, layer is dominated by no point, and is therefore disregarded. Only the most robust solution dominates the second layer, hence gets the whole slice. The third layer is dominated by an additional point, and both solutions get half of the share. The procedure continues up to the layer where every point reaches the necessary robustness, this layer is distributed evenly among all points.

affects the hypervolume, if no other point dominating the partition U at least as robust as the removed point remains in the population.

On the other hand, lets assume only the most robust solution x_1 is removed. By doing this, the hypervolume decreases by $\lambda(U) \cdot (\varphi(r(x_1)) - \varphi(r(x_2)))$, which is non zero if the robustness of x_1 is more desirable than the one of x_2. Clearly, this loss has to be fully attributed to point x_1, as no other point is removed. Now lets extend this to more than one point being removed. Assume points x_1, x_2, and x_4 are removed. As seen before, the loss of x_4 does not affect the hypervolume since x_3 (which is more robust) stays in the set. So in a set of points remaining in the population, the most robust individual sets a cutoff. For all individuals above this cutoff, i.e., for all individuals being less robust, the hypervolume does not decrease if these individuals are removed. The total loss of $I_H^{\varphi,w}$ is $\lambda(U) \cdot (\varphi(r(x_1)) - \varphi(r(x_3)))$. The question now is, how to distribute the loss among solutions. The share $\lambda(U) \cdot (\varphi(r(x_1)) - \varphi(r(x_2)))$ is only due to x_1, hence it is fully attributed to x_1. The share between $\varphi(x_2)$ and $\varphi(x_3)$ is dominated by both x_1 and x_2, so the portion is evenly split. This procedure continues for all robustness levels below the cutoff, see Fig. 10(b).

Probability of Loosing Hypervolume Now that it is known how to distribute the partition U among points for a particular selection of points, one has to consider all possible subsets of UP, i.e., subsets of points dominating U, and calculate the probability that the subset is lost. Let p denote the total number of points, let $n := |UP|$ denote the number of points dominating U, and let k denote the number of points to be removed, i.e., $k = p - \alpha$. Not all $\binom{n}{k}$ subsets have to be considered separately, but they can be summarized into classes c_i with $0 \leq i \leq n-1$, where i denotes the position of the cutoff level, see Fig. 11. More specifically, c_v contains all subsets where

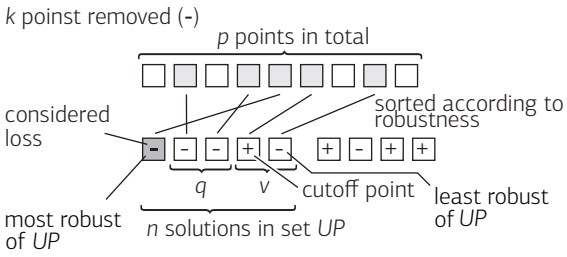


Figure 11: Illustration of class c_m: from p points, n dominate the region under consideration. The cutoff point is denoted as v. Besides the considered solution, q points need to be removed below the cutoff. In total k points are removed. In the example, $p = 9$, $v = 2$, $q = 2$, $n = 5$, and $k = 5$.

the most robust solution from UP not being removed is the vth least robust solution among all solutions in UP. For $v = 0$, all solutions dominating U are removed. For example, let $(\chi_1, \ldots, \chi_p)$ represent different subsets of A, where $\chi_i \in \{0, 1\}$ denotes the absence or presence respectively of solution x_i, and $\chi_i = \times$ denotes both cases (don't care). In the considered example, $c_0 = (0, 0, 0, 0, \times, \ldots, \times)$, $c_1 = (0, 0, 0, 1, \times, \ldots, \times)$, $c_2 = (0, 0, 1, \times, \times, \ldots, \times)$, and $c_3 = (0, 1, \times, \times, \times, \ldots, \times)$. Note that as for the regular HypE calculation, the fitness of a solution is determined under the assumption that this solution is removed, therefore, the subset c_4 (no solution removed from UP) is not possible.

To derive the probability that a subset being removed belongs to the class c_m, consider one particular way this can happen: the first q individuals are removed from below the cutoff, i.e., are more robust. The remaining $k - q$ points are then removed from above the cutoff or from the set $A \backslash UP$. This is one of $\binom{k-1}{q}$ equally probable combinations to obtain a cutoff level v, so that the obtained probability has to be multiplied by $\binom{k-1}{q}$[7] in the end.

A cutoff of v means, besides the considered point $q = n - v - 1$ points are removed from below the cutoff level. The probability that these q individuals are removed in the first q removal steps is:

$$P_1 = \frac{q}{p-1} \cdot \frac{q-1}{p-1} \cdots \frac{1}{p-q} = \frac{q! \cdot (p-1-q)!}{(p-1)!} \quad . \tag{19}$$

For the remaining $k - 1 - q$ points, any of the $p - n$ individuals not dominating the partition may be selected, as well as any of the $v - 1$ individuals above the cutoff, i.e., solutions less robust than the cutoff. The cutoff itself may not be removed as this would change the level v. So the probability for the second portion of points to be picked accordingly is:

$$P_2 = \frac{p-n+v-1}{p-q-1} \cdot \frac{p-n+v-2}{p-q-2} \cdots \frac{p-k}{p-k+1} = \frac{(p-n+v-1)!(p-k)!}{(p-k-1)!(p-q-1)!} \tag{20}$$

[7] Again, note that the solutions whose fitness needs to be determined, is assumed to be removed and belongs to the first q individuals, otherwise it would induce no loss in hypervolume. That is why the binomial coefficient considers the *k-1* set instead of *k*.

Multiplying P_1 (Eq. 19), P_2 (Eq. 20) and the number of combinations $\binom{k-1}{q}$ gives the final probability

$$P_v(p,q,k) = P_1 \cdot P_2 \cdot \binom{k-1}{q} = \frac{(p-k)!}{(p-1)!} \cdot q! \cdot \frac{(p-q-2)!}{(p-k-1)!} \cdot \frac{(k-1)!}{q!(k-1-q)!}$$
$$= (p-1)\frac{(p-q-2)!}{(p-1)!}\frac{(k-1)!}{(k-1-q)!} = (p-1)\prod_{i=p-q-1}^{p-1}\frac{1}{i}\prod_{i=k-q}^{k-1} i \ . \qquad (21)$$

For $v = 0$ and $p = n$ the last line is undefined, in this case, $P_0(n,q,k) = 1$ holds.

Example A.1. *Consider four solutions a, b, c and d with robustness $r(a) = 0.8$, $r(b) = 0.9$, $r(c) = 1.05$ and $r(d) = 1.2$. Let the robustness constraint be $\eta = 1$, and let the desirability φ be defined according to Eq. 13 with $\theta = 0.1$ and assume two solutions need to be removed. Now consider a sample dominated by a, c and d. This gives $p = 4$, $n = 3$ and $k = 2$. Since only two individuals are to be removed, the probability for having $v = 0$, i.e, all three individuals dominating the sample are removed, is 0. The probability for $v = 1$, i.e., another solution dominating the sample is removed besides the considered individual, is $1/3$. In this case, the first robustness layer extends from $r(a) = .8$ to $r(c) = 1.05$. This gives a value of $\frac{1}{3}\left(\varphi^{0.1}(.8) - \varphi^{0.1}(1.05)\right) = 0.253$ which is completely attributed to a since only this solution reaches the degree of robustness. The second layer extends from $r(c) = 1.05$ to $r(d) = 1.2$ and half of the value $\frac{1}{3}\left(\varphi^{0.1}(1.05) - \varphi^{0.1}(1.2)\right) = 0.079$ is added to the fitness of a and c respectively. The probability for $v = 2$ is $2/3$ (either b or d can be removed, but not c). The contribution $\frac{2}{3}\left(\varphi^{0.1}(.8) - \varphi^{0.1}(1.05)\right) = 0.506$ is completely added to the fitness of a.*

A.2 Test Problems

In this appendix two new classes of problems for robustness investigations are presented: first, in Sec. A.2.1 a real world mechanical problem is stated. Secondly, in Sec. A.2.2 a novel test problem suite is presented to test the performance of algorithms with respect to different robustness landscapes.

A.2.1 Truss Bridge Problem

First, the truss bridge problem is stated. Then, a problem-specific evolutionary algorithm is presented.

Problem Statement The task is to build a bridge over a river. Between two banks, n equally long decks have to be supported by a steel truss[8]. A uniform load is assumed over the decks that lead to $n-1$ equal force vectors, see Fig. 12. The first objective of the truss problem is to maximize the structural efficiency—the ratio of load carried by the bridge without elastic failure to the total bridge mass, i.e., costs. The river is considered environmentally sensitive and therefore no supporting structures are allowed below the water level. Furthermore, to limit the intervention in the natural scenery, the second objective is to minimize the rise of the bridge, measured from the decks at the center of the bridge[9].

[8] The decks are of 5 meters long.

[9] The height is arbitrarily defined at the middle of the bridge and not over the entire span width, to promote bridges very different to those optimizing the structural efficiency, and which tend to have the largest height at the center of the bridge.

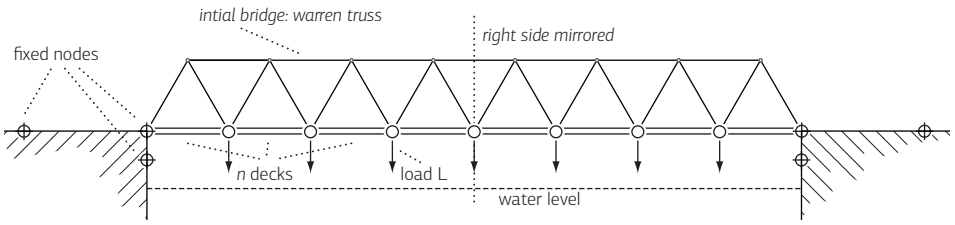


Figure 12: Illustration of the truss bridge problem. Between the two banks with predefined abutments, n decks with equal load have to be supported by a steel truss. As starting point, the individuals of the evolutionary algorithm are initialized to the shown Warren truss without verticals. At each bank, two supplementary fixed nodes are available to support the bridge additionally.

The bridge is considered two dimensional, i.e., all nodes and bar lie within a two dimensional plane. The slender members (referred to as bars) are connected at revolute joints (referred to as nodes). Half of the external load on the decks is applied to each of the two end joints and the weight of the members is considered insignificant compared to the loads and is therefore omitted. Hence, no torsional forces are active and all forces on members are tensile or compressive. For detailed information on truss bridges see W.F. Chen and L. Duan (1999).

In contrast to other well known truss problems, like for instance the ten bar truss problem (Deb, 2001), the nodes or bars are not specified in advanced, neither are they restricted to discrete positions like as in Ben-Tal and Nemirovski (2003). In fact, all kinds of geometries are possible which renders the problem much harder than the above mentioned ten bar truss problem. The only restriction is, that the predefined decks can not be changed in any way. In addition to the two endpoints, two additional fixed nodes at each bank can, but do not need to be, added to the truss[10].

The truss is made only from steel with yield strength 690 MPa and density 7800 kg/m^3. The maximum area of the members is 0.04 m^2 (radius 0.2 m), and the minimum area is set to $2.5{\cdot}10^{-5}$ m^2, (radius of 5 mm). The decks have a fixed cross-sectional area of 0.02 m^2.

Evolutionary Algorithm In the following an evolutionary algorithm is presented tailored to the steel truss bridge problem stated above. The algorithm consists of (i) a general representation which can model all possible bridges, (ii) an initialization of solutions, (iii) the calculation of the objective values, and (iv) different mutation operators to generate new solutions.

Representation The representation consists of variable length lists. The first list contains all nodes. A node is thereby determined by its position (x, y), the degrees of freedom of the node (i.e., whether the node is fixed or not), and the load attached to this node—the latter is non-zero only for the $n - 1$ predefined joints between the decks. The second list contains the members that consist of references to the two endpoints, and the cross-sectional area of the bar. Since the problem is mirror-symmetrical, only

[10]The additional fixed nodes are located 2.5 m below the edge of the abutment and 7.5 m to the left and right of the edge respectively.

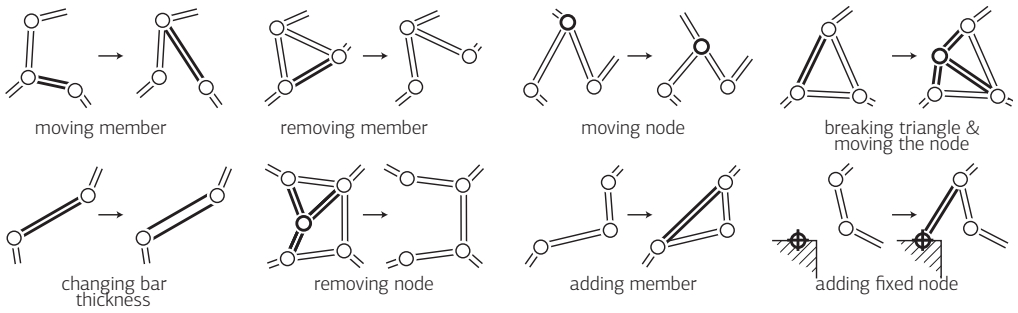


Figure 13: Illustrates the eight mutation operators used to optimize the bridge problem.

the left half of the bridge is represented and solved, see below.

Initialization As a starting point, all solutions are set to a Warren truss without verticals, and with equilateral triangles. This ensures that the initial bridges are statically determinate and stable. Of course, the risk increases that the diversity of solutions is limited unnecessarily. As the results in Sec. 5.2 on page 23 show this is not the case though—the solutions found vary a lot from the initial Warren truss.

Calculating the Objective Function To determine the first objective function, the structural efficiency, matrix analysis of the truss is performed; more specifically, the matrix force method is used to determine the internal forces of all members. Given their area, the weakest link can then be identified which defines the maximum load of the bridge. If the bridge is statistically undetermined, i.e., the matrix becomes singular, the bridge is classified infeasible. No repairing mechanism is used in this case. The weight of the bridge, on the other hand, is determined by summing up the product of length, area, and density of all bars. For the weight, the members constituting the deck of the bridge are also included. Finally, dividing the maximum load by the density gives the first objective, i.e., the structural efficiency.

The maximum vertical distance of a member or node above the deck level, measured at the center of the bridge, gives the second objective. The rise of the bridge is to be minimized.

Because of the symmetry of the problem, only the left half of the bridge is represented and solved. To this end, all nodes lying on the mirror axis, the center of the bridge, are fixed in the horizontal dimension. This models the fact, that due to symmetry, the horizontal component of the force vectors at these nodes is zero. All members except those on the mirror axis are considered twice in terms of cost, since they have a symmetric counterpart. On the other hand, the internal load of members on the symmetry axis is doubled after matrix analysis, since the right, not considered, half of the bridge will contribute the same load as the left half.

Mutation Operators Due to the complex representation of the bridges no crossover but only mutation operators are applied to generate new bridges. In each generation, one of the following eight mutation functions is used, where the probability of an operator is evolved by self-adaptation (Deb, 2001):

moving member A member is randomly picked and its endpoints are changed, such that the member ends up at a different place.

removing member A randomly chosen member is removed. Nodes that are no longer connected are removed too.

adding member Two nodes are picked randomly and a member is added between them.

moving nodes The location of a node is moved uniformly within the interval [-1 m,1 m]×[-1 m,1 m].

removing node A randomly chosen node is removed; all members connected to that node are removed as well.

adding fixed node A node from the set of fixed nodes is picked, and connected to a randomly chosen (non-fixed) node.

breaking triangle and moving node This mutation operator should help the formation of repeated triangular patterns, which are known to be beneficial because triangles can not be distorted by stress. An existing triangle is chosen, then one of its three sides is divided in the middle by adding a new node. This node is then connected by the corresponding median of the triangle. Since this new member would get zero force, the new node is moved additionally by the operator *moving node*.

changing member area This mutation operator randomly picks a member and changes its area by factor ρ, where $\rho \sim U(0.5, 1.5)$ is randomly distributed between 0.5 and 1.5.

Fig. 13 illustrates the eight mutation operators. In addition to these operators, with a probability of 50% the cross-sectional areas of the bridge are optimized according to matrix analysis, i.e., each cross-sectional area is decreased as far as the maximum load carried does not decrease.

Noise Many different sources of uncertainty are conceivable for the truss problem, e.g., differing location of the nodes due to imprecise construction, varying member area because of manufacturing imperfection or changing external load distributions. The present thesis considers random perturbations of the yield strength of members. The thicker a bar thereby is, the larger the variance of the noise. The reasoning behind this assumption is that material properties are harder to control the larger a structure is. The model $\sigma_{\text{UTS}} \sim \sigma_{\text{UTS}} \cdot U(1 - (r^2)\delta, 1 + (r^2)\delta)$ is used, where r is the radius of the bar and σ_{UTS} denotes the yield strength of a member. As robustness measure, the maximum deviation according to Eq. 5 is used. However, in contrast to Sec. 5 where a sampling procedure is used to estimate the worst case, the worst case is determined analytically: for each member, the yield strength is set to the minimum value according to the noise model, i.e., $\sigma^{w}_{\text{UTS}} = \sigma_{\text{UTS}} \cdot (1 - (r^2))\delta$. In all experimental comparisons, δ was set to 100.

A.2.2 BZ Robustness Test Problem Suite

Existing test problem suites like WFG (Huband et al., 2006) or DTLZ (Deb et al., 2002b) feature different properties—like non-separability, bias, many-to-one mappings and multimodality. However, these problems have no specific robustness properties, and the robustness landscape is not known. For that reason, six novel test problems are proposed denoted as Bader-Zitzler (BZ) that have different, known robustness characteristics. These problems, BZ1 to BZ6, allow to investigate the influence of different

robustness landscapes on the performance of the algorithms. All, except for BZ5, share the following simple structure:

$$\left.\begin{array}{ll} \text{Minimize} & f_i(x) = \dfrac{x_i}{\|\sum_{i=1}^{k} x_i\|_\beta} \cdot \big(1 + S(g(x))\big) \quad 1 \le i \le d \\ \text{with} & g(x) = \dfrac{1}{n-k} \displaystyle\sum_{i=k+1}^{n} x_i \\ \text{subject to} & 0 \le x_i \le 1 \text{ for } i = 1, 2, \ldots, n \end{array}\right\} \tag{22}$$

The first k decision variables are position related, the last $n-k$ decision variables determine the distance to the Pareto front. The Pareto front is reached for $x_i = 0$, $k+1 \le i \le n$, which leads to $S(g(x)) = 0$. The front has has the form $\big(f_1(x)^\beta + \ldots + f_d(x)^\beta\big)^{1/\beta} = 1$. The parameter β thereby specifies the shape of the Pareto front: for $\beta > 1$ the shape is convex, for $\beta = 1$ it is linear and for $0 < \beta < 1$ the shape is concave. The distance to the Pareto front is given by $S(g(x))$, where $g(x)$ is the mean of the distance related decision variables $x_{k+1}, \ldots, x_n$ (an exception is BZ5, where S is a function of $g(x)$ and the variance $\sigma^2 = \mathrm{Var}(\{x_1, \ldots, x_k\})$).

The distance to the front, i.e., $S(g(x))$, depends on a fast oscillating cosine function that causes the perturbations of the objective values and where its amplitude determines the robustness of a solution. In the following, realization of S are listed, and choice of parameter β for the six test problems BZ1 to BZ6 and discuss their robustness landscape.

In the following, let $h := g(x)$

BZ1 For the first test problem, the distance to the front subject to $h := g(x)$ is

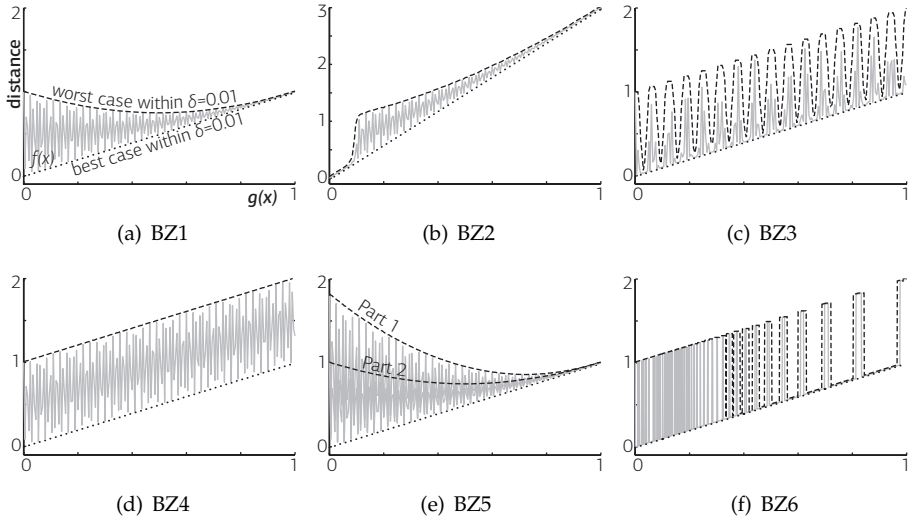
$$S(h) = h + \big((1-h) \cdot \cos(1000h)\big)^2 \tag{23}$$

Fig. 14(a) shows the function S as a function of h, as well as the maximum and minimum within a neighborhood of B_δ (see Sec. 2.2). As for all BZ test problems, the (best case) distance to the front linearly decreases with decreasing d. The difference to the worst case, on the other hand, goes in the opposite direction and increases. This gives a continuous trade-off between the objective values $f(x)$ (better for smaller values of h) and the robustness $r(x)$ (better the larger h). The parameter β is set to 1 which gives a linear front shape.

BZ2 For the second test problem, $\beta = 2$ describing a sphere-shaped Pareto front, the distance to which is given by:

$$S(h) = 3h + \frac{1}{1 + \exp(-200(h - 0.1))} \cdot \big((1-h) \cdot \cos(1000h)\big)^2 \tag{24}$$

Fig. 14(b) shows the distance as a function of h. As in BZ1, the robustness first decreases with decreasing h. However, around $h = 0.1$ the exponential distribution kicks in, and the amplitude of the cosine function becomes very small such that the Pareto front and its adjacencies are robust. BZ2 tests, whether an algorithm is able to overcome the decreasing robustness as approaching the Pareto front, or if the solutions are driven away from the Pareto front to increase their robustness.



(a) BZ1 (b) BZ2 (c) BZ3

(d) BZ4 (e) BZ5 (f) BZ6

Figure 14: Distance to the front (gray line) as a function of $g(x)$ (abscissa), see Eq. 22. The solid and dashed line represent the minimal and maximal distance respectively within the interval $[g(x-\delta), g(x+\delta)]$ with $\delta = 0.01$.

BZ3 For the third instance of BZ the distance to the front is a product of two cosine terms:

$$S(h) = h + \big(\cos(50h)\cos(1000h)\big)^4 \ . \tag{25}$$

The concave Pareto front ($\beta = 0.5$) is non robust. However, by increasing the distance to the front the term $\cos(50h)$ periodically leads to robust fronts, see Fig. 14(c). An algorithm therefore has to overcome many robust local fronts before reaching the robust front that is closest to the Pareto front.

BZ4 For BZ4, the amplitude of the oscillation term does not change, see Fig. 14(d):

$$S(h) = h + \cos(1000h)^2 \ . \tag{26}$$

Therefore, the robustness does not change with h. The only way to minimize $r(x)$ (Eq. 5) is to choose a h for which $\cos(1000h)^2$ is close to the worst case. The shape of the Pareto front is convex with $\beta = 3$.

BZ5 The distance to the front for the fifth BZ not only depends on the aggregated variable h, but also on the variance of the distance related decision variables $\sigma^2 = \mathrm{Var}(\{x_1, \ldots, x_k\})$:

$$S(h, \sigma^2) = \begin{cases} h + \big((1-h)\cos(1000h)\big)^2 & \sigma^2 < 0.04 \\ h + 1.8\big((1-h)\cos(1000h)\big)^2 & \text{otherwise} \end{cases} \tag{27}$$

This gives two different degrees of robustness for any choice of h. Depending on the location in the objective space, the distance to the Pareto front (given by $\beta = 0.3$) therefore varies for a given robustness level: it is smaller where $\sigma^2 < 0.04$ and larger where the variance exceeds 0.04.

BZ6 The last instance of the BZ suite uses a step function as distance $S(h)$, see Fig. 14(f),

$$S(h) = \begin{cases} h+1 & \cos\left(\frac{1000.0}{(0.01+h)h\pi}\right) > 0.9 \\ h & \text{otherwise} \end{cases} . \tag{28}$$

This leads to different robust regions whose width decrease with decreasing distance to the front. Therewith, the ability of an algorithm to determine the robustness of a solution is tested. For example, when the number of samples to determine the robustness is small, the edges of a robust region might not be detected and a non-robust solutions is misclassified as robust. As for BZ2 the Pareto front is a sphere ($\beta = 2$).

References

Abraham, B., editor (1998). *Quality Improvement Through Statistical Methods (Statistics for Industry and Technology)*. Birkhäuser Boston, 1 edition.

Auger, A., Bader, J., Brockhoff, D., and Zitzler, E. (2009a). Articulating User Preferences in Many-Objective Problems by Sampling the Weighted Hypervolume. In Raidl, G. et al., editors, *Genetic and Evolutionary Computation Conference (GECCO 2009)*, pages 555–562, New York, NY, USA. ACM.

Auger, A., Bader, J., Brockhoff, D., and Zitzler, E. (2009b). Theory of the Hypervolume Indicator: Optimal μ-Distributions and the Choice of the Reference Point. In *Foundations of Genetic Algorithms (FOGA 2009)*, pages 87–102, New York, NY, USA. ACM.

Bader, J. and Zitzler, E. (2008). HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. TIK Report 286, Computer Engineering and Networks Laboratory (TIK), ETH Zurich.

Bader, J. and Zitzler, E. (2009). HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. *Evolutionary Computation*. to appear.

Beer, M. and Liebscher, M. (2008). Designing Robust Structures–a Nonlinear Simulation Based Approach. *Computers and Structures*, 86(10):1102–1122.

Ben-Tal, A. and Nemirovski, A. (2003). Robust optimization – methodology and applications. *Mathematical Programming*, 92(3):453–480.

Beume, N. and Rudolph, G. (2006). Faster S-Metric Calculation by Considering Dominated Hypervolume as Klee's Measure Problem. Technical Report CI-216/06, Sonderforschungsbereich 531 Computational Intelligence, Universität Dortmund. shorter version published at IASTED International Conference on Computational Intelligence (CI 2006).

Beyer, H.-G. and Sendhoff, B. (2007). Robust optimization - A comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33-34):3190–3218.

Branke, J. and Schmidt, C. (2003). Selection in the Presence of Noise. *Lecture Notes in Computer Science*, pages 766–777.

Bringmann, K. and Friedrich, T. (2008). Approximating the Volume of Unions and Intersections of High-Dimensional Geometric Objects. In Hong, S. H., Nagamochi, H., and Fukunaga, T., editors, *International Symposium on Algorithms and Computation (ISAAC 2008)*, volume 5369 of *LNCS*, pages 436–447, Berlin, Germany. Springer.

Brockhoff, D., Friedrich, T., and Neumann, F. (2008). Analyzing Hypervolume Indicator Based Algorithms. In Rudolph, G. et al., editors, *Conference on Parallel Problem Solving From Nature (PPSN X)*, volume 5199 of *LNCS*, pages 651–660. Springer.

Burke, E., Causmaecker, P. D., Maere, G. D., Mulder, J., Paelinck, M., and Vanden Berghe, G. (2009). A multi-objective approach for robust airline scheduling. *Computers and Operations Research*.

Chen, W., Allen, J., Tsui, K.-L., and Mistree, F. (1996). A Procedure For Robust Design: Minimizing Variations Caused By Noise Factors And Control Factors. *ASME Journal of Mechanical Design*, 118:478–485.

Das, I. (2000). Robustness Optimization for Constrained Nonlinear Programming Problems. *Engineering Optimization*, 32(5):585–618.

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 Schoenauer, M. et al., editors, *Conference on Parallel Problem Solving from Nature (PPSN VI)*, volume 1917 of *LNCS*, pages 849–858. Springer.

Deb, K. and Gupta, H. (2005). Searching for Robust Pareto-Optimal Solutions in Multi-objective Optimization. In *Evolutionary Multi-Criterion Optimization*, volume 3410/2005 of *Lecture Notes in Computer Science*, pages 150–164. Springer.

Deb, K. and Gupta, H. (2006). Introducing Robustness in Multi-Objective Optimization. *Evolutionary Computation*, 14(4):463–494.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002a). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2002b). Scalable Multi-Objective Optimization Test Problems. In *Congress on Evolutionary Computation (CEC 2002)*, pages 825–830. IEEE Press.

Egorov, I., Kretinin, G., and Leshchenko, I. (2002). How to Execute Robust Design Optimization. In *9th AIAA/ISSMO Symposium and Exhibit on Multidisciplinary Analysis and Optimization*.

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)*, volume 3410 of *LNCS*, pages 62–76. Springer.

Fleischer, M. (2003). The measure of Pareto optima. Applications to multi-objective metaheuristics. In Fonseca, C. M. et al., editors, *Conference on Evolutionary Multi-Criterion Optimization (EMO 2003)*, volume 2632 of *LNCS*, pages 519–533, Faro, Portugal. Springer.

Fonseca, C. M. and Fleming, P. J. (1998). Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms—Part I: A Unified Formulation. *IEEE Transactions on Systems, Man, and Cybernetics*, 28(1):26–37.

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, Sheraton Vancouver Wall Centre Hotel, Vancouver, BC Canada. IEEE Press.

Friedrich, T., Horoba, C., and Neumann, F. (2009). Multiplicative Approximations and the Hypervolume Indicator. In Raidl, G. et al., editors, *Genetic and Evolutionary Computation Conference (GECCO 2009)*, pages 571–578. ACM.

Ge, P., Stephen, C., and Bukkapatnam, S. (2005). Supporting Negotiations in the Early Stage of Large-Scale Mechanical System Design. *Journal of Mechanical Design*, 127:1056.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, Massachusetts.

Guimaraes, F., Lowther, D., and Ramirez, J. (2006). Multiobjective Approaches for Robust Electromagnetic Design. *Magnetics, IEEE Transactions on*, 42(4):1207–1210.

Gunawan, S. and Azarm, S. (2004). On a Combined Multi-Objective and Feasibility Robustness Method for Design Optimization. *Proceedings of 10th AIAA/ISSMO MDO*.

Gunawan, S. and Azarm, S. (2005). Multi-objective robust optimization using a sensitivity region concept. *Structural and Multidisciplinary Optimization*, 29(1):50–60.

Hamann, A., Racu, R., and Ernst, R. (2007). Multi-Dimensional Robustness Optimization in Heterogeneous Distributed Embedded Systems. In *13th IEEE Real Time and Embedded Technology and Applications Symposium, 2007. RTAS'07*, pages 269–280.

Harzheim, E. (2005). *Ordered Sets (Advances in Mathematics)*. Springer.

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. (2001). Evolutionary multi-objective ranking with uncertainty and noise. In *Evolutionary Multi-Criterion Optimization*, Lecture Notes in Computer Science, pages 329–343. Springer Berlin.

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

Jin, Y. and Branke, J. (2005). Evolutionary Optimization In Uncertain Environments—A Survey. *IEEE Transactions on Evolutionary Computation*, 9(3):303–317.

Jin, Y. and Sendhoff, B. (2003). Trade-Off between Performance and Robustness: An Evolutionary Multiobjective Approach. In Fonseca, C. M. et al., editors, *EMO 2003*, volume 2632, pages 237–251. Springer.

Jürgen Branke (1998). Creating Robust Solutions by Means of Evolutionary Algorithms. In A. E. Eiben and T. Bäck and M. Schoenauer and H.-P. Schwefel, editor, *Parallel Problem Solving from Nature – PPSN V*, pages 119–128, Berlin. Springer. Lecture Notes in Computer Science 1498.

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.

Kouvelis, P. and Yu, G. (1997). *Robust Discrete Pptimization and its Applications*. Kluwer Academic Publishers.

Kunjur, A. and Krishnamurty, S. (1997). A Robust Multi-Criteria Optimization Approach. *Mechanism and Machine Theory*, 32(7):797–810.

Li, M., Azarm, S., and Aute, V. (2005). A Multi-objective Genetic Algorithm for Robust Design Optimization. In *GECCO 2005*, pages 771–778. ACM.

Mulvey, J., Vanderbei, R., and Zenios, S. (1995). Robust Optimization of Large-Scale Systems. *Operations research*, pages 264–281.

Parkinson, A., Sorensen, C., and Pourhassan, N. (1993). A General Approach for Robust Optimal Design. *Journal of Mechanical Design*, 115(1):74–80.

S. Kotz and S. Nadarajah (2001). *Extreme Value Distributions: Theory and Applications*. World Scientific Publishing Company, 1st edition.

Soares, G. L., Adriano, R. L. S., Maia, C. A., Jaulin, L., and Vasconcelos, J. A. (2009a). Robust Multi-Objective TEAM 22 Problem: A Case Study of Uncertainties in Design Optimization. *IEEE Transactions on Magnetics*, 45:1028–1031.

Soares, G. L., Parreiras, R. O., Jaulin, L., and Maia, J. A. V. C. A. (2009b). Interval Robust Multi-objective Algorithm. *Nonlinear Analysis*.

Srinivas, N. and Deb, K. (1994). Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248.

Taguchi, G. (1986). *Introduction to Quality Engineering: Designing Quality into Products and Processes*. Quality Resources.

Teich, J. (2001). Pareto-Front Exploration with Uncertain Objectives. In *Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, pages 314–328, London, UK. Springer.

Tsui, K. (1999). Robust Design Optimization for Multiple Characteristic Problems. *International Journal of Production Research*, 37(2):433–445.

Tsutsui, S. and Ghosh, A. (1997). Genetic Algorithms with a Robust Solution Searching Scheme. *IEEE Trans. on Evolutionary Computation*, 1(3):201–208.

Wagner, T., Beume, N., and Naujoks, B. (2007). Pareto-, Aggregation-, and Indicator-based Methods in Many-objective Optimization. In Obayashi, S. et al., editors, *Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*, volume 4403 of *LNCS*, pages 742–756, Berlin Heidelberg, Germany. Springer. extended version published as internal report of Sonderforschungsbereich 531 Computational Intelligence CI-217/06, Universität Dortmund, September 2006.

W.F. Chen and L. Duan (1999). *Bridge Engineering Handbook*. CRC, 1 edition.

While, L., Bradstreet, L., Barone, L., and Hingston, P. (2005). Heuristics for Optimising the Calculation of Hypervolume for Multi-objective Optimisation Problems. In *Congress on Evolutionary Computation (CEC 2005)*, pages 2225–2232, IEEE Service Center, Edinburgh, Scotland. IEEE Press.

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.

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 Obayashi, S. et al., editors, *Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*, volume 4403 of *LNCS*, pages 862–876, Berlin. 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, Zurich, Switzerland.

Zitzler, E., Thiele, L., and Bader, J. (2008). SPAM: Set Preference Algorithm for Multiobjective Optimization. In Rudolph, G. et al., editors, *Conference on Parallel Problem Solving From Nature (PPSN X)*, volume 5199 of *LNCS*, pages 847–858. Springer.

Zitzler, E., Thiele, L., and Bader, J. (2009). On Set-Based Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*. to appear.

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.