Substitute Distance Assignments in NSGA-II for Handling Many-Objective Optimization Problems

Mario Köppen and Kaori Yoshida

Kyushu Institute of Technology, Dept. Artificial Intelligence,
680-4, Kawazu, Iizuka, Fukuoka 820-8502 Japan
{mkoeppen, kaori}@pluto.ai.kyutech.ac.jp

Abstract. Many-objective optimization refers to optimization problems with a number of objectives considerably larger than two or three. In this paper, a study on the performance of the Fast Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) for handling such many-objective optimization problems is presented. In its basic form, the algorithm is not well suited for the handling of a larger number of objectives. The main reason for this is the decreasing probability of having Pareto-dominated solutions in the initial external population. To overcome this problem, substitute distance assignment schemes are proposed that can replace the crowding distance assignment, which is normally used in NSGA-II. These distances are based on measurement procedures for the highest degree, to which a solution is nearly Pareto-dominated by any other solution: like the number of smaller objectives, the magnitude of all smaller or larger objectives, or a multi-criterion derived from the former ones. For a number of many-objective test problems, all proposed substitute distance assignments resulted into a strongly improved performance of the NSGA-II.

1 Introduction

Recently, there has been increasing awareness for the specific application of evolutionary multi-objective optimization algorithms to problems with a number of objectives considerably larger than two or three. Fleming et al. [8] note the common appearance of such problems in design optimization, and suggested the use of the term *many-objective optimization*. Most evolutionary multi-objective optimization algorithms (EMOs) show a rather decreasing performance, or rapidly increasing search effort for an increasing number of objectives. Other problems with the handling of many objectives are related to the missing means for performance assessment, to difficulties in visualizing results, and to the low number of existing, well-studied test problems. The DTLZ suite of test problems [6,7] defines most of their problems for an arbitrary number of objectives. Results here have been reported for up to 8 objectives [10]. The Pareto-Box problem [12] was also defined for an arbitrary number of objectives, and results were given for up to 15 objectives.

S. Obayashi et al. (Eds.): EMO 2007, LNCS 4403, pp. 727–741, 2007.

The reason for the decreasing algorithm performance is strongly related to the (often even exponentially) growing problem complexity. This growing complexity can be measured by several means. One example for this is, if considering a randomly initialized population, the rapidly decreasing probability of having a pair of solutions, where one solution Pareto-dominates the other. Within the unit hypercube, the expectation value for the number of non-dominated solutions among m randomly selected solutions can be computed by [12]:

$$e_m(n) = m - \sum_{k=1}^{m} \frac{(-1)^{k+1}}{k^{n-1}} \binom{m}{k} \tag{1}$$

where m stands for the number of individuals, and n for the number of objectives. For example, for 15 objectives and 10 individuals, the expectation value for the number of dominated solutions is already as low as 0.0027.

Among the most successful and most often applied EMOs we find the Fast Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) [3,5]. But the poor performance of the NSGA-II algorithm for a large number of objectives has already been reported as well, see e.g. [10,9]. This can be considered a kind of misfortune, as otherwise, the NSGA-II is one of the most attractive EMOs today, due to its simple structure, its availability, the elaborated design of its operations [1], the existence of experience in practical applications, and its excellent performance on the majority of test problems.

This paper attempts to overcome this drawback by analyzing the reasons for NSGA-II's failure in the many-objective optimization domain, and by providing corresponding countermeasures. The main approach, as will be more detailed in section 2, is to replace the crowding distance assignment that is used for secondary ranking among individuals of the same rank. Four methods will be considered here, which all suit better to a larger number of objectives. Section 3 will present results for the convergence metric and Pareto front coverage for a number of many-objective test problems, and section 4 will render conclusions from these results.

2 Substitute Distance Assignments in NSGA-II

2.1 Structure of NSGA-II Algorithm

The outline of the NSGA-II algorithm can be seen in the following listing. Here, we are focussing on a multi-objective minimization problems.

NSGA-II:

$R_t = P_t \cup Q_t$	combine parent and children population
$F = \texttt{fast\_nondominated\_sort}(R_t)$	$F = (F_1, F_2, \cdots)$ all non-dominated fronts of R_t
$P_{t+1} = \emptyset$, $i = 1$	init next parent population
while $\vert P_{t+1}\vert < N$ do	until the parent population is filled

`secondary_ranking_assignment`(F_i)	calculate ranking values in F_i
$P_{t+1} = P_{t+1} \cup F_i$, $i = i + 1$	include i-th non-dominated front in the
end	parent population
Sort($P_{t+1}, \geq_n$)	sort in descending order using $\geq_n$
$P_{t+1} = P_{t+1}[0 : N]$	choose the first N elements of P_{t+1}
$Q_{t+1} =$ `make_new_population`(P_{t+1})	use selection, crossover and mutation
$t = t + 1$	to create a new population Q_{t+1}

For each generation t, the algorithm maintains an external population of N parent individuals P_t and creates a child population Q_t from the parents. Both populations, fused together, are lexicographically sorted by two different global ranking measures. The first is the non-dominated sorting, as result of the procedure `fast_nondominated_sort`. For details of its implementation, see [3]. The main outcome of this procedure is the assignment of a *rank* to each solution in the set R_t. Two solutions of the same rank do not dominate each other, but for each solution of rank $r > 1$, there exists at least one dominating individual of lower rank. The rank 1 is assigned to all solutions that are in the Pareto set. Thus, the rank value implies a total ordering of the set of solutions in the algorithm for each generation.

To yield a more competitive ordering, NSGA-II also assigns a secondary ranking measure to each solution. So far, only the `crowding-distance-assignment` has been considered, as given in the following listing:

CROW-DIST: `crowding-distance-assignment`(I)

$l = \|I\|$	number of solutions in I
for each i, set $I[i].dist = 0$	initialize distance
for each objective m do	
$I = \text{sort}(I, m)$	sort using each objective value
$I[1].dist = I[l].dist = \infty$	so that boundary points are always selected
for $i = 2$ to $(l - 1)$ do	for all other points
$I[i].dist = I[i].dist + (I[i+1].m - I[i-1].m)$	
end	
end	larger *dist* count better

This distance measure is well suited for a later stage of the algorithms' application, where the population is already close to the true Pareto front of the problem (hopefully). It forces the solutions to keep distance to their neighboring solutions in objective space. Using this distance in addition to the ranking, the comparison of two solutions is based on the ordering relation $\geq_n$:

$$i \geq_n j \text{ if } (i_{rank} < j_{rank}) \text{ or } ((i_{rank} = j_{rank}) \text{ and } (i_{dist} > j_{dist})) \qquad (2)$$

2.2 NSGA-II and Many Objectives

In the case of a larger number of objectives, the performance of NSGA-II is notably dropping, down to a level, where its behavior resembles more or less a random search [12]. Considering the plot in fig. 1, some insight into this phenomenon can be yielded. For an initial parent population of 100 individuals, ranks have been computed. The plot shows the average number of rank 1 solutions over 100 such random initializations and with increasing number of objectives for the DTLZ2 and DTLZ6 problems. For more than 2 or 3 objectives, the amount of rank 1 solutions sharply increases. For the (also known to be more complex) DTLZ6 problem, the rank 1 rapidly accounts for more than 90 percent of the population. For the partial ordering used in the NSGA-II, this means that most of the ranking now is delegated to the secondary ranking assignment, i.e. the crowding distance comparison. However, measuring crowding in an initial population is randomized as well, and as a result, the algorithm gets stuck right from the beginning.

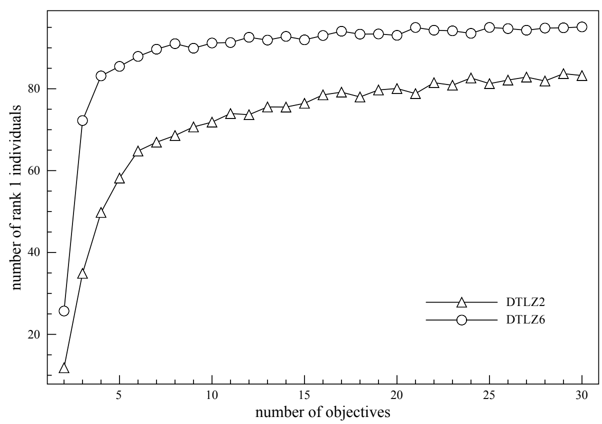

Fig. 1. Average number of rank 1 individuals in an initial random parent population of 100 individuals for the DTLZ2 and DTLZ6 problems with 2 to 30 objectives

It seems suitable to consider a different way for secondary ranking assignment in the first (explorative) generations of the algorithm, in order to avoid the algorithm getting stuck. This will be discussed in the next subsection.

2.3 Secondary Ranking Assignment by Pareto Dominance Degrees

As it was already pointed out in the introduction, with increasing number of objectives the appearance of Pareto-dominance among the solutions becomes more and more unlikely. However, two solutions can be close to the situation where one solution Pareto-dominates the other. As a basic idea, we are going to measure this kind of closeness and use such measurements instead of the crowding distance for the secondary ranking in the NSGA-II algorithm.

The degree, by which a solution A is nearly-dominated by a solution B, can be related to more than one criterion. Basically, the following independent cases can be considered:

– the number of smaller or larger objectives;
– the magnitude of all smaller or larger objectives; or
– a multi-criterion based on the former ones.

In the following, we are going to consider measurements for all these cases. The measurements take advantage of the fact that in NSGA-II, due to the non-dominated sorting, the secondary ranking is only applied to solution sets I where no solution Pareto-dominates any other solution of the same set: $pareto\_set(I) = I$.

Subvector dominance (SV-DOM): given two solutions A and B, the procedure $svd(A, B)$ directly counts the number of objectives of B that are smaller than the corresponding objectives in A. For each solution $I[i]$ in a set I of solutions, the largest such value among all other solutions is assigned as distance value to $I[i]$. The smaller this value, the smaller is the number of lower objectives that appear among all other members of the set I. Such a solution is more close to being not Pareto-dominated by any other solution. For a strongly Pareto-dominated solution, its distance equals the number of objectives. In [2], such a measure was used for the so-called *efficiency of order k*-selection among Pareto optimal solutions. The pseudo-code for computing SV-DOM is as follows:

SV-DOM: `subvector-dominance-assignment`(I)

def $svd(i, j)$	comparing solution i with j
$cnt = 0$	initialize counter
for each objective m do	
$cnt = cnt + 1$ if $I[j].m < I[i].m$	count number of smaller objectives
return cnt	
end	
for each $i = 1, \cdots, \vert I\vert$ do	for all solutions $I[i]$
set $I[i].dist = 0$	initialize distance
for each $j \neq i$ do	among all other solutions j
$v = svd(i, j)$	find the one with the largest
if $I[i].dist < v$ then $I[i].dist = v$	number of smaller objectives;
end	this j gives the distance value for i
end	smaller *dist* count better

-eps-dominance ($-\epsilon$-DOM): for two solutions A and B of the solution set, the procedure $mepsd(A, B)$ considers all objectives of B that are larger than the corresponding objectives of A (i.e. worse). It computes the smallest value ϵ, which, if subtracted from all objectives of B, makes B Pareto-dominating A. This corresponds to the concept of additive ϵ-dominance. For each solution $I[i]$ in a set I of solutions, the smallest such value among all other solutions is assigned as distance value to $I[i]$. The larger this distance for a solution, the higher the "effort" that would be needed to make the other solutions Pareto-dominating the former. For a Pareto-dominated solution, the distance is 0. The ϵ-DOM distance can also be computed as follows:

-eps-DOM: `meps-dominance-assignment`(I)

```
def mepsd(i, j)                                  comparing solution i with j
  max = 0                                        initialize maximum variable
  for each objective m do
    if I[j].m > I[i].m then                      for all larger objectives
      max = max [I[j].m − I[i].m, max]           get largest differing objective
  end
  return max
end
for each i = 1, ··· , |I| do                     for all solutions I[i]
  set I[i].dist = ∞                              initialize distance
  for each j ≠ i do                              among all other solutions j
    v = mepsd(i, j)                              find the one with the smallest
    if I[i].dist > v then I[i].dist = v          maximal differing larger objective;
  end                                            this j gives the distance value for i
end                                              larger dist count better
```

Fuzzy Pareto dominance (FPD): Given two solutions A and B, this procedure accounts for all objectives of B that are also larger than the corresponding objectives of A (i.e. worse). Instead of seeking the maximum difference (as we did for ϵ-DOM), and thus basing the comparison onto a single objective only, we are going to fuse all the magnitudes of larger objectives into a single value. The procedure equals the Fuzzy-Pareto-Dominance relation as presented in [11]. It uses the notion of bounded division of two reals x and y from $[0, 1]$:

$$\left[\frac{x}{y}\right] = \begin{cases} 1, & \text{if } y \le x \\ x/y, & \text{if } x < y \end{cases} \tag{3}$$

All bounded quotients of corresponding objectives in A and B are multiplied. For a smaller objective in B, this gives a factor of 1. Thus, if A is Pareto-dominated by B, the measure becomes 1. For each solution $I[i]$ in a set I of solutions, the largest product value from all other solutions is assigned as distance value to $I[i]$. The smaller this value, the lower the degree by which a solution is dominated by any other solution in I. The pseudo-code for FPD distance measure is as follows:

FPD: `fuzzy-pareto-dominance-assignment`(I)

def $fpd(i, j)$	comparing solution i with j
$cv = 1$	initialize comparison value
for each objective m do	
$cv = cv \cdot [I[i].m/I[j].m]$	multiply bounded quotient
end	
return cv	
end	
for each $i = 1, \cdots, \|I\|$ do	for all solutions $I[i]$
set $I[i].dist = 0$	initialize distance
for each $j \neq i$ do	among all other solutions j
$v = fpd(i, j)$	find the one with the largest
if $I[i].dist < v$ then $I[i].dist = v$	comparison value to i;
end	this j gives the distance value for i
end	smaller *dist* count better

Sub-objective dominance count (SOD-CNT): None of the methods introduced so far regards for *all* aspects of the ranking relation between two solutions. If the comparison is based on all larger objectives, the information about smaller objectives is neglected, and vice versa. If the number of larger objectives is considered, nothing is known about the difference in the magnitudes of these objectives. Thus, we are also considering a multi-criterion here, and provide a distance assignment procedure for such a multi-criterion ranking.

Taking any solution A of a (non-dominated) solution set I, we derive a set S_A of all pairs of two single-criterion distance measures to all other solutions B of the set. In this study, we take the pair $M - svd(A, B)$ (M is the number of objectives) from SV-DOM distance and $mepsd(A, B)$ from $-\epsilon$-DOM distance. This set S_A has a Pareto set, which is composed of all solutions that "perform well" against A. Each solution in I gets the number of occurrences in all the possible Pareto sets PSO_A assigned. The higher this number, the more often the corresponding solution "performs well" against some other solution in I. In pseudo-code:

SOD-CNT: `subobjective-dominance-count-assignment`(I)

for each i, set $I[i].dist = 0$	initialize distance
for each $i = 1, \cdots, \|I\|$ do	for all solutions $I[i]$
$S_i = \{(M - svd(i, j), mepsd(i, j)) \mid j \neq i\}$	
	all pairs of subvector dominance
	and -eps-dominance distances
$POS_i = pareto\_set(S_i)$	get the Pareto set of S_i
for each $j \in PSO_i$ do	for each solution j in that Pareto set
$I[j].dist = I[j].dist + 1$	increment counter of solution j;
end	larger *dist* count better

For better understanding, we will provide an example for the computation of SOD-CNT. Consider the four vectors $A = (2, 1, 3), B = (1, 2, 4), C = (3, 3, 1)$ and $D = (1, 4, 1)$. The values for $M - svd(row, column)$ and $meps(row, column)$ are given in the following tables:

	(2,1,3)	(1,2,4)	(3,3,1)	(1,4,1)
(2,1,3)	-	2	2	1
(1,2,4)	1	-	2	2
(3,3,1)	1	1	-	2
(1,4,1)	2	2	2	-

$M - svd(row, column)$

	(2,1,3)	(1,2,4)	(3,3,1)	(1,4,1)
(2,1,3)	-	1	2	3
(1,2,4)	1	-	2	2
(3,3,1)	2	3	-	1
(1,4,1)	2	3	2	-

$mepsd(row, column)$

For example, the entry "2" in the second column of the first row in the left-hand table indicates that the solution $(1, 2, 4)$ has one smaller objective than $(2, 1, 3)$. Thus, the entry is $3 - 1 = 2$. The corresponding entry in the right-hand table indicates that at least 1 has to be subtracted from all objectives in $(1, 2, 4)$ to make it Pareto-dominating $(2, 1, 3)$.

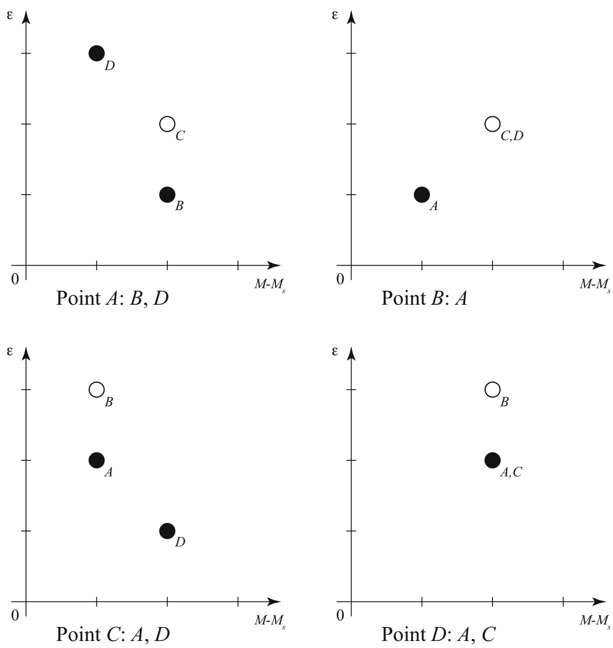

Fig. 2. Example for the computation of the sub-objective dominance count (SOD-CNT) secondary ranking measure

Figure 2 shows the following evaluation for each solution. For example, for solution A, we read the three pairs $(2, 1)$ for comparing with B, $(2, 2)$ for comparing with C, and $(1, 3)$ for comparing with D from the tables above. The Pareto set of these three pairs is the set $\{(2, 1), (1, 3)\}$, which refers to the solutions B and D (black circles in the figure). Doing this for all four solutions, A appears three times in such a Pareto set, B one time, C one time and D two times. This equals the distance assignment to the four solutions, and gives A to be of higher value for the secondary ranking, followed by D, and B, C having lowest ranking value.

2.4 Using the Substitute Distance Assignments

In the NSGA-II algorithm, the four proposed distance assignment procedures are used the same way as the crowding distance, as given by eq. (2). For SV-DOM and FPD, the ">" has to be replaced by "<", as for these procedures, smaller values count better.

3 Results

3.1 Convergence Metric

In this subsection, we present some results that were obtained using the newly introduced substitute distance assignments. As test problems, DTLZ2, DTLZ3 and DTLZ6 for 2, 8 and 15 objectives have been used. Since these test problems are well covered in literature, and also for limited space reason, we are not going to provide details of the definitions of these test problems here. For details, the reader is kindly referenced to the literature [6,7]. Also, the genuine many-objective Pareto-Box test problem, as introduced in [12], was studied. Here, to any point $x \in [0, 1]^M$, the M objectives $|x_i - 0.5|$ were assigned.

As performance measure, the convergence metric [4] was used. In case of DTLZ2, DTLZ3 and DTLZ6, this measure simplifies to $|I| - 1$, where I is a solution (vector of objectives). In case of the Pareto-Box problem, the convergence metric can be also simply computed by $|I|$, as the task here is to come close to the mid-point of the unit hypercube.

The settings for the NSGA-II algorithm were the same as used in [10]: crossover probability 0.7, distribution index for SBX 15, mutation probability $1/M$, and distribution index for polynomial mutation 20. However, due to a better fit to the many-objective optimization domain, search effort was kept small. In all cases, a population of 20 individuals was used, and each experiment went over 300 generations. This is equal to the smallest settings that were used in [10].

The results listed in table 1 were achieved by averaging the minimal convergence metric of a population over the 300 generations for 30 runs each. The reason that no archive was used is as follows: for a larger number of objectives, any solution tends to be included in the archive, as Pareto dominance is becoming more unlikely. Thus, the procedure for reducing archive size equals more or less the procedure of elitist selection in the population itself, and it is easily observed that the values of the convergence metric for both sets do not differentiate much. The presented substitute distance assignments could be considered for adding to an archive in a many-objective optimization problem as well. This

Table 1. Results of the application of the substitute distance assignments to common test problems with increasing complexity

Obj.	CROW-DIST	SV-DOM	-ϵ-DOM	FPD	SOD-CNT
Convergence Metric for Pareto-Box					
2	$(3 \pm 3) \cdot 10^{-5}$	$(5 \pm 7) \cdot 10^{-5}$	$(3 \pm 4) \cdot 10^{-5}$	$(7 \pm 9) \cdot 10^{-5}$	$(8 \pm 10) \cdot 10^{-5}$
8	0.49 ± 0.04	$(16 \pm 18) \cdot 10^{-4}$	0.028 ± 0.006	$(6 \pm 15) \cdot 10^{-4}$	$(8 \pm 5) \cdot 10^{-4}$
15	0.98 ± 0.07	0.02 ± 0.01	0.066 ± 0.008	0.09 ± 0.08	0.005 ± 0.001
Convergence Metric for DTLZ2					
2	$(9 \pm 2) \cdot 10^{-4}$	$(3 \pm 2) \cdot 10^{-4}$	$(8 \pm 2) \cdot 10^{-4}$	$(5 \pm 2) \cdot 10^{-4}$	$(7 \pm 4) \cdot 10^{-5}$
8	0.80 ± 0.07	$(3 \pm 2) \cdot 10^{-4}$	0.029 ± 0.007	0.15 ± 0.06	$(11 \pm 10) \cdot 10^{-5}$
15	0.81 ± 0.06	0.002 ± 0.002	0.06 ± 0.01	0.3 ± 0.1	$(14 \pm 17) \cdot 10^{-5}$
Convergence Metric for DTLZ3					
2	22 ± 9	20 ± 10	15 ± 9	16 ± 10	16 ± 6
8	890 ± 60	50 ± 20	40 ± 20	230 ± 50	30 ± 10
15	990 ± 80	80 ± 30	60 ± 20	400 ± 100	30 ± 10
Convergence Metric for DTLZ6					
2	0.7 ± 0.2	0.7 ± 0.2	0.51 ± 0.06	0.8 ± 0.3	0.6 ± 0.1
8	9.05 ± 0.09	7.4 ± 0.6	6.6 ± 0.9	8.7 ± 0.3	3.8 ± 0.9
15	9.05 ± 0.08	8.1 ± 0.4	8.2 ± 0.5	8.9 ± 0.1	4.8 ± 0.8

is the topic of an on-going study of the authors. The $\pm$-values in table 1 refer to the corresponding standard deviation from the sample average.

The results clearly demonstrate the better performance of *all* substitute distance assignments, even in the case of two objectives. Between the substitutes, SOD-CNT achieves the best results, and FPD the worst (but generally still better than CROW-DIST). Table 2 gives a comparison to results from literature. This shows the modified NSGA-II also to be highly competitive, especially regarding the comparable low effort that was needed to achieve the given convergence metric values.

3.2 Pareto Front Coverage

Having found substantially better convergence metric values in the former subsection, the question about Pareto front coverage has to be considered as well. However, the quantitative assessment of the coverage is not simple. So far, several ad hoc approaches for certain test problems, with a focus on visualization have been presented (as e.g. in [10]). For doing similarly for many objectives, we propose a class of test problems that allows for easy visualization and evaluation of Pareto front coverage, which is referred to as *P* problem* for indicating the variable number of points from which the objectives are derived.

Given is a set P of m points P_i in the Euclidian plane (the case of two dimensional Euclidian space is completely sufficient for the present analysis). The feature space F equals the Euclidian plane, where the points P_i are located. The objective space O is an m-dimensional vector space. For a given point x in

Table 2. Comparison of results with the results reported in [10]. Column 2 also lists average convergence metric values for 1000 randomly initialized vectors. The entries in the columns entitled Effort are population size times number of generations that were used to achieve the reported results.

Obj.	**Random**	**PESA**[10]	**NSGA-II**[10]	**Effort**	**SOD-CNT**	**Effort**
Convergence Metric for DTLZ2						
2	0.83	0.00008	0.00180	$20 \cdot 300$	0.00007	$20 \cdot 300$
8	0.84	0.00689	2.30766	$600 \cdot 600$	0.00011	$20 \cdot 300$
15	0.83	-	-	-	0.00014	$20 \cdot 300$
Convergence Metric for DTLZ3						
2	1077.7	22.52023	21.32032	$20 \cdot 500$	16.0	$20 \cdot 300$
8	1082.2	7.23062	1753.41364	$600 \cdot 1000$	30.0	$20 \cdot 300$
15	1079.3	-	-	-	30.0	$20 \cdot 300$
Convergence Metric for DTLZ6						
2	9.10	0.79397	0.63697	$20 \cdot 500$	0.6	$20 \cdot 300$
8	9.08	6.32247	10.27306	$600 \cdot 1000$	3.8	$20 \cdot 300$
15	9.08	-	-	-	4.8	$20 \cdot 300$

the feature space, its objective vector $o(x)$ is the vector with the components $o_i = d(x, P_i)$ for $i = 1$ to m, where $d(a, b)$ is the Euclidian distance of two points $a, b \in F$. Thus, the objectives to minimize are the distances to a given collection of points, where the distance to any of these point is treated as an independent objective.

The Pareto set of this problem, i.e. the set of feature vectors giving objective vectors that are not dominated by any other feasible objective vector (in other words, are closest to all points P_i), equals the convex closure of the points P_i. Here, convex closure means the union of the volume enclosed by the convex hull and the convex hull itself.

To see this, consider the left subfigure of fig. 3. Consider any point X in the plane that does not belong to the convex closure of the points P_i. Assume that the convex hull of the points P_i is being established by the poly-line $\overline{A_1 A_2 \ldots A_n A_{n+1=1}}$, where each $A_i \in P$. As X is outside the enclosed area, there must be a connection $\overline{A_i A_{i+1}}$ such that all points of P are either on the connecting line, or on the opposite side of the connecting line than X. Dropping a perpendicular from X to the connecting line, one can see that any point Y between X and the line is more close to any point on the opposite side of the connecting line, and more close to any point on the connecting line as well. Thus, for any point X outside the convex closure there is a point that is more close to all points of P, and X does not belong to the Pareto set.

To see that none of the points of the convex closure dominates any other, consider the right subfigure of fig. 3. By connecting any two points U and V of the convex closure and drawing the perpendiculars to this line trough U and through V, the convex closure is segmented into three parts. There is at least one point of the point set A (and thus of P) located to the l.h.s. of the perpendicular

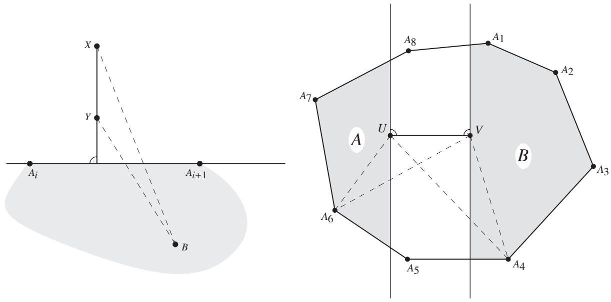

Fig. 3. Illustrating the proof that the Pareto set of the P* problem for the points A_i is the convex closure of these points

through U (indicated by encircled A in the figure), or located on this line, and there is at least one point of A belonging to the r.h.s. of the perpendicular through V or on it (indicated by encircled B). Otherwise, the shape would not be convex. Now, point U is more close to any point of A than V, and point V is more close to any point of B than U. Neither U nor V can dominate the other.

Having thus a rather simple solution structure in the feature space (not objective space, which is high-dimensional), the problem is worth a study for a heuristic algorithm for several reasons:

- the number of objectives can be easily scaled
- by reducing the area enclosed by the convex closure, the effort for random search (the “Monte-Carlo Barrier”) can be easily increased
- typical performance measures (as average distance to Pareto front, number of individuals belonging to the Pareto front) can be directly computed
- as the feature space is two-dimensional, the results can be directly visualized; however, the extension to higher-dimensional spaces is straightforward
- the search space is not bounded
- the problem is a continuous optimization problem
- boundary conditions can be directly included
- crowding in objective space directly corresponds to crowding in feature space
- modeling of algorithm behavior seems feasible
- by using the distance to the center of gravity of the points instead, a comparison to the single-objective case becomes possible

We have studied the performance of the modified NSGA-II algorithms on such a 15-objective P* problem.

Figure 4 gives a result that demonstrates how differently the considered methods are behaving. Note that this figure shows the Pareto front of the algorithms and the test problem in feature space, and not in the (15-dimensional) objective space. The best coverage of the polygon is achieved with the FPD method, closely followed by $-\epsilon$-DOM. These are the methods that employed the magnitudes of larger objectives of solutions directly. SVD shows a rather small coverage

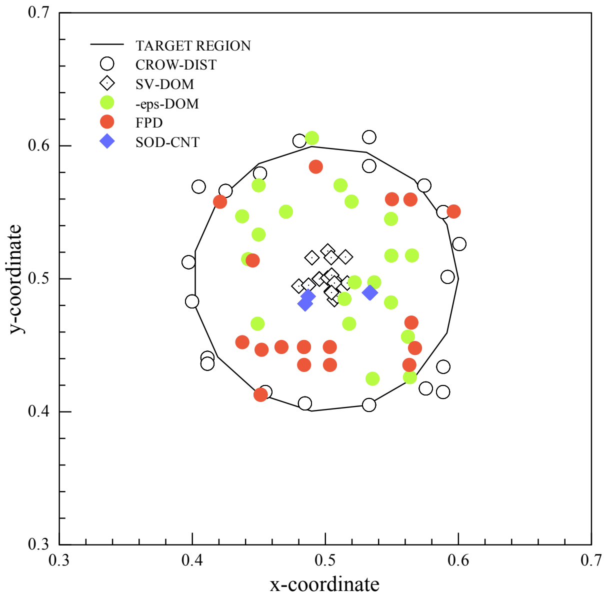

Fig. 4. Final populations using the considered secondary ranking methods on a 15-objective P* problem after 100 generations. Population size is 20.

of the Pareto front, and SOD-CNT, having by far the best convergence metric values, nearly collapses into a single point. Notable also the distribution of the "default" crowding distance measure: as the crowding distance, by construction, keeps extreme individuals in the objective space, it favors individuals that are near to the corners of the polygon. For the P* test problem, this feature of the crowding distance is obviously a drawback.

4 Conclusions

We have studied a number of modifications of the NSGA-II algorithm, to make this algorithm better capable of solving many-objective optimization problems. The modifications were substitutes for the crowding distance assignment, based on closeness of solutions to the case that one solution Pareto-dominates the other. The results and experiences, also taking non-measurable aspects into account, can be summarized as follows:

SV-DOM. This distance measure is very easy to implement, and needs the lowest computation time. It showed the second best results for the convergence metric, but failed in the Pareto front coverage study.

$-\epsilon$-DOM. This distance measure can also be easily implemented, and needs a little more computational effort than SV-DOM. Convergence metric performance was average among the considered modifications, but is accompanied by a good Pareto front coverage. Altogether, this makes this method a good trade-off among all the studied modifications.

FPD. This method has a rather high computational effort, and it also has some formal weaknesses regarding issues of division by 0 etc. Among the studied modifications, the convergence metric performance was worst (however, still better than the crowding distance), but it demonstrated the best Pareto set coverage for the P* problem.

SOD-CNT. The convergence metric of this method is excellent. This good result is weakened by a very poor Pareto front coverage, and also higher computational effort. Moreover, in contrary to the other methods, the processing time is not predictable, as the size of the Pareto sets for the single solutions may vary. During the experiments, we faced an unexpected very long processing time for larger populations and a small number of objectives. Also, if the non-dominated sets gets smaller, the differentiation among the individuals by this measure becomes low. In a few cases, we could even observe convergence of the algorithm, despite of the use of a mutation operation. Some issues regarding this approach, which showed a very good convergence metric, still need further investigation.
To summarize, the results of this study indicate two promising strategies for the application of NSGA-II to many-objective optimization problems: first is to replace the crowding distance completely by $-\epsilon$-dominance distance, second is to use the sub-objective dominance count distance SOD-CNT for the first generations of the algorithm, as long as most of the individuals get rank 1 assigned, and switch to the crowding distance, once the SOD-CNT values tends to be equalized over the whole population.

Acknowledgment

A researcher involved in this study has been supported by a JSPS grant.

References

1. Meghna Babbar, Ashvin Lakshmikantha, and David E. Goldberg. A Modified NSGA-II to Solve Noisy Multiobjective Problems. In James Foster, editor, *2003 Genetic and Evolutionary Computation Conference. Late-Breaking Papers*, pages 21–27, Chicago, Illinois, USA, July 2003. AAAI.
2. I. Das. A preference ordering among various pareto optimal alternatives. *Structural and Multidisciplinary Optimization*, 18(1):30–35, 1999.
3. Kalyanmoy Deb, Samir Agrawal, Amrit Pratab, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, Juan Julian Merelo, and Hans-Paul Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.

4. Kalyanmoy Deb and Sachin Jain. Running Performance Metrics for Evolutionary Multi-Objective Optimization. In Lipo Wang, Kay Chen Tan, Takeshi Furuhashi, Jong-Hwan Kim, and Xin Yao, editors, *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL'02)*, volume 1, pages 13–20, Orchid Country Club, Singapore, November 2002. Nanyang Technical University.
5. Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.
6. Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable Test Problems for Evolutionary Multi-Objective Optimization. Technical Report 112, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 2001.
7. Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable Multi-Objective Optimization Test Problems. In *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 825–830, Piscataway, New Jersey, May 2002. IEEE Service Center.
8. Peter Fleming, Robin C. Purshouse, and Robert J. Lygoe. Many-Objective Optimization: An Engineering Design Perspective. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, pages 14–32, Guanajuato, México, March 2005. Springer. Lecture Notes in Computer Science Vol. 3410.
9. Crina Grosan. Multiobjective adaptive representation evolutionary algorithm (MAREA) - a new evolutionary algorithm for multiobjective optimization. In A. Abraham, Bernard De Baets, Mario Köppen, and Bertram Nickolay, editors, *Applied Soft Computing Technologies: The Challenge of Complexity*, Advances in Soft Computing, pages 113–121. Springer Verlag, Germany, 2006.
10. V. Khare, X. Yao, and K. Deb. Performance Scaling of Multi-objective Evolutionary Algorithms. In Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele, editors, *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, pages 376–390, Faro, Portugal, April 2003. Springer. Lecture Notes in Computer Science. Volume 2632.
11. Mario Köppen and Raul Vicente Garcia. A fuzzy scheme for the ranking of multivariate data and its application. In *Proceedings of the 2004 Annual Meeting of the NAFIPS (CD-ROM)*, pages 140–145, Banff, Alberta, Canada, 2004.
12. Mario Köppen, Raul Vicente-Garcia, and Betram Nickolay. Fuzzy-Pareto-Dominance and Its Application in Evolutionary Multi-objective Optimization. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, pages 399–412, Guanajuato, México, March 2005. Springer. Lecture Notes in Computer Science Vol. 3410.