Ranking-Dominance and Many-Objective Optimization

Saku Kukkonen, *Student Member, IEEE* and Jouni Lampinen

***Abstract*— An alternative relation to Pareto-dominance is studied. The relation is based on ranking a set of solutions according to each separate objective and an aggregation function to calculate a scalar fitness value for each solution. The relation is called as ranking-dominance and it tries to tackle the curse of dimensionality commonly observed in multi-objective optimization. Ranking-dominance can be used to sort a set of solutions even for a large number of objectives when the Pareto-dominance relation cannot distinguish solutions from one another anymore. This permits the search to advance even with a large number of objectives.**

Experimental results indicate that in some cases the selection based on ranking-dominance is able to advance the search towards the Pareto-front better than the selection based on Pareto-dominance. However, in some cases it is also possible that the search does not proceed into direction of the Pareto-front because the ranking-dominance relation permits deterioration of individual objectives.

The results also show that when the number of objectives increases, the selection based on just Pareto-dominance without diversity maintenance is able to advance the search better than with diversity maintenance. Therefore, diversity maintenance connives at difficulties solving problems with a high number of objectives.

I. Introduction

During the last two decades, evolutionary algorithms (EAs) have gained popularity since EAs are capable of dealing with difficult objective functions, which are, *e.g.*, discontinuous, non-convex, multi-modal, and non-differentiable. EAs have also gained popularity in solving multi-objective optimization problems (MOOPs). A term *multi-objective* is used when the number of objectives is more than one. A term *many-objective* is used when the number of objectives is more than two or three (the term is not settled yet).

A MOOP with M objectives can be presented in the form [1, p. 5]:

$$\text{minimize} \quad \{f_1(\vec{x}), f_2(\vec{x}), \ldots, f_M(\vec{x})\},$$

where $\vec{x}$ is a decision vector containing decision variables. Maximization can be easily transformed to minimization, thereby the formulation above is without loss of generality. Minimization of objectives is assumed through this paper.

Multi-objective EAs (MOEAs) are capable of providing multiple solution candidates in a single run that is desirable with MOOPs [2]. However, it has been noticed that the performance of many MOEAs declines fast when the number of objectives increases. Probable reason for this is that many MOEAs, such as the elitist non-dominated sorting genetic algorithm (NSGA-II) [3], the strength Pareto evolutionary algorithm (SPEA2) [4], and generalized differential evolution 3 (GDE3) [5], use Pareto-dominance [6] in their selection operation. If a solution can be improved in such a way that at least one objective improves and the other objectives do not decline, then the new solution Pareto-dominates the original solution. More formally, vector $\vec{x}_1$ Pareto-dominates $\vec{x}_2$, *i.e.*, $\vec{x}_1 \prec_P \vec{x}_2$ iff

$$\forall i : f_i(\vec{x}_1) \leq f_i(\vec{x}_2) \ \wedge \ \exists i : f_i(\vec{x}_1) < f_i(\vec{x}_2).$$

When the number of objectives increases, Pareto-dominating solutions become more seldom and the search based on Pareto-dominance decelerates. This effect is referred as a *curse of dimensionality* and it is illustrated in Fig. 1, where proportions of non-dominated solutions from sets of randomly generated solutions inside a unit hypercube are illustrated. A non-dominated solution is a solution, which is not Pareto-dominated by any other solution. Figure 1 shows results with three different set sizes, and it can be observed that when the number of objectives reaches 20, all the solutions belong to the same non-dominated set of solutions and the search based on Pareto-dominance stagnates. The curse of dimensionality is probably the biggest problem in current MOEAs and it prevents them to be used for problems with more than few objectives. This paper tries to tackle the curse of dimensionality by studying an alternative dominance relation to be used instead of Pareto-dominance.

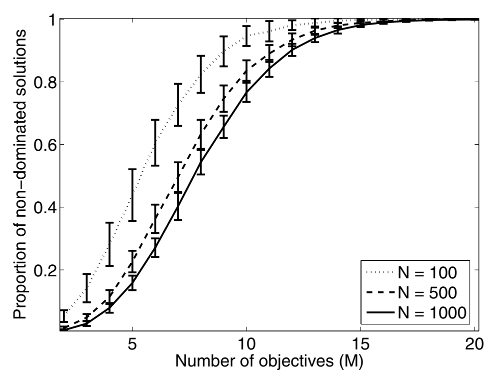

Fig. 1. Proportion of non-dominated solutions with respect to the number of objectives in a set of randomly generated solutions (average and standard deviation values from 101 repetitions).

Since researchers have been aware of the curse of dimensionality, there already exist several approaches for the problem. One of the earliest approaches controls the dominance area to ease ranking of solutions [7]. This idea has also been extended recently [8]. Fuzzification of the Pareto-dominance relation has been studied in [9], [10]. An approach using several different scalarization functions to give a set of

The authors are with the Department of Information Technology, Lappeenranta University of Technology, P.O. Box 20, FIN-53851 Lappeenranta, Finland; email: saku.kukkonen@lut.fi.

1-4244-1340-0/07$25.00 ©2007 IEEE

fitness values to solutions has been developed in [11]. In the approach, fitness values of scalarization functions are ranked and solutions having the best fitness rank for the largest number of the scalarization functions are preferred. One recent approach is comparing two solutions based on which one is better according to a larger number of objectives without violating individual objectives too much [12]. Another recent approach is measuring which one of two solutions is closer to dominate the other [13].

II. Non-Dominated Sorting

A common concept to sort solutions in multi-objective optimization is to use non-dominated sorting [2, pp. 33 – 44]. Non-dominated sorting divides a given set of solutions into different non-dominated sets and gives different classes for different non-dominated sets to reflect their mutual order. First, non-dominated solutions from a given set are searched and they are given a class label, which indicates that they belong to the first non-dominated set. Then, solutions of the first non-dominated set are removed from the original set of solutions and non-dominated solutions are searched from the remaining solutions. These solutions belong to the second non-dominated set. After this, both first and second non-dominated sets are removed from the original set of solutions and non-dominated solutions are found from the remaining solutions, and so fort. This process continues as long as there exist unsorted points.

A result of non-dominated sorting for a set of solutions in a two-objective space is illustrated in Fig. 2. The first non-dominated set has class label 1, the second one has class label 2, etc. Eight solutions form four non-dominated sets. Solutions belonging to the first non-dominated set are considered to be the best solutions in the sense of Pareto-dominance, and better solutions according to their non-dominated set class are preferred in selection operations. As it was noted in Section I, when number of objectives increases, more and more solutions belong to the same non-dominated set and ordering solutions becomes harder. Therefore, the Pareto-dominance relation cannot be used to sort out solutions when the number of objectives becomes large.

III. Ranking-Dominance

An alternative to the Pareto-dominance relation is discussed here. The idea is to find ranks of solutions in terms of each separate objective and use these ranks with a suitable aggregation function to calculate fitness values for the solutions. If an aggregation function is denoted as agg, then the fitness value can be calculated as follows:

Definition 1: Value R_{agg} of vector $\vec{x}_j$ is calculated as:

$$\begin{aligned} R_{agg}(\vec{x}_j) =& agg\left(rank\left(f_1(\vec{x}_j)\right), rank\left(f_2(\vec{x}_j)\right),\right. \\ & \left. \ldots, rank\left(f_M(\vec{x}_j)\right)\right), \end{aligned}$$

where $rank\left(f_i(\vec{x}_j)\right)$ returns the rank of $\vec{x}_j$ in a set of solutions according to objective f_i. Rank value 1 means the best objective value and rank value N means the worst objective

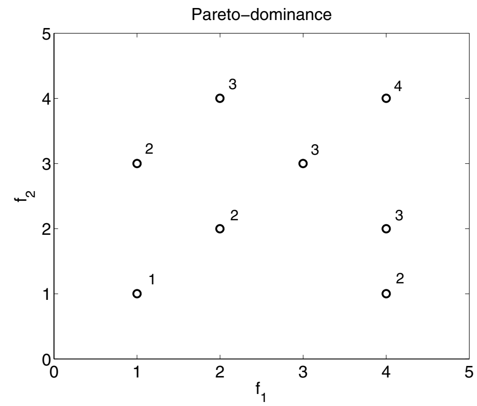

Fig. 2. Non-dominated sets calculated according to non-dominated sorting based on Pareto-dominance.

value. The use of objective ranks has been discussed already in [14] but in a bit different context.

Two aggregation functions, a sum of elements and a minimum of elements, are proposed to be used. With these functions fitness values R_{sum} and R_{min} are calculated in the following way:

$$\begin{aligned} R_{sum}(\vec{x}_j) &= \sum_{i=1}^{M} rank\left(f_i(\vec{x}_j)\right) \\ R_{min}(\vec{x}_j) &= \min_{i=1,\ldots,M} rank\left(f_i(\vec{x}_j)\right). \end{aligned}$$

Figure 3 shows an example of how the R_{sum} and R_{min} values are calculated in the case of five solution candidates with three objectives. Figures 4 and 5 show the R_{sum} and R_{min} values[1] for the same set of solutions as in Fig. 2. Solutions belonging to the same non-dominated set in the sense of Pareto-dominance do not necessarily have a same R_{sum} or R_{min} value. From Fig. 4 it can be noticed that the Pareto-dominating solutions have smaller R_{sum} values compared to the solutions they dominate. This does not hold always for R_{min} values as it can be observed from Fig. 5.

Now, based on the definition of R_{agg}, a new dominance relation can be defined:

Definition 2: If $R_{agg}(\vec{x}_1) < R_{agg}(\vec{x}_2)$, then $\vec{x}_1$ *ranking-dominates* $\vec{x}_2$, and the relation is expressed as $\vec{x}_1 \prec_{R_{agg}} \vec{x}_2$.

An interesting and important theorem for the ranking-dominance relation with the sum aggregation function can be given:

Theorem 1: If $\vec{x}_1 \prec_P \vec{x}_2$, then $\vec{x}_1 \prec_{R_{sum}} \vec{x}_2$.

Proof: $\vec{x}_1 \prec_P \vec{x}_2 \iff \forall i : f_i(\vec{x}_1) \leq f_i(\vec{x}_2) \wedge \exists i : f_i(\vec{x}_1) < f_i(\vec{x}_2) \iff \forall i : rank\left(f_i(\vec{x}_1)\right) \leq rank\left(f_i(\vec{x}_2)\right) \wedge \exists i : rank\left(f_i(\vec{x}_1)\right) < rank\left(f_i(\vec{x}_2)\right) \implies \sum_{i=1}^{M} rank\left(f_i(\vec{x}_1)\right) < \sum_{i=1}^{M} rank\left(f_i(\vec{x}_2)\right) \iff R_{sum}(\vec{x}_1) < R_{sum}(\vec{x}_2) \iff \vec{x}_1 \prec_{R_{sum}} \vec{x}_2$. ■

Thus, it holds that if $\vec{x}_1$ Pareto-dominates $\vec{x}_2$, then also $R_{sum}(\vec{x}_1) < R_{sum}(\vec{x}_2)$, and the ranking-dominance relation

[1] Solutions with a same objective value share a same rank.

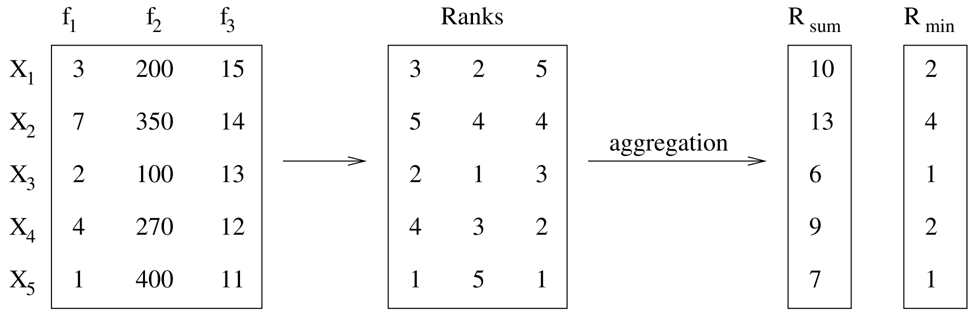

Fig. 3. Calculation of the R_{sum} and R_{min} values with an example of five solution candidates with three objectives.

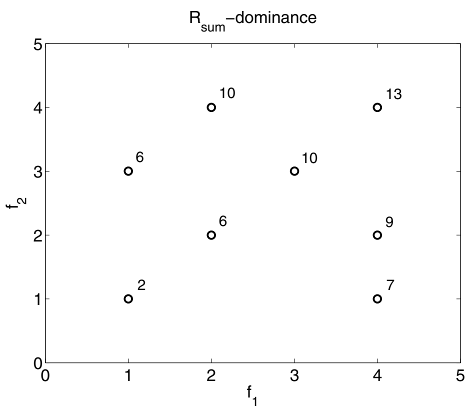

Fig. 4. A set of solutions and the corresponding R_{sum} values.

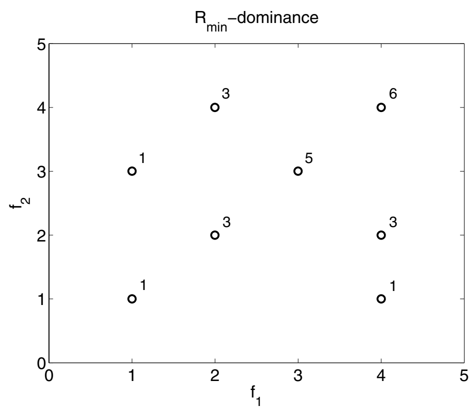

Fig. 5. A set of solutions and the corresponding R_{min} values.

with the sum aggregation function will prefer a Pareto-dominating solution over a Pareto-dominated solution. Opposite does not hold, *i.e.*, $R_{sum}(\vec{x}_1) < R_{sum}(\vec{x}_2)$, does not necessarily imply that $\vec{x}_1$ Pareto-dominates $\vec{x}_2$ (as it can be also noticed from Fig. 4). The ranking-dominance relation with the sum aggregation function is *complete* with respect to $\prec_P$ and *compatible* with a relation $\not\prec_P$ [15]. This means that $\vec{x}_1 \prec_P \vec{x}_2 \implies \vec{x}_1 \prec_{R_{sum}} \vec{x}_2$, and $\vec{x}_1 \prec_{R_{sum}} \vec{x}_2 \implies \vec{x}_2 \not\prec_P \vec{x}_1$. Same properties also hold for hypervolume [15], which is a commonly used performance measurement for the MOEAs but computationally expensive to calculate when the number of objectives increases.

It should be noted that Theorem 1 above does not hold for an arbitrary aggregation function, *e.g.*, a maximum of ranks. Theorem 1 holds for the min aggregation function, if objective values are different thus different ranks can be assigned unambiguously.

Another theorem can be given about the ranking-dominance relation with the sum and min aggregation functions:

Theorem 2: The relation $\prec_{R_{agg}}, agg = \{sum, min\}$ is *transitive*, *i.e.*, $\vec{x}_1 \prec_{R_{agg}} \vec{x}_2 \wedge \vec{x}_2 \prec_{R_{agg}} \vec{x}_3$ implies that $\vec{x}_1 \prec_{R_{agg}} \vec{x}_3$, when $agg = \{sum, min\}$.
Proof for the theorem is given for the sum aggregation function, and the theorem can be proven similarly for the min aggregation function.

Proof: $\vec{x}_1 \prec_{R_{sum}} \vec{x}_2 \wedge \vec{x}_2 \prec_{R_{sum}} \vec{x}_3 \implies \sum_{i=1}^{M} rank\,(f_i(\vec{x}_1)) < \sum_{i=1}^{M} rank\,(f_i(\vec{x}_2)) < \sum_{i=1}^{M} rank\,(f_i(\vec{x}_3)) \implies \sum_{i=1}^{M} rank\,(f_i(\vec{x}_1)) < \sum_{i=1}^{M} rank\,(f_i(\vec{x}_3)) \iff \vec{x}_1 \prec_{R_{sum}} \vec{x}_3$ ■

Ranking-dominance does not restrict deterioration of a solitary objective value with the aggregation functions above. Ranking-dominance can be applied also in situations, where objective values cannot be calculated but only their order can be determined (*e.g.*, situation, where a human is acting as a fitness function). Also, ranking-dominance is independent from the scale of objective values as demonstrated in Fig. 3. Since R_{agg} is set-dependent, it cannot be calculated without other solutions. However, already a set of two solutions is enough to calculate the R_{agg} values and therefore a tournament of two solutions can be performed.

Ranking-dominance can be used to sort a set of solutions even when the number of objectives is large. Figures 6 and 7 illustrate ranking-dominance values for a randomly generated set of 1000 solutions with different number of objectives. The solutions are sorted according to their ranking-dominance values, and the R_{sum} values are divided with the number of objectives to have overlapping curves in Fig. 6. A logarithmic scale is used for R_{min} to ease observation in Fig. 7. When

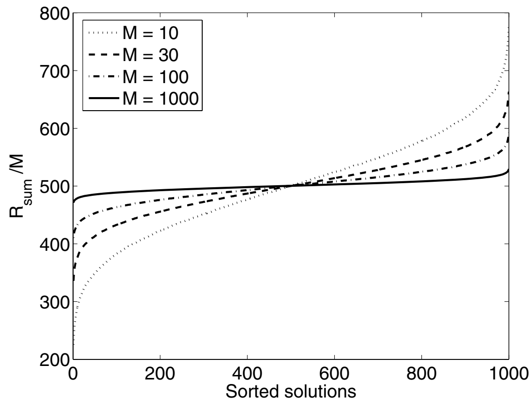

Fig. 6. R_{sum}/M values in sets of 1000 sorted solutions (median values from 101 repetitions).

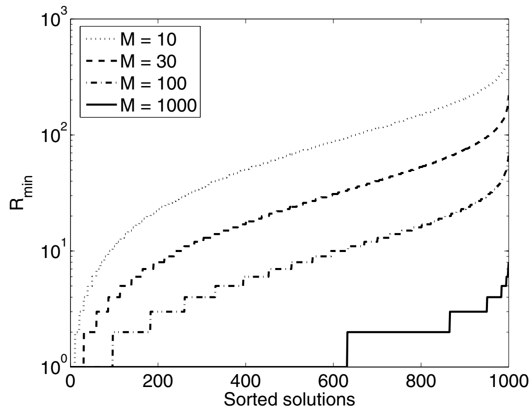

Fig. 7. R_{min} values in sets of 1000 sorted solutions (median values from 101 repetitions).

the number of objectives increases, more and more solutions have a common R_{min} value. Still, a set of solutions can be sorted to several groups based on R_{min} even for a large number of objectives when the Pareto-dominance relation is not able to create distinction anymore.

Sorting a set of solutions using ranking-dominance is a relatively fast operation. Ranking solutions with respect to an objective means sorting objective values. Complexity of sorting is $O(N \log N)$ and sorting is repeated for each objective. After calculation of aggregated values, solutions are sorted according to these values. Total time complexity of ranking-dominance sorting is $O(MN \log N)$, which is considerably less than time complexity class $O\left(N \log^{M-1} N\right)$ of non-dominated sorting [16].

Ranking-dominance can be applied also for constraints. Some existing constraint handling techniques apply Pareto-dominance in the space of constraint (violation) functions [17], [18]. Replacing Pareto-dominance relation with ranking-dominance relation might speed up the search, especially, if a problem contains many conflicting constraints.

IV. Multi-Objective Optimization with Ranking-Dominance

Sorting based on ranking-dominance can be used instead of non-dominated sorting based on Pareto-dominance in MOEAs. However, convergence towards the actual Pareto-optimal front is just a single desired quality of a MOEA. A good MOEA should also be able to maintain diversity along the Pareto-front [2, pp. 426–430]. For example, in NSGA-II at the end of each generation a parent and child population are combined. The combined population is sorted using non-dominated sorting, and a part of the non-dominated sets are selected for the next generation starting from the best non-dominated set. If some non-dominated set does not totally fit in, then required amount of solutions are removed according to diversity among the solutions in this non-dominated set. This approach increases nicely diversity preservation during generations (for optimization problems with a couple of objectives). This is because at the early generations there are many non-dominated sets in the combined population and diversity maintenance has only small effect. When the solutions converge close to the Pareto-front, the size of the non-dominated sets increase and it is likely that already the best non-dominated set of combined population is too large and needs to be reduced [2, pp. 245–246].

Balancing between convergence towards to the Pareto-front and diversity among solutions is difficult. If convergence is emphasized too much, diversity of solutions might be lost. On the other hand, emphasizing diversity preservation will slow convergence and may cause stagnation of the search. In NSGA-II, convergence and diversity preservation are balanced automatically. However, when the number of objectives increases, diversity preservation is more dominating operation and the search slows down.

When the ranking-dominance relation described in Section III is applied instead of Pareto-dominance, diversity preservation becomes difficult since there is no clear indicator when and how much to perform diversity maintenance. One way to solve the problem could be to increase diversity maintenance gradually during generations. A maximum number of generations is the most usual stopping criterion for an EA. Therefore, amount of diversity preservation can be set according to current generation and the known maximum number of generations. A linearly increasing diversity maintenance along generations is illustrated in Fig. 8. At the first generation, the population of the next generation is selected from the combined parent and child population just based on the ranking-dominance relation. In the half of generations, half of the population of the next generation is selected based on the ranking-dominance relation from the combined parent and child population. The rest half of the population is filled with the best members according to diversity from the remaining individuals of the combined parent and child population. At the final generation, individuals from the combined parent and child population are selected just according to diversity.

Of course, the increase of diversity maintenance could also

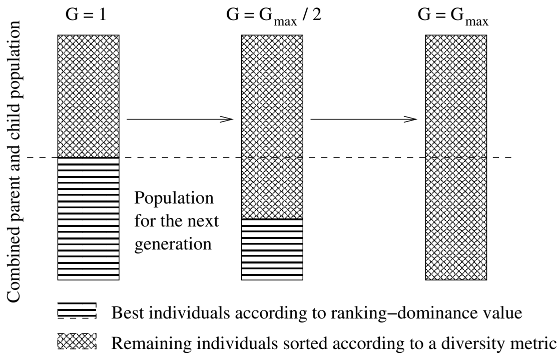

Fig. 8. Linearly along generations increasing diversity maintenance for an EA, where a population for the next generation is selected from the combined parent and child population.

be other than linear. Diversity maintenance should prefer less crowded individuals over more crowded. This can be done effectively and efficiently just based on the distance to the nearest neighbors as presented in [19].

V. Experiments

The ranking-dominance relation was tested using a set of test functions commonly known as DTLZ problems [20]. These minimization problems are synthetic and scalable, thus the number of objectives can be changed. Problems DTLZ1–6 were selected for testing with 2–10, 15, 20, 25, 30, 35, 40, 45, and 50 objectives. Function values of Pareto-optimal solutions are known for these problems. For DTLZ1, the objective function values of a Pareto-optimal solution lie on the linear hyper-plane: $\sum_{i=1}^{M} f_i = 0.5$, and for the rest of the problems, the objective function values of a Pareto-optimal solution lie on the hyper-sphere: $\sum_{i=1}^{M} {f_i}^2 = 1$. Therefore convergence can be easily measured with an error value $E = 1/N \sum_{j=1}^{N} \left(\sum_{i=1}^{M} f_i - 0.5 \right)$ for DTLZ1 and $E = 1/N \sum_{j=1}^{N} \left(\sum_{i=1}^{M} {f_i}^2 - 1 \right)$ for the other DTLZ problems. For the Pareto-optimal solutions $E = 0$, and for non-Pareto-optimal solutions $E > 0$. Shape of the Pareto-front is a surface in hyperspace for DTLZ1–4 but a curve for DTLZ5,6.

The ranking-dominance relation was implemented in generalized differential evolution 3 (GDE3) [5], [21], which was then used to solve the test problems. GDE3 is an extension of differential evolution (DE) [22] for constrained multi-objective optimization. Basically, the evolutionary part of the algorithm is DE and the multi-objective part is from the NSGA-II. This combination has been shown to give benefit over the NSGA-II with rotated problems [23] and some practical problems [24].

When the ranking-dominance relation was applied, non-dominated sorting was not needed any more. Also, no diversity maintenance technique was used in convergence tests. Therefore, after each generation, the combined parent and child population was sorted and reduced just based on ranking-dominance values. Population size of 200 was used and evolution was continued for 1000 generations. The higher number of objectives made the problems harder. Therefore, the population size and number of generations were greater than usually used with the DTLZ problems.

Results from repetition tests are shown in Fig. 9. In comparison, an early version of GDE [25] was used. This version does not have any kind of diversity maintenance and the selection is based on just the Pareto-dominance relation. Also, GDE3 with the Pareto-dominance selection and a diversity maintenance technique from [19] suited for many-objective optimization was used in comparison. The diversity maintenance technique uses an efficient search method to find the nearest neighbors and crowding information calculated from the two nearest neighbors (2-NN). All the experiments were performed using control parameter values $CR = F = 0.2$.

With the DTLZ1–3 problems, it can be observed that ranking-dominance (GDE3, R_{sum} and GDE3, R_{min}) leads to converged solutions even when the number of objectives becomes 50. Convergence of GDE deteriorates steadily. Same observations also hold for DTLZ4 except that also the search using ranking-dominance does not reach the Pareto-front in given generations when the number of objectives is more than 25. Both aggregation functions perform similarly, the min function being a slightly better. An interesting observation is that GDE3 with the diversity maintenance technique (GDE3, 2-NN) stagnates when the number of objectives increases. The reason for this is probably the fact that diversity maintenance becomes a dominating operation when the number of objectives increases, and this prevents progression of the search. Therefore, the curse of dimensionality for the current MOEAs such as GDE3 and the NSGA-II is not just because of Pareto-dominance but also because of the growth of diversity maintenance with respect to the number of objectives.

Results for DTLZ5,6 differ greatly from above. Simple GDE performs best with a large number of objectives and its performance is also relatively good with a few objectives. Ranking-dominance with the min aggregation function performs better than with the sum aggregation function but both perform badly compared to results with DTLZ1-4. The reason is that ranking-dominance relation allows deterioration of individual objective values if other objective values are improved. In the case of DTLZ5,6, solutions get “drifted” away from the Pareto-front when the most of the objective values are slightly improved at the same time while one objective value gets relatively much worse. Figure 10 shows the error value through generations and illustrates this phenomenon. It also shows better performance with the min aggregation function than the sum aggregation function. The phenomenon is of course undesired but hard to avoid because ranking-dominance permits deterioration of individual objectives. A different aggregation function and/or suitable combination of the ranking-dominance and Pareto-dominance relations might alleviate the problem.

When the obtained results with ranking-dominance are examined more carefully, it can be noticed that the solutions

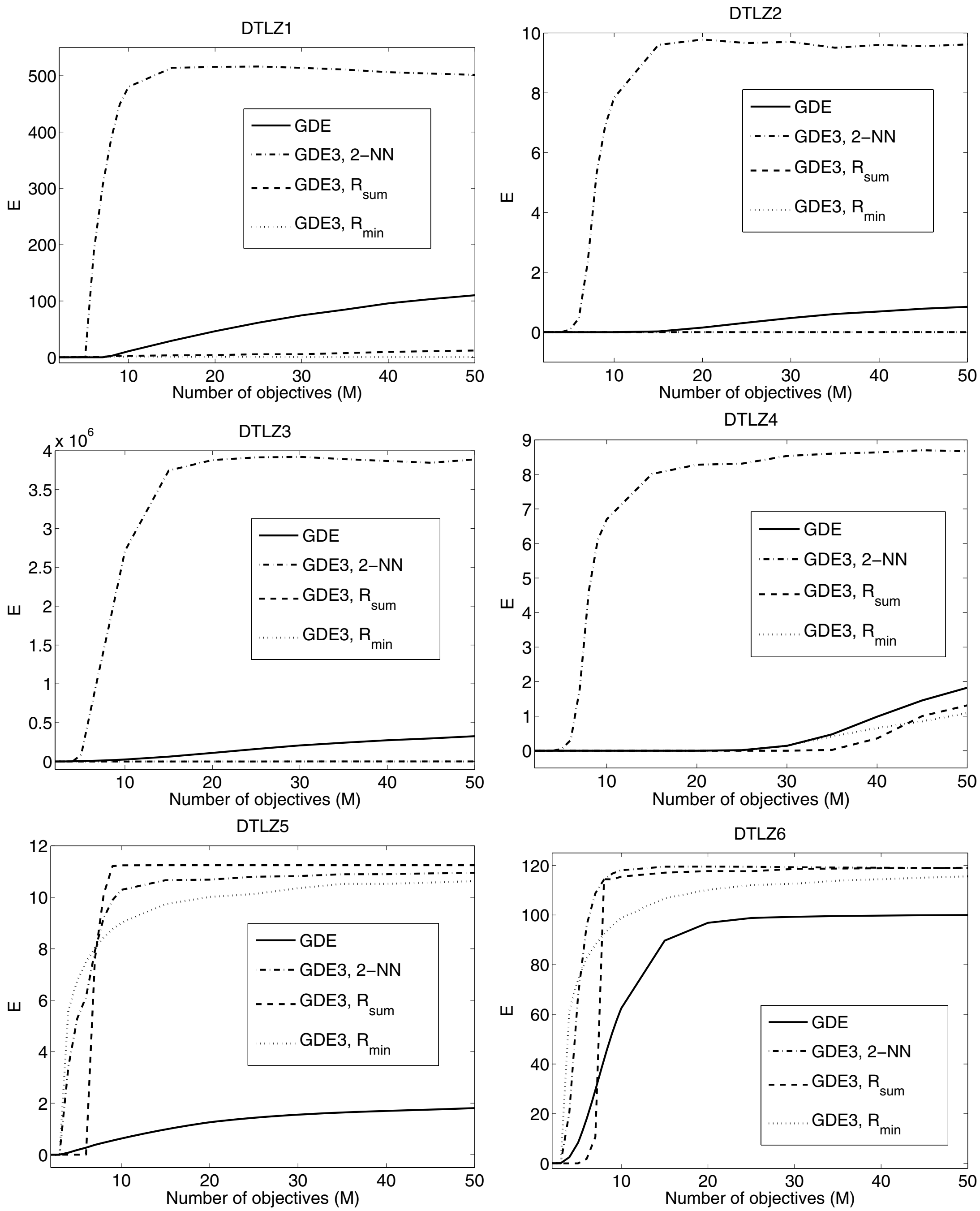

Fig. 9. Median convergence curves from 101 repetitions for the problems solved with four methods: GDE, GDE3 with diversity preservation (2-NN), and GDE3 with ranking-dominance (R_{sum} & R_{min}).

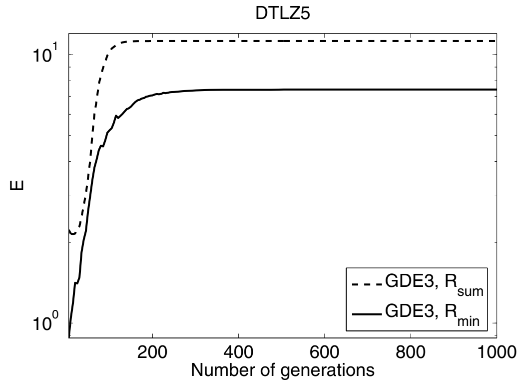

Fig. 10. Error value E through generations for DTLZ5 (median of 101 repetitions).

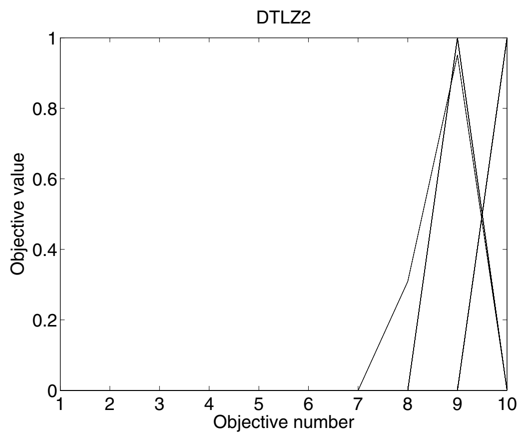

Fig. 11. A value path for 10-objective DTLZ2 solved using ranking-dominance with the min aggregation function.

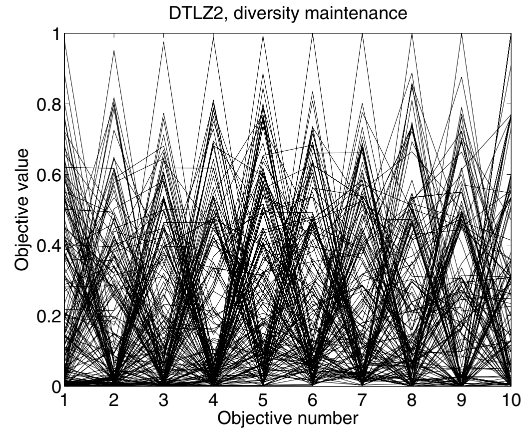

Fig. 12. A value path for 10-objective DTLZ2 solved using ranking-dominance with the min aggregation function and diversity preservation.

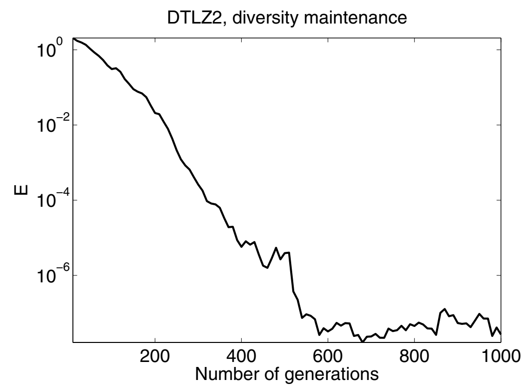

Fig. 13. Error value E through generations for the search leading to the result shown in Fig. 12.

have converged to a few or only a single point, *i.e.*, diversity is lost. Figure 11 shows a value path [2, pp. 316–318] for 10-objective DTLZ2 solved using the R_{min} aggregation function, and it can be observed that all the 200 solutions have converged to three points.

Diversity maintenance can be incorporated in a way described in Section IV. Instead of linear growth, diversity maintenance is increased according to a power-law: if a level 1 means that all the individuals for the next generation are selected based on ranking-dominance and level 0 means that all the individual for the next generation are selected based on diversity, then level at generation G is calculated as $level(G) = \left(\left(G_{max} - G\right)/G_{max}\right)^{0.25}$. The value path for 10-objective DTLZ2 solved using the min aggregation function and the 2-NN diversity preservation technique from [19] is shown in Fig. 12. Now, solutions cover almost full value range for each objective. Figure 13 shows the error value through generations for the search leading to the result shown in Fig. 12. Final value of 2.7×10^{-8} indicates that solutions are very close to the actual Pareto-front.

When the number of objectives increases, reaching the Pareto-front becomes harder and more generations are needed. Also, balancing between convergence and diversity preservation becomes harder but not impossible as the result in Fig. 12 shows. This is an encouraging observation since finding a set of converged and diverse solutions for 10-objective DTLZ2 has been considered to be a very hard task for a modern MOEA [26], [27].

VI. Conclusions

An alternative dominance relation to Pareto-dominance has been studied. The relation is based on ranking a set of solutions according to each separate objective, and an aggregation function is used to calculate a scalar fitness value for each solution. The relation is called as ranking-dominance and it can be used to sort a set of solutions even for a large number of objectives when Pareto-dominance relation cannot distinguish solutions from one another anymore.

Two aggregation functions have been considered: a sum of ranks and a minimum of ranks. The ranking-dominance relation with the first aggregation function is complete with respect to the Pareto-dominance relation. Also, ranking-dominance with this aggregation function is compatible with the Pareto-non-dominance relation. These ensure that use of the ranking-dominance relation with the sum of ranks does not violate Pareto-dominance in the sense that Pareto-dominating solutions are always preferred over Pareto-dominated.

With respect to the ranking-dominance relation a concept of balancing between convergence and diversity has been proposed. The idea is simply to increase diversity maintenance gradually during generations.

Experimental results indicate that in some cases the selection based on ranking-dominance is able to advance the search towards the Pareto-front better than the selection based on Pareto-dominance. However, in some cases it is also possible that the search does not proceed into direction of the Pareto-front because the ranking-dominance relation permits deterioration of individual objectives. The results also show that when the number of objectives increases, the selection based on just Pareto-dominance without diversity maintenance is able to advance the search better than with diversity maintenance. Therefore, diversity maintenance connives at difficulties solving problems with a high number of objectives.

The ranking-dominance relation appears to be a promising alternative to the Pareto-dominance relation. However, more research is needed to improve the ranking-dominance relation and more experiments are needed to evaluate its performance.

REFERENCES

[1] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston: Kluwer Academic Publishers, 1998.

[2] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. Chichester, England: John Wiley & Sons, 2001.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, April 2002.

[4] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Proceedings of the Third Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems (EUROGEN 2001)*, Athens, Greece, Sept 2002, pp. 95–100.

[5] S. Kukkonen and J. Lampinen, "GDE3: The third evolution step of Generalized Differential Evolution," in *Proceedings of the 2005 Congress on Evolutionary Computation (CEC 2005)*, Edinburgh, Scotland, Sept 2005, pp. 443–450. [Online]. Available: http://www.iitk.ac.in/kangal/papers/k2005013.pdf, 6.11.2006

[6] V. Pareto, *Cours D'Economie Politique*. Geneve: Libraire Droz, 1964 (the first edition in 1896).

[7] K. Ikeda, H. Kita, and S. Kobayashi, "Failure of Pareto-based MOEAs: Does non-dominated really mean near to optimal," in *Proceedings of the 2001 Congress on Evolutionary Computation (CEC 2001)*, Seoul, Korea, May 2001, pp. 957–962.

[8] H. Sato, H. E. Aguirre, and K. Tanaka, "Controlling dominance area of solutions and its impact on the performance of MOEAs," in *Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*, Matsushima, Japan, March 2007, pp. 5–20.

[9] M. Farina and P. Amato, "Fuzzy optimality and evolutionary multiobjective optimization," in *Proceedings of the second International Conference on Evolutionary Multi-Criterion Optimization*, Faro, Portugal, April 2003, pp. 58–72.

[10] M. Köppen, R. Vicente-Garcia, and B. Nickolay, "Fuzzy-Pareto-dominance and its application in evolutionary multi-objective optimization," in *Proceedings of the third International Conference on Evolutionary Multi-Criterion Optimization*, Guanajuato, Mexico, March 2005, pp. 399–412.

[11] E. J. Hughes, "Multiple single objective Pareto sampling," in *Proceedings of the 2003 Congress on Evolutionary Computation (CEC 2003)*, Canberra, Australia, Dec 2003, pp. 2678–2684.

[12] A. Sülflow, N. Drechsler, and R. Drechsler, "Robust multi-objective optimization in high dimensional spaces," in *Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*, Matsushima, Japan, March 2007, pp. 715–726.

[13] M. Köppen and K. Yoshida, "Substitute distance assignments in NSGA-II for handling many-objective optimization problems," in *Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*, Matsushima, Japan, March 2007, pp. 727–741.

[14] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evolutionary Computation*, vol. 3, no. 1, pp. 1–16, 1995.

[15] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, April 2003.

[16] M. T. Jensen, "Reducing the run-time complexity of multiobjective EAs: The NSGA-II and other algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 5, pp. 503–515, Oct 2003.

[17] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms–part I: A unified formulation," *IEEE Transactions on Systems, Man, and Cybernetics–Part A: Systems and Humans*, vol. 28, no. 1, pp. 26–37, Jan 1998.

[18] J. Lampinen, "A constraint handling approach for the Differential Evolution algorithm," in *Proceedings of the 2002 Congress on Evolutionary Computation (CEC 2002)*, Honolulu, Hawaii, May 2002, pp. 1468–1473.

[19] S. Kukkonen and K. Deb, "A fast and effective method for pruning of non-dominated solutions in many-objective problems," in *Proceedings of the 9th International Conference on Parallel Problem Solving from Nature (PPSN IX),*, Reykjavik, Iceland, Sept 2006, pp. 553–562.

[20] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, *Evolutionary Multiobjective Optimization*. London: Springer-Verlag, 2005, ch. Scalable Test Problems for Evolutionary Multiobjective Optimization, pp. 105–145. [Online]. Available: http://www.iitk.ac.in/kangal/papers/k2001001.ps.gz, 28.1.2007

[21] S. Kukkonen and K. Deb, "Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems," in *Proceedings of the 2006 Congress on Evolutionary Computation (CEC 2006)*, Vancouver, BC, Canada, July 2006, pp. 3995–4002.

[22] K. V. Price, R. Storn, and J. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Berlin: Springer-Verlag, 2005.

[23] A. Iorio and X. Li, "Solving rotated multi-objective optimization problems using Differential Evolution," in *Proceedings of the 17th Australian Joint Conference on Artificial Intelligence (AI 2004)*, Cairns, Australia, Dec 2004, pp. 861–872.

[24] M. D. Johnston, "Multi-objective scheduling for NASA'a future deep space network array," in *Proceedings of the 5th International Workshop on Planning and Scheduling for Space (IWPSS 2006)*, Baltimore, MD, USA, Sept 2006, pp. 27–35.

[25] J. Lampinen, "DE's selection rule for multiobjective optimization," Lappeenranta University of Technology, Department of Information Technology, Tech. Rep., 2001. [Online]. Available: http://www.it.lut.fi/kurssit/03-04/010778000/MODE.pdf, 15.2.2006

[26] V. R. Khare, X. Yao, and K. Deb, "Performance scaling of multi-objective evolutionary algorithms," in *Proceedings of the second International Conference on Evolutionary Multi-Criterion Optimization*, Faro, Portugal, April 2003, pp. 376–390.

[27] K. Deb and D. K. Saxena, "Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems," in *Proceedings of the 2006 Congress on Evolutionary Computation (CEC 2006)*, Vancouver, BC, Canada, July 2006, pp. 3353–3360.