Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Multicriteria Decision Making (MCDM 2007)

Variants of Differential Evolution for Multi-Objective Optimization

Karin Zielinski and Rainer Laur

Abstract— **In multi-objective optimization not only fast convergence is important, but it is also necessary to keep enough diversity so that the whole Pareto-optimal front can be found. In this work four variants of Differential Evolution are examined that differ in the selection scheme and in the assignment of crowding distance. The assumption is checked that the variants differ in convergence speed and amount of diversity. The performance is shown for 1000 consecutive generations, so that different behavior over time can be detected.**

I. Introduction

In contrast to single-objective optimization that tries to determine one global best solution for an optimization problem, generally several trade-off solutions are generated in multi-objective optimization. These solutions should fulfil the following goals: They should be as close to the Pareto-optimal front as possible and furthermore they should have a good diversity, meaning that the extent of non-dominated solutions is as large as possible and the distribution of non-dominated solutions is as uniformly as possible. In this work four variants of a multi-objective Differential Evolution (DE) algorithm are examined that differ in the selection scheme and in the calculation of crowding distance. The variants are expected to result in different convergence speed and in different amounts of diversity. In former work the variants were tested using only one performance measure besides visual examination, and furthermore results were checked for only two generation numbers. In this work the performance of the DE variants is evaluated in dependence on the number of generations to allow a more complete assessment of their behavior, based on several well known bi-objective test functions. More informative performance measures were applied to make a more thorough discussion possible.

This work is organized as follows: Section II gives an overview about Differential Evolution, including a description of the here used variants. The experimental settings are explained in Section III. Results are given in Section IV and conclusions are presented in Section V.

II. Differential Evolution

DE was developed in 1995 for unconstrained single-objective optimization [1]. Because of the demands of real-world problems that often include constraints and multiple objectives, several enhancements have been done by various researchers, expanding DE for constrained optimization of single-objective [2], [3], [4] and multi-objective problems [5], [6], [7]. In the following first the basic single-objective algorithm is described. Afterwards, the four variants for handling multi-objective optimization problems are specified.

A. Basic Single-Objective Differential Evolution

Using DE the population members are randomly initialized in the beginning of an optimization run. The individuals are evolved by employing the operators mutation, recombination and selection which are described in the following. In this work variant DE/rand/1/bin [8] is used that employs three randomly chosen population members for mutation:

$$\vec{v}_i = \vec{x}_{r_1} + F \cdot (\vec{x}_{r_2} - \vec{x}_{r_3}) \tag{1}$$

The individuals are real-valued vectors with dimension D that equals the number of objective function parameters. The factor F is a user-defined control parameter that is usually chosen from the interval $F \in [0, 1]$. The indices r_1, r_2, r_3 denote three mutually different population members that are also unequal to the currently regarded individual $\vec{x}_i$. Mutation is conducted for every individual of the population ($i \in \{0, NP-1\}$ where NP is the population size that has to be chosen by the user).

Each mutated vector $\vec{v}_i$ is recombined with the corresponding target vector $\vec{x}_i$ to build the trial vector $\vec{u}_i$:

$$u_{i,j} = \begin{cases} v_{i,j} & \text{if } rand_j \leq CR \text{ or } j = k \\ x_{i,j} & \text{otherwise} \end{cases} \tag{2}$$

CR is the third control parameter of DE. It is chosen from the interval $CR \in [0, 1]$. Based on the comparison with a random variable $rand_j$ from the same interval, it is decided from which vector the trial vector inherits its components. It is ensured that the trial vector differs from the target vector in at least one component by randomly determining $k \in \{0, D–1\}$ for every population member in each generation.

In unconstrained single-objective optimization, vectors for the subsequent generation $G + 1$ are selected based on their objective function value. Each trial vector is compared with the respective target vector and the one with the lower objective function value (for minimization problems as in this work) is chosen for the next generation.

B. Multi-Objective Differential Evolution

Mutation and recombination can be conducted in the same way for single-objective and multi-objective optimization. However, the selection process has to be modified for multi-objective optimization because now multiple objectives exist. For this purpose the dominance relation can be used. A vector $\vec{x}$ dominates a vector $\vec{y}$ if

$$\forall i \in \{0, \ldots, D-1\} : x_i \leq y_i \tag{3}$$

$$\exists i \in \{0, \ldots, D-1\} : x_i < y_i \tag{4}$$

The dominance relation can also be extended for constrained problems [9]. However, in this work only unconstrained multi-objective test problems are used.

Karin Zielinski and Rainer Laur are with the Institute for Electromagnetic Theory and Microelectronics (ITEM), University of Bremen, Germany (email: {zielinski,rlaur}@item.uni-bremen.de).

1-4244-0702-8/07/$20.00 ©2007 IEEE

1) Selection Procedures: When designing a multi-objective optimization algorithm on the basis of DE, and the advantages of NSGA-II[1] [9] should be exploited, several techniques can be adopted like fast non-dominated sorting and diversity preservation based on crowding distance. However, a characteristic of single-objective DE is the direct comparison of target vector and trial vector during the selection procedure that is not present in NSGA-II. The question arises which technique is better: Adding all newly generated vectors to the population as it is done in NSGA-II or keeping the direct comparison between target vector and trial vector and only adding vectors if target and trial vector are non-dominated. Therefore, both methods are compared to each other in this work. Variant 1 is similar to the original DE selection scheme, meaning that the trial vector $\vec{u}_i$ instantly replaces the corresponding target vector $\vec{x}_i$ if $\vec{u}_i$ dominates $\vec{x}_i$. The trial vector is discarded if $\vec{x}_i$ dominates $\vec{u}_i$. If the vectors are non-dominated, the trial vector is added to the population for later sorting. Variant 2 is closer to NSGA-II by abandoning the direct comparison of trial and target vectors and instead adding all trial vectors to the population.

DE is an elitist algorithm because in a comparison of two vectors always the one with the better properties wins. However, elitism is even more pronounced in the second variant compared to the first variant. This is due to the fact that the direct comparison between target vectors and trial vectors allows discarding of individuals which are relatively good in comparison to the whole population but worse than the competitor. In contrast, only the best NP individuals are kept using variant 2. Therefore, it is assumed that the second variant will converge faster, but the first variant will be more successful for complex problems due to a higher diversity.

2) Assignment of Crowding Distance: Using variant 2, the population size equals $2 \cdot NP$ after applying the evolutionary operators. Depending on the number of non-dominated target and trial vectors, the population size will be in $\{NP, 2 \cdot NP\}$ for variant 1. The population size is set back to NP by a procedure based on non-domination and crowding distance. Crowding distance is a measure for the closeness of individuals. For the outermost solutions of the non-dominated front it is set to infinity because the extent of fronts should be as large as possible. For other vectors it is calculated as follows:

$$dist_{\vec{x}_i} = \sum_{j=0}^{m-1} \frac{(f_{j,i+1} - f_{j,i-1})}{(f_{j,max} - f_{j,min})} \tag{5}$$

where f_j corresponds to the jth objective function, m equals the number of objective functions and the population is sorted according to the mth objective function value, respectively.

In order to set the population size back to NP, first a fast non-dominated sorting is conducted as in NSGA-II [9]. The population is ordered into non-dominated fronts $\mathcal{F}_0 \dots \mathcal{F}_l$, so that members of one front are non-dominated by each other and $\mathcal{F}_0$ contains the best non-dominated solutions. The next generation is built by subsequently adding fronts beginning at $\mathcal{F}_0$. Generally, there will be a front $\mathcal{F}_a$ with $a \in \{0, l\}$ that cannot be added completely to the next generation without exceeding NP. Therefore, individuals from $\mathcal{F}_a$ featuring the largest crowding distances are chosen for the next generation.

For the calculation of crowding distances often only members of front $\mathcal{F}_a$ are regarded (this is also done in variant A here). This approach may be unfavorable if an individual is far away from members of its own front but very close to an individual of another front $\mathcal{F}_i$ with $0 \leq i < a$, so as a result this individual is considered to contribute much to diversity using variant A when in fact it does not. Furthermore, the outermost individuals of front $\mathcal{F}_a$ are always kept using variant A regardless of how near other solutions are because of their crowding distance of infinity that they get in order to facilitate a large extent of the front. However, there may be solutions in other fronts that already provide an extent of equal or even larger size, so it may not be necessary to keep the outermost solutions of front $\mathcal{F}_a$. Therefore, an algorithm variant B is examined that also includes individuals of other fronts $\mathcal{F}_i$ with $0 \leq i \leq a$ during the calculation of crowding distances for individuals from front $\mathcal{F}_a$.

If only one front exists, variants A and B are equal. In other cases, variant B is expected to result in a higher diversity. In [10] an example is given that shows a situation for which variant B avoids gaps in the distribution of individuals that would have been caused by using variant A. The effect is assumed to be especially pronounced in the beginning of an optimization run because there will be many fronts. In [11] it is stated that it may be disadvantageous if algorithms first try to reach the Pareto-optimal front and only afterwards work on finding a good spread of solutions. Using algorithm variant B, this unfavorable behavior is expected to be prevented.

III. Experimental Settings

For the multi-objective DE algorithm two different selection schemes and two different ways of calculating crowding distances were introduced. In the following these properties are combined to form four algorithm variants that will be evaluated in this work (no comparison with NSGA-II is done here because the competitiveness of multi-objective DE with NSGA-II was already shown elsewhere, see e.g. [7]):

- 1A: Trial vectors and corresponding target vectors are compared directly (original DE selection scheme), calculation of crowding distance like in NSGA-II.
- 1B: Trial vectors and corresponding target vectors are compared directly (original DE selection scheme), adapted calculation of crowding distance.
- 2A: All trial vectors are added to the population, calculation of crowding distance like in NSGA-II.
- 2B: All trial vectors are added to the population, adapted calculation of crowding distance.

In former work [10] the four algorithm variants were examined using test functions CONSTR, SRN and TNK from [9]. However, these functions are considered to be rather easy to optimize, so there was the need to do examinations with more difficult problems. Furthermore, besides visual

[1] Non-dominated Sorting Genetic Algorithm II

comparisons only the spacing metric was used to evaluate the performance of the algorithm variants in [10]. As this performance measure is able to make statements about the distribution of solutions but not about convergence properties, other performance measures are also used in this work. Moreover, performance measures are not only shown for some selected generations but continuously over time.

Because algorithm variants should be compared, the set coverage metric $\mathcal{C}(A, B)$ [12] is used as performance measure here. This metric determines the percentage of members of a set B that are dominated by members of set A:

$$\mathcal{C}(A, B) = \frac{|b \in B | \exists a \in A : a \preceq b|}{|B|} \tag{6}$$

Generally $\mathcal{C}(A, B) \neq 1 - \mathcal{C}(B, A)$, so both $\mathcal{C}(A, B)$ and $\mathcal{C}(B, A)$ have to be considered. The set coverage metric results in values $\mathcal{C}(A, B) \in [0, 1]$ where $\mathcal{C}(A, B) = 1$ means that all vectors of B are dominated by vectors of A or they are equal to each other, and $\mathcal{C}(A, B) = 0$ means that no vectors of B are dominated by or equal to vectors of A.

Hypervolume is a commonly used performance measure that evaluates both the closeness to the Pareto-optimal front as well as diversity [13]. The volume of a hypercube limited by the obtained non-dominated solutions and a reference point is calculated in objective space. For the reference point usually a vector is built whose components are equal or worse than the worst objective function values that were found. A better performance is indicated by a larger hypervolume. Because objective functions can have a different range of values, it is ensured that no objective function is favored by normalizing all values using the minimum and maximum objective function values that were found in all optimization runs. In this work all values are scaled to the interval [0,1] and the reference point is set to (1.1, 1.1).

The spacing metric as described in [13] was also checked but due to lack of space the results are not shown here. The same holds for the maximum spread metric (also specified in [13]). However, none of these metrics revealed significant differences between the algorithm variants.

Some of the most widely used multi-objective test functions are T_1, T_2, T_3, T_4 and T_6 [14], so they are used as basis for this examination. The first three functions have a dimension of $D = 30$ whereas the dimension of the latter two functions is $D = 10$, respectively. All optimization problems are bi-objective. 100 independent runs are conducted for every algorithm variant and each optimization problem where the initial population is equal for all four variants, respectively. For the control parameters standard settings of $F = 0.7$, $CR = 0.9$ and $NP = 100$ are used and no effort was made to tune the parameters for the examined test problems. Because it can be assumed that the behavior of the algorithm variants differs over time, e.g. convergence may be faster for certain variants, averages of all performance measures are shown over time up to $G_{max} = 1000$ generations, similar as in [15]. Standard deviations of performance measures were also calculated as a function of the generation G, but they are not displayed due to lack of space.

Another possibility to evaluate the performance of multiple runs apart from calculating averages of performance measures is to join the obtained Pareto-optimal solutions of all runs at a specific generation and compute performance measures for the combined Pareto-optimal set [16]. In this case diversity measures like spacing do not make sense, but the set coverage metric and hypervolume can be evaluated.

IV. Results

For functions T_1, T_2, T_3 and T_6 the Pareto-optimal fronts as shown in [14] were successfully generated. In Figs. 1-4 the combined non-dominated solutions of all independent runs are shown for three chosen generations. There are noticeable differences between the algorithm variants regarding the non-dominated solutions after 100 generations, and especially for T_6 still after 200 generations, but after 1000 generations all algorithm variants have converged to the Pareto-optimal front and no differences can be seen any more.

For function T_4 not enough diversity was generated (the population converged to one point in the search space), so the results for T_4 are not shown. One possible reason for this behavior could be unsuitable parameter settings.

In Table I the largest and smallest objective function values are shown that were found during all optimization runs. They were used for the normalization of objective function values before calculating the average hypervolume shown in Figs. 5(a)-8(a). Unfortunately, almost no differences can be seen between the algorithm variants. It is assumed that the scaling may be unsuitable, so for some chosen generations the average hypervolume is also shown using only objective function values of non-dominated solutions of the respective generation for normalization (Figs. 5(b)-8(b)). Apart from the average hypervolume also the hypervolume of the combined non-dominated solutions of all 100 independent runs is shown in Figs. 5(c)-8(c) for some chosen generations. Again, only objective function values of non-dominated solutions of the respective generations are used for the normalization. The hypervolume of combined solutions over time is not shown due to lack of space. However, it looks very similar to the average hypervolume over time. Due to lack of space the maximum and minimum objective function values that were used for the normalization before calculating hypervolume for Figs. 5(b)-8(b) and Figs. 5(c)-8(c) are not shown here. However, it should be noted that for each given generation different values were used for normalization, thus the hypervolume shown in the figures can be smaller for later generations than for earlier generations although in fact the

TABLE I

Maximum and minimum objective function values

Function	$f_{1,min}$	$f_{1,max}$	$f_{2,min}$	$f_{2,max}$
T_1	0	1	0.00430141	6.51019
T_2	0	1	0.00813649	7.00497
T_3	0	0.901791	-0.75995	6.47247
T_6	0.280775	1	0.00998032	9.3748

hypervolume increases over time if the same normalization is used, see Figs. 5(a)-8(a). Although the normalization of solutions was changed, still only small differences can be seen between the algorithm variants when hypervolume is used as a performance measure. For T_6 it is indicated that variant 2B converges faster than the other variants but otherwise no definitive statements can be made. Two different conclusions are possible from these results: Either the hypervolume metric in the form applied here is unsuitable to detect differences for this examination or there are no or only small differences to observe. In the following it is checked if the set coverage metric is able to clarify the matter.

In Figs. 9-12 the average set coverage metric is shown for all combinations of algorithms. Often the differences between algorithm variants are rather small, so it is difficult to draw conclusions from these results. For T_1 1A is better than 1B in a certain range of generations, furthermore 2B better than 2A, 1A better than 2A, and 1B better than 2A (for the latter two combinations the reverse relations hold in earlier generations), but for most of the other functions the algorithm variants show different behavior. The most pronounced differences can be seen for T_6 where 2A and 2B are inferior to 1A and 1B (however, again the first generations show different relations than later stages).

The development of the set coverage metric for the combined solutions of all independent runs (see Figs. 13-16) is less smooth than for the average set coverage metric, and more differences between the algorithm variants can be seen. This is due to the fact that regarding the combined non-dominated solutions means that only the best solutions of 100 independent runs are shown, so runs with bad performance are not included here, but they affect the average set coverage metric. Based on the set coverage metric of combined solutions, variant 2B is mostly better than 2A, especially in early stages (Figs. 13(a)-16(a)). For the comparison of 1A and 1B the results differ for different optimization problems as well as the considered generation (Figs. 13(a)-16(a)), thus it cannot be decided which is better. Furthermore, variant 1B is generally preferable to 2A, and variant 1A is mostly better than 2A (Figs. 13(b)-16(b)). For most functions variant 1A is better than 2B, only for T_3 opposed behavior is shown (Figs. 13(c)-16(c)). The comparison of variants 1B and 2B is mostly inconclusive with the exception of T_6 for which 1B is better than 2B (Figs. 13(c)-16(c)).

V. Conclusions

In this work four variants of multi-objective Differential Evolution have been compared that differ in the selection scheme and the assignment of crowding distance. Instead of showing results of only few discrete generations, the performance of the algorithm variants is continuously monitored over 1000 generations. Taking the set coverage metric of combined solutions as basis, the results of variants 1A and 1B are generally better than the results of 2A and 2B, indicating that it is preferable to use the original DE selection scheme instead of adding all trial vectors to the population. As 2B is generally better than 2A but no conclusive statements can be made regarding variants 1B and 1A, it is not clear if the adapted crowding distance calculation brings advantages. For future work the multi-objective DE could be further improved e.g. by considering methods like the improved pruning in [17] or by addressing lateral diversity issues like in [18].

Unfortunately, the hypervolume metric did not provide conclusive results here. The reasons for this will be further investigated in future work.

References

[1] R. Storn and K. Price, "Differential Evolution - A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces," International Computer Science Institute, Berkeley, Tech. Rep. TR-95-012, 1995.

[2] J. Lampinen and R. Storn, "Differential Evolution," in *New Optimization Techniques in Engineering*, G. C. Onwubolu and B. Babu, Eds. Berlin Heidelberg: Springer-Verlag, 2004, pp. 123–166.

[3] T. Takahama and S. Sakai, "Constrained Optimization by the ϵ Constrained Differential Evolution with Gradient-Based Mutation and Feasible Elites," in *Proceedings of the Congress on Evolutionary Computation*, 2006, pp. 308–315.

[4] R. L. Becerra and C. A. Coello Coello, "Optimization with Constraints using a Cultured Differential Evolution Approach," in *Proceedings of the Conference on Genetic and Evolutionary Computation*, 2005.

[5] K. Parsopoulos, D. Tasoulis, N. Pavlidis, V. Plagianakos, and M. Vrahatis, "Vector Evaluated Differential Evolution for Multiobjective Optimization," in *Proceedings of the IEEE 2004 Congress on Evolutionary Computation*, 2004.

[6] H. A. Abbass, "The Self-Adaptive Pareto Differential Evolution Algorithm," in *Proceedings of the IEEE Congress on Evolutionary Computation*, 2002.

[7] T. Robič and B. Filipič, "DEMO: Differential Evolution for Multiobjective Optimization," in *Proceedings of the Conference on Evolutionary Multiobjective Optimization*, 2005.

[8] K. V. Price, "An Introduction to Differential Evolution," in *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds. London: McGraw-Hill, 1999, pp. 79–108.

[9] K. Deb, A. Pratap, S. Agrawal, and T. Meyarian, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[10] K. Zielinski, D. Peters, and R. Laur, "Constrained Multi-Objective Optimization Using Differential Evolution," in *Proceedings of the Third International Conference on Computational Intelligence, Robotics and Autonomous Systems*, Singapore, 2005.

[11] K. Deb, A. Pratap, and T. Meyarivan, "Constrained Test Problems for Multi-Objective Evolutionary Optimization," in *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, 2001.

[12] E. Zitzler, "Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications," Ph.D. dissertation, Swiss Federal Institute of Technology Zurich, 1999.

[13] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley, 2001.

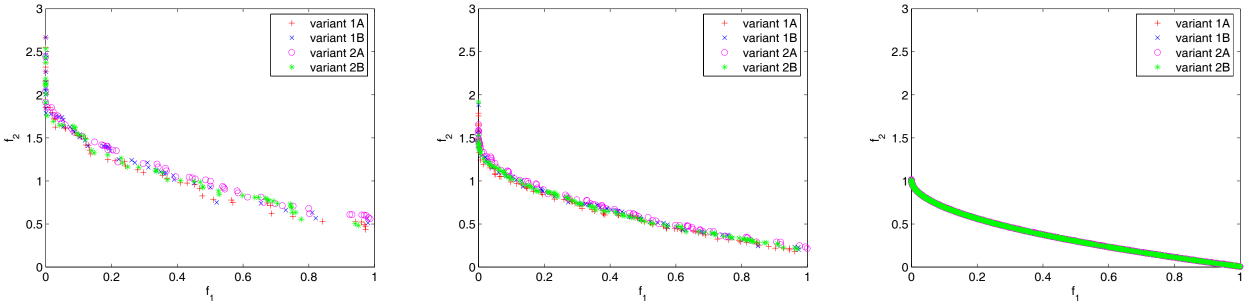
[14] E. Zitzler, K. Deb, and L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.

[15] L. T. Bui, K. Deb, H. A. Abbass, and D. Essam, "Dual Guidance in Evolutionary Multi-objective Optimization By Localization," in *Proceedings of the 6th International Conference on Simulated Evolution and Learning*, 2006.

[16] M. Sato, H. E. Aguirre, and K. Tanaka, "Effects of δ-Similar Elimination and Controlled Elitism in the NSGA-II Multiobjective Evolutionary Algorithm," in *Proceedings of the Congress on Evolutionary Computation*, 2006, pp. 3980–3987.

[17] S. Kukkonen and K. Deb, "Improved Pruning of Non-Dominated Solutions Based on Crowding Distance for Bi-Objective Optimization Problems," in *Proceedings of the IEEE Congress on Evolutionary Computation*, 2006, pp. 3995–4002.

[18] K. Deb and T. Goel, "Controlled Elitist Non-dominated Sorting Genetic Algorithms for Better Convergence," in *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, 2001, pp. 67–81.

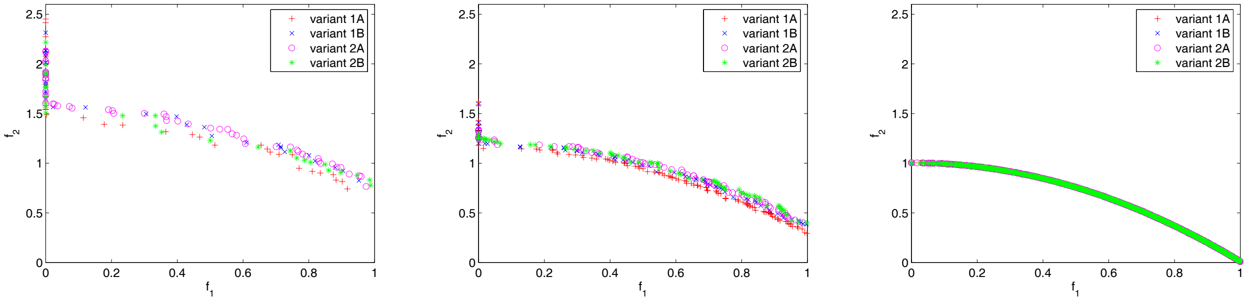


(a) After 100 generations

(b) After 200 generations

(c) After 1000 generations

Fig. 1. Non-dominated solutions for T_1

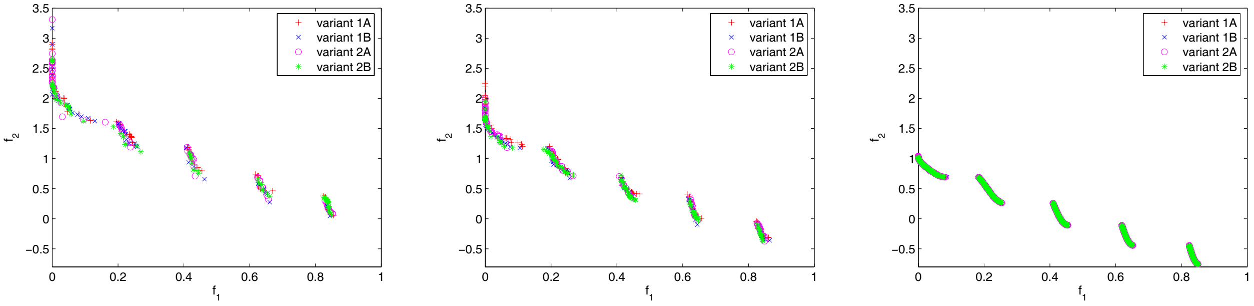


(a) After 100 generations

(b) After 200 generations

(c) After 1000 generations

Fig. 2. Non-dominated solutions for T_2

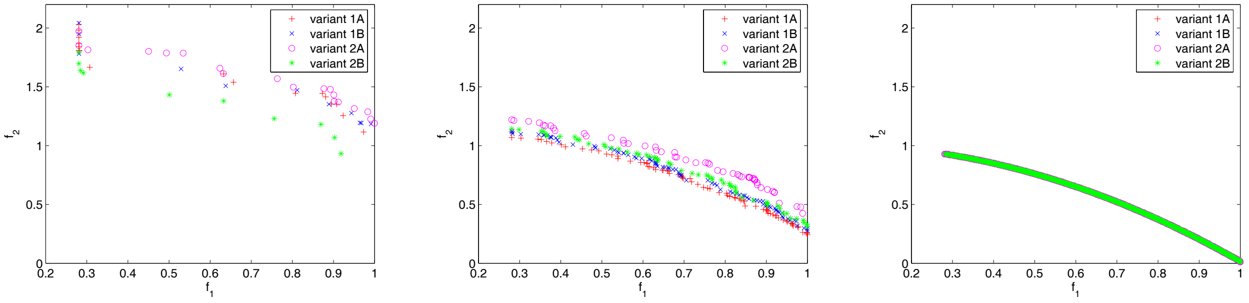


(a) After 100 generations

(b) After 200 generations

(c) After 1000 generations

Fig. 3. Non-dominated solutions for T_3

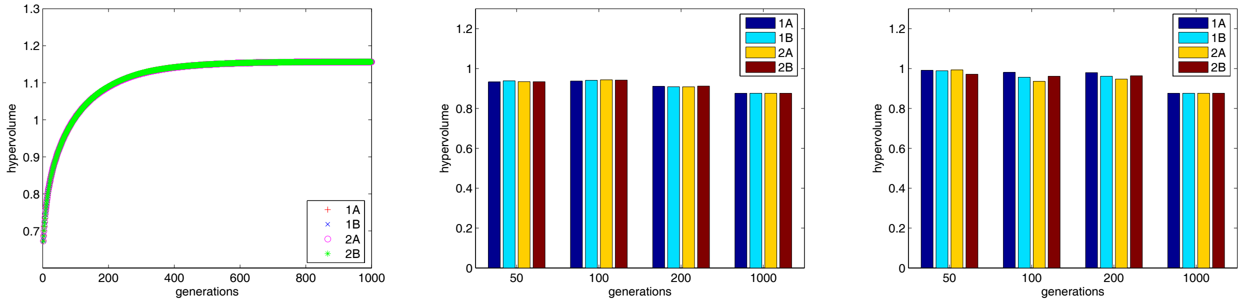


(a) After 100 generations

(b) After 200 generations

(c) After 1000 generations

Fig. 4. Non-dominated solutions for T_6

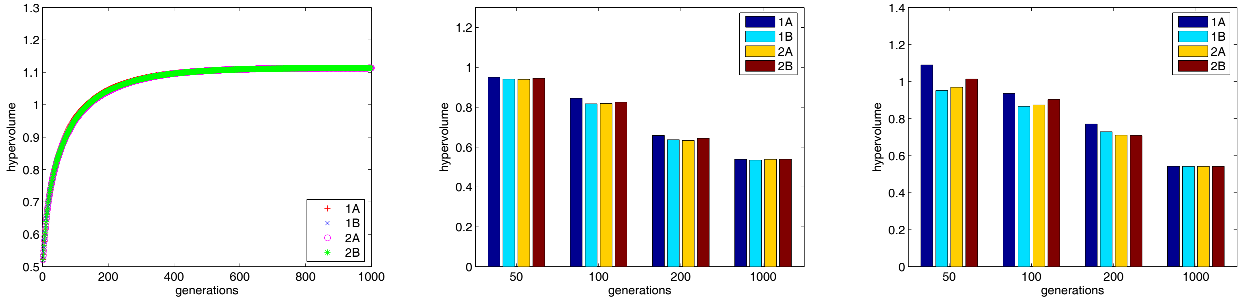


(a) Hypervolume over time

(b) Average hypervolume

(c) Hypervolume of combined solutions

Fig. 5. Hypervolume for T_1

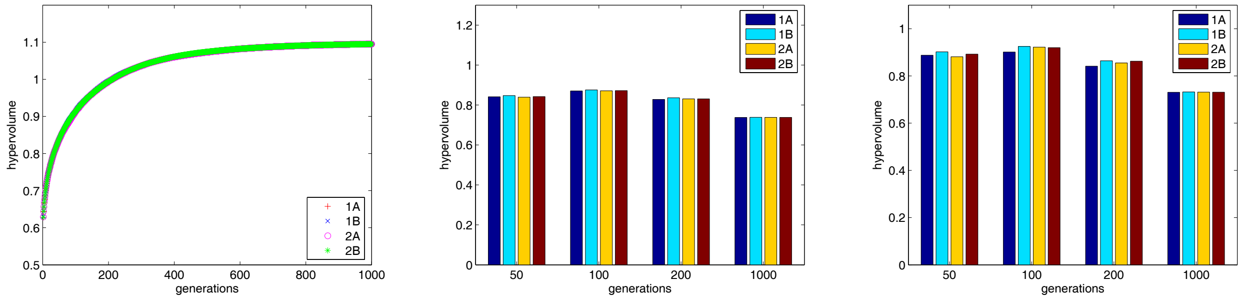


(a) Hypervolume over time

(b) Average hypervolume

(c) Hypervolume of combined solutions

Fig. 6. Hypervolume for T_2

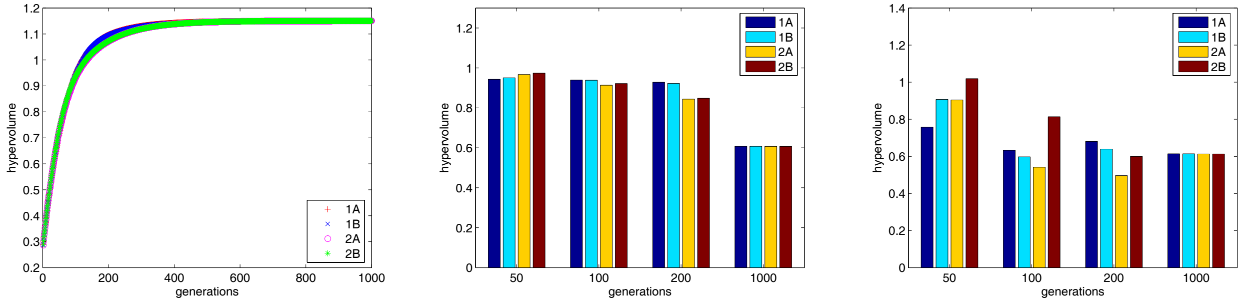


(a) Hypervolume over time

(b) Average hypervolume

(c) Hypervolume of combined solutions

Fig. 7. Hypervolume for T_3

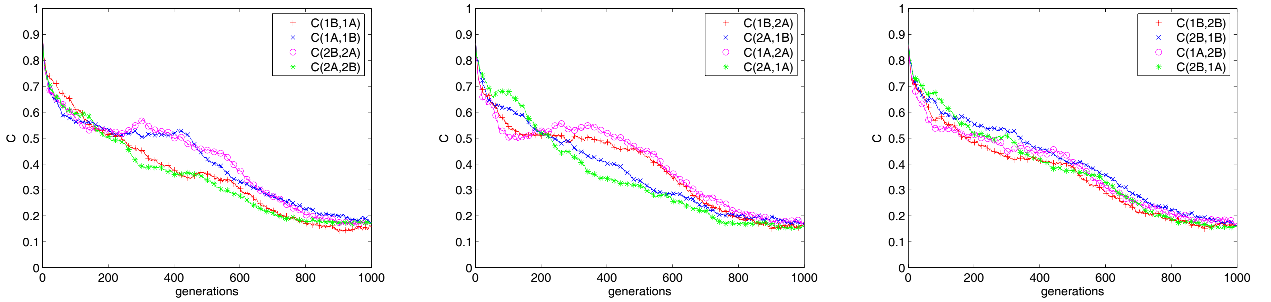


(a) Hypervolume over time

(b) Average hypervolume

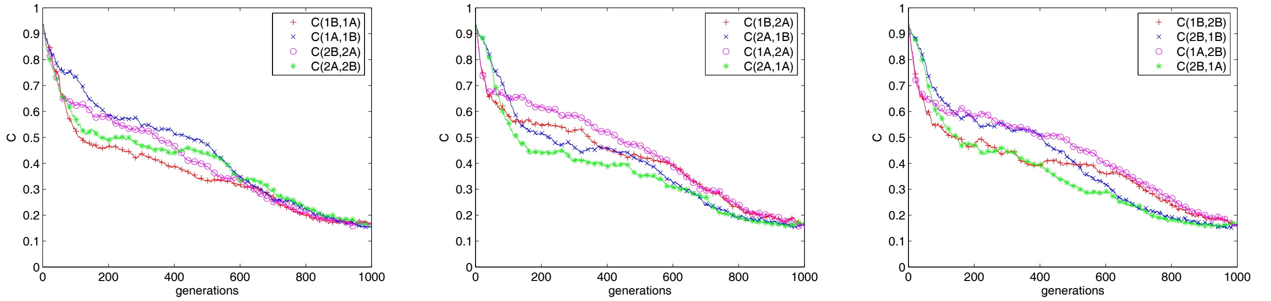
(c) Hypervolume of combined solutions

Fig. 8. Hypervolume for T_6



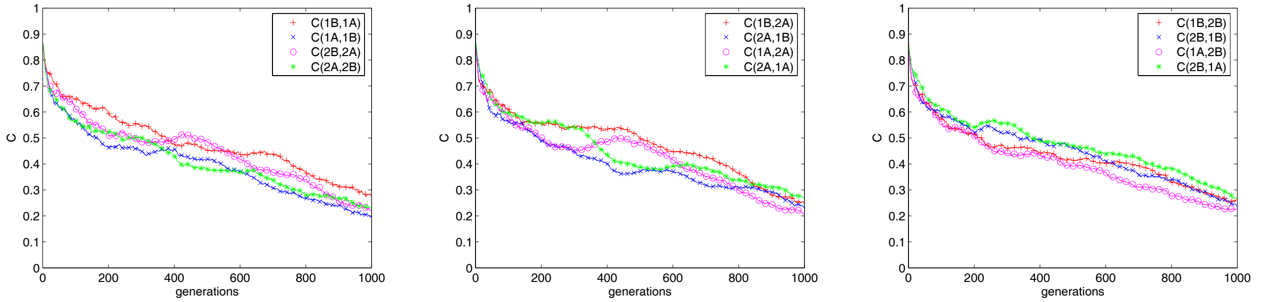
(a) Comparison of 1A and 1B; also 2A and 2B (b) Comparison of 1B and 2A; also 1A and 2A (c) Comparison of 1B and 2B; also 1A and 2B

Fig. 9. Average set coverage metric for T_1



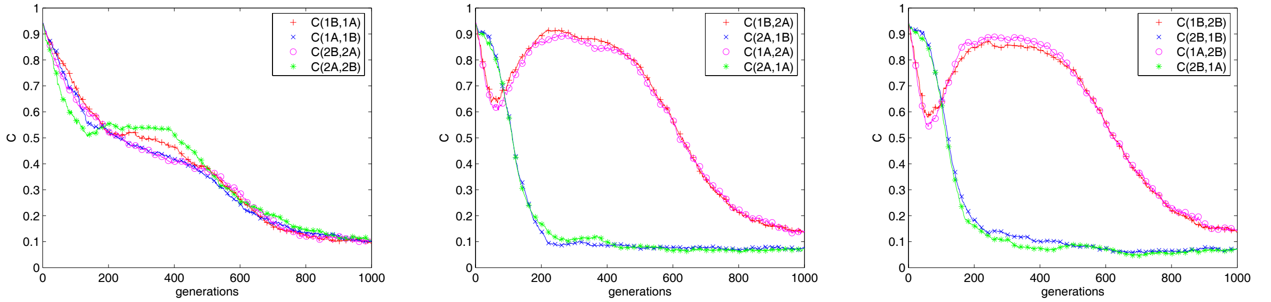
(a) Comparison of 1A and 1B; also 2A and 2B (b) Comparison of 1B and 2A; also 1A and 2A (c) Comparison of 1B and 2B; also 1A and 2B

Fig. 10. Average set coverage metric for T_2



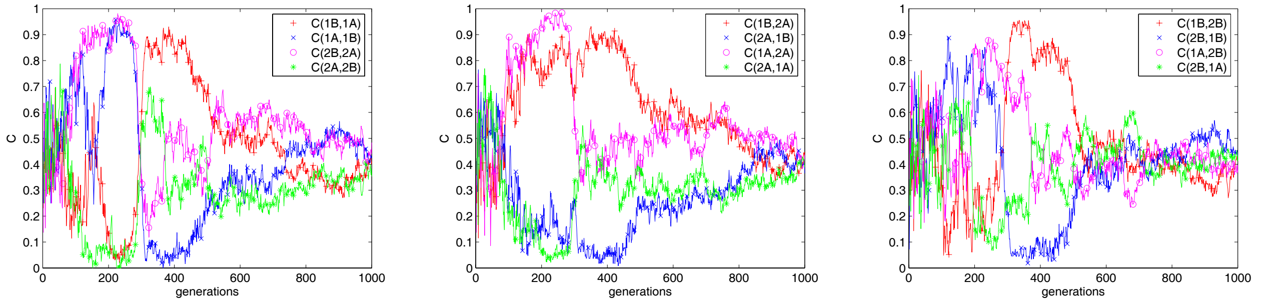
(a) Comparison of 1A and 1B; also 2A and 2B (b) Comparison of 1B and 2A; also 1A and 2A (c) Comparison of 1B and 2B; also 1A and 2B

Fig. 11. Average set coverage metric for T_3



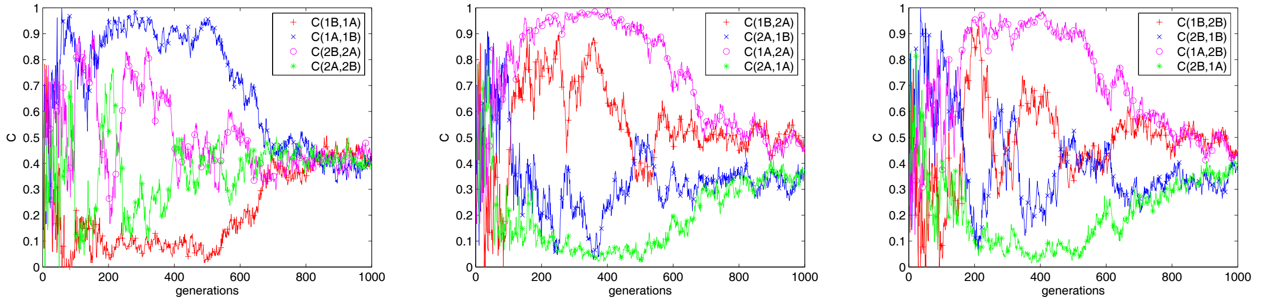
(a) Comparison of 1A and 1B; also 2A and 2B (b) Comparison of 1B and 2A; also 1A and 2A (c) Comparison of 1B and 2B; also 1A and 2B

Fig. 12. Average set coverage metric for T_6



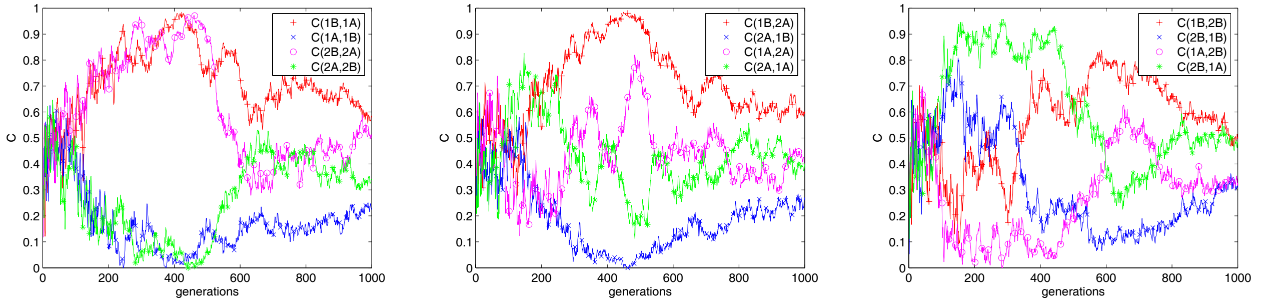
(a) Comparison of 1A and 1B; also 2A and 2B (b) Comparison of 1B and 2A; also 1A and 2A (c) Comparison of 1B and 2B; also 1A and 2B

Fig. 13. Set coverage metric of combined solutions for T_1



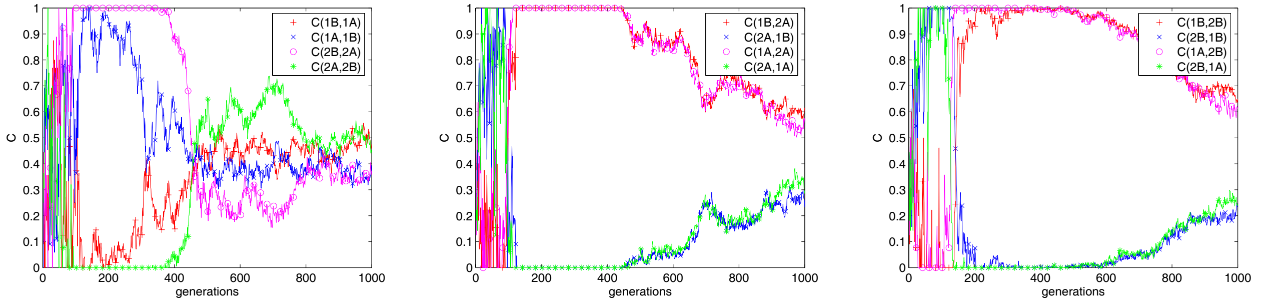
(a) Comparison of 1A and 1B; also 2A and 2B (b) Comparison of 1B and 2A; also 1A and 2A (c) Comparison of 1B and 2B; also 1A and 2B

Fig. 14. Set coverage metric of combined solutions for T_2



(a) Comparison of 1A and 1B; also 2A and 2B (b) Comparison of 1B and 2A; also 1A and 2A (c) Comparison of 1B and 2B; also 1A and 2B

Fig. 15. Set coverage metric of combined solutions for T_3



(a) Comparison of 1A and 1B; also 2A and 2B (b) Comparison of 1B and 2A; also 1A and 2A (c) Comparison of 1B and 2B; also 1A and 2B

Fig. 16. Set coverage metric of combined solutions for T_6