The Distribution Index in Polynomial Mutation for Evolutionary Multiobjective Optimisation Algorithms: An Experimental Study

Mohammad M. Hamdan
Department of Computer Science
Yarmouk University
Irbid 21163
Jordan
Email: hamdan@yu.edu.jo

***Abstract*—Polynomial mutation is an important operator in evolutionary multiobjective optimisation algorithms. Here, an experimental study is conducted on the behaviour of this operator by changing the settings of the distribution index parameter. The results are interesting in one of the most popular algorithms (NSGA-II) using two performance metrics: hypervolume and spread.**

I. Introduction

Different setting of the parameters of a system could yield gain in system performance in case of minimisation, maximisation or even equilibrium. In real systems, there are different objectives to be configured during the optimisation process. However, these objective could be conflicting and difficult to minimise or maximise all at the same time. Therefore, a set of tradeoffs could be found and the user could determine the settings he prefers. This is known as Multiobjective Optimisation Problems (MOPs) [2].

Evolutionary Multiobjective Optimisation Algorithms (EMOAs) [1] have been known of being sucessfull in solving MOPs. The Non-dominated Sorting Genetic Algorithm II (NSGA-II) [5] is one of the fastest breaking algorithms in this domain. The aim of this algorithm is to find multiple Pareto optimal solutions with good diversity. It follows the standard operators of genetic algorithm in terms of selection, crossover and mutation. NSGA-II uses Polynomial mutation as the mutation operator.

The work presented in this paper looks at the settings of the polynomial mutation operator. We aim to show whether or not to use the common settings of the operator parameters.

The rest of this paper is organised as follows. Section II describes the polynomial mutation. The experimental environment is presented in Section III. The results are shown in Section IV. Finally, Section V concludes the paper.

II. The Polynomial Mutation

The polynomial mutation was first proposed in [3]. It is shown in Algorithm 1. Mutation probability is set by the variable P_m, n is number of decision variables and η_m is distribution index which can take any non-negative value. For each decision variable x_i, box constraints are defined in $[x_i^{Lower}, x_i^{Upper}]$. The polynomial mutation works as follows. Each decision variable X_i has a probability P_m to be perturbed. Using the procedure described in Algorithm 1, a mutated variable gets its new value. It is common in the literature to set the mutation parameters: P_m and η_m to $1/n$ and 20, respectively.

```
i ← 0 ;
repeat
    r ← U[0,1] ;
    if r ≤ P_m then
        δ_1 ← (X_i − X_i^Lower) / (X_i^Upper − X_i^Lower) ;
        δ_2 ← (X_i^Upper − X_i) / (X_i^Upper − X_i^Lower) ;
        r ← U[0,1] ;
        δ_q ← { [(2r) + (1 − 2r)*
                 (1 − δ_1)^(η_m+1)]^(1/(η_m+1)) − 1            if r ≤ 0.5
                1 − [2(1 − r) + 2.(r − 0.5)*
                 (1 − δ_2)^(η_m+1)]^(1/(η_m+1))                otherwise
        X_i ← X_i + δ_q.(X_i^Upper − X_i^Lower)
    end
until i + + == n;
```

$$\delta_q \leftarrow \begin{cases} [(2r) + (1-2r)*(1-\delta_1)^{\eta_m+1}]^{\frac{1}{\eta_m+1}} - 1 & \text{if } r \leq 0.5 \\ 1 - [2(1-r) + 2.(r-0.5)*(1-\delta_2)^{\eta_m+1}]^{\frac{1}{\eta_m+1}} & \text{otherwise} \end{cases}$$

Algorithm 1: The Polynomial Mutation.

Previous work on this operator has showed that there are two variants in the implemented systems [7]. In this study we use the latest version (highly-disruptive polynomial mutation [8]).

Figure 1 illustrates the effect of using different values for the distribution index η_m. Here, smaller values (i.e. $\eta_m = 1$) is considered a strong mutation and results in new values that are far away from original variable value. On the other hand, using bigger values ($\eta_m \geq 15$) gives more probability of generating new values that are very close to parent. Strong mutation has better chances of escaping local optima.

III. Experimental Environment

To validate the proposed idea in this work, we used the JMetal version 4.0 [6] framework for the experiments. Here

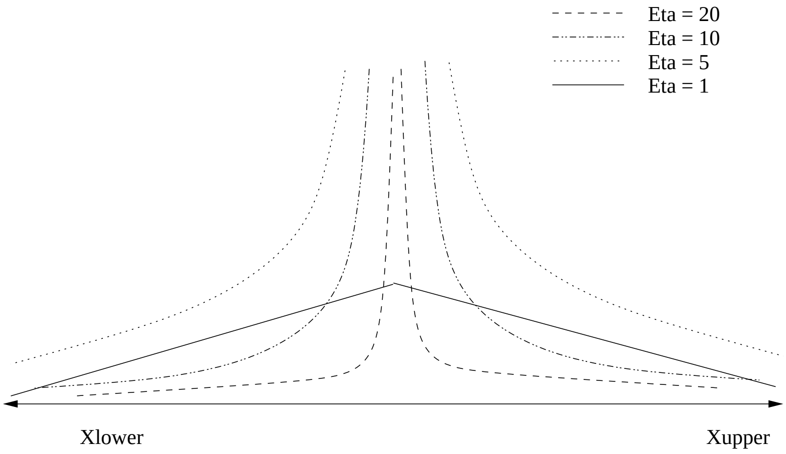

Fig. 1. The effect of distribution index η_m on mutated values.

NSGA-II is used for the study on the following artificial test suites: Zitzler-Deb-Theile (ZDT) [10], Deb-Thiele-Laumanns-Zitzler (DTLZ) [4] and Walking-Fish-Group (WFG) [9]. The standard number of variables and objects were used (as defined in JMetal). The mutation probability is $P_m = 1/n$. We used different settings for η_m as follows: 1, 5, 10, 15, 20, 50 and 100. Here, small values should give a strong mutation while large values should give a smooth mutation. We used 25000 function evaluations and the results were averaged over 100 independent runs. For measuring the difference between different settings of the distribution index η_m parameter, we used two metrics: hypervolume [11] and spread [5].

IV. Results

Table I shows the hypervolume and spread results for NSGA-II using different settings for the distribution index η_m. The first best is highlighted in dark Grey while the second best in light Grey. Regarding hypervolume metric, NSGA-II prefered strong mutation for up to 12 out of 21 problems. Also, for spread metric, the strong mutation was better for up to 18 out of 21 problems.

It is clearly noted that for most of the problems and for the three test suites NSGA-II prefers to use a strong mutation, i.e. $\eta_m = 1$. This finding suggests that it is better to use a strong mutation rather than a smooth mutation ($\eta_m \geq 20$).

This is an interesting finding that would change the recommended value for distribution index. It is important to note that many EMOAs use a value of either 15 or 20 [8]. The corresponding box plots for hypervolume and spread are shown in Figures 2 and 3, respectively. Again the box plots confirm that a strong mutation is better for most problems and for both performance metrics.

V. Conclusion and Future Work

Evolutionary algorithms are sensitive to parameter setting. The behavior and results can change easily from one problem to another or when using different programming languages and systems (e.g. non uniformity of random variable generator). Also, using previously suggested values for parameters could be misleading. It is better to experiment with your own algorithm and problems before running the main experiments. As it was shown in this paper, it was found that a strong mutation using a distribution index value of $\eta_m = 1$ gave better results to the common configuration of using $\eta_m = 20$. It would be useful if the same experiment in this paper is used with other EMOAs and the shifted version of the problems used in this study.

References

[1] Carlos A. Coello Coello, David A. Van Veldhuizen, and Gary B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, May 2002. ISBN 0-3064-6762-3.

[2] K. Deb. *Multi-objective optimization using evolutionary algorithms*. Wiley, Chichester, UK, 2001.

[3] K. Deb and M. Goyal. A combined genetic adaptive search (geneas) for engineering design. *Computer Science and Informatics*, 26(4):30–45, 1996.

[4] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Test Problems for Evolutionary Multi-Objective Optimization. In A. Abraham, R. Jain, and R. Goldberg, editors, *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, chapter 6, pages 105–145. Springer, 2005.

[5] Kalyanmoy Deb, Sameer Agarwal Amrit Pratap, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.

[6] Juan J. Durillo, Antonio J. Nebro, Francisco Luna, Bernabé Dorronsoro, and Enrique Alba. jMetal: A Java Framework for Developing Multi-Objective Optimization Metaheuristics. Technical Report ITI-2006-10, Departamento de Lenguajes y Ciencias de la Computación, University of Málaga, E.T.S.I. Informática, Campus de Teatinos, December 2006.

[7] Mohammad Hamdan. On the disruption-level of polynomial mutation for evolutionary multi-objective optimisation algorithms. *Computing and Informatics*, 28(2009), 2009.

[8] Mohammad Hamdan. A dynamic polynomial mutation for evolutionary multi-objective optimization algorithms. *International Journal on Artificial Intelligence Tools*, 20(1):209–219, 2011.

[9] S. Huband, P. Hingston, L. Barone, and L. While. A review of multiobjective test problems and a scalable test problem toolkit. *Evolutionary Computation, IEEE Transactions on*, 10(5):477 –506, oct. 2006.

[10] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Embirical results. *Evolutionary Computing*, 8(2):173–195, 2000.

[11] Eckart Zitzler and Lothar Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, November 1999.

TABLE I

HYPERVOLUME: MEAN AND STANDARD DEVIATION (ABOVE) AND SPREAD: MEAN AND STANDARD DEVIATION (BELOW)

	$\eta_m = 1$	$\eta_m = 5$	$\eta_m = 10$	$\eta_m = 15$	$\eta_m = 20$	$\eta_m = 50$	$\eta_m = 100$
ZDT1	$6.60e-01_{3.3e-04}$	$6.59e-01_{3.5e-04}$	$6.59e-01_{3.2e-04}$	$6.59e-01_{3.6e-04}$	$6.59e-01_{3.8e-04}$	$6.59e-01_{3.3e-04}$	$6.59e-01_{2.8e-04}$
ZDT2	$3.26e-01_{3.0e-04}$	$3.26e-01_{3.6e-04}$	$3.26e-01_{3.5e-04}$	$3.25e-01_{4.2e-04}$	$3.26e-01_{3.2e-04}$	$3.26e-01_{3.1e-04}$	$3.26e-01_{3.4e-04}$
ZDT3	$5.15e-01_{3.8e-04}$	$5.15e-01_{1.6e-04}$	$5.15e-01_{3.8e-04}$	$5.14e-01_{6.4e-04}$	$5.15e-01_{3.7e-04}$	$5.15e-01_{1.6e-04}$	$5.15e-01_{1.5e-04}$
ZDT4	$6.56e-01_{3.3e-03}$	$6.54e-01_{3.6e-03}$	$6.53e-01_{3.8e-03}$	$6.53e-01_{4.0e-03}$	$6.54e-01_{3.6e-03}$	$6.55e-01_{3.7e-03}$	$6.55e-01_{3.1e-03}$
ZDT6	$3.89e-01_{1.3e-03}$	$3.88e-01_{1.5e-03}$	$3.87e-01_{1.7e-03}$	$3.86e-01_{1.8e-03}$	$3.88e-01_{1.5e-03}$	$3.89e-01_{1.6e-03}$	$3.89e-01_{1.7e-03}$
DTLZ1	$6.16e-01_{2.5e-01}$	$6.19e-01_{2.4e-01}$	$6.23e-01_{2.2e-01}$	$5.64e-01_{2.6e-01}$	$6.46e-01_{2.1e-01}$	$6.05e-01_{2.4e-01}$	$6.47e-01_{2.2e-01}$
DTLZ2	$3.74e-01_{6.2e-03}$	$3.74e-01_{6.1e-03}$	$3.73e-01_{6.4e-03}$	$3.73e-01_{6.2e-03}$	$3.74e-01_{5.5e-03}$	$3.74e-01_{6.0e-03}$	$3.74e-01_{4.9e-03}$
DTLZ3	$0.00e+00_{0.0e+00}$	$0.00e+00_{0.0e+00}$	$0.00e+00_{0.0e+00}$	$0.00e+00_{0.0e+00}$	$0.00e+00_{0.0e+00}$	$0.00e+00_{0.0e+00}$	$0.00e+00_{0.0e+00}$
DTLZ4	$3.70e-01_{3.8e-02}$	$3.75e-01_{4.9e-03}$	$3.75e-01_{4.8e-03}$	$3.67e-01_{5.3e-02}$	$3.67e-01_{5.3e-02}$	$3.70e-01_{3.8e-02}$	$3.74e-01_{5.0e-03}$
DTLZ5	$9.29e-02_{2.3e-04}$	$9.28e-02_{2.2e-04}$	$9.27e-02_{2.0e-04}$	$9.26e-02_{2.2e-04}$	$9.28e-02_{1.9e-04}$	$9.28e-02_{2.2e-04}$	$9.28e-02_{2.4e-04}$
DTLZ6	$0.00e+00_{0.0e+00}$	$0.00e+00_{0.0e+00}$	$0.00e+00_{0.0e+00}$	$0.00e+00_{0.0e+00}$	$0.00e+00_{0.0e+00}$	$0.00e+00_{0.0e+00}$	$0.00e+00_{0.0e+00}$
DTLZ7	$2.80e-01_{4.2e-03}$	$2.80e-01_{3.5e-03}$	$2.79e-01_{5.1e-03}$	$2.77e-01_{4.9e-03}$	$2.79e-01_{4.0e-03}$	$2.80e-01_{4.2e-03}$	$2.80e-01_{3.9e-03}$
WFG1	$5.22e-01_{9.8e-02}$	$5.43e-01_{8.8e-02}$	$5.15e-01_{1.0e-01}$	$4.97e-01_{1.1e-01}$	$5.19e-01_{9.1e-02}$	$5.22e-01_{8.6e-02}$	$5.15e-01_{8.2e-02}$
WFG2	$5.62e-01_{1.4e-03}$	$5.62e-01_{1.4e-03}$	$5.62e-01_{1.3e-03}$	$5.62e-01_{1.3e-03}$	$5.62e-01_{1.4e-03}$	$5.62e-01_{1.4e-03}$	$5.62e-01_{1.4e-03}$
WFG3	$4.41e-01_{2.9e-04}$	$4.41e-01_{3.1e-04}$	$4.41e-01_{3.1e-04}$	$4.40e-01_{2.8e-04}$	$4.41e-01_{3.6e-04}$	$4.41e-01_{3.2e-04}$	$4.41e-01_{2.9e-04}$
WFG4	$2.17e-01_{3.3e-04}$	$2.17e-01_{3.0e-04}$	$2.17e-01_{3.9e-04}$	$2.17e-01_{3.7e-04}$	$2.17e-01_{3.2e-04}$	$2.17e-01_{3.3e-04}$	$2.17e-01_{3.4e-04}$
WFG5	$1.95e-01_{3.4e-04}$	$1.95e-01_{8.3e-04}$	$1.95e-01_{2.6e-04}$	$1.95e-01_{3.1e-04}$	$1.95e-01_{3.2e-04}$	$1.95e-01_{7.5e-04}$	$1.95e-01_{3.5e-04}$
WFG6	$2.00e-01_{9.8e-03}$	$1.99e-01_{1.2e-02}$	$1.99e-01_{1.0e-02}$	$2.00e-01_{9.5e-03}$	$2.00e-01_{1.2e-02}$	$1.97e-01_{1.2e-02}$	$1.99e-01_{9.3e-03}$
WFG7	$2.09e-01_{3.1e-04}$	$2.09e-01_{2.6e-04}$	$2.09e-01_{3.4e-04}$	$2.09e-01_{3.4e-04}$	$2.09e-01_{3.5e-04}$	$2.09e-01_{3.4e-04}$	$2.09e-01_{3.1e-04}$
WFG8	$1.53e-01_{1.5e-02}$	$1.53e-01_{1.5e-02}$	$1.52e-01_{1.3e-02}$	$1.55e-01_{1.7e-02}$	$1.52e-01_{1.3e-02}$	$1.53e-01_{1.5e-02}$	$1.54e-01_{1.5e-02}$
WFG9	$2.37e-01_{1.5e-03}$	$2.37e-01_{1.6e-03}$	$2.37e-01_{1.4e-03}$	$2.37e-01_{1.3e-03}$	$2.37e-01_{1.5e-03}$	$2.37e-01_{1.7e-03}$	$2.37e-01_{1.6e-03}$

ZDT1	$3.52e-01_{2.7e-02}$	$3.69e-01_{3.3e-02}$	$3.97e-01_{3.3e-02}$	$4.23e-01_{3.4e-02}$	$3.65e-01_{3.4e-02}$	$3.70e-01_{3.4e-02}$	$3.70e-01_{2.8e-02}$
ZDT2	$3.50e-01_{2.8e-02}$	$3.78e-01_{3.0e-02}$	$4.09e-01_{3.6e-02}$	$4.35e-01_{3.4e-02}$	$3.83e-01_{3.0e-02}$	$3.76e-01_{3.1e-02}$	$3.79e-01_{3.1e-02}$
ZDT3	$7.46e-01_{1.4e-02}$	$7.45e-01_{1.6e-02}$	$7.47e-01_{1.3e-02}$	$7.53e-01_{1.4e-02}$	$7.49e-01_{1.4e-02}$	$7.47e-01_{1.3e-02}$	$7.47e-01_{1.7e-02}$
ZDT4	$3.41e-01_{3.2e-02}$	$3.99e-01_{3.8e-02}$	$4.50e-01_{4.2e-02}$	$4.87e-01_{4.6e-02}$	$3.95e-01_{3.0e-02}$	$3.97e-01_{4.6e-02}$	$3.96e-01_{3.9e-02}$
ZDT6	$3.30e-01_{2.5e-02}$	$3.62e-01_{3.2e-02}$	$3.87e-01_{3.2e-02}$	$4.13e-01_{3.4e-02}$	$3.52e-01_{2.9e-02}$	$3.55e-01_{3.1e-02}$	$3.63e-01_{3.2e-02}$
DTLZ1	$8.59e-01_{1.5e-01}$	$8.94e-01_{1.8e-01}$	$9.35e-01_{2.3e-01}$	$9.35e-01_{1.7e-01}$	$8.75e-01_{1.6e-01}$	$9.23e-01_{2.3e-01}$	$9.03e-01_{1.6e-01}$
DTLZ2	$6.98e-01_{4.9e-02}$	$6.99e-01_{5.2e-02}$	$7.12e-01_{4.6e-02}$	$7.12e-01_{4.7e-02}$	$6.98e-01_{4.9e-02}$	$7.08e-01_{4.5e-02}$	$6.96e-01_{5.2e-02}$
DTLZ3	$9.41e-01_{1.3e-01}$	$1.04e+00_{1.3e-01}$	$1.18e+00_{1.3e-01}$	$1.23e+00_{1.5e-01}$	$1.03e+00_{1.2e-01}$	$1.05e+00_{1.2e-01}$	$1.06e+00_{1.2e-01}$
DTLZ4	$6.74e-01_{5.5e-02}$	$6.71e-01_{4.5e-02}$	$6.70e-01_{4.6e-02}$	$6.89e-01_{6.8e-02}$	$6.75e-01_{6.3e-02}$	$6.83e-01_{5.2e-02}$	$6.80e-01_{4.5e-02}$
DTLZ5	$4.29e-01_{5.4e-02}$	$4.45e-01_{5.3e-02}$	$4.65e-01_{4.9e-02}$	$4.95e-01_{4.6e-02}$	$4.48e-01_{4.8e-02}$	$4.49e-01_{4.3e-02}$	$4.38e-01_{4.4e-02}$
DTLZ6	$8.11e-01_{5.1e-02}$	$8.25e-01_{5.3e-02}$	$8.19e-01_{5.2e-02}$	$8.33e-01_{5.3e-02}$	$8.13e-01_{4.9e-02}$	$8.07e-01_{5.1e-02}$	$8.14e-01_{4.9e-02}$
DTLZ7	$7.44e-01_{4.6e-02}$	$7.44e-01_{4.8e-02}$	$7.43e-01_{5.7e-02}$	$7.55e-01_{5.5e-02}$	$7.44e-01_{5.1e-02}$	$7.38e-01_{5.4e-02}$	$7.40e-01_{5.5e-02}$
WFG1	$7.10e-01_{4.1e-02}$	$7.16e-01_{5.5e-02}$	$7.27e-01_{5.0e-02}$	$7.46e-01_{6.3e-02}$	$7.09e-01_{4.0e-02}$	$7.24e-01_{6.3e-02}$	$7.20e-01_{4.8e-02}$
WFG2	$7.85e-01_{1.1e-02}$	$7.90e-01_{1.1e-02}$	$7.93e-01_{1.7e-02}$	$7.95e-01_{1.9e-02}$	$7.88e-01_{1.1e-02}$	$7.88e-01_{1.2e-02}$	$7.90e-01_{1.1e-02}$
WFG3	$5.60e-01_{2.0e-02}$	$5.78e-01_{2.2e-02}$	$5.97e-01_{2.3e-02}$	$6.17e-01_{2.5e-02}$	$5.82e-01_{2.3e-02}$	$5.82e-01_{2.1e-02}$	$5.77e-01_{2.3e-02}$
WFG4	$3.65e-01_{2.4e-02}$	$3.86e-01_{2.8e-02}$	$4.08e-01_{3.0e-02}$	$4.23e-01_{3.1e-02}$	$3.90e-01_{2.8e-02}$	$3.82e-01_{2.7e-02}$	$3.81e-01_{3.3e-02}$
WFG5	$3.93e-01_{2.9e-02}$	$4.14e-01_{3.4e-02}$	$4.28e-01_{2.8e-02}$	$4.42e-01_{3.0e-02}$	$4.10e-01_{3.2e-02}$	$4.13e-01_{3.1e-02}$	$4.16e-01_{2.9e-02}$
WFG6	$3.62e-01_{2.8e-02}$	$3.83e-01_{3.3e-02}$	$4.06e-01_{3.3e-02}$	$4.26e-01_{3.6e-02}$	$3.84e-01_{3.6e-02}$	$3.86e-01_{3.0e-02}$	$3.86e-01_{2.9e-02}$
WFG7	$3.61e-01_{3.4e-02}$	$3.84e-01_{3.2e-02}$	$3.96e-01_{3.4e-02}$	$4.07e-01_{3.1e-02}$	$3.80e-01_{3.2e-02}$	$3.84e-01_{3.0e-02}$	$3.85e-01_{2.8e-02}$
WFG8	$6.12e-01_{5.1e-02}$	$6.44e-01_{4.6e-02}$	$6.74e-01_{4.3e-02}$	$6.73e-01_{3.9e-02}$	$6.50e-01_{4.3e-02}$	$6.45e-01_{4.1e-02}$	$6.43e-01_{4.5e-02}$
WFG9	$3.79e-01_{2.7e-02}$	$4.01e-01_{3.3e-02}$	$4.06e-01_{2.9e-02}$	$4.17e-01_{3.2e-02}$	$3.96e-01_{3.2e-02}$	$3.98e-01_{3.1e-02}$	$3.93e-01_{2.9e-02}$

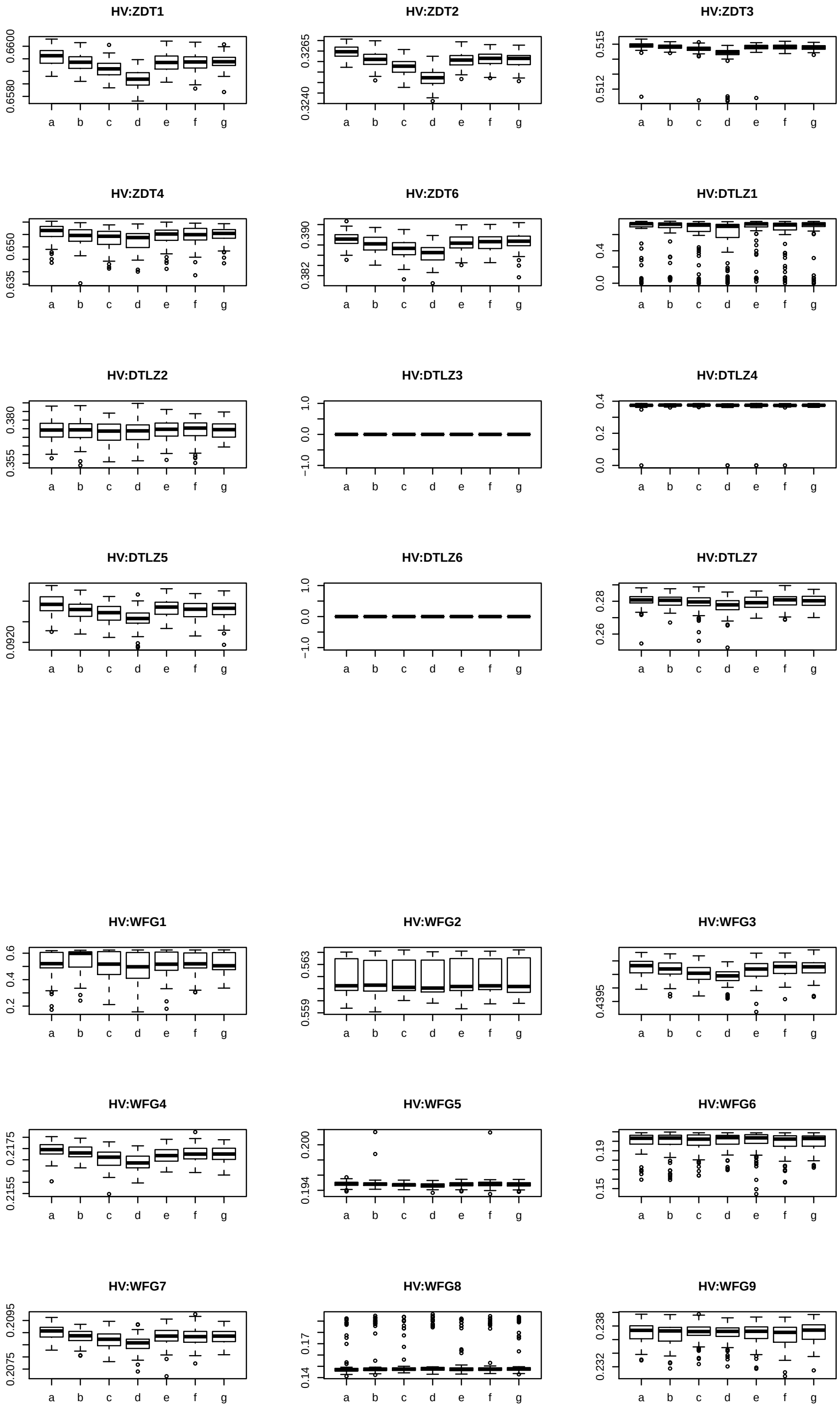

Fig. 2. The box plots for the hypervolume metric for ZDT, DTLZ and WFG test suites.

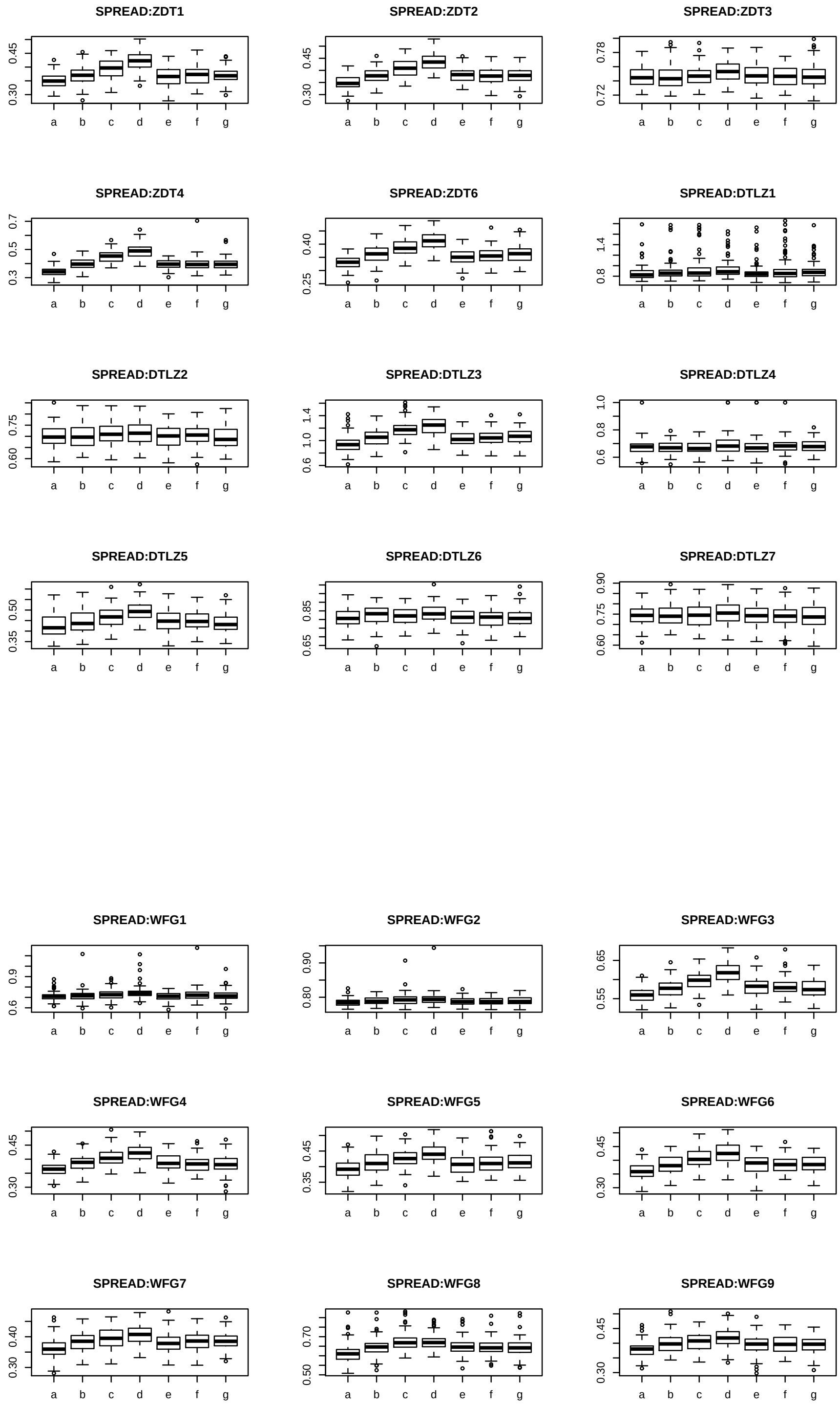

Fig. 3. The box plots for the spread metric for ZDT, DTLZ and WFG test suites.