Misleading Pareto Optimal Front Diversity Metrics: Spacing and Distribution

Christiaan Scheepers
Department of Computer Science
University of Pretoria
South Africa
Email: cscheepers@acm.org

Andries P. Engelbrecht
Department of Computer Science
University of Pretoria
South Africa
Email: engel@cs.up.ac.za

***Abstract*—The spacing metric by Schott and the distribution metric by Goh and Tan are often used to quantify the quality of the Pareto optimal front (POF) solution diversity. This paper presents a hypothesis that both the spacing and distribution metrics suffer from a pairwise grouping problem. This pairwise grouping problem leads to inaccurate measurement results that give a false indication of the POF solution diversity. In order to verify the hypothesis, a new diversity metric based on crowding distance is introduced. The vector evaluated particle swarm optimization (VEPSO) algorithm is used to evaluate the three POF solution diversity metrics. Results from the new diversity metric verify the presented hypothesis.**

I. Introduction

Many real life optimization problems consist of multiple objectives that need to be optimized. In many cases, the objectives are often in conflict with one another. These problems are called multi-objective optimization problems (MOPs). MOPs do not have a single solution, but rather have a set of solutions, referred to as the Pareto optimal front (POF). Multi-objective optimization (MOO) algorithms can be used to find sets of solutions, or POFs, for a MOP.

Since the introduction of VEGA [1] by Schaffer in 1995, various MOO algorithms, including NSGA-II [2], SPEA [3], PAES [4], MOEA/D [5], OMOPSO [6], SMPSO [7], and VEPSO [8], have been developed. Performance metrics can be used to determine which of these algorithms perform well for a given MOP. Performance metrics quantify how well a MOO algorithm solve a MOP. Various performance metrics have been developed to measure the quality of the found POFs. In general, a MOO algorithm has two goals [9]:

1) Find a set of solutions as close to the true POF as possible.
2) Find a set of solutions with as high a diversity as possible.

In more recent research a clear distinction is drawn between metrics that measure the diversity-, versus the uniformity of the solutions that make up the POF. Wang *et al.* [10] investigated diversity metrics for many-objective optimization problems, MOPs with more than three objectives, and classified the metrics as measuring diversity, uniformity, or both.

While newer metrics have been developed [10], spacing by Schott [11] and distribution by Goh and Tan [12] are often used [12]–[17] to quantify and compare the diversity of the solutions found by a MOO algorithm.

This paper presents a hypothesis that a weakness exists in the well-known distribution and spacing metrics that lead to inaccurate quantification of the POF diversity. A new diversity metric based on distribution and spacing, without the hypothesized flaw, is defined to evaluate the hypothesis. Experimental results verifying the hypothesis, using the vector evaluated genetic algorithm (VEGA) [1] inspired vector evaluated particle swarm optimization (VEPSO) algorithm [8], is presented and discussed.

The remainder of this paper is organized as follows. Section II describes the well-known distribution and spacing metrics. Section III presents a hypothesis that a weakness exists in the definitions of the distribution and spacing metrics. Section IV introduces the crowding distribution metric that is used to verify the aforementioned hypothesis. Section V discusses all the components that make up the VEPSO algorithm. Section VI presents the knowledge transfer strategies used in this study. Section VII discusses the four archive deletion approaches used in the experimental work presented in this paper. Section VIII describes the experimental procedure and test sets. Section IX presents an analysis and discussion of the results obtained from the experimental work. Finally, section X presents the findings and conclusions.

II. Pareto Optimal Front Diversity Metrics

Schott [11] introduced the spacing metric in 1995 to quantify the diversity of the POF. The spacing metric is formally defined as follows:

$$S = \sqrt{\frac{1}{|Q|-1}\sum_{i=1}^{|Q|}\left(\bar{d}-d_i\right)^2} \tag{1}$$

with

$$d_i = min_{\vec{x_j}\in Q \wedge j\neq i}\sum_{k=1}^{K}|x_{i,k}-x_{j,k}| \tag{2}$$

and

$$\bar{d} = \sum_{i=1}^{|Q|}\frac{d_i}{|Q|} \tag{3}$$

where Q is the set of solutions that make up the POF.

Goh and Tan [12] introduced the distribution metric in 2007, based on the spacing metric. The distribution metric is formally defined as follows:

$$D = \frac{1}{|Q|}\sqrt{\frac{1}{|Q|}\sum_{i=1}^{|Q|}\left(d_i - \bar{d}\right)^2} \tag{4}$$

with

$$\bar{d} = \frac{1}{|Q|}\sum_{j=1}^{|Q|} d_j \tag{5}$$

where d_i is the Euclidean distance in objective space between i and its nearest neighbor in Q.

Note that the calculations for both these metrics are very similar and the main difference is the switch from Manhattan distance, in S, to Euclidean distance, in D.

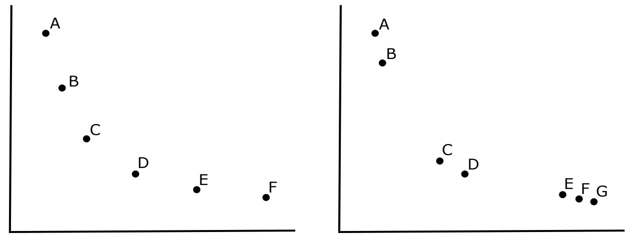
When the solutions are near uniformly spread, the resulting distribution, D, and spacing, S, metric values will be small.

III. Pairwise Hypothesis

A more detailed analysis of the distribution, D, and spacing, S, calculations reveals a potential weakness. Distribution calculates d_i as the Euclidean distance in objective space between solution i and its nearest neighbor in Q. Similarly, spacing calculates d_i as the Manhattan distance between solution i and its nearest neighbor in Q. In each case, the distance between solution i and its nearest neighbor in Q is used in the diversity calculation.

Using the nearest neighbor solution in objective space creates a pairwise combination problem where two solutions select each other during the diversity calculation. Fig. 1 depicts two example POFs for discussion purposes. Fig. 1(a) illustrates a well-spread POF with a nearly equal distribution of solutions. Potential nearest neighbor pairings for the first example are (A, B), (B, C), (D, E), and (E, F). Note that the (C, D) pairing is not used in any calculation as it is not the nearest neighbor pairing. In the second example illustrated in Fig. 1(b), the solutions that make up the POF are clustered. In this case, the nearest neighbor pairings would be (A, B), (C, D), (E, F), and (F, G). The diversity calculations will not take the larger distances between B and C or D and E into consideration, which in turn, would lead to misleading diversity measurement values. As long as the distances between the solutions that make up the nearest neighbor pairings are close to equal, the spacing and distribution measurement values would indicate a good diversity.

Adapting the distribution and spacing calculations to take more than one nearest neighbor into account would not solve the problem. In Fig. 1(b) a grouping of three solutions, (E, F, G), can easily be formed without taking the distance between D and E into account. Similarly, in actual POFs, solutions can be grouped into groups of three or more solutions that would still be susceptible to the same problem as the pairwise groupings; that is, not to take all the distances between solutions into account.



(a) Nearly equal distribution (b) Clustered distribution

Fig. 1. POF solution distribution examples

IV. Crowding Distance based Distribution

In order to verify the hypothesis that the nearest neighbor pairwise groupings affect the resulting distribution and spacing metric a new diversity metric is introduced. The new diversity metric is based on crowding distance and avoids the pairwise problem by using sorted sets to determine the next neighbor for the distance calculation. The crowding distribution is defined as follows:

$$C = \frac{1}{|Q|-1}\sum_{i=1}^{|Q|}|\bar{d} - d_i| \tag{6}$$

with

$$\bar{d} = \frac{1}{|Q|}\sum_{j=1}^{|Q|} d_j \tag{7}$$

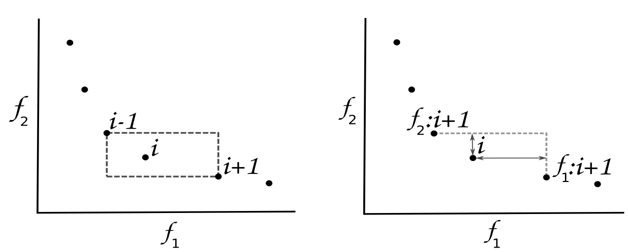
and

$$d_i = \sum_{k=1}^{K}\left(d_{i,k} - x_{i,k}\right) \tag{8}$$

where

$$d_{i,k} = \begin{cases} \min\{x_{j,k}\} & \text{if } \vec{x_j} \in Q : x_{j,k} > x_{i,k} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

and K is the number of objectives. d_i is calculated for solution $\vec{x_i} \in Q \subseteq \mathbb{R}^K$, similar to the crowding distance, as the sum of the difference between $x_{i,k}$ and $x_{i+1,k}$ where $x_{i+1,k}$ is the next solution in the set Q ordered by the k'th objective. Fig. 2 illustrates the crowding distance hypercube and the distances used by the crowding distribution calculation. $f_k : i{+}1$ denotes the next solution in the set Q ordered by the k'th objective.



(a) Crowding distance (b) Crowding distribution

Fig. 2. Crowding distance and distribution calculation hypercubes

The crowding distribution is not subject to the pairwise problem due to the use of the crowding distance based calculation of d_i.

It should be noted that the distance calculations used in the distribution and spacing calculations cannot be changed to use the next solution from a sorted, by objective function, set Q, as this approach will not scale to more than two objectives. The proposed crowding distribution metric can be used on problems with more than two objectives.

V. Vector Evaluated Particle Swarm Optimization

VEPSO [8], [18]–[20] is an extension of particle swarm optimization (PSO) [21] to deal with MOPs. Each objective of the MOP is represented and optimized by a separate swarm. The swarms exchange information through the use of a knowledge transfer strategy (KTS). The KTS is implemented by replacing the global guide, $\vec{\hat{y}}_i(t)$, in the PSO velocity update equation with a guide selected by the KTS. The VEPSO velocity update equation is formally defined as follows:

$$\begin{aligned}\vec{v}_i(t+1) =& w\vec{v}_i(t-1) + c_1\vec{r}_1(t)(\vec{y}_i(t) - \vec{x}_i(t)) \\ &+ c_2\vec{r}_2(t)(\vec{\hat{y}}_i(t) - \vec{x}_i(t))\end{aligned} \tag{10}$$

where $\vec{v}_i(t)$ is the velocity of particle i at iteration t, w is the inertia weight, c_1 and c_2 are acceleration constants, $\vec{r}_1(t)$ and $\vec{r}_2(t)$ are random vectors with components sampled uniformly from $(0, 1)$ at iteration t, $\vec{x}_i(t)$ is the position of particle i at iteration t, $\vec{y}_i(t)$ is the local (also referred to as the personal best position) guide of particle i at iteration t, and $\vec{\hat{y}}_i(t)$ is the global guide of particle i at iteration t.

At the end of each iteration, the non-dominated solutions are inserted into the archive if they are not dominated by any solution already in the archive. Solutions in the archive that are dominated by the newly inserted solutions are removed. Once the archive reaches the specified size limit, a deletion approach is used to remove solutions from the archive until the size limit criterion are satisfied again.

The VEPSO algorithm (for two objectives) can be summarized as follows:

Create and initialise all particles in swarms S_1 and S_2.
repeat
  for all swarms $S \in \{S_1, S_2\}$ **do**
    for all particles P_i in the swarm S **do**
      Select global guide, $\vec{\hat{y}}_i(t)$, for swarm S using a KTS.
      Update particle velocity using equation 10.
      Update particle position.
      Update the archive if the new particle position is non dominated.
    end for
  end for
until all swarms converge **or** iteration limit is reached.

VI. Knowledge Transfer Strategies

The experimental work presented in this paper made use of two KTSs. The first KTS, the *Random Personal Best* KTS, selects the global guide, $\vec{\hat{y}}_i(t)$, randomly from a randomly selected swarm [22]. The random KTS has been shown to perform well for a variety of problems [22].

The second KTS, the *Parent Centric Crossover Archive* (PCXA) KTS, calculates the global guide, $\vec{\hat{y}}_i(t)$, as the offspring of the parent centric crossover (PCX) operator [23] applied to three randomly selected non-dominated solutions from the archive [24]. The PCXA KTS have been shown to perform well in terms of POF solution diversity for a variety of problems [24].

VII. Archive Management

Four different archive deletion approaches were used in the experimental work presented in this paper.

The first deletion approach is the *Crowding Distance* based deletion approach. Deb *et al.* [2] introduced the crowding distance to estimate the density of solutions surrounding a particular solution. The crowding distance is calculated as the average distance to the solutions either side of the specified solution along each of the objectives. The solution with the lowest crowding distance is removed from the archive. Crowding distance based archives are used by the OMOPSO [6] and SMPSO [7] algorithms.

The second deletion approach is the *Distance Metric* [25] deletion approach. The distance metric deletion approach removes the solution with the lowest relative distance in the archive. The relative distance is calculated as follows:

$$f_{del,i} = \sum_{\forall j \neq i} \left(\sqrt{\sum_{k=1}^{K} \left(\frac{x_{i,k} - x_{j,k}}{max_k - min_k} \right)^2} \right)^{-1} \tag{11}$$

where $f_{del,i}$ is the relative distance for solution i in the archive, K is the number of objectives, $x_{i,k}$ and $x_{j,k}$ is the k'th objective function value for the i'th and j'th solutions in the archive respectively, max_k and min_k are the maximum and minimum values reached by an archive member for the k'th objective.

The third deletion approach is the *Nearest Neighbor* deletion approach [16]. The nearest neighbor deletion approach removes the solution with the lowest nearest neighbor distance. The nearest neighbor distance is calculated similarly to the distance metric, except, instead of using all the solutions, only the n nearest neighbors are used. Similar to [16], n is set to 2 for the experimental work presented in this paper.

Finally, the fourth deletion approach is the *Random* deletion approach. The random deletion approach randomly selects solutions to be removed from the archive.

VIII. Experimental Setup

The experimental work presented in this study analyzed the distribution, spacing, and crowding distribution metrics with each of the four archive deletion approaches described earlier with an archive size limit of 50. Each of the algorithms ran for 2000 iterations. Results were taken over 30 independent runs. Each of the subswarms had 50 particles. Known well-performing values for the inertia weight, $w = 0.729844$, and the acceleration constants, $c_1 = c_2 = 1.49618$, were used [26].

The Zitzler, Deb and Thiele (ZDT) [27] test set was used to provide a mix of challenges to evaluate the algorithms. Table I present a summary of the properties of each of the problems in the ZDT test set.

All algorithm and problem implementations were done and executed using the CIlib framework [28], [29].

IX. Analysis

Figs. 3(a) – 15(e) depict the measurement values for the three metrics, the number of solutions in the archive for ZDT1 through ZDT6 over 2000 iterations as well as a selection of the resulting POFs.

For VEPSO (Random) on ZDT1, ZDT2, and ZDT3, the crowding distribution, C, measurement values stopped decreasing around iteration 400 for the distance metric and random archive approach algorithms. The discontinuation in the decrease of the measurement values for C deviates from the measurement values for distribution, D, and spacing, S where a continued decrease can be noted. From Figs. 5(a) – 5(c) it can be noted that the archive limit is reached around iteration 400. This corresponds to where the C measurement values start to deviate from the D and S measurement values. Figs. 7(a) – 8(e) show the POFs obtained for iterations 100, 250, 500, 750, and 1000 for the VEPSO (Random) algorithm with the distance metric archive deletion and nearest neighbor archive deletion approaches. A large gap in the POF in Fig. 7(e) can be noted. The nearest neighbor deletion approach POF, in turn, shows a much better spread of solutions. The POFs clearly confirms that the C measurement values represent the actual spread of solutions whereas the D and S measurement values give a misleading indication of the actual spread of solutions.

The POFs shown in Figs. 9(a) – 10(e) confirm that a similar misleading indication between the actual spread and the D and S measurement values exists. In this case a degredation in the visible diversity of the solutions can be noted between the POF for iteration 500 shown in Fig. 10(c) and the POF at iteration 1000 shown in Fig. 10(e).

The ZDT4 results were erratic for all three diversity metrics. It should also be noted that none of the ZDT4 algorithms reached the archive size limit. The ZDT6 C results show that the distance metric and random archive deletion approach algorithms had a higher diversity than the crowding or nearest neighbor algorithms. The D and S results did not reveal the same trend. The POFs shown in Figs. 11(a) – 12(e) confirm that the D and S measurement values are misleading and that there is a notable difference in the spread of solutions as indicated by the C measurement values.

For VEPSO (PCXA) on ZDT1, the distance metric and random archive deletion approaches had higher C values than that of the crowding distance and nearest neighbor archive deletion approach algorithms. Again, this pattern does not match the D results where all the algorithms performed similarly. For ZDT1 the random archive deletion approach algorithm had an unstable S value as visible in Fig. 4(a). Similar to the ZDT1 result, the S values for ZDT2 were unstable for the random archive deletion approach. No notable difference in D values could be noted for the four algorithms. The distance metric and random archive deletion approach had slightly higher C values.

For ZDT3 the C measurement values again show varying performance for the four algorithms whereas the S and D measurement values showed no notable difference between the four algorithms. Figs. 13(a) – 15(e) clearly show that the algorithms achieved notably different spreads as reflected in the C measurement values. Fig. 14(e) is an excellent example of the pairwise grouping problem. Clusters of solutions are clearly visible and the corresponding S and D measurement values give no indication of the degraded spread of solutions.

Similar to the VEPSO (Random) results for ZDT4, the diversity metrics were erratic. For ZDT6, in contrast to the D and S results, the C values showed notable differences in the results achieved by the four algorithms.

Overall, the results indicate that the distribution, D, and spacing, S, measurement values were misleading whereas the crowding distribution, C, measurement values gave a more accurately indication of the actual spread of solutions on the POF. The results confirm the hypothesis that the distribution, D, and spacing, S, metrics are susceptible to a pairwise grouping problem where larger distances in the actual spread of solutions are ignored. This in turn leads to misleading measurement values.

TABLE I
Properties of the ZDT Problems

Name	Separability	Modality	Geometry
ZDT1	separable	unimodal	convex
ZDT2	separable	unimodal	concave
ZDT3	separable	unimodal/multimodal	disconnected
ZDT4	separable	unimodal/multimodal	convex
ZDT6	separable	multimodal	concave

X. Findings and Conclusions

This paper presented an analysis of the distribution and spacing metrics. A detailed analysis of the distribution and spacing calculations revealed a potential design flaw. It is hypothesized that this potential design flaw, the pairwise grouping problem, could lead to an inaccurate measurement of the POFs diversity. In order to test the hypothesis, a new diversity metric, crowding distribution, was introduced. The VEPSO algorithm was used to evaluate the performance of all three metrics. Two knowledge transfer strategies, namely random and PCXA, was used in conjunction with four archive deletion approaches.

The experimental results showed that the crowding distribution results did not exhibit the same behavior as the distribution and spacing results. This deviation in the results along with an analysis of the POFs confirms the hypothesis presented in this paper. The distribution and spacing metrics are unable to accurately measure the POF solution diversity. The newly introduced crowding distribution reported the diversity more

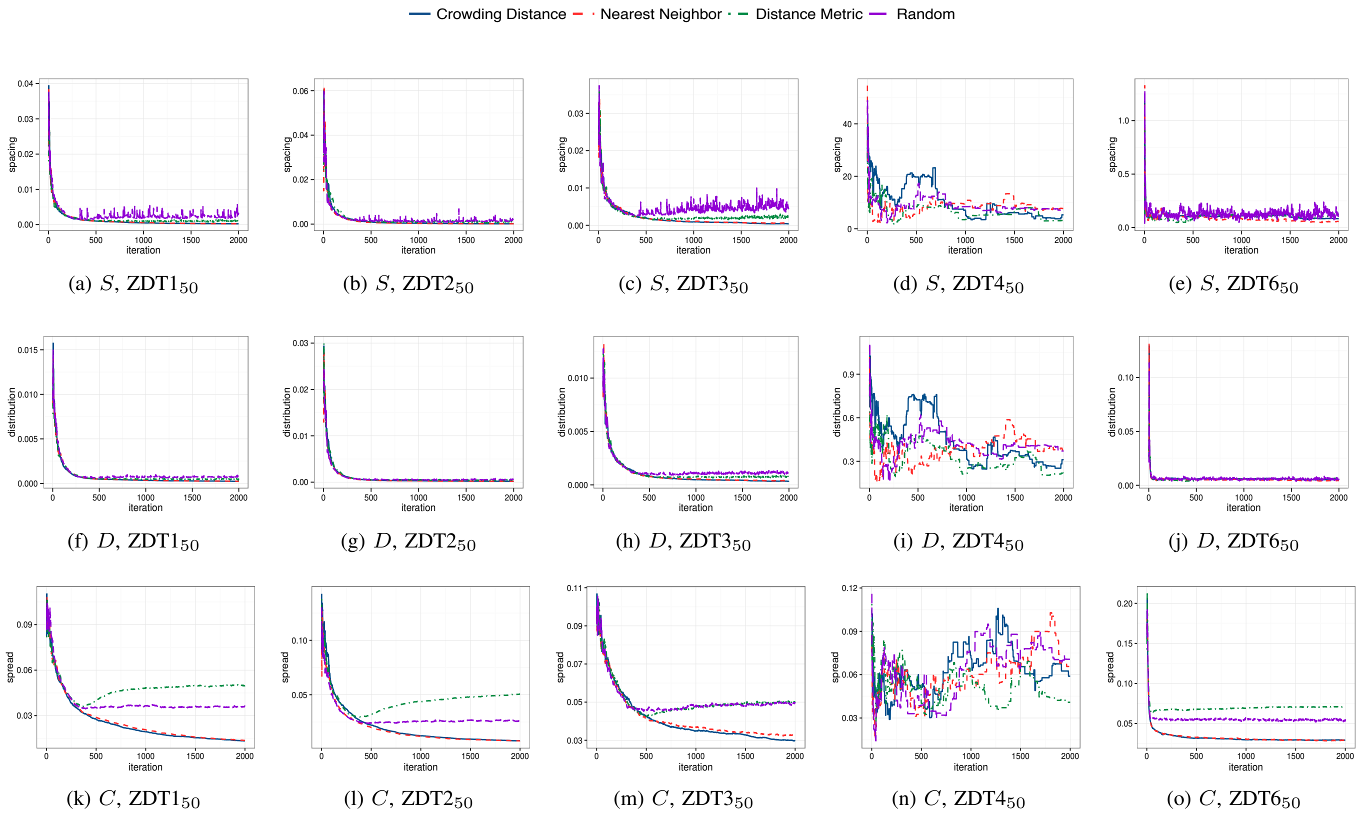


Fig. 3. VEPSO (Random) Spacing, S, Distribution, D, Crowding Distribution, C, with archive size 50 for ZDT1 through ZDT6

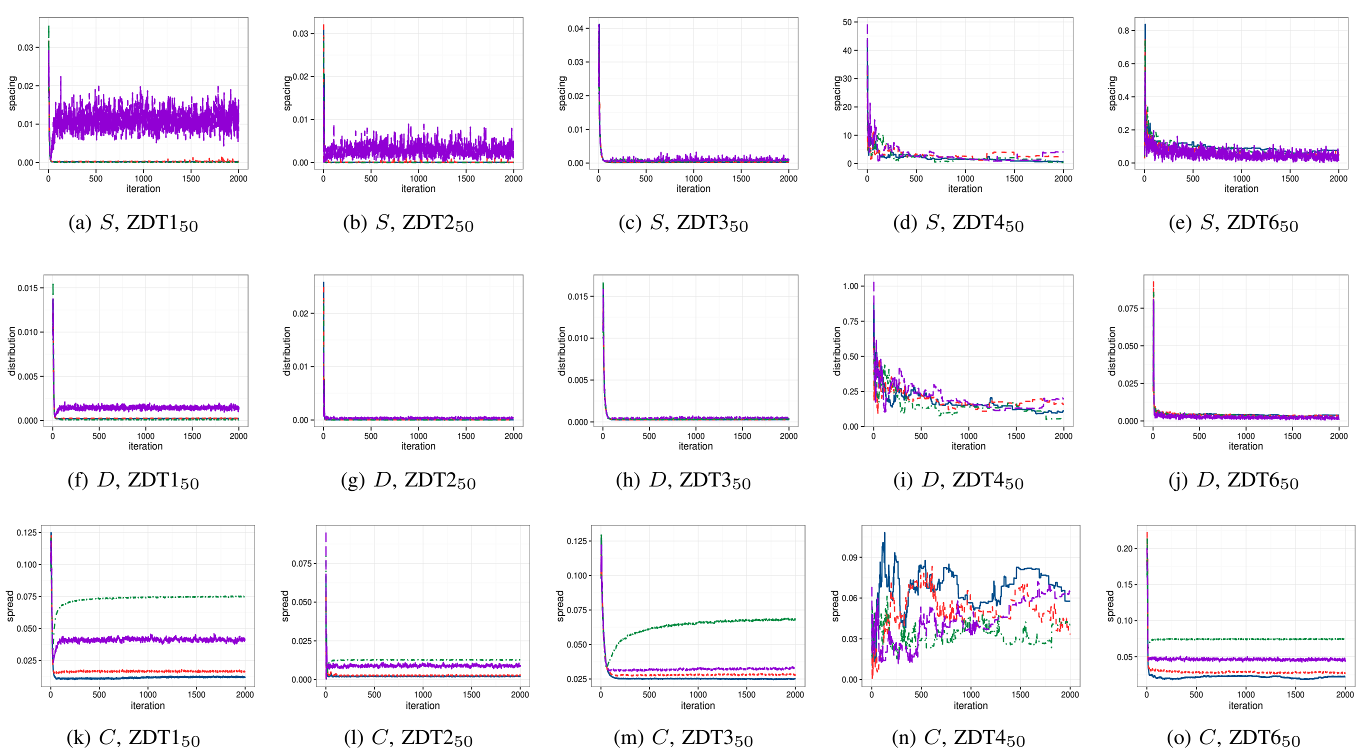
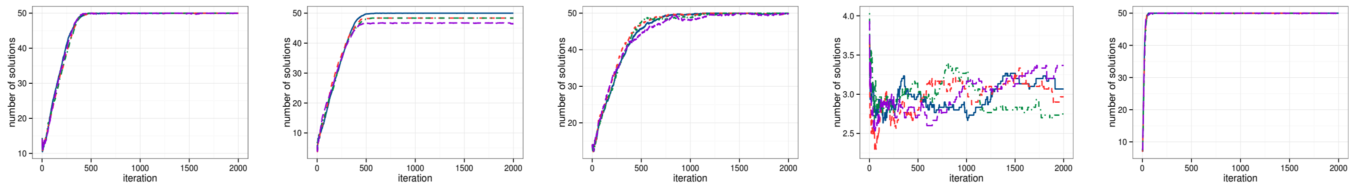
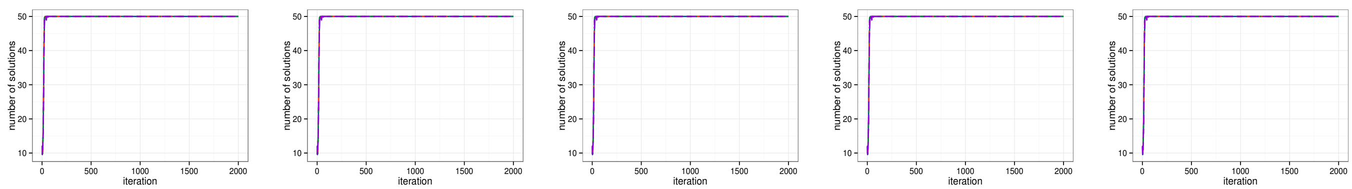


Fig. 4. VEPSO (PCXA) Spacing, S, Distribution, D, Crowding Distribution, C, with archive size 50 for ZDT1 through ZDT6



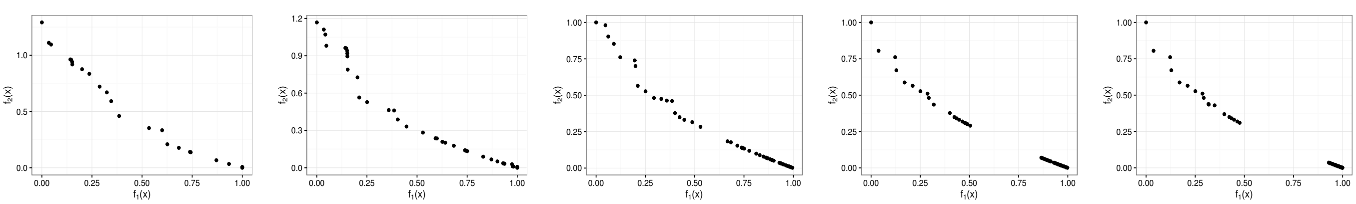
(a) ZDT1 (b) ZDT2 (c) ZDT3 (d) ZDT4 (e) ZDT6

Fig. 5. VEPSO (Random) number of solutions



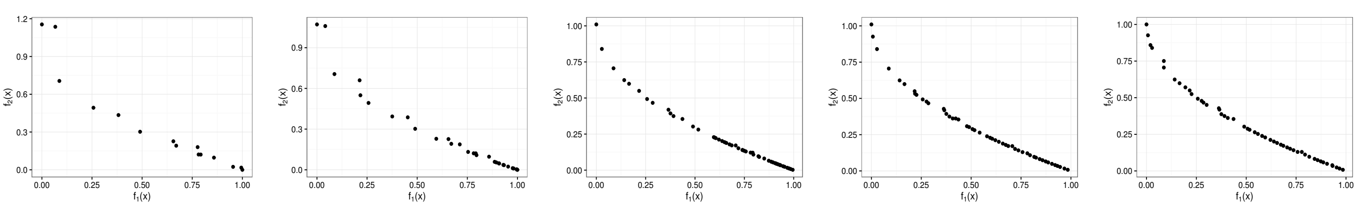
(a) ZDT1 (b) ZDT2 (c) ZDT3 (d) ZDT4 (e) ZDT6

Fig. 6. VEPSO (PCXA) number of solutions



(a) $t = 100$ (b) $t = 250$ (c) $t = 500$ (d) $t = 750$ (e) $t = 1000$

Fig. 7. VEPSO (Random) with distance metric archive deletion POF for ZDT1



(a) $t = 100$ (b) $t = 250$ (c) $t = 500$ (d) $t = 750$ (e) $t = 1000$

Fig. 8. VEPSO (Random) with nearest neighbor archive deletion POF for ZDT1

accurately and was nonsusceptible to the pairwise grouping problem.

From the experimental results presented in this paper, it can be concluded that using distribution and spacing, as presented in this paper, should be avoided and alternative metrics should be used instead.

The development of new performance metrics that more accurately measure the quality of the POF for comparisons between algorithms is currently underway.

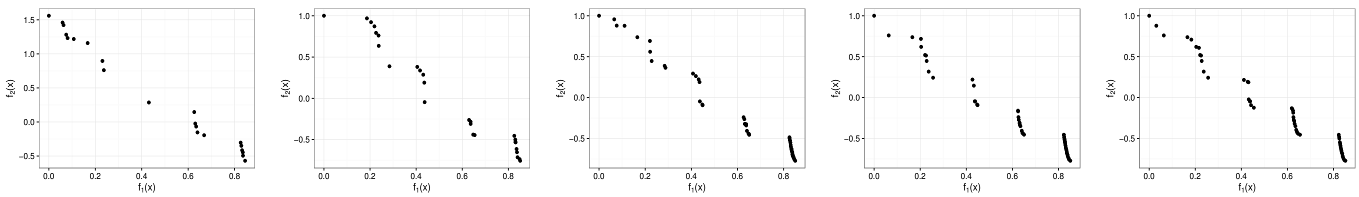
REFERENCES

[1] J. Schaffer, "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms," in *Proceedings of the 1st International Conference on Genetic Algorithms*, 1985, pp. 93–100.

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

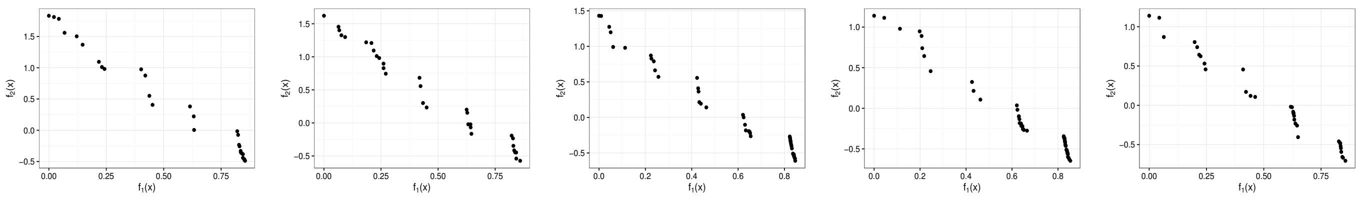
[3] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[4] J. D. Knowles and D. W. Corne, "The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Pareto Multiobjective Optimisation,"



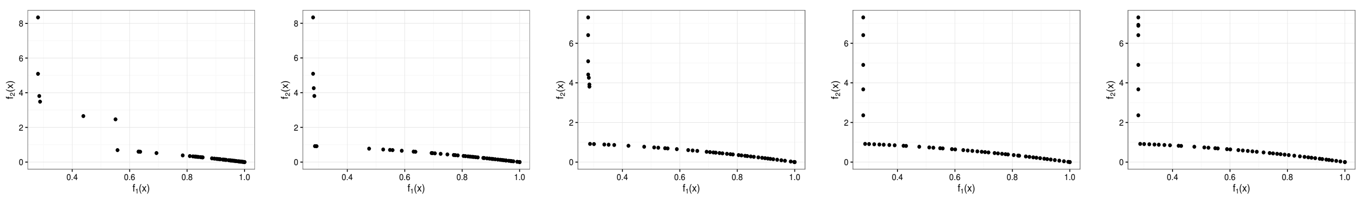
(a) $t = 100$ (b) $t = 250$ (c) $t = 500$ (d) $t = 750$ (e) $t = 1000$

Fig. 9. VEPSO (Random) with crowding distance archive deletion POF for ZDT3



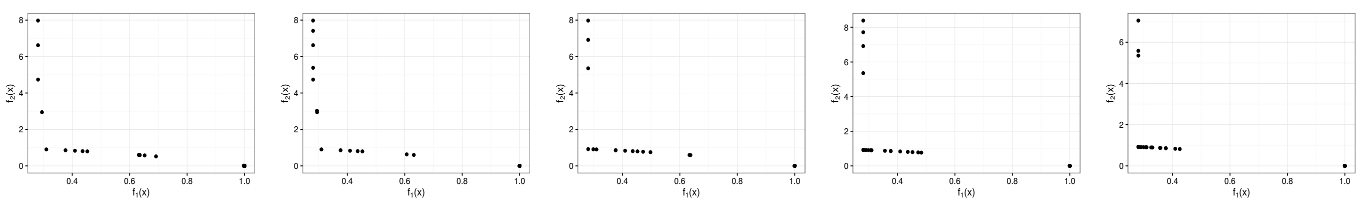
(a) $t = 100$ (b) $t = 250$ (c) $t = 500$ (d) $t = 750$ (e) $t = 1000$

Fig. 10. VEPSO (Random) with random archive deletion POF for ZDT3



(a) $t = 100$ (b) $t = 250$ (c) $t = 500$ (d) $t = 750$ (e) $t = 1000$

Fig. 11. VEPSO (Random) with crowding distance archive deletion POF for ZDT6



(a) $t = 100$ (b) $t = 250$ (c) $t = 500$ (d) $t = 750$ (e) $t = 1000$

Fig. 12. VEPSO (Random) with distance metric archive deletion POF for ZDT6

in *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 1, 1999.

[5] Q. Zhang and H. Li, “MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition,” *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[6] M. R. Sierra and C. A. Coello Coello, “Improving PSO-Based Multi-objective Optimization Using Crowding, Mutation and -Dominance,” in *Evolutionary Multi-Criterion Optimization: Third International Conference, EMO 2005, Guanajuato, Mexico, March 9-11, 2005. Proceedings*, C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, vol. 3410, pp. 505–519.

[7] A. Nebro, J. Durillo, J. García-Nieto, C. A. Coello Coello, F. Luna, and E. Alba, “SMPSO: A New PSO-based Metaheuristic for Multi-objective Optimization,” in *Proceedings of the IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*, no. 2, 2009, pp. 66–73.

[8] K. E. Parsopoulos and M. N. Vrahatis, “Recent approaches to global optimization problems through Particle Swarm Optimization,” *Natural Computing*, vol. 1, no. 2, pp. 235–306, 2002.

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

[10] H. Wang, Y. Jin, and X. Yao, “Diversity Assessment in Many-Objective Optimization,” *IEEE Transactions on Cybernetics*, vol. PP, no. 99, pp. 1–13, 2016.

[11] J. R. Schott, “Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization,” MSc Thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1995.

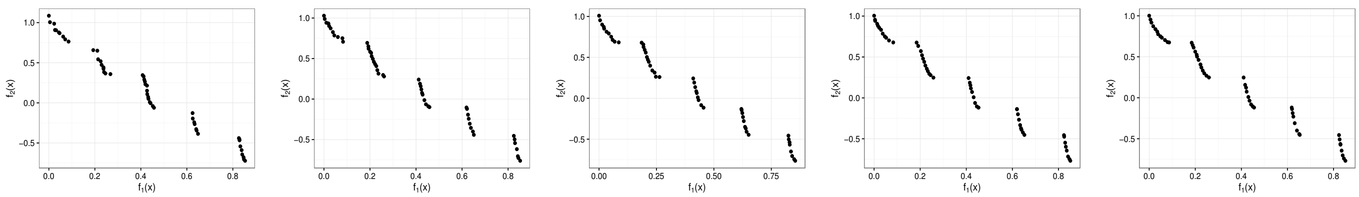
[12] C. K. Goh and K. C. Tan, “An Investigation on Noisy Environments in Evolutionary Multiobjective Optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 3, pp. 354–381, 2007.

[13] D. A. Van Veldhuizen and G. B. Lamont, “On Measuring Multiobjective Evolutionary Algorithm Performance,” in *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 1, 2000, pp. 204–211.

[14] C. A. Coello Coello, G. T. Pulido, and M. S. Lechuga, “Handling multiple objectives with particle swarm optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, 2004.

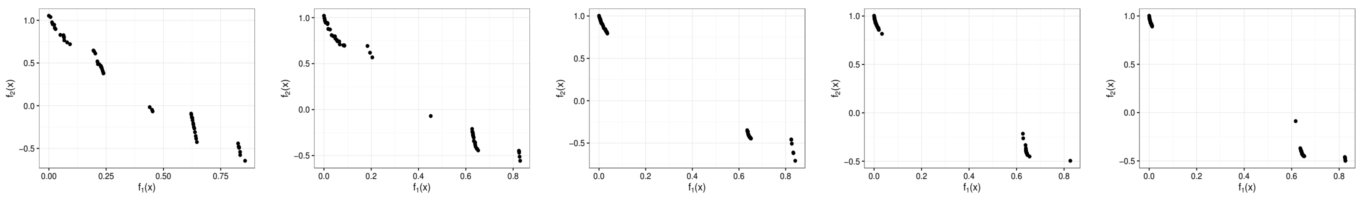
[15] M. Helbig and A. P. Engelbrecht, “Performance measures for dynamic multi-objective optimisation algorithms,” *Information Sciences*, vol. 250, pp. 61–81, 2013.

[16] K. R. Harrison, A. P. Engelbrecht, and B. M. Ombuki-Berman, “A Scalability Study of Multi-Objective Particle Swarm Optimizers,” in *Proceedings of the IEEE Congress on Evolutionary Computation*, 2013, pp. 189–197.



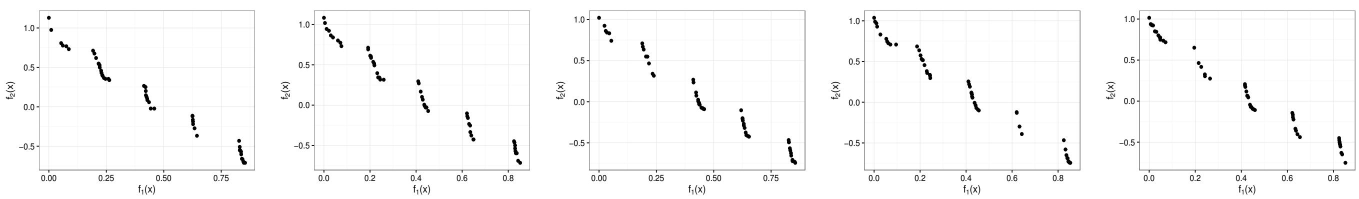
(a) $t = 100$ (b) $t = 250$ (c) $t = 500$ (d) $t = 750$ (e) $t = 1000$

Fig. 13. VEPSO (PCXA) with crowding distance archive deletion POF for ZDT3



(a) $t = 100$ (b) $t = 250$ (c) $t = 500$ (d) $t = 750$ (e) $t = 1000$

Fig. 14. VEPSO (PCXA) with distance metric archive deletion POF for ZDT3



(a) $t = 100$ (b) $t = 250$ (c) $t = 500$ (d) $t = 750$ (e) $t = 1000$

Fig. 15. VEPSO (PCXA) with random archive deletion POF for ZDT3

[17] J. Maltese, B. M. Ombuki-Berman, and A. P. Engelbrecht, “Co-operative Vector-Evaluated Particle Swarm Optimization for Multi-objective Optimization,” *Proceedings of the IEEE Symposium Series on Computational Intelligence*, pp. 1294–1301, 2015.

[18] K. E. Parsopoulos and M. N. Vrahatis, “Particle swarm optimization method in multiobjective problems,” in *Proceedings of the ACM Symposium on Applied Computing*, 2002, pp. 603–607.

[19] K. E. Parsopoulos, D. Tasoulis, N. Pavlidis, V. Plagianakos, and M. N. Vrahatis, “Vector Evaluated Differential Evolution for Multiobjective Optimization,” in *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 1, 2004, pp. 204–211.

[20] K. E. Parsopoulos, D. Tasoulis, and M. N. Vrahatis, “Multiobjective Optimization using Parallel Vector Evaluated Particle Swarm Optimization,” in *Proceedings of the IASTED International Conference on Artificial Intelligence and Applications*, vol. 2, 2004, pp. 823–828.

[21] J. Kennedy and R. C. Eberhart, “Particle Swarm Optimization,” in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.

[22] J. Grobler, “Particle swarm optimization and differential evolution for multi objective multiple machine scheduling,” Master’s thesis, University of Pretoria, 2009.

[23] K. Deb, A. Anand, and D. Joshi, “A Computationally Efficient Evolutionary Algorithm for Real-Parameter Optimization,” *Evolutionary Computation*, vol. 10, no. 4, pp. 371–395, 2002.

[24] K. R. Harrison, B. M. Ombuki-Berman, and A. P. Engelbrecht, “Knowledge transfer strategies for vector evaluated particle swarm optimization,” in *International Conference on Evolutionary Multi-Criterion Optimization*, 2013, pp. 171–184.

[25] T. Bartz-Beielstein, P. Limbourg, J. Mehnen, K. Schmitt, K. E. Parsopoulos, and M. N. Vrahatis, “Particle Swarm Optimization for Pareto Optmization with Enhanced Archiving Techniques,” in *Proceedings of the IEEE Congress on Evolutionary Computation*, 2003, pp. 1780–1787.

[26] R. Eberhart and Y. Shi, “Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization,” in *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 1, no. 2, 2000, pp. 84–88.

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

[28] G. Pampara, A. Engelbrecht, and T. Cloete, “CIlib: A collaborative framework for Computational Intelligence algorithms-Part I,” in *Proceedings of the IEEE International Joint Conference on Neural Networks*, 2008, pp. 1750–1757.

[29] ——, “CIlib: A collaborative framework for Computational Intelligence algorithms-Part II,” in *Proceedings of the IEEE International Joint Conference on Neural Networks*, 2008, pp. 1764–1773.