Various Degrees of Steadiness in NSGA-II and Their Influence on the Quality of Results

Maxim Buzdalov
ITMO University
49 Kronverkskiy ave.
Saint-Petersburg, Russia
buzdalov@rain.ifmo.ru

Vladimir Parfenov
ITMO University
49 Kronverkskiy ave.
Saint-Petersburg, Russia
parfenov@mail.ifmo.ru

ABSTRACT

Steady-state evolutionary algorithms are often favoured over generational ones due to better scalability in parallel and distributed environments. However, in certain conditions they are able to produce results of better quality as well.

We consider several ways to introduce various "degrees of steadiness" in the NSGA-II algorithm, some of which have not been known in literature, and show experimentally (on a corpus of 21 test problems) the presence of a general trend: algorithms with more steadiness yield better results.

CCS Concepts

•**Theory of computation** → *Bio-inspired optimization; Random search heuristics;*

Keywords

NSGA-II, multi-objective, steady-state.

1. INTRODUCTION

For multi-objective evolutionary algorithms, steady-state versions typically have significantly higher computation complexity, measured as number of supporting operations per evaluation, than generational versions. This happens because procedures which update algorithm's computation state have to be run not once per generation but once per individual. For example, the NSGA-II algorithm [3], which has an $O(N^2K)$ complexity of its generational version (where N is the generation size and K is the problem dimension), or $O(N \log^{K-1} N)$ if implemented more efficiently [1], becomes $\Theta(N)$ times slower when implemented as steady-state [5].

This problem limits both theoretical and practical interest in steady-state approaches for this area. However, recent advances introduced algorithms and data structures which made certain steady-state approaches efficient in theory and practice, which motivated further research.

The historically first approach to decrease computational complexity for steady-state NSGA-II is called "Efficient Non-dominated Level Update" and described in a technical report by Li, Deb, Zhang and Kwong [4]. Although the worst-case complexity of a single element insertion is still $O(N^2K)$, a worst-case complexity of $O(N\sqrt{N}K)$ was proven in [4] if elements are distributed evenly among non-dominated layers.

Another approach is based on an algorithm called "Incremental Non-Dominated Sorting" (INDS), which is currently works for bi-objective problems only. The algorithm and the data structure are presented in [7], and an INDS-based steady-state version of NSGA-II is presented and analysed in [2]. The insertion and removal complexities are at most $O(N)$ and are provably lower under several conditions.

Nebro and Durillo [5] showed that the steady-state version of the NSGA-II algorithm typically demonstrates better convergence and diversity than the generational one (within equal budgets for fitness function evaluations). In this paper we experimentally show that this trend spreads over many different ways of introducing steadiness to this algorithm: the "more steady" the algorithm is, the better results it yields. As we use the approach from [2], computational running times are almost the same regardless steadiness, so we concentrate on result quality only.

This work was financially supported by the Government of Russian Federation, Grant 074-U01.

GECCO '15 July 11-15, 2015, Madrid, Spain

ACM ISBN 978-1-4503-3488-4/15/07.

DOI: http://dx.doi.org/10.1145/2739482.2764887

2. DEGREES OF STEADINESS

On a very high level, the classical NSGA-II algorithm (and possibly many other algorithms) can be described as an iterative algorithm where each iteration manipulates with a set of solutions S of size N in the following way:

1. $A \leftarrow$ a set of N solutions sampled from S.
2. $B \leftarrow$ individuals from A after crossover and mutation.
3. $S \leftarrow$ best N individuals of $S \cup B$.

Note that the last line can be read as *bulk insertion* of new individuals into S and then *bulk removal* of individuals which are worse by rank or by crowding distance.

However, bulk removal can remove many promising individuals. For example, consider individuals $A = (0, 1)$, $B = (0.3, 0.7)$, $C = (0.6, 0.4)$, $D = (0.65, 0.35)$, $E = (1, 0)$. If two worst individuals have to be removed, then bulk removal deletes C and D. However if we remove worst individuals *one by one*, e.g. remove the worst one and then again remove the worst one, then the first individual to remove is C, and the second one is B. The resulting set $\{A, D, E\}$ has a better value of the hypervolume indicator than $\{A, B, E\}$. We call this approach *bulk insertion, steady removal*.

Table 1: Results of experiments. Notation: PSS – pure steady state, SISR – steady insertion, steady removal, BISR – bulk insertion, steady removal, BIBR – bulk insertion, bulk removal (the classical scheme).

Problem	PSS	SISR	BISR	BIBR	Problem	PSS	SISR	BISR	BIBR
DTLZ1	$4.927 \cdot 10^{-1}$	$4.927 \cdot 10^{-1}$	$4.924 \cdot 10^{-1}$	$4.915 \cdot 10^{-1}$	DTLZ2	$2.108 \cdot 10^{-1}$	$2.108 \cdot 10^{-1}$	$2.104 \cdot 10^{-1}$	$2.092 \cdot 10^{-1}$
DTLZ3	$1.387 \cdot 10^{-1}$	$7.812 \cdot 10^{-2}$	$8.089 \cdot 10^{-2}$	$5.233 \cdot 10^{-2}$	DTLZ4	$2.108 \cdot 10^{-1}$	$2.108 \cdot 10^{-1}$	$2.105 \cdot 10^{-1}$	$2.092 \cdot 10^{-1}$
DTLZ5	$2.108 \cdot 10^{-1}$	$2.108 \cdot 10^{-1}$	$2.104 \cdot 10^{-1}$	$2.092 \cdot 10^{-1}$	DTLZ6	$2.107 \cdot 10^{-1}$	$2.107 \cdot 10^{-1}$	$2.107 \cdot 10^{-1}$	$2.093 \cdot 10^{-1}$
DTLZ7	$3.089 \cdot 10^{-1}$	$3.089 \cdot 10^{-1}$	$3.089 \cdot 10^{-1}$	$3.087 \cdot 10^{-1}$					
WFG1	$2.066 \cdot 10^{-1}$	$2.433 \cdot 10^{-1}$	$2.403 \cdot 10^{-1}$	$2.470 \cdot 10^{-1}$	WFG2	$5.577 \cdot 10^{-1}$	$5.577 \cdot 10^{-1}$	$5.577 \cdot 10^{-1}$	$5.573 \cdot 10^{-1}$
WFG3	$4.416 \cdot 10^{-1}$	$4.416 \cdot 10^{-1}$	$4.414 \cdot 10^{-1}$	$4.408 \cdot 10^{-1}$	WFG4	$2.105 \cdot 10^{-1}$	$2.105 \cdot 10^{-1}$	$2.101 \cdot 10^{-1}$	$2.089 \cdot 10^{-1}$
WFG5	$1.791 \cdot 10^{-1}$	$1.791 \cdot 10^{-1}$	$1.789 \cdot 10^{-1}$	$1.780 \cdot 10^{-1}$	WFG6	$2.004 \cdot 10^{-1}$	$2.039 \cdot 10^{-1}$	$2.029 \cdot 10^{-1}$	$2.017 \cdot 10^{-1}$
WFG7	$2.106 \cdot 10^{-1}$	$2.106 \cdot 10^{-1}$	$2.102 \cdot 10^{-1}$	$2.089 \cdot 10^{-1}$	WFG8	$1.479 \cdot 10^{-1}$	$1.481 \cdot 10^{-1}$	$1.481 \cdot 10^{-1}$	$1.471 \cdot 10^{-1}$
WFG9	$2.088 \cdot 10^{-1}$	$2.090 \cdot 10^{-1}$	$2.085 \cdot 10^{-1}$	$2.070 \cdot 10^{-1}$					
ZDT1	$6.617 \cdot 10^{-1}$	$6.613 \cdot 10^{-1}$	$6.610 \cdot 10^{-1}$	$6.596 \cdot 10^{-1}$	ZDT2	$3.284 \cdot 10^{-1}$	$3.279 \cdot 10^{-1}$	$3.277 \cdot 10^{-1}$	$3.265 \cdot 10^{-1}$
ZDT3	$5.156 \cdot 10^{-1}$	$5.154 \cdot 10^{-1}$	$5.153 \cdot 10^{-1}$	$5.148 \cdot 10^{-1}$	ZDT4	$6.574 \cdot 10^{-1}$	$6.567 \cdot 10^{-1}$	$6.567 \cdot 10^{-1}$	$6.560 \cdot 10^{-1}$
ZDT6	$3.970 \cdot 10^{-1}$	$3.913 \cdot 10^{-1}$	$3.914 \cdot 10^{-1}$	$3.907 \cdot 10^{-1}$					

The next degree of steadiness would be to add individuals from B one at a time and to remove the worst individual immediately after each insertion. We call it *steady insertion, steady removal*. Finally, the last degree of steadiness is the usual steady-state version, or *pure steady-state*. The only difference between the latter two versions is that the former samples many elements at a time before their insertion, while the latter samples one element at a time.

3. EXPERIMENTS

We evaluated[1] four mentioned versions of NSGA-II with different degrees of steadiness on 21 test problems (ZDT1–ZDT4 and ZDT6, DTLZ1–DTLZ7, WFG1–WFG9, all mentioned in [5]). There were 1000 runs of each version on each problem. Median values of the hypervolume indicator (evaluated as in papers [2, 5]) are presented in Table 1.

One can see that in all problems except six (DTLZ3, WFG1, WFG6, WFG8, WFG9, ZDT6) the median values do not decrease as steadiness increase. If we exclude the pure steady state version, there are only three problems (DTLZ3, WFG1, ZDT6) where the order is "broken".

In order to compare the results in a more proper way, we conducted the Wilcoxon signed rank test for all pairs of algorithms for every single problem. Each test was performed thrice: with "two sided", "less" and "greater" alternative hypotheses (in notation of the R [6] language). If for the "two-sided" test the p-value is less than 0.008 (this is how we adapt the usual value of 0.05 to six tests per problem), we consider the difference to be significant, and in this case exactly one of "less" and "greater" tests, which produces the p-value less than 0.008 shows which algorithm is better. The problems could be split into four classes:

- all statistical comparisons are strict and have the "right" directions: DTLZ2, DTLZ5, DTLZ7, WFG5, WFG7, ZDT1–ZDT3 (8 problems);
- all strict statistical comparisons have the "right" directions: DTLZ1, DTLZ3, DTLZ4, WFG2–WFG4, ZDT4, ZDT6 (8 problems);
- there is exactly one strict comparison which has the "wrong" direction: DTLZ6, WFG9 (2 problems);
- other problems: WFG1, WFG6, WFG8 (3 problems).

[1]Source code is available at GitHub: https://github.com/mbuzdalov/papers/tree/master/2015-gecco-nsga-ii-ss

We also noticed that for non-pure steady-state approaches no statistical comparison showed the "wrong" direction (that is, all "broken" differences between the corresponding medians were statistically insignificant).

We hope that the presented results will encourage the multi-objective evolutionary community to pay more attention to steady-state algorithms, as they are capable of producing better results, and nowadays they are not as time-consuming as they used to be.

4. REFERENCES

[1] M. Buzdalov and A. Shalyto. A provably asymptotically fast version of the generalized Jensen algorithm for non-dominated sorting. In *International Conference on Parallel Problem Solving from Nature*, number 8672 in Lecture Notes in Computer Science, pages 528–537. 2014.

[2] M. Buzdalov, I. Yakupov, and A. Stankevich. Fast implementation of the steady-state NSGA-II algorithm for two dimensions based on incremental non-dominated sorting. In *Proceedings of Genetic and Evolutionary Computation Conference*, 2015 (to appear).

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast Elitist Multi-Objective Genetic Algorithm: NSGA-II. *Transactions on Evolutionary Computation*, 6:182–197, 2000.

[4] K. Li, K. Deb, Q. Zhang, and S. Kwong. Efficient non-domination level update approach for steady-state evolutionary multiobjective optimization. Technical report, 2014.

[5] A. J. Nebro and J. J. Durillo. On the effect of applying a steady-state selection scheme in the multi-objective genetic algorithm NSGA-II. In *Nature-Inspired Algorithms for Optimisation*, number 193 in Studies in Computational Intelligence, pages 435–456. Springer Berlin Heidelberg, 2009.

[6] R Core Team. R: A language and environment for statistical computing. http://www.R-project.org/, 2013.

[7] I. Yakupov and M. Buzdalov. Incremental non-dominated sorting with $O(N)$ insertion for the two-dimensional case. In *Proceedings of IEEE Congress on Evolutionary Computation*, 2015 (to appear).