Incorporating User Preferences in Many-Objective Optimization using Relation Epsilon-Preferred

Nicole Drechsler[1], André Sülflow[2] and Rolf Drechsler[1,3]
[1]*Institute of Computer Science, University of Bremen, 28359 Bremen, Germany*
[2]*solvertec GmbH, 28359 Bremen, Germany*
[3]*Cyber-Physical Systems, DFKI GmbH, 28359 Bremen, Germany*

Keywords: Many-Objective Optimization, Nurse Rostering Problem, Relation ε-Preferred, User Preferences.

Abstract: During the last 10 years, many-objective optimization problems, i.e. optimization problems with more than three objectives, are getting more and more important in the area of multi-objective optimization. Many real-world optimization problems consist of more than three mutually dependent subproblems, that have to be considered in parallel. Furthermore, the objectives have different levels of importance. For this, priorities have to be assigned to the objectives. In this paper we present a new model for many-objective optimization called *Prio-ε-Preferred*, where the objectives can have different levels of priorities or user preferences. This relation is used for ranking a set of solutions such that an ordering of the solutions is determined. *Prio-ε-Preferred* is controlled by a parameter ε, that is problem specific and has to be adjusted experimentally by the designer. Therefore, we also present an extension called *Adapted-ε-Preferred* (AEP), that determines the ε values automatically without any user interaction. To demonstrate the efficiency of our approach, experiments are performed.

1 INTRODUCTION

Many real-world optimization problems consist of several mutually dependent subproblems that have to be optimized in parallel. The so called *Multi-Objective Optimization* (MOO) problems and approaches for solving them have been intensively studied in the past. For this, in the area of *Evolutionary Algorithms* (EAs) many models and algorithms for MOO have been presented (Fonseca and Fleming, 1995; Zitzler and Thiele, 1999; Deb, 2001; Bader and Zitzler, 2011). If more than three optimization objectives are considered, the corresponding MOO problems are called *Many-Objective Optimization* problems in literature. Especially, real-world optimization problems often have more than three objectives (Drechsler et al., 2001; Hughes, 2007; Pizzuti, 2012). Furthermore, several MOO problems in industrial applications consist of subproblems that have different levels of importance. The importance of a subproblem is specified by the user and different methods exist to model these user preferences or priorities (Schmiedle et al., 2001; Wickramasinghe and Li, 2009; Wagner and Trautmann, 2012). Considering both, many-objective optimization problems and user preferences, there is a need for algorithms that can combine these properties.

One classical approach to deal with multiple optimization criteria is the *weighted sum* approach. It is often used in industrial applications, since it is easy to implement and on a first view scales well (see e.g. (Burke et al., 2004)). If only a small region of the *Pareto-front* is of interest, *weighted sums* can be used to control the optimization process. The objectives' priorities can be set by the choice of the weights. In the context of *Many-Objective Optimization* this method reaches its limit, because it is hard to determine the weights, such that the search is guided in the desired direction (Drechsler et al., 2001; Geiger, 2009). Additionally, the weighted sum approach is unable to find compromise solutions of concave *Pareto* fronts.

In evolutionary MOO one of the first approaches was the use of *Pareto-optimal elements* (Goldberg, 1989). Here, the goal is to explore the *Pareto-set* of a given MOO problem, such that as many elements as possible of the *Pareto-set* are calculated. To guide the search, a basic *Dominates* relation is defined, that is used to compare the solution elements. Based on this relation many approaches for MOO have been in-

Drechsler N., Sülflow A. and Drechsler R..
Incorporating User Preferences in Many-Objective Optimization using Relation Epsilon-Preferred.
DOI: 10.5220/0004496000670074
In *Proceedings of the 5th International Joint Conference on Computational Intelligence* (ECTA-2013), pages 67-74
ISBN: 978-989-8565-77-8

tensively studied (Fonseca and Fleming, 1995; Zitzler and Thiele, 1999; Deb, 2001). But often, these methods only consider two or three optimization objectives (Deb, 2001). If many-objective optimization problems are considered, these methods have several drawbacks. For example in (Deb, 2001) it is reported, that the number of individuals in the *Pareto-set* increases with the number of optimization objectives. Experiments have shown, that for 20 objectives the ranking of solutions is nearly impossible. For this, the ratio of solutions that cannot be distinguished using the *Dominates* relation is almost 100%. But, if EAs are used, a ranking of the solutions is necessary to guide the search.

To overcome these problems in many-objective optimization several approaches have been presented (see e.g. (Fleming et al., 2005; Corne and Knowles, 2007; Ishibuchi et al., 2008; Brockhoff and Zitzler, 2009; Bader and Zitzler, 2011)). A promising approach in evolutionary many-objective optimization is objective reduction based on the *Dominates* relation (Brockhoff and Zitzler, 2009). There, the considered objectives are reduced during decision making while preserving the dominance structure of the considered optimization problem as much as possible. In (Brockhoff and Zitzler, 2009) the reduction concept has been studied for the multi-objective knapsack problem, for DTLZ test functions (Deb et al., 2005) with up to 25 objectives and for a radar waveform problem with nine objectives (Hughes, 2007). The hypervolume based MOO has also been extended to many-objective optimization. A method that approximates the hypervolume indicator is presented in (Bader and Zitzler, 2011). In (Drechsler et al., 2001; di Pierro et al., 2007; Li and Wong, 2009; Sülflow et al., 2007) relations are presented that distinguish between solutions that are incomparable if the *Dominates* relation is considered. An overview and a comparison of these methods is given in (Corne and Knowles, 2007; Ishibuchi et al., 2008).

Furthermore, approaches are presented that consider user preferences in many-objective optimization (Wickramasinghe and Li, 2009; Auger et al., 2009; Wagner and Trautmann, 2012). In (Wickramasinghe and Li, 2009) a user-defined distance metric is used to guide the search on the basis of the *Dominates* relation. The incorporation of user preferences to the hypervolume approach has been investigated in (Auger et al., 2009; Wagner and Trautmann, 2012).

The approaches in (Drechsler et al., 2001; Schmiedle et al., 2001; Sülflow et al., 2007) are based on a relation called *Preferred*. Relation *Preferred* is a refinement of relation *Dominates*, i.e. a ranking of solution elements that are incomparable using the *Dominates* relation is enabled. This results in a better guided search if EAs are used. In (Drechsler et al., 2001) the model has been applied to an optimization problem from the area of computer-aided design of integrated circuits. There, five optimization objectives have been considered in parallel. Experiments have shown, that *Preferred* clearly outperforms the *Dominates* relation and an approach based on a *weighted sum*. In (Schmiedle et al., 2001) *Preferred* is extended, such that it can also handle different levels of priorities. The model is applied to an approach that makes use of *Genetic Programming* (Koza, 1992) in computer-aided designed of integrated circuits.

An extension of *Preferred*, the so-called relation ε-*Preferred*, has been introduced in (Sülflow et al., 2007). For this, *Preferred* is enriched by a parameter ε, where ε defines a radius for each objective. If an objective is outside this region, the corresponding elements are "punished". In these examinations experiments are performed, where a complex scheduling problem is considered, i.e. the *Nurse Rostering Problem* (NRP) is solved using an EA. The NRP is of high practical relevance and consists of several constraints, i.e. resource planning for employees in a hospital has to be performed. In the experiments, an example from a hospital that consists of 26 optimization objectives, has been considered. It turns out, that two approaches based on the *Dominates* relation and *Nondominated Sorting* (NSGA-II) could be improved with respect to quality significantly. Additionally, ε-*Preferred* outperforms *Preferred* with respect to quality and robustness.

In this article relation ε-*Preferred* is considered and further extended, such that it can handle different levels of priorities. Optimization problems like e.g. the NRP consist of several objectives with problem specific user preferences. For this, the new relation model *Prio*-ε-*Preferred* is formally introduced. This relation is used to guide the search of an EA, where the parameter ε has to be set by the developer. It is shown by experiments that the convergence behavior of the algorithm and the quality of the results depend on the choice of ε. Thus, a new method is presented, that allows to determine good choices of ε automatically without user interaction. The resulting method *Adapted*-ε-*Preferred* (AEP) automatically adapts parameter ε such that the same quality as the "hand-crafted" results can be obtained without user interaction.

To demonstrate the efficiency of *Prio*-ε-*Preferred* and AEP several experiments are performed. The methods are used to guide the search of an EA for the NRP, using benchmarks from (Benchmarks, 2012), where the user preferences are given. Considering these benchmarks, the number of optimization objectives ranges from 9 to 17[1]. In the experiments different choices of ε are examined such the potential of *Prio*-ε-*Preferred* is shown. A comparison to relation *Dominates* (Goldberg, 1989) and method NSGA-II (Deb, 2001) shows that *Prio*-ε-*Preferred* clearly outperforms these approaches[2]. Then the influence of parameter ε is studied in detail. The new method AEP is applied to the considered benchmarks. It is shown, that the automatic method AEP is as good as *Prio*-ε-*Preferred*, where parameter ε is adjusted manually. Notice, that for AEP no user interaction is required.

The paper is structured as follows: In Section 2 previous work is reviewed. Then, the considered application, the NRP is explained in Section 3. The new model *Prio*-ε-*Preferred* is described in Section 4. Then in Section 5 experimental evaluations are performed that demonstrate the properties of the *Prio*-ε-*Preferred* relation. In Section 6 the method AEP is introduced and experimental results are discussed. A summary of the proposed relation and methods is given in Section 7.

2 PRELIMINARIES

To make the article self-contained, we briefly give an overview on basic techniques in evolutionary multi-objective optimization and on relations proposed in this field for comparison.

2.1 Multi-objective Optimization

In general, many optimization problems consist of several mutually dependent and conflicting subproblems, i.e. usually the improvement in one objective leads to the deterioration of another objective.

A multi-objective optimization problem is defined as follows: Given a search space Ω, an evaluation function $f : \Omega \to \mathbb{R}^m$ is defined to calculate the fitness vector $F(A) : \forall A \in \Omega$ of size m. Then we have to minimize (or maximize) the elements of $F(A)$. In the following we assume, without loss of generality, that F has to be minimized for all objectives. According to (Goldberg, 1989) it holds:

[1]In contrast to (Sülflow et al., 2007) benchmarks from (Benchmarks, 2012) are taken. This makes the results comparable to other approaches.

[2]A comparison to a many-objective approach with user preferences like e.g. performed in (Auger et al., 2009) can not be directly given, because different modellings of user preferences are used. But, the modelling of user preferences as presented here in combination with the hypervolume indicator is an interesting task for future work.

Definition 1. Let $A, B \in \Omega$.
$A \prec_{dominates} B :\Leftrightarrow$
$\exists j : F_j(A) < F_j(B) \wedge F_i(A) \leqslant F_i(B), 1 \leqslant i \leqslant m.$

Based on this, we can describe the `Pareto set` χ as

$$\forall p \in \chi : \nexists q \in \Omega : q \prec_{dominates} p.$$

As can be seen from the definition above, if two elements $A, B \in \Omega$ are compared with the *Dominates* relation, then A dominates B only if it is less or equal to B in all objectives and if it is better in at least one objective. All elements $A \in \Omega$ in the Pareto-set are equal or *not comparable*. Usually, all points in this set are of interest for the decision maker or designer.

2.2 Relation *Preferred*

To perform a refinement of the *Dominates* relation *Preferred* is defined. Using the *Preferred* relation a set of solutions can be classified into so-called *Satisfiability Classes* (SC) (Drechsler et al., 2001). Relation *Preferred* respects the number of objectives in which A differs from B:

Definition 2. Let $A, B \in \Omega$. $A \prec_{preferred} B :\Leftrightarrow$ $|\{i : F_i(A) < F_i(B), 1 \leqslant i \leqslant m\}| > |\{j : F_j(B) < F_j(A), 1 \leqslant j \leqslant m\}|$.

Using Definition 2 we are able to compare elements $A, B \in \Omega$ pairwise more precisely. A is *preferred* to B ($A \prec_{preferred} B$) iff i ($i \leq n$) components of A are smaller than the corresponding components of B and only j ($j < i$) components of B are smaller than the corresponding components of A. We use a graph representation for the relation, where each element is a node and "preferences" are given by edges. Relation *Preferred* is not a partial order, because it is not transitive, i.e. the relation graph can have cycles. Solutions that are included in a cycle are denoted as incomparable. They are computed by a linear time graph algorithm (Drechsler et al., 2001).

2.3 Relation ε-*Preferred*

To improve the robustness of relation *Preferred* in many-objective optimization ε-*Preferred* has been introduced (Sülflow et al., 2007). Hence, the idea is to define fitness limits $\varepsilon_i, 1 \leq i \leq m$, for each dimension.

Definition 3. Let $A, B \in \Omega$ and ε_i, $1 \leqslant i \leqslant m$

$A \prec_{\varepsilon-exceed} B \Leftrightarrow$
$|\{i : F_i(A) < F_i(B) \wedge |F_i(A) - F_i(B)| > \varepsilon_i\}| >$
$|\{j : F_j(A) > F_j(B) \wedge |F_j(A) - F_j(B)| > \varepsilon_j\}|.$

ε-*exceed* additionally takes the distance of the solutions components into account. Using ε-*exceed* the extension ε-*Preferred* is defined as follows:

Definition 4. Given two solutions $A, B \in \Omega$.
$A \prec_{\varepsilon-preferred} B \Leftrightarrow$
$A \prec_{\varepsilon-exceed} B \vee (B \not\prec_{\varepsilon-exceed} A \wedge A \prec_{preferred} B)$

First it is counted how often a solution exceeds the ε-limits and the better solution is determined. If both solutions are in the given range *Preferred* is used for comparison.

Example 1. Consider some solution vectors from $\mathbb{R}^3$, i.e. the results of three objective functions:
(7,0,9) (8,7,1) (1,9,6)

Additionally, let $\varepsilon_i = 5, 1 \leq i \leq 3$. (7,0,9) $\prec_{\varepsilon-preferred}$ (8,7,1), because for the second objective it holds $|0-7| > \varepsilon_2$, where solution (7,0,9) "wins", and for the third it holds $|9-1| > \varepsilon_3$, where solution (8,7,1) "wins". Since each solution has an ε-exceeding objective, *Preferred* is used for comparison. The same argumentation holds for (8,7,1) $\prec_{\varepsilon-preferred}$ (1,9,6) and (1,9,6) $\prec_{\varepsilon-preferred}$ (7,0,9).

3 THE APPLICATION: UTILIZATION PLANNING PROBLEM

The problem of utilization planning, i.e. the *Nurse Rostering Problem* (NRP) (Burke et al., 2004; Burke et al., 2012), is very complex and cannot fully be described with all details. For this, we briefly highlight the main aspects to give an idea of the underlying optimization problem.

The problem is to determine a schedule for employees at a hospital. In our examinations schedules for up to 16 persons for a planning period of 30 days are computed. The computation of the fitness can be roughly categorized in three main areas:

1. Rules resulting from ergonomics, e.g. having regular shifts
2. Restrictions by law, e.g. maximal hours of work per day or maximal working days per month
3. Rules of the nurse station, e.g. sufficient nurses per shift

Some of these constraints are "hard" in the sense that they have to be fulfilled, while others are "soft", i.e. they improve the fitness, but also without them valid schedules result. All together up to 30 optimization objectives are influencing the fitness function. Each one might have a different influence, e.g. some are linear while others are exponential regarding their influence.

In our application we make use of benchmarks for the nurse rostering problem that are reported in (Burke et al., 2012). The benchmarks are available from (Benchmarks, 2012). There, the rules of the nurse station are modeled as hard constraints. The remaining optimization objectives are considered as soft constraints, where a weight for each rule is given in the benchmark. Thus, a weighted sum can be constructed in such a way that a single fitness value is calculated for each employee and schedule. Following this, we get one fitness value for each employee that has to be optimized in parallel. The fitness values are stored in an m-dimensional vector, where $m = \#employees + 1$. One dimension gives the objective function for the hard constraint and the remaining dimensions the soft constraints for each employee.

Name/Day	1	2	3	4	5	6	7	8	9	...	30
C. Meyer	S	S	D	D	V	V	V	-	N	...	N
J. Smith	D	D	-	-	N	N	N	-	-	...	V
J. Doe	-	-	D	D	D	D	-	-	-	...	D
J. Blow	F	F	-	L	L	L	-	-	F	...	-
J. Bloggs	S	V	V	-	-	N	N	N	-	...	-
...											

Figure 1: Example nurse rostering schedule.

Example 2. To give a better understanding of the approach in Figure 1 a sketch of a schedule is given. Depending on the grade of training the optimization algorithm assigns exactly one shift to a person per day. In this example all given shifts are marked with a letter. These letters have the following meaning: Day shift (D), Night shift (N), Late shift (L), Vacation (V), Free shift (F) and Stand-by shift (S). For more details see (Burke et al., 2004; Burke et al., 2012; Benchmarks, 2012).

4 RELATION *Prio-ε-Preferred*

Most often in real-world applications the objectives have different priorities, i.e. one objective is more important than another objective. For this it is of high relevance to model the priorities during the optimization process. For the NRP the user preferences have been pointed out in the previous section. In the following we propose a new model that combines multiple optimization criteria with user preferences.

4.1 Definition

To model priorities of optimization objectives in this approach a new relation *Prio-ε-Preferred* is defined. In (Schmiedle et al., 2001) relation *Priority Preferred* has been defined that combines relation *Preferred* with priorities. Following this, the model proposed in this approach combines relation ε-*Preferred* with a lexicographic ordering of the objectives.

Let us assume that priorities $1,2,\ldots,k$ are assigned to the m objectives in an ascending ordering, i.e. the lower the index $i, 1 \leq i \leq k$, the higher the priority.

Definition 5. Let $p = (p_1,\ldots,p_k)$ be a priority vector. p_i determines the number of objectives that have priority i. The priority of an objective is calculated by the function $pr : \{1,\ldots,m\} \to \{1,\ldots,k\}$. The subvector of objectives $c|_i$ of priority i is defined as

$$c|_i \in \mathbb{R}^{p_i}, c|_i = (c_r,\ldots,c_s),$$

where

$$r = \sum_{j=1}^{i-1} p_j + 1 \wedge s = \sum_{j=1}^{i} p_j.$$

For $A,B \in \Omega$ relation $\prec_{\varepsilon-priopref}$ (*Prio*-ε-*Preferred*) is defined by:
$A \prec_{\varepsilon-priopref} B :\Leftrightarrow$
$\exists j \in \{1,\ldots,k\} : A|_j \prec_{\varepsilon-preferred} B|_j \wedge$
$(\forall h < j : A|_h \not\prec_{\varepsilon-preferred} B|_h \wedge B|_h \not\prec_{\varepsilon-preferred} A|_h)$

More informally, *Prio*-ε-*Preferred* considers the objectives that have the same priority and then performs comparisons using the ε-*Preferred* relation on these objectives. The objectives are considered in descending order of their priority. A solution A is better than B with respect to relation *Prio*-ε-*Preferred* if the objectives with highest priority of A are ε-*Preferred* to B. If they are not comparable or equal, the objectives with the next priority in descending ordering are considered and compared using the ε-*Preferred* relation. This is done until the better solution, A or B, is found or all priorities are considered. If no better solution is found, A and B are denoted as incomparable.

Example 3. Let us consider a problem with 5 objectives and 3 different priorities. Let $c = (c_1,c_2,c_3,c_4,c_5)$ a solution vector and $p = (1,3,1)$ a priority vector, i.e. one objective has priority 1 (i.e. $p_1 = 1$), three objectives have priority 2 ($p_2 = 3$) and one objective has priority 3 ($p_3 = 1$). This leads to the function pr with $pr(1) = 1$, $pr(2) = 2$, $pr(3) = 2$, $pr(4) = 2$, and $pr(5) = 3$ what means that the first objective has priority 1, the second objective priority 2, and so on. For priority 2 the projection is $c|_2 \in \mathbb{R}^3, c|_2 = (c_2,c_3,c_4)$, since $r = 1+1 = 2$ and $s = 1+3 = 4$.

Now, let us consider two solution vectors, $A = (2,7,0,9,15)$ and $B = (2,1,9,6,5)$. Then it holds, that $B \prec_{\varepsilon-priopref} A$. For this, first the objectives with priority 1 are compared. Since they are equal, next the objectives with priority 2 are compared with relation $\prec_{\varepsilon-preferred}$, i.e. $(1,9,6) \prec_{\varepsilon-preferred} (7,0,9)$ (see Example 1) which leads to $B \prec_{\varepsilon-priopref} A$. The last objective does not have to be considered anymore, because it has lowest priority and the decision, which solution is better with respect to relation $\prec_{\varepsilon-priopref}$, has already been made.

Then, after a pairwise comparison using *Prio*-ε-*Preferred* analogously to (Sülflow et al., 2007) a relation graph is constructed. Then the strongly-connected components are computed to perform a final ranking as described in (Sülflow et al., 2007).

4.2 Methods

To test and compare the methods proposed in this paper, a framework of an *Evolutionary Algorithm* (EA) that optimizes schedules of the NRP has been used. Details of the representation of the individuals and the genetic operators are left out due to page limitation. The objective function measures the fitness of each individual/schedule, i.e. the rules given by the benchmarks are evaluated and the hard constraint for the schedule and soft constraints for each employee are calculated. These optimization rules are given in the benchmarks. The first objective of the m-dimensional fitness vector gives the hard constraint and the remaining objectives the soft constraints. The priorities for the optimization objectives are set such that the first objective, the hard constraint, is of highest priority 1 and the soft constraints for the employees have priority 2. Using the notation from Definition 5, for the two priorities we have $p = (1,m-1)$, $pr(1) = 1$ and $pr(i) = 2, 2 \leq i \leq m$.

5 EXPERIMENTAL RESULTS

In this section we give an insight into the behavior of the many-objective optimization methods for the nurse rostering problem presented in this paper. For the experiments, the algorithms are applied to benchmarks that are taken from (Benchmarks, 2012).

To compare the results of the high dimensional optimization, we use a weighted sum approach to transform the fitness vector back to one dimension. The justification of the weights results from the experience of experts and they are given in the benchmarks. To measure the influence of random seeds on the results, the random number generator has been initialized with 10 different values. The results in the following give the average value for these 10 runs. The average values of the results are used for comparison of the methods in the following. In all experiments the population size is set to 50 and the EA runs for 5000 generations.

Table 1: Fitness for generation 5000.

Benchmark	#Objectives	Method	*AVG*	Quality
GPost	9	NSGA-II	36993	0%
		PrioPref	10512	71%
		Prio εPref -5000	10485	72%
		Weighted Sum	7159	80%
Millar-2Shift-DATA1	9	NSGA-II	2140	0%
		PrioPref	2670	-24%
		Prio εPref -5000	2720	-27%
		Weighted Sum	1310	39%
ORTEC01	17	NSGA-II	92022	0%
		PrioPref	12976	86%
		Prio εPref -5000	15448	83%
		Weighted Sum	9132	90%
Valouxis-1	17	NSGA-II	114604	0%
		PrioPref	21048	82%
		Prio εPref -5000	19872	83%
		Weighted Sum	13986	88%

Table 2: Comparison of different epsilon values after 5000 generations.

Benchmark	Prio εPref -5000		Prio εPref -1000		Prio εPref -500		Prio εPref -10	
	AVG	Quality	AVG	Quality	AVG	Quality	AVG	Quality
GPost	10485	0%	9014	14%	7925	24%	8502	19%
Millar-2Shift	2720	0%	1970	28%	1540	43%	2340	14%
ORTEC01	15448	0%	11181	28%	11275	27%	14496	6%
Valouxis-1	19872	0%	17508	12%	17018	14%	17954	10%

First, the presented approaches PrioPref and Prio εPref are compared to the well-known method NSGA-II . The results are summarized in Table 1. The epsilon value in Prio εPref is set to 5000 by the user. Experiments have shown that this setting is a good starting point for our investigations. It can be seen that both methods based on relation *Preferred* outperform NSGA-II enormously in most cases. Only for benchmark *Millar-2Shift-DATA1* NSGA-II has a better performance, but for the other cases it is improved by more than 70%. In row *Weighted Sum* additionally the results for the weighted sum approach, i.e. a single objective optimization approach, are given. The results obtained by the weighted sum are better than the results calculated by the multi-objective optimization methods. This can be explained by the fact that the benchmarks are designed such that the weights are directly given, which is an advantage of the weighted sum approach.

For the experiments above a relatively high epsilon value has been used. In a next series of experiments we briefly discuss the influence of alternative choices, the results are summarized in Table 2. By this, directions for improvements are pointed out (see also Section 6).

As reference for the experiments we use the epsilon value of 5000 from the previous section. The results are given in column Prio εPref -5000. In a first run denoted in column Prio εPref -1000 the epsilon values of the optimization objectives were set to 1000. Even by these first experiments, the quality could be improved by more than 10% to nearly 30%. The trend holds, if epsilon value 500 is considered. Especially for benchmark *Millar-2Shift-DATA1*[3] improvements of more than 40% can be observed. If the value of ε was too low, e.g. 10 as given in Table 2 in column Prio εPref -10, the quality of the results is decreasing. For all considered benchmarks the average value of the fitness function is worse than the quality for epsilon values 1000 and 500, respectively. The experiments also show that the best choice of the epsilon value depends on the considered benchmark. For *ORTEC01* the best average results are determined with epsilon value 1000, while the best results for *GPost*, *Millar-2Shift-DATA1* and *Valouxis-1* are obtained for epsilon value 500. Thus, there is a need for methods to determine a good epsilon value for the bench-

[3] In Table 2 and Table 3 *Millar-2Shift* is the abbreviation for benchmark *Millar-2Shift-DATA1*.

Table 3: Comparison of methods for adapted epsilon values after 5000 generations.

Benchmark	NSGA-II		Prio εPref -1000		Prio εPref -500		SEP		AEP	
	AVG	Quality	AVG	Quality	AVG	Quality	AVG	Quality	AVG	Quality
GPost	36993	0%	9014	75%	7925	79%	15755	57%	7557	80%
Millar-2Shift	2140	0%	1970	8%	1540	28%	2400	-12%	1590	26%
ORTEC01	92022	0%	11181	88%	11275	88%	14058	85%	11672	87%
Valouxis-1	114604	0%	17508	85%	17018	85%	22594	80%	16692	85%

mark under consideration automatically. In summary, based on relation *Prio-ε-Preferred* the quality measured by the fitness value could be significantly improved. Even for example *Millar-2Shift-DATA1* better results have been calculated. But as can be seen by the experiments, there is a need to determine good problem specific epsilon values. For this, in the next section an approach is presented that sets the epsilon values automatically, i.e. no user interaction is needed anymore.

6 ADAPTATION OF EPSILON VALUES

In this section first a description of the automatic adaptation of the epsilon values is given. Then experimental results are carried out to demonstrate the efficiency of the approach.

6.1 The Idea

In the last section we gave insight into the influence of the epsilon values during the optimization process. Altogether, the choice of the epsilon values has a large influence on the quality of the results. Now, the problem is to find a good epsilon value, such that the algorithm has its best performance. For this, two methods *Separated ε-Preferred* (SEP) and *Adapted ε-Preferred* (AEP) have been developed that are introduced in the following. Starting from an initial point, both methods reduce the epsilon values used throughout the algorithm automatically.

For method SEP for each objective one separated epsilon value is provided. It is determined such that for each objective the average fitness value over the whole population is calculated:

$$\varepsilon_j = \sum_{i=1}^{|P|} \frac{Ind_{i,j}}{|P|}, 1 \leq j \leq m,$$

where m is the number of objectives, $|P|$ is the size of the population and $Ind_{i,j}$ is the j-th objective of the i-th individual in population P. The epsilon values $\varepsilon_j, 1 \leq j \leq m$, are updated in each generation.

In contrast, for method AEP one epsilon value for all objectives is determined. Therefore, one individual out of the best *Satisfiability Class* (SC) derived by Prio εPref is randomly chosen. The new epsilon value is determined by the average value of all objectives of that individual:

$$\varepsilon = \sum_{j=1}^{m} \frac{Ind_{best,j}}{m},$$

where $Ind_{best,j}$ is the j-th objective of a randomly chosen individual out of the best *SC*. The epsilon value is updated in each generation. The idea behind this method is that individuals can be distinguished by relation *Prio-ε-Preferred*, if the difference of the individuals exceeds the calculated average range.

6.2 Experimental Evaluation

The results of methods SEP and AEP are summarized in Table 3. For comparison, the results that are derived from NSGA-II and Prio εPref with epsilon values 1000 and 500 are shown in the table. First, we take a closer look to column SEP where the results of separated epsilon values are summarized. The average is better than the average values derived by NSGA-II , but worse than the results derived by Prio εPref -500 and Prio εPref -1000. In column AEP the method for automatic setting of epsilon values shows the same performance than the algorithms that resulted from setting the epsilon values manually. If doing so, many experiments have to be performed to find a good choice of epsilon. In contrast, if using AEP epsilon is set automatically and no user interaction is necessary. Even in two cases (*GPost, Valouxis-1*) the best average quality over all considered epsilon values could be improved. For this it is recommended to use method AEP.

7 CONCLUSIONS

Many-objective optimization is becoming more important in real-world applications. In this article an industrial scheduling problem has been examined. It consists of up to 17 optimization objectives that have to be optimized in parallel. The objectives have different levels of importance that are defined by the user.

A new relation model called *Prio-ε-Preferred* for incorporating user preferences in many-objective optimization has been presented and experimentally evaluated. It was shown, that high quality results could be obtained, i.e. the standard method *NSGA-II* was improved by more than 80%.

Furthermore, a new automatic technique for determining the parameters for relation *Prio-ε-Preferred* automatically has been developed. In this context, the epsilon values have been reduced dynamically during the optimization run. Experiments showed that best quality results could be obtained while adjusting the parameters automatically.

REFERENCES

Auger, A., Bader, J., Brockhoff, D., and Zitzler, E. (2009). Articulating user preferences in many-objective problems by sampling the weighted hypervolume. In *Genetic and Evolutionary Computation Conference*, pages 555–562.

Bader, J. and Zitzler, E. (2011). HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19(1):45–76.

Benchmarks (2012). Employee scheduling benchmark data set: http://www.cs.nott.ac.uk/ tec/nrp/. Technical report, ASAP, School of Computer Science, The University of Nottingham, UK.

Brockhoff, D. and Zitzler, E. (2009). Objective reduction in evolutionary multiobjective optimization: Theorie and applications. *Evolutionary Computation*, 17(2):135–166.

Burke, E., Causmaecker, P. D., Berghe, G., and Landeghem, H. (2004). The state of the art of nurse rostering. *Journal of Scheduling*, 7:441–499.

Burke, E., Curtois, T., Qu, R., and Vanden-Berghe, G. (2012). Problem model for nurse rostering benchmark instances. Technical report, ASAP, School of Computer Science, University of Nottingham, UK.

Corne, D. and Knowles, J. (2007). Techniques for highly multiobjective optimization: Theorie and applications. In *Genetic and Evolutionary Computation Conference*, pages 773–780.

Deb, K. (2001). *Multi-objective Optimization using Evolutionary Algorithms*. John Wiley and Sons, New York.

Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2005). Scalable test problems for evolutionary multiobjective optimization. In *In Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, pages 105–145.

di Pierro, F., Khu, S., and Savic, D. (2007). An investigation on preference order ranking scheme for multiobjective optimization. *IEEE Trans. on Evolutionary Comp.*, 11(1):17–45.

Drechsler, N., Drechsler, R., and Becker, B. (2001). Multiobjective optimisation based on relation favour. In *Int'l Conference on Evolutionary Multi-Criterion Optimization*, pages 154–166.

Fleming, P., Purshouse, R., and Lygoe, R. (2005). Many-objective optimization: An engineering design perspective. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 14–32.

Fonseca, C. and Fleming, P. (1995). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1):1–16.

Geiger, M. (2009). Multi-criteria curriculum-based course timetabling - a comparison of a weighted sum and a reference point based approach. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 290–304.

Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison-Wesley Publisher Company, Inc.

Hughes, E. (2007). Radar waveform optimization as a many-objective application benchmark. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 700–714.

Ishibuchi, H., Tsukamoto, N., and Nojima, Y. (2008). Evolutionary many-objective optimization: A short review. In *IEEE Congress on Evolutionary Computation*, pages 2424–2431.

Koza, J. (1992). *Genetic Programming - On the Programming of Computers by means of Natural Selection*. MIT Press.

Li, X. and Wong, H. (2009). Logic optimality for multiobjective optimization. *Applied Mathematics and Computation*, 215:3045–3056.

Pizzuti, C. (2012). A multiobjective genetic algorithm to find communities in complex networks. *IEEE Trans. on Evolutionary Comp.*, 16(3):418–430.

Schmiedle, F., Drechsler, N., Große, D., and Drechsler, R. (2001). Priorities in multi-objective optimization for genetic programming. In *Genetic and Evolutionary Computation Conference*, pages 129–136.

Sülflow, A., Drechsler, N., and Drechsler, R. (2007). Robust multi-objective optimization in high-dimensional spaces. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 715–726.

Wagner, T. and Trautmann, H. (2012). Integration of preferences in hypervolume-based multiobjective evolutionary algorithms by means of desirability functions. *IEEE Trans. on Evolutionary Comp.*, 14(5):688–701.

Wickramasinghe, U. and Li, X. (2009). A distance metric for evolutionary many-objective optimization algorithms using user-preferences. In *22nd Australasian Joint Conference on Advances in Artificial Intelligence (AI'09)*, pages 443–453.

Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Trans. on Evolutionary Comp.*, 3(4):257–271.