Improved Pruning of Non-Dominated Solutions Based on Crowding Distance for Bi-Objective Optimization Problems

Saku Kukkonen and **Kalyanmoy Deb**
Kanpur Genetic Algorithms Laboratory (KanGAL)
Indian Institute of Technology Kanpur
Kanpur, PIN 208 016, India
saku.kukkonen@lut.fi, {saku, deb}@iitk.ac.in

KanGAL Report Number 2006007

Abstract

In this paper an algorithm for pruning a set of non-dominated solutions is proposed. The algorithm is based on the crowding distance calculation used in the elitist non-dominated sorting genetic algorithm (NSGA-II). The time complexity class of the new algorithm is estimated and in most cases it is the same as for the original pruning algorithm. Numerical results also support this estimate.

For used bi-objective test problems, the proposed pruning algorithm is demonstrated to provide better distribution compared to the original pruning algorithm of NSGA-II. However, with tri-objective test problems there is no improvement and this study reveals that crowding distance does not estimate crowdedness well in this case and presumably also in cases of more objectives.

1 Introduction

Pruning a set of non-dominated solutions is a common task for multi-objective evolutionary algorithms (MOEAs) such as the strength Pareto evolutionary algorithm (SPEA2) [21] and the elitist non-dominated sorting genetic algorithm (NSGA-II) [5]. An idea is to prune a non-dominated set to have desired number of solutions in such a way that remaining solutions would have as good diversity as possible, *i.e.*, the spread of extreme solutions is as high as possible and the relative distance between solutions is as equal as possible. Probably the best way to obtain a good distribution would be to use some clustering algorithm. However, this is computationally expensive since clustering algorithms take usually time $O\left(MN^2\right)$ to prune a set of size N with M objectives [9]. In NSGA-II the pruning of non-dominated solutions is done in time $O\left(MN\log N\right)$ based on *crowding distance (*CD*)*. However, this method often gives non-optimal distribution as it is demonstrated later in this paper. Therefore, an improved pruning based on *CD* is proposed here.

Besides NSGA-II, *CD* has been used as a distribution maintenance method in many other MOEAs [1, 10–13, 16–18, 20, 22]. There also exist studies where the original *CD* has been slightly modified to use, *e.g.*, different kind of normalization techniques for objective values [14], polar coordinates in the objective space [19], problem specific measures [3], and a different distance metric for crowdedness calculation [7].

This paper continues with the following parts: In Section 2 the original pruning algorithm of NSGA-II is described and its drawback in some cases is demonstrated. Section 3 describes the proposed pruning algorithm and its time complexity is analyzed in Section 4. The proposed method is tested in Section 5 and observations are discussed in Section 6. Finally, conclusions are given in Section 7.

2 Pruning of Non-Dominated Solutions in NSGA-II

At the end of one generation of NSGA-II, the size of the population is twice bigger than the original size. This bigger population is pruned based on the non-dominated sorting [4, pp. 33–44], [9] and *CD*. *CD* for a member of a non-dominated set tries to approximate the perimeter of a cuboid formed by using the nearest neighbors of the member. The cuboid of a non-dominated set member i in the case of two objectives is illustrated in Figure 1 (objectives are minimized in this paper). For a member of non-dominated set, *CD* is calculated by finding distance between two nearest solutions on either side of the member along each of the objectives. These distances are normalized dividing them by the difference between maximum and minimum values of corresponding objectives, and then these normalized distances are summed up giving a *CD* for the corresponding member. For those members of the non-dominated set, which have maximum or minimum value for any objective, *CD* is assigned to have an infinite value. Finally, the members of the non-dominated set are sorted in monotonically decreasing order according to *CD*s and a desired number of members having the largest *CD* values are selected.

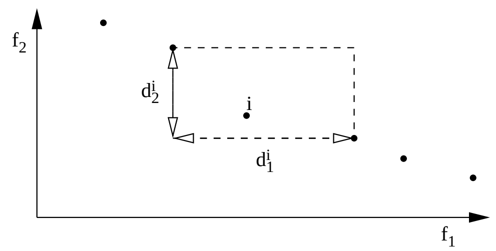

Figure 1: An example of the cuboid of a solution i in the case of two objectives.

For selecting n members out of N based on *CD*, members are first sorted according to objective values taking time $O(N \log N)$ for each of M objective. Then *CD*s are calculated taking time $O(MN)$. Finally, members are sorted according to the *CD* values taking time $O(N \log N)$. The overall computation time of the algorithm is dominated by the first sorting of members according to the objective values for each objective taking time $O(MN \log N)$.

Although the idea of this algorithm is reasonable, it does not provide good result in all cases. In Figure 2 a non-dominated set of 11 members and the six selected members according to *CD*s calculated for the members are presented in the case of two objectives. As it can be seen, leaving out the rest five members having the smallest *CD*s leaves a gap in the resulting set and it is clear that this set of solutions is not optimal in the sense of distribution. There are also other cases where pruning based on *CD* does not provide good distribution [7].

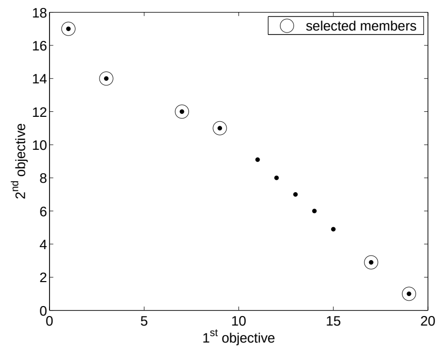

Figure 2: A set of 11 non-dominated members from which six are selected based on the pruning method used in NSGA-II.

3 Proposed Algorithm

The proposed algorithm first calculates *CD*s for the members of a non-dominated set. Instead of selecting n members having the largest *CD* values, $N - n$ members having the smallest *CD* values are removed one by one, updating the *CD* values for the remaining members of the set after each removal. The efficient implementation of this algorithm needs an implementation of a priority queue such as a heap [2, pp. 140–152]. The proposed algorithm is:

PRUNING OF NON-DOMINATED SET

input: a non-dominated set $\mathcal{F}$,
    the size n of a desired pruned set
output: elements of a heap H

calculate *CD* for each member of the set $\mathcal{F}$
create a data structure D containing information about neighbors on either side of the members of $\mathcal{F}$ along each objectives
create an ascending heap H from the members of $\mathcal{F}$ using *CD*s as ordering keys
while $|H| > n$
    remove an element with a minimum *CD* value from H and update H
    update D to have correct neighbor information for the neighbors of the removed element
    for all the neighbors of the removed element
        calculate a new *CD*

9 replace old *CD* value in H with the new one and update H

The output of this algorithm is demonstrated for the same non-dominated set as with the original pruning algorithm. Figure 3 shows the output, which is better distributed than in Figure 2. The proposed method would, presumably, also improve the performance of other methods based on *CD*, *e.g.*, those mentioned in Section 1.

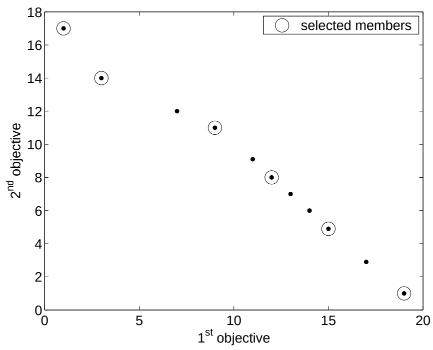

Figure 3: A set of 11 non-dominated members from which six are selected based on the proposed pruning algorithm.

4 Complexity Analysis of the Algorithm

The first operation in line 1 can be done in time $O(MN \log N)$ as in the original algorithm. Creating data structure D containing the neighborhood information along each of the objectives in line 2 can be done in time $O(MN)$. The creation of a heap takes time $O(N)$. The while-loop in lines 4–9 is executed at most N times, on an average $N/2$ times. Removing a minimum element from the heap and updating the heap to have correct structure in line 5 takes time $O(\log N)$. Updating D in line 6 takes time $O(M)$ since the removed element has at most $2M$ neighbors along the objectives. Calculating new *CD* for a neighbor element with the help of D in line 8 takes time $O(M)$ since there are at most $2M$ objective values to be taken for calculations. Updating *CD* value to the heap and updating the heap to have correct structure in line 9 takes time $O(\log N)$. Therefore, the computation time for the whole for-loop in lines 7–9 is bounded by $O(M \log N)$ or $O\left(M^2\right)$, whichever is greater. The while-loop in lines 4–9 is bounded by $O(MN \log N)$ or $O\left(M^2 N\right)$, whichever is greater. These are also overall complexities for the whole algorithm. In many cases, where $M < \log N$, the former complexity dominates. This leads to complexity class $O(MN \log N)$ for the algorithm. Anyway, for large N the proposed algorithm is faster than the typical clustering methods except in some rare cases when $M > N$.

5 Experiments

The original and proposed pruning algorithms were implemented in the Generalized Differential Evolution 3 (GDE3) [11], which was then used to solve test problems. GDE3 is an extension of Differential Evolution (DE) [15] for constrained multi-objective optimization. Roughly speaking, the evolutionary part of the algorithm is DE and the multi-objective part is from NSGA-II [5]. This kind of combination has been shown to give benefit over NSGA-II with rotated problems [8]. Further, GDE3 has some other improvements over NSGA-II, and an interested reader is advised to look reference [11].

In repetition tests, the diversity of obtained result was measured using the spacing (S) metric [4, pp. 327–328] and variance of *CD*s. The spacing metric measures the standard deviation of distances (according to Manhattan, *i.e.*, L_1 distance metric) from each vector to nearest vector in the obtained non-dominated set. A smaller value of S is better and for an ideal distribution $S = 0$. Also, the CPU time needed for pruning and for the entire process was registered. The hardware used was a PC with 2.6 GHz Pentium 4 CPU & 512 MB RAM, and the operating system was Linux.

5.1 Bi-Objective Problems

The pruning algorithms were used to solve the bi-objective test problems ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 [4, pp. 356–360], using population size 100 and 1000 generations. Control parameter values used in GDE3 were a crossover control parameter $CR = 0.2$ and a mutation factor $F = 0.2$ for all the problems except ZDT4. For ZDT4 the control parameters values used were $CR = 0.0$ and $F = 0.5$.

Sets of solutions to these problems are shown in Figures 4–8, where the solutions generated by using the proposed pruning method have been sifted by -0.05

units along both objectives to alleviate observation. As it can be seen, the solutions obtained with the proposed pruning method have better distribution.

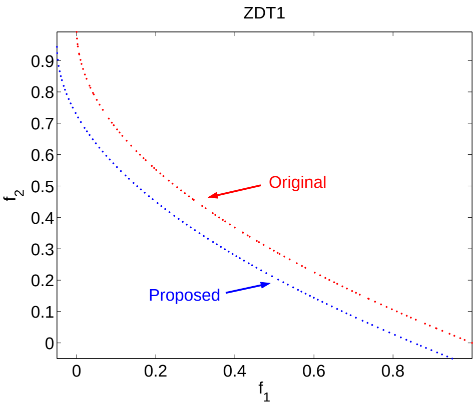

Figure 4: A result for ZDT1 using the original and proposed pruning methods (the solutions obtained by the proposed method are deliberately moved towards the origin for clarity).

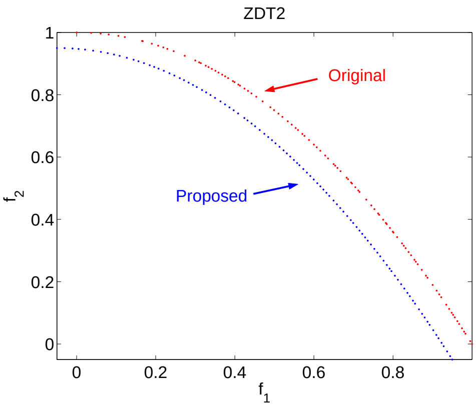

Figure 5: A result for ZDT2 using the original and proposed pruning methods (the solutions obtained by the proposed method are deliberately moved towards the origin for clarity).

Solving the bi-objective optimization problems was repeated 100 times. The numerical results, *i.e.*, mean and standard deviation values are reported in Table 1. The spacing metric suggests that the results obtained with the proposed pruning method are almost three times better compared to those with the original method in terms of diversity, while the total required CPU time

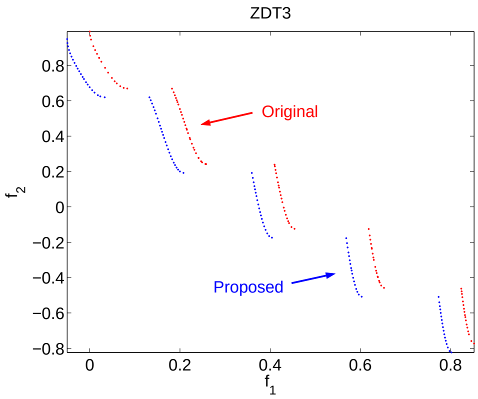

Figure 6: A result for ZDT3 using the original and proposed pruning methods (the solutions obtained by the proposed method are deliberately moved towards the origin for clarity).

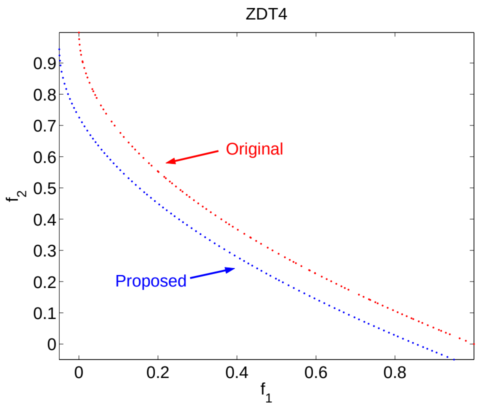

Figure 7: A result for ZDT4 using the original and proposed pruning methods (the solutions obtained by the proposed method are deliberately moved towards the origin for clarity).

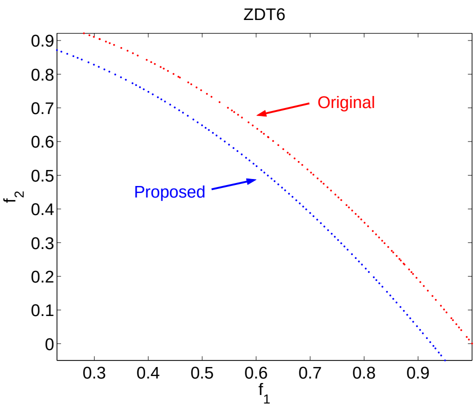

Figure 8: A result for ZDT6 using the original and proposed pruning methods (the solutions obtained by the proposed method are deliberately moved towards the origin for clarity).

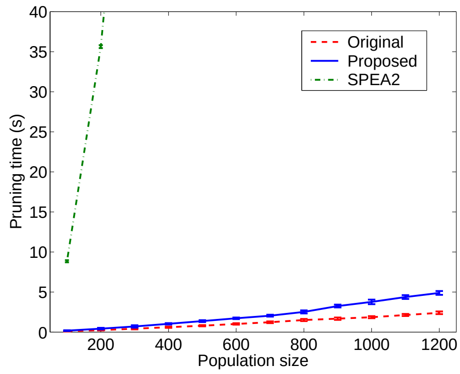

Figure 9: Mean pruning times (standard deviations as error bars) measured from 100 runs for solving ZDT1 problem with different population sizes.

has increased less than 10%. Table 1 also contains corresponding results for SPEA2[1]. Control parameter values used with SPEA2 were $\eta_c = 15$ controlling a SBX crossover operator, $\eta_m = 20$ controlling polynomial mutation, $1/n$ for a variable mutation probability (n is the number of decision variables in a problem), 0.5 for a variable swap probability, 0.5 for variable recombination probability, 1.0 for an individual mutation probability, and 1.0 for an individual recombination probability. For ZDT2 the variable recombination probability was 1.0.

Interestingly, it can be noticed that according to measures, the diversity obtained with SPEA2 is between the original and proposed pruning methods. The time complexity of the pruning methods was verified experimentally using the ZDT1 problem. The measured pruning times for various population sizes are shown in Figure 9. It can be observed that the proposed pruning method takes about twice the time of the original method and the complexity classes of these methods are the same, whereas the pruning time in SPEA2 is much more and increases drastically when the population size increases.

5.2 Tri-Objective Problems

The proposed pruning method was also tested with a set of tri-objective test problems, DTLZ1, DTLZ2, DTLZ4, DTLZ5, and DTLZ7 [6], using population size 300 and 1000 generations. Control parameter values used were same as for the most of the ZDT problems. The numerical results from 100 repetition runs are shown in Table 2. From these results it was noticed that there was no significant improvement in the values of spacing although the results with the proposed pruning method were better (except for DTLZ7) in term of variance of *CD*s. SPEA2 provided the best distribution according to the spacing measure but worst according to the variance of *CD*s. Moreover, there was no significant difference between the original and proposed methods when results were evaluated visually, whereas SPEA2 provided much better results (Figures 10–15). The only exception was DTLZ5, where the Pareto-front is curved in the objective space; in this case the proposed pruning method outperformed the original method and SPEA2. Bad observed performance of the proposed pruning method was unexpected and led to closer analysis of *CD* in the case of three objectives.

In Figures 16 and 17 sets of non-dominated solutions to be pruned are shown for the DTLZ1 and DTLZ2 problems. Both Figures also show three randomly selected solutions with their neighbors along each objective. It can be noticed that these neighbors do not necessarily locate near by corresponding solutions and therefore the calculated *CD*s do not properly estimate the crowdedness of solutions. Presumably, the same observation is extendable for cases with more than three objectives. *CD* gives relatively good estimation about crowdedness in the bi-objectives cases because the nondominance property (which implies a monotonic rela-

[1]SPEA2 implementation was taken from the PISA web site: http://www.tik.ee.ethz.ch/pisa/

Table 1: Mean and standard deviation values of spacing, variance of *CD*s, and CPU times for ZDT test problems measured from 100 independent runs

Problem	Method	Spacing	$\sigma^2(CD)$	Total time (s)	Pruning time (s)
ZDT1	Original	$6.4240e-03 \pm 5.5946e-04$	$1.2755e-04 \pm 2.5046e-05$	$\mathbf{1.4877e+00 \pm 1.3246e-02}$	$\mathbf{1.0530e-01 \pm 3.1669e-02}$
	Proposed	$\mathbf{2.5348e-03 \pm 2.6001e-04}$	$\mathbf{2.1163e-05 \pm 3.6732e-06}$	$1.5940e+00 \pm 1.5699e-02$	$1.8720e-01 \pm 4.1368e-02$
	SPEA2	$3.2063e-03 \pm 2.9122e-04$	$4.2059e-05 \pm 8.0237e-06$	$1.1756e+01 \pm 7.0249e-02$	$7.6469e+00 \pm 1.9474e-01$
ZDT2	Original	$6.3904e-03 \pm 5.4882e-04$	$1.3523e-04 \pm 2.8336e-05$	$\mathbf{1.4718e+00 \pm 6.8048e-03}$	$\mathbf{1.0521e-01 \pm 3.1254e-02}$
	Proposed	$\mathbf{2.5981e-03 \pm 2.8669e-04}$	$\mathbf{2.1138e-05 \pm 3.7108e-06}$	$1.5747e+00 \pm 6.6290e-03$	$1.8227e-01 \pm 3.5693e-02$
	SPEA2	$3.4023e-03 \pm 3.6818e-04$	$4.9467e-05 \pm 1.0613e-05$	$1.2291e+01 \pm 9.7998e-02$	$8.1818e+00 \pm 2.2531e-01$
ZDT3	Original	$4.3573e-03 \pm 4.1398e-04$	$2.0291e-03 \pm 5.1891e-05$	$\mathbf{1.3838e+00 \pm 5.4643e-03}$	$\mathbf{8.5400e-02 \pm 2.9074e-02}$
	Proposed	$\mathbf{1.5039e-03 \pm 1.8103e-04}$	$\mathbf{2.0169e-03 \pm 2.0486e-05}$	$1.4421e+00 \pm 6.7112e-03$	$1.2820e-01 \pm 3.1667e-02$
	SPEA2	$3.0892e-03 \pm 3.4177e-04$	$2.0962e-03 \pm 3.0930e-05$	$1.1500e+01 \pm 7.2117e-02$	$7.3613e+00 \pm 2.2574e-01$
ZDT4	Original	$6.0789e-03 \pm 5.5549e-04$	$1.0398e-04 \pm 2.1932e-05$	$\mathbf{9.5180e-01 \pm 5.7525e-03}$	$\mathbf{8.6200e-02 \pm 2.6772e-02}$
	Proposed	$\mathbf{2.1344e-03 \pm 3.1260e-04}$	$\mathbf{1.4066e-05 \pm 3.6445e-06}$	$9.8360e-01 \pm 6.8931e-03$	$1.2440e-01 \pm 2.6980e-02$
	SPEA2	$2.8862e-03 \pm 2.9266e-04$	$3.2085e-05 \pm 6.8718e-06$	$1.0639e+01 \pm 9.3916e-02$	$6.3440e+00 \pm 2.8208e-01$
ZDT6	Original	$6.6047e-03 \pm 6.5536e-04$	$1.3420e-04 \pm 2.8494e-05$	$\mathbf{1.2147e+00 \pm 5.7753e-03}$	$\mathbf{9.6869e-02 \pm 3.4866e-02}$
	Proposed	$\mathbf{2.6662e-03 \pm 2.6590e-04}$	$\mathbf{2.2287e-05 \pm 3.5135e-06}$	$1.3088e+00 \pm 6.7090e-03$	$1.7740e-01 \pm 4.2035e-02$
	SPEA2	$3.1010e-03 \pm 3.0937e-04$	$3.8489e-05 \pm 8.6676e-06$	$1.1662e+01 \pm 5.7654e-02$	$7.5451e+00 \pm 2.0404e-01$

Table 2: Mean and standard deviation values of spacing, variance of *CD*s, and CPU times for DTLZ test problems measured from 100 independent runs

Problem	Method	Spacing	$\sigma^2(CD)$	Total time (s)	Pruning time (s)
DTLZ1	Random	$3.0980e-02 \pm 3.7708e-03$	$2.8078e-04 \pm 1.3032e-04$	$1.4591e+01 \pm 1.5585e-01$	$6.4080e-01 \pm 8.4539e-02$
	Original	$2.3927e-02 \pm 1.1410e-03$	$2.5804e-05 \pm 2.8357e-06$	$\mathbf{1.3648e+01 \pm 1.1168e-01}$	$\mathbf{6.2460e-01 \pm 6.3444e-02}$
	Proposed	$2.3549e-02 \pm 1.0009e-03$	$\mathbf{9.7780e-06 \pm 9.0294e-07}$	$1.4872e+01 \pm 9.3362e-02$	$1.3115e+00 \pm 8.0634e-02$
	SPEA2	$\mathbf{1.1661e-02 \pm 1.1200e-02}$	$3.7368e-04 \pm 8.7105e-04$	$1.2211e+02 \pm 1.2640e-01$	$7.6797e+01 \pm 1.7521e-01$
DTLZ2	Random	$3.8560e-02 \pm 3.5166e-03$	$2.1920e-04 \pm 8.9905e-05$	$1.4465e+01 \pm 1.3946e-01$	$6.7530e-01 \pm 7.5324e-02$
	Original	$2.8914e-02 \pm 1.3827e-03$	$2.1855e-05 \pm 1.9725e-06$	$\mathbf{1.3536e+01 \pm 3.3132e-02}$	$\mathbf{6.4540e-01 \pm 7.4947e-02}$
	Proposed	$2.8014e-02 \pm 1.5111e-03$	$\mathbf{1.0638e-05 \pm 8.4747e-07}$	$1.4831e+01 \pm 4.8988e-02$	$1.3456e+00 \pm 1.4674e-01$
	SPEA2	$\mathbf{1.2919e-02 \pm 6.9397e-04}$	$8.1446e-05 \pm 1.0243e-05$	$1.3095e+02 \pm 7.7438e-02$	$8.7535e+01 \pm 1.5309e-01$
DTLZ4	Random	$4.6069e-02 \pm 1.0088e-02$	$1.7806e-03 \pm 4.7662e-04$	$\mathbf{1.3135e+01 \pm 4.1494e-01}$	$\mathbf{6.3520e-01 \pm 7.2995e-02}$
	Original	$2.8452e-02 \pm 1.2664e-03$	$2.2104e-05 \pm 2.0978e-06$	$1.3813e+01 \pm 5.8934e-02$	$6.4060e-01 \pm 6.9614e-02$
	Proposed	$2.7879e-02 \pm 1.3367e-03$	$\mathbf{1.0864e-05 \pm 7.9148e-07}$	$1.5101e+01 \pm 6.2203e-02$	$1.3591e+00 \pm 1.2607e-01$
	SPEA2	$\mathbf{1.3022e-02 \pm 7.2771e-04}$	$9.2581e-05 \pm 1.1494e-05$	$1.3124e+02 \pm 1.0552e-01$	$8.7665e+01 \pm 1.7151e-01$
DTLZ5	Random	$7.1387e-03 \pm 2.6948e-03$	$5.1590e-04 \pm 2.9150e-04$	$1.3403e+01 \pm 1.3308e-01$	$6.2730e-01 \pm 7.1941e-02$
	Original	$3.5364e-03 \pm 1.9787e-04$	$3.8692e-05 \pm 4.6751e-06$	$\mathbf{1.2149e+01 \pm 8.9120e-02}$	$\mathbf{5.7600e-01 \pm 6.6999e-02}$
	Proposed	$\mathbf{1.6077e-03 \pm 1.5687e-04}$	$\mathbf{5.8746e-06 \pm 9.1744e-07}$	$1.3575e+01 \pm 7.2202e-02$	$1.1437e+00 \pm 7.8130e-02$
	SPEA2	$2.3039e-03 \pm 1.1166e-04$	$2.4633e-05 \pm 2.5317e-06$	$1.2363e+02 \pm 8.3786e-02$	$7.7715e+01 \pm 1.2489e-01$
DTLZ7	Random	$4.1365e-02 \pm 1.8823e-02$	$4.6935e-03 \pm 1.3886e-03$	$1.3011e+01 \pm 1.3106e-01$	$6.3580e-01 \pm 7.4633e-02$
	Original	$2.3658e-02 \pm 2.6283e-03$	$\mathbf{1.5495e-03 \pm 2.1474e-04}$	$\mathbf{1.2066e+01 \pm 2.2497e-02}$	$\mathbf{5.5920e-01 \pm 6.4834e-02}$
	Proposed	$2.3932e-02 \pm 2.3022e-03$	$1.5639e-03 \pm 1.9189e-04$	$1.3029e+01 \pm 4.1459e-02$	$1.0860e+00 \pm 8.3545e-02$
	SPEA2	$\mathbf{1.5140e-02 \pm 8.2394e-04}$	$1.9248e-03 \pm 1.0965e-04$	$1.3342e+02 \pm 6.7649e-02$	$8.8697e+01 \pm 1.0649e-01$

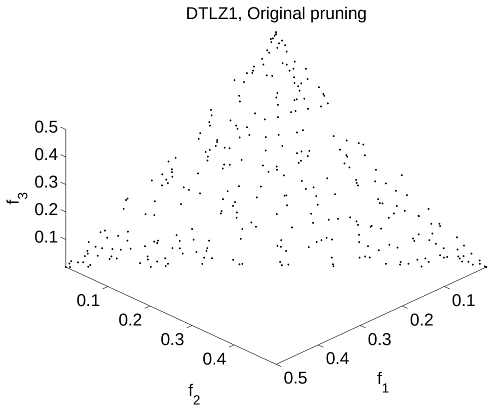

Figure 10: A set of solutions for DTLZ1 using the original pruning method.

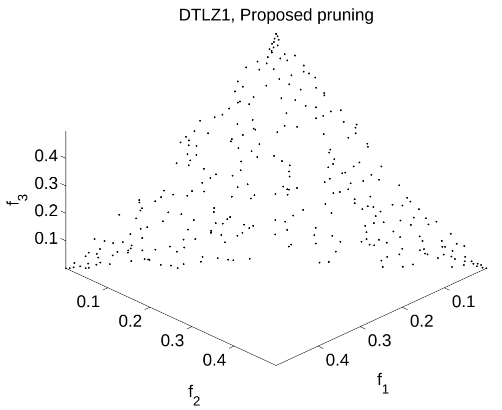

Figure 11: A set of solutions for DTLZ1 using the proposed pruning method.

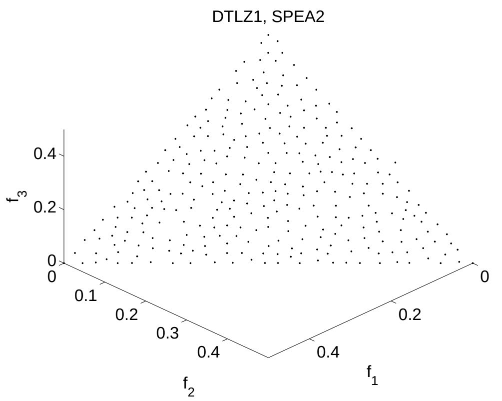

Figure 12: A set of solutions for DTLZ1 using SPEA2.

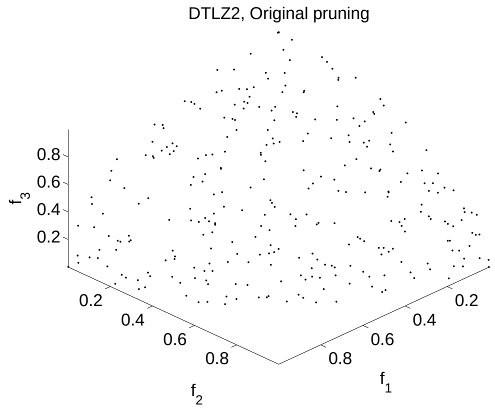

Figure 13: A set of solutions for DTLZ2 using the original pruning method.

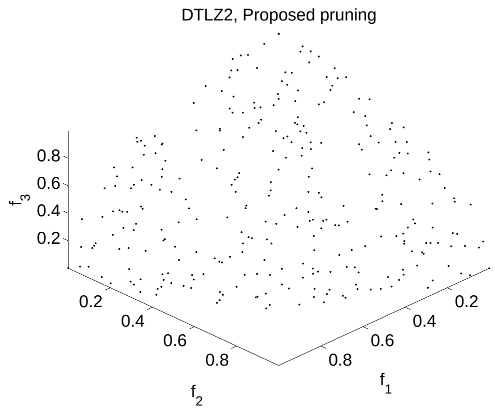

Figure 14: A set of solutions for DTLZ2 using the proposed pruning method.

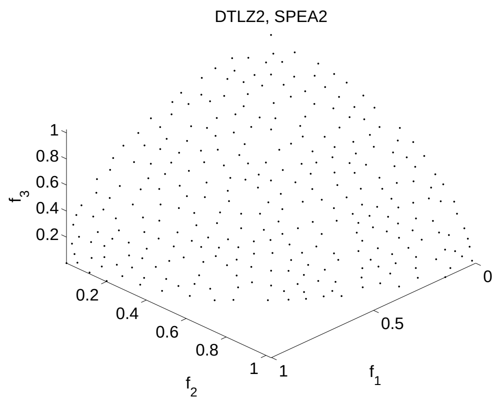

Figure 15: A set of solutions for DTLZ2 using SPEA2.

tion between solutions in the objective space) causes solutions to come close together along both the objectives. With three or more objectives this does not need to hold at all, even though it holds with DTLZ5 where the non-dominating solutions form a curve instead of a surface.

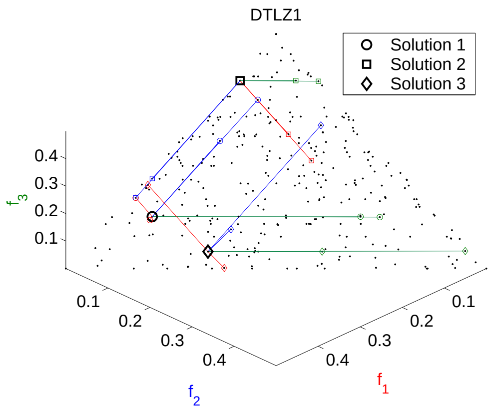

Figure 16: A set of solutions to be pruned for DTLZ1 and three solutions with their neighbors according to individual objectives.

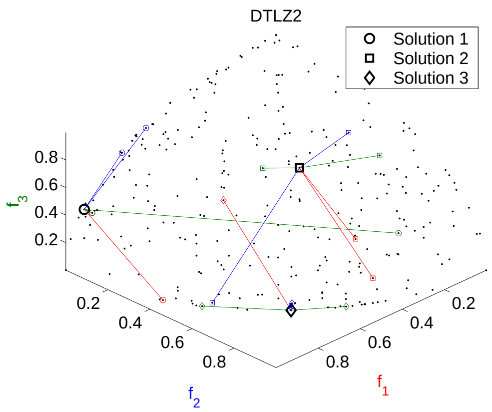

Figure 17: A set of solutions to be pruned for DTLZ2 and three solutions with their neighbors according to individual objectives.

For comparison, the original pruning method was also implemented in such a way that random values for *CD*s were used instead of calculated values. For each non-dominated set member having maximum or minimum value for any objective, *CD* was assigned an infinite value. The numerical results for DTLZ problems using this random pruning approach are also shown in Table 2. According to the results, the random pruning performs notably worse than the pruning method based on the calculated *CD* values. Therefore *CD* contains some knowledge about crowdedness although the estimation is far from perfect based on Figures 16 and 17.

The proposed method did not improve diversity in the DTLZ7 problem according to the variance of *CD*s, and in the ZDT3 problem the improvement according to *CD* was modest. The probable reason for this is the fact that the Pareto-front for these problems is discontinuous in contrast to the other problems. Visually observed, the diversity was also improved for ZDT3.

6 Discussion

According to observations, the diversity handling method of NSGA-II should be modified if a good diversity is desired for problems having more than two objectives. Worse obtained diversity with *CD* than with a diversity maintenance technique based on clustering, has already been observed earlier in the case of optimization problems with three objectives [6] but the reason for the bad performance has not been studied or demonstrated earlier according to authors' knowledge. These observations mean that also other methods (*e.g.*, those mentioned in Section 1) applying *CD* or its modification and using neighbors along each objectives to estimate crowdedness, do not provide good distribution when the number of objectives is larger than two. Use of conventional clustering techniques lead to high time complexity such as in SPEA2.

This leads to conclusion that a new pruning method should be developed, particularly for many (> 2) objective problems. One way to do this would be to use an efficient nearest neighbor method to compute a distance metric with a few (probably two) nearest solutions (in the Euclidean sense) of every member of a non-dominated set in the normalized objective space. Thereafter, solutions having a larger distance metric can be preferred to maintain diversity in the non-dominated set. The extreme solutions can be preserved in the same manner as in the pruning algorithms based on *CD*, and the pruning technique should take care of updating crowdedness values for remaining members of the non-dominated set after each extraction of the most crowded member. Other ideas are also possible, but this paper has indicated the need for such studies, particularly for large number of objectives.

Better pruning method is not only likely to provide a better distribution, but may also help in a faster convergence because better distribution helps to observe the characteristics of the objective function space better.

7 Conclusions

In this paper an algorithm improving pruning of non-dominated set of solutions is proposed. The algorithm is based on crowding distance and it is demonstrated to provide better distribution for the pruned set in bi-objective cases than the original algorithm used in NSGA-II.

Time complexity for the new algorithm is estimated and it is bounded by $O(MN \log N)$ or $O(M^2 N)$, whichever is greater. In practice N is often much larger than M leading to the same complexity class $O(MN \log N)$ as the original pruning algorithm.

The original and proposed pruning methods were implemented and their time complexity classes were observed to be the same. The methods were compared using test problems and diversity metrics. According to the results, the improvement for the obtained diversity is considerable in bi-objective cases. However, there is no improvement in most tri-objective cases because the crowding distance metric does not estimate crowdedness properly in these cases.

The development of an efficient pruning method for problems with more than two objectives will be a subject of future research.

Acknowledgments

Authors wish to thank Dr. M. Laumanns and Mr. S. Bleuler for their help with SPEA2. First author gratefully acknowledges support from the East Finland Graduate School in Computer Science and Engineering (ECSE), Technological Foundation of Finland, Finnish Cultural Foundation, and Centre for International Mobility (CIMO). This work was done during the visit of first author to KanGAL under the Indo-Finnish exchange student programme. First author also wishes to thank KanGAL researchers for their help and fruitful discussions.

References

[1] H. E. Aguirre and K. Tanaka. Selection, drift, recombination, and mutation in multiobjective evolutionary algorithms on scalable mnk-landscapes. In *Proceedings of the 3rd International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, pages 355–369, Guanajuato, Mexico, March 2005.

[2] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. Prentice-Hall, 1990.

[3] B. de la Iglesia, A. Reynolds, and V. J. Rayward-Smith. Developments on a multi-objective metaheuristic (MOMH) algorithm for finding interesting sets of classification rules. In *Proceedings of the 3rd International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, pages 826–840, Guanajuato, Mexico, March 2005.

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

[5] K. Deb, A. Pratap, S. 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] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. *Evolutionary Multiobjective Optimization*, chapter Scalable Test Problems for Evolutionary Multiobjective Optimization, pages 105–145. Springer-Verlag, London, 2005.

[7] N. Hallam, P. Blanchfield, and G. Kendall. Handling diversity in evolutionary multiobjective optimisation. In *Proceedings of the 2005 Congress on Evolutionary Computation (CEC 2005)*, pages 2233–2240, Edinburgh, Scotland, Sept 2005.

[8] A. Inorio 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)*, pages 861–872, Cairns, Australia, Dec 2004.

[9] M. T. Jensen. Reducing the run-time complexity of multiobjective EAs: The NSGA-II and other algorithms. *IEEE Transactions on Evolutionary Computation*, 7(5):503–515, Oct 2003.

[10] S. Kukkonen and J. Lampinen. An extension of Generalized Differential Evolution for multi-objective optimization with constraints. In *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII)*, pages 752–761, Birmingham, England, Sept 2004.

[11] 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)*, pages 443–450, Edinburgh, Scotland, Sept 2005.

[12] N. K. Madavan. Multiobjective optimization using a Pareto Differential Evolution approach. In *Proceedings of the 2002 Congress on Evolutionary Computation (CEC 2002)*, pages 1145–1150, Honolulu, Hawaii, May 2002.

[13] K. Maneeratana, K. Boonlong, and N. Chaiyaratana. Multi-objective optimisation by co-operative co-evolution. In *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII)*, pages 772–781, Birmingham, England, Sept 2004.

[14] G. K. M. Pedersen and D. E. Goldberg. Dynamic uniform scaling for multiobjective genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation (GECCO 2004)*, pages 11–23, Seattle, WA, USA, June 2004.

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

[16] C. R. Raquel and P. C. Naval Jr. An effective use of crowding distance in multiobjective particle swarm optimization. In *Proceedings of the Genetic and Evolutionary Computation (GECCO 2005)*, pages 257–264, Washington, DC, USA, June 2005.

[17] T. Robič and B. Filipič. DEMO: Differential Evolution for multiobjective optimization. In *Proceedings of the 3rd International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, pages 520–533, Guanajuato, Mexico, March 2005.

[18] K. Sastry, M. Pelikan, and D. E. Goldberg. Limits of scalability of multiobjective estimation of distribution algorithms. In *Proceedings of the 2005 Congress on Evolutionary Computation (CEC 2005)*, pages 2217–2224, Edinburgh, Scotland, Sept 2005.

[19] H. Sato, H. Aguirre, and K. Tanaka. On the locality of dominance and recombination in multiobjective evolutionary algorithms. In *Proceedings of the 2005 Congress on Evolutionary Computation (CEC 2005)*, pages 451–466, Edinburgh, Scotland, Sept 2005.

[20] M. R. Sierra and C. A. Coello Coello. Improving pso-based multi-objective optimization using crowding, mutation and e-dominance. In *Proceedings of the 3rd International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, pages 505–519, Guanajuato, Mexico, March 2005.

[21] 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)*, pages 95–100, Athens, Greece, Sept 2002.

[22] X. Zou, M. Liu, L. Kang, and J. He. A high performance multi-objective evolutionary algorithm based on the principles of thermodynamics. In *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII)*, pages 922–931, Birmingham, England, Sept 2004.