Improving the NSGA-II Performance with an External Population

Krzysztof Michalak

Department of Information Technologies,
Institute of Business Informatics,
Wroclaw University of Economics, Wroclaw, Poland
krzysztof.michalak@ue.wroc.pl

Abstract. The NSGA-II algorithm is among the best performing ones in the area of multiobjective optimization. The classic version of this algorithm does not utilize any external population. In this work several techniques of reintroducing specimens from the external population back to the main one were proposed. These techniques were tested on multiobjective optimization problems named ZDT-1, ZDT-2, ZDT-3, ZDT-4 and ZDT-6. Algorithm performance was evaluated with the hypervolume measure commonly used in the literature. Experiments show that reintroducing specimens from the external population improves the performance of the algorithm.

Keywords: multiobjective optimization, NSGA-II, external population

1 Introduction

This paper deals with multiobjective optimization problems. Such problems can be formalized as follows:

$$\begin{array}{l} \text{minimize } F(x) = (f_1(x), \ldots, f_m(x))^T \\ \text{subject to } x \in \Omega, \end{array} \tag{1}$$

where:
Ω - decision space,
f_i - i-th optimization criterion.

Most often there exists no single $x_0 \in \Omega$ which minimizes all $f_i, i \in \{1, \ldots, m\}$ simultaneously, so there is no possibility of simply choosing "the best" solution. However, solutions can be compared to each other using Pareto domination relation. Given two points x_1 and x_2 in the decision space Ω for which:

$$\begin{array}{l} F(x_1) = (f_1(x_1), \ldots, f_m(x_1))^T \\ F(x_2) = (f_1(x_2), \ldots, f_m(x_2))^T \end{array} \tag{2}$$

we say that x_1 *dominates* x_2 ($x_1 \succ x_2$) iff:

$$\begin{array}{l} \forall i \in 1, \ldots, m : f_i(x_1) \leq f_i(x_2) \\ \exists i \in 1, \ldots, m : f_i(x_1) < f_i(x_2) \end{array} \tag{3}$$

A solution x is said to be *nondominated* iff:

$$\neg \exists x' \in \Omega : x' \succ x. \qquad (4)$$

Solving a multiobjective optimization problem usually means finding a *Pareto set* of nondominated solutions or a *Pareto front* of points in the objective space R^m which correspond to solutions from the Pareto set.

There are many optimization techniques that can be used to find Pareto fronts. Evolutionary methods are particularly well-suited for this task because they work with an entire population of solutions among which the Pareto domination relation can be established. One of the most successful multiobjective evolutionary algorithms known in the literature is the NSGA-II algorithm [2], which uses Pareto domination relation to induce ordering among specimens, which in turn is used for selecting winners in a binary tournament procedure.

In the original version, the NSGA-II algorithm does not utilize any external population. However, external populations have been used to improve performance in other algorithms such as SPEA [11] and SPEA-2 [10]. In this paper NSGA-II-EXT algorithm - a modified version of the NSGA-II algorithm - is proposed which stores all the nondominated solutions found so far in an external population. The proposed algorithm includes an additional step in which some or all specimens from the external population are reintroduced to the main population. The NSGA-II-EXT algorithm was tested with various external population merging techniques and compared to the regular NSGA-II algorithm. A comparison with the SPEA-2 algorithm was also performed because the SPEA-2 algorithm also involves merging of the external population with the main population. In experiments performed by other authors, the NSGA-II algorithm was tested on ZDT problems and it was found to outperform PAES and SPEA algorithms [2]. Therefore, these two algorithms have not been included in the tests.

2 External Population Merging

External population is used in some multiobjective genetic algorithms such as SPEA [11] and SPEA-2 [10]. Also, a similar concept of an archive which stores all nondominated specimens found so far by the algorithm is often used to generate a comprehensive overview of possible solutions of the solved problem.

In this paper NSGA-II-EXT - a modified version of the NSGA-II algorithm is proposed in which, after each generation, the current population P is appended to the external population EP and then some specimens from the EP can be reintroduced to the original population P. In the proposed algorithm the external population EP is at the same time involved in the working of the algorithm and plays the role of an archive which stores the nondominated solutions encountered during the entire run of the algorithm.

An overview of the modified algorithm is given in Algorithm 1.

Algorithm 1 The NSGA-II-EXT algorithm.

```
$P$ = InitPopulation()
$EP = \emptyset$
Evaluate($P$)
Rank($P$)
for $t = 2 \to N_{Gen}$ do
    $P_{parent}$ = BinaryTournament($P$)
    $P_{offspring}$ = GenerateOffspring($P_{parent}$)
    Evaluate($P_{offspring}$)
    $P = P \cup P_{offspring}$

    $EP$ = AddToEP($EP, P$)
    $P$ = MergeEP($P, EP$)
    $P$ = Reduce($P$)
end for
```

The following procedures are used in the proposed algorithm:

Evaluate - calculates values of the objective functions for specimens from a given population,

Rank - assigns nondominated front ranks to specimens based on the domination relation and calculates the crowding distance in each front separately,

BinaryTournament - selects parents for crossover and mutation using a binary tournament procedure based on the nondominated front rank and the crowding distance - the same procedure as in the NSGA-II algorithm,

GenerateOffspring - generates offspring from parents, most often using crossover and mutation operators. In some cases other operators may be used. For example the inver-over operator is often applied in the case of the Travelling Salesman Problem [6]. In the experiments presented in this paper the SBX crossover operator [8] and the polynomial mutation [3] were used because these two operators are dedicated for real-domain problems.

AddToEP - adds the current population P to the external population EP. In this procedure specimens from EP dominated by specimens in P are removed and nondominated specimens from P are added to EP.

MergeEP - reintroduces specimens from the external population EP back to the main population P. Several different reintroduction procedures were tested in this paper:

– **none** - no EP merging. This is equivalent to the regular NSGA-II algorithm,
– **all** - all specimens from EP are added to P,
– **rand** - $q\%$ of specimens from EP selected at random are added to P,
– **cd** - crowding distance (CD) is calculated for specimens in $P \cup EP$, then $q\%$ of specimens from EP with the largest CD values are added to P.

Reduce - performs elitist population reduction used in the NSGA-II algorithm.

Preliminary tests were also performed using a technique based on the Average Linkage clustering method used in the SPEA algorithm [11]. However, due to slow, incremental nature of this clustering procedure this technique resulted in a very long running time and has not been, therefore, extensively tested.

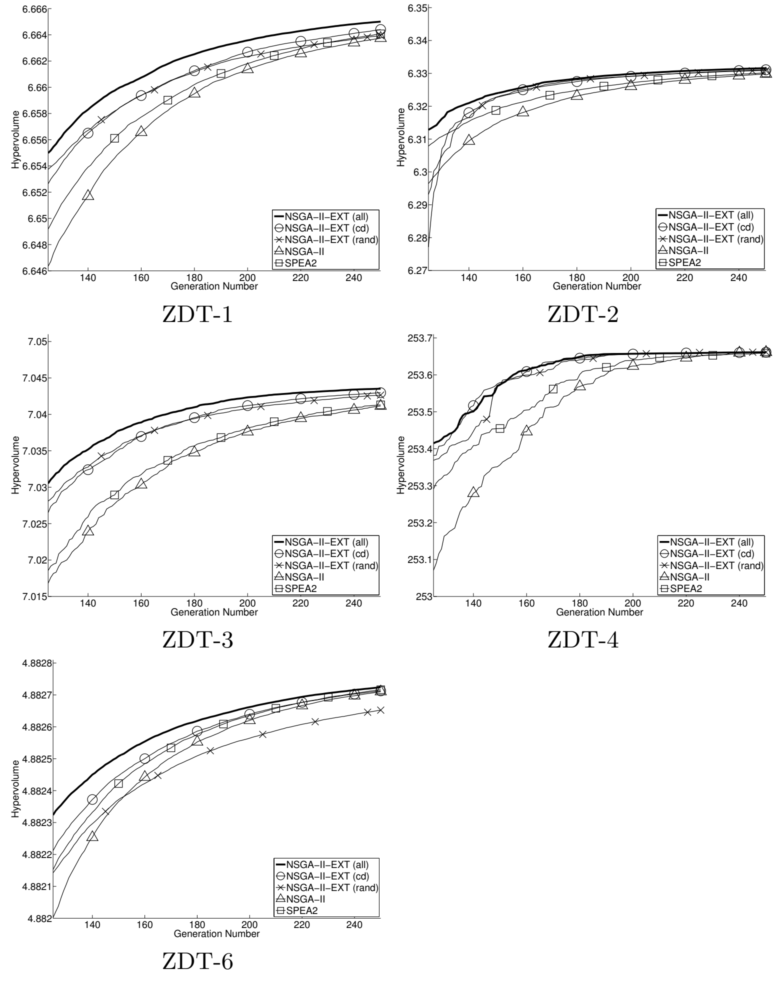
3 Experiments and Results

In the experiments the NSGA-II-EXT algorithm was tested with four external population merging methods: **all**, **cd**, **rand** and **none** (which makes the NSGA-II-EXT algorithm equivalent to the regular NSGA-II). The fraction $q\%$ of EP reintroduced into the main population in the **cd** and **rand** methods was set to 75%. The proposed value of this parameter has been set so as to differentiate the **cd** and **rand** versions of the algorithm from the **all** version. The results presented in this paper suggest that reintroducing the entire population is the most beneficial, so high values of the $q\%$ parameter should work best. However, a thorough examination of the influence of the $q\%$ parameter on the working of the algorithm has been left as a further work.

The experiments were done on several well known data sets used in the literature for testing multiobjective genetic algorithms, named ZDT-1, ZDT-2, ZDT-3, ZDT-4, ZDT-6 [9, 12]. The ZDT test suite also contains the ZDT-5 problem, but this problem was not used for tests in this paper because it uses a different encoding (binary strings as opposed to real-value encoding used in other ZDT suite test problems). Binary encoding can be handled by both the original NSGA-II algorithm and by the version proposed in this paper. However, binary-encoded problems require different operators than real-valued problems. The work on the former class of optimization problems has been left out for another paper.

For each data set and each external population merging scheme 30 iterations of the test were performed. Following [5] the probabilities of crossover and mutation were set to $P_{cross} = 0.9$ and $P_{mut} = 0.033$. Based on parameter settings in [9] the population size was set to $N_{pop} = 100$. The number of generations for all tests was set to $N_{gen} = 250$. Combined with population size N_{pop} this resulted in a total of $N_{eval} = 25000$ objective function evaluations. Following both [2] and [9] the distribution indexes for both the SBX crossover operator and polynomial mutation were set to $\eta_{cross} = 20$ and $\eta_{mut} = 20$.

During the tests the contents of the external population EP was stored after each generation. The quality of results was assessed using the hypervolume measure [4, 13] which is often used for comparing the performance of multi-objective evolutionary algorithms. The hypervolume is the Lebesgue measure of the portion of the objective space that is dominated by a set of solutions collectively. In two and three dimensions the hypervolume corresponds to the area and volume respectively. To calculate the hypervolume for a given set of solutions the Hypervolume by Slicing Objectives (HSO) algorithm [7] was used. In order to be able to plot hypervolume changes in time a single nadir point was calculated for each test problem from all generations and test runs of all algorithms. Using this common nadir point as a reference, hypervolume values were calculated for each generation in each test run. Plots of hypervolume values against the generation number for all test problems are presented in Figure 1. These graphs show hypervolume values starting from the half of the number of generations because quick changes of the hypervolume in the first half of each run extend the scale and make the graph too compressed to be readable. It can be seen in the graphs that in most cases the new algorithm converges faster than the NSGA-II and SPEA2 algorithmns, which could not achieve equally good results in the given 250 generations. Only in the case of the ZDT-4 problem all the algortihms attained an approximately the same hypervolume in 250 generations, but the NSGA-II-EXT worked faster than NSGA-II and SPEA2.



ZDT-1

ZDT-2

ZDT-3

ZDT-4

ZDT-6

Fig. 1. The hypervolume plotted against the generation number.

The performance of different variants of the NSGA-II-EXT algorithm was verified with a statistical test. Since hypervolume values obtained in the tests are not normally distributed a Kruskall-Wallis test [1] was performed which does not make an assumption about distribution normality. The Kruskall-Wallis test verifies the null hypothesis that two distributions have equal medians. Therefore, median values of the hypervolume obtained at the 250th generation were calculated from 30 repetitions of the test and used for comparing the algorithms.

Results of the statistical tests are presented in Tables 1-5. These tables present a comparison of the algorithms listed in the first column of each table to the NSGA-II and the SPEA-2 algorithms. P-values given in the tables are calculated for the null hypothesis that hypervolume distributions obtained using the compared algorithms have equal medians. The interpretation of the numerical results was determined as follows. If the median of the hypervolume distribution obtained using the algorithm listed in the left column is lower than the median obtained using the NSGA-II (or the SPEA-2 respectively) then the interpretation is ”worse” regardless of the p-value. If the median of the hypervolume distribution obtained using the algorithm listed in the left column is not lower than the median obtained using the NSGA-II (or SPEA-2 respectively) then the interpretation is ”significant” if the p-value does not exceed 0.05 or ”insignificant” otherwise. If the result obtained by the algorithms listed in the first column is interpreted as significantly better that the result obtained by the NSGA-II or SPEA-2 the corresponding fields in the table are shaded in gray.

Table 1. Results of statistical comparison of the proposed algorithms based on hypervolume values obtained on the ZDT-1 test problem

Algorithm	**median**	**std**	**compared to NSGA-II**		**compared to SPEA-2**	
			p-value	interp.	p-value	interp.
NSGA-II	6.6637	0.00027870	—		0.00017317	worse
NSGA-II-EXT (cd)	6.6644	0.00025231	2.7927e-009	significant	7.4755e-006	significant
NSGA-II-EXT (rand)	6.6639	0.00033312	0.12057	insignificant	0.041325	worse
NSGA-II-EXT (all)	6.6650	0.00022047	2.8719e-011	significant	4.2855e-011	significant
SPEA-2	6.6641	0.00021345	0.00017317	significant	—	

Table 2. Results of statistical comparison of the proposed algorithms based on hypervolume values obtained on the ZDT-2 test problem

Algorithm	**median**	**std**	**compared to NSGA-II**		**compared to SPEA-2**	
			p-value	interp.	p-value	interp.
NSGA-II	6.3298	0.00049425	—		0.00010095	worse
NSGA-II-EXT (cd)	6.3311	0.0004308	3.8787e-011	significant	7.4386e-009	significant
NSGA-II-EXT (rand)	6.3309	0.00041632	9.4449e-011	significant	4.9898e-007	significant
NSGA-II-EXT (all)	6.3316	0.00031698	2.8719e-011	significant	5.2283e-011	significant
SPEA-2	6.3301	0.00041895	0.00010095	significant	—	

Table 3. Results of statistical comparison of the proposed algorithms based on hypervolume values obtained on the ZDT-3 test problem

Algorithm	**median**	**std**	**compared to NSGA-II**		**compared to SPEA-2**	
			p-value	interp.	p-value	interp.
NSGA-II	7.0412	0.00040727	—		0.34404	worse
NSGA-II-EXT (cd)	7.0430	0.015934	1.0241e-007	significant	6.2639e-008	significant
NSGA-II-EXT (rand)	7.0426	0.015917	1.0241e-007	significant	6.2639e-008	significant
NSGA-II-EXT (all)	7.0435	0.0095554	5.3167e-010	significant	4.403e-010	significant
SPEA-2	7.0413	0.013224	0.34404	insignificant	—	

Table 4. Results of statistical comparison of the proposed algorithms based on hypervolume values obtained on the ZDT-4 test problem

Algorithm	**median**	**std**	**compared to NSGA-II**		**compared to SPEA-2**	
			p-value	interp.	p-value	interp.
NSGA-II	6.6613	0.97866	—		0.30071	insignificant
NSGA-II-EXT (cd)	6.6614	1.5487	0.56421	insignificant	0.086346	insignificant
NSGA-II-EXT (rand)	6.6618	0.031004	0.28711	insignificant	0.017299	significant
NSGA-II-EXT (all)	6.6613	0.031059	0.47792	worse	0.028663	significant
SPEA-2	6.6590	0.037741	0.30071	worse	—	

4 Conclusions

In this paper the NSGA-II-EXT algorithm - a modified version of the NSGA-II algorithm was proposed which stores all nondominated solutions found so far in an external population EP. Specimens from the external population are reintroduced to the main population at the end of each generation. Several methods of such merging were tested: no merging at all, selection of a fraction of specimens from EP with the highest value of crowding distance, random selection of a fraction of specimens from EP and merging of the entire external population EP with the main population.

The performance of the NSGA-II-EXT algorithm with different merging schemes, the original NSGA-II algorithm and the SPEA-2 algorithm was compared using test problems ZDT-1, ZDT-2, ZDT-3, ZDT-4 and ZDT-6. The com-

Table 5. Results of statistical comparison of the proposed algorithms based on hypervolume values obtained on the ZDT-6 test problem

Algorithm	**median**	**std**	**compared to NSGA-II**		**compared to SPEA-2**	
			p-value	interp.	p-value	interp.
NSGA-II	4.8827	1.0912e-005	—		0.0025611	worse
NSGA-II-EXT (cd)	4.8827	9.9921e-006	0.58436	insignificant	0.0084977	worse
NSGA-II-EXT (rand)	4.8827	1.5931e-005	3.5098e-011	worse	2.8719e-011	worse
NSGA-II-EXT (all)	4.8827	8.4028e-006	2.2902e-008	significant	0.0010302	significant
SPEA-2	4.8827	9.4905e-006	0.0025611	significant	—	

parison was done using hypervolume measure commonly used in the literature. The NSGA-II-EXT (all) algorithm (the variant with merging of the entire external population) is among the best performing algorithms tested in this paper. For all tests problems except ZDT-4 the NSGA-II-EXT (all) outperformed the NSGA-II algorithm in terms of hypervolume. The SPEA-2 algorithm was outperformed by NSGA-II-EXT (all) in all the tests. Both results were verified using the Kruskall-Wallis test to be statistically significant.

Overall, the NSGA-II-EXT (all) variant of the algorithm proposed in this paper seems to be able to generate many new solutions of the tested multiobjective problems and to produce Pareto fronts with good hypervolume values in the case of most of the problems tested in this paper, compared to algorithms known from the literature and compared to other variants of the NSGA-II-EXT algorithm based on different external population recombination techniques.

References

1. Corder, G.W., Foreman, D.I.: Nonparametric Statistics for Non-Statisticians: A Step-by-Step Approach. New Jersey: Wiley (2009)
2. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. on Evol. Comp. 6, 182–197 (2002)
3. Hamdan, M.: On the disruption-level of polynomial mutation for evolutionary multi-objective optimisation algorithms. Comput. and Inf. 29(5), 783–800 (2010)
4. Purshouse, R.: On the evolutionary optimisation of many objectives. Ph. D. thesis, The University of Sheffield, United Kingdom (2003)
5. Sharma, D., Kumar, A., Deb, K., Sindhya, K.: Hybridization of SBX based NSGA-II and sequential quadratic programming for solving multi-objective optimization problems. In: IEEE Congress on Evol. Comp. pp. 3003–3010. IEEE (2007)
6. Tao, G., Michalewicz, Z.: Inver-over operator for the TSP. In: in Proc. PPSN. pp. 803–812. Springer (1998)
7. While, L., Hingston, P., Barone, L., Huband, S.: A faster algorithm for calculating hypervolume. IEEE Trans. on Evol. Comp. 10(1), 29–38 (2006)
8. Yang, L., Yang, C., Liu, Y.: Particle swarm optimization with simulated binary crossover. In: Intelligent Systems Design and Engineering Applications (ISDEA), 2014 Fifth International Conference on. pp. 710–713 (2014)
9. Zhang, Q., Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Trans. on Evol. Comp. 11(6), 712–731 (2007)
10. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In: Giannakoglou, K., et al. (eds.) Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001). pp. 95–100. International Center for Numerical Methods in Engineering (CIMNE) (2002)
11. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans. on Evol. Comp. 3(4), 257–271 (1999)
12. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation 8(2), 173–195 (2000)
13. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multiobjective optimizers: An analysis and review. IEEE Trans. on Evol. Comp. 7, 117–132 (2002)