Padding the Dimensions for Knowledge Transfer in the Dynamic Vector Evaluated Particle Swarm Optimisation Algorithm

Mardé Helbig
Department of Computer Science
University of Pretoria
Hatfield, South Africa
Email: mhelbig@cs.up.ac.za

***Abstract*—Most real world problems have more than one objective, with at least two objectives in conflict with one another and at least one objective that is dynamic in nature. The dynamic vector evaluated particle swarm optimisation (DVEPSO) algorithm is a co-operative algorithm, where each sub-swarm solves only one objective function and therefore, each sub-swarm optimises only a sub-set of decision variables. Knowledge is shared amongst the sub-swarms when the particles' velocity is updated, by using the position of the global guide of the sub-swarm or of another sub-swarm. The global guide can only provide information about the decision variables that are applicable to the objective function that its sub-swarm is optimising. Therefore, padding is required for the other decision variables. This paper investigates various padding approaches, namely using the sub-swarm's global best, using the personal best (pbest) of another particle in the sub-swarm, using the global best (gbest) of another sub-swarm or performing parent-centric crossover on another particle's position, pbest and gbest. Results indicate that using a random gbest or pbest performed well in fast changing environments, and using the sub-swarm's gbest performed well in slowly changing environments.**

I. Introduction

Optimisation problems with more than one objective, with at least two objectives in conflict, and where the environment changes over time are referred to as dynamic multi-objective optimisation problems (DMOOPs). This paper focusses on boundary constrained DMOOPs. Due to the conflicting objectives a single solution does not exist. Therefore, a dynamic multi-objective optimisation algorithm (DMOA) has to find a set of trade-off solutions for each environment.

Greeff and Engelbrecht [6] proposed a co-operative particle swarm optimisation (PSO) algorithm to solve DMOOPs. Each sub-swarm only optimises one objective and then knowledge is shared amongst the sub-swarms. The knowledge sharing occurs through the velocity update of a sub-swarm's particles by using the global guide of either the particle's own sub-swarm or of another sub-swarm. If the global guide is from another sub-swarm, it can only provide information or knowledge about the decision variables that its sub-swarm is optising, i.e. the decision variables that are applicable to the objective function that the sub-swarm is optimising. Therefore, for the velocity update, the selected global guide may not contain information for all dimensions (decision variables) that is required by the particles. Padding should then be applied to the missing dimensions. Previous versions of dynamic vector evaluated particle swarm optimisation (DVEPSO) used the sub-swarm's global best (gbest) for padding. However, this approach may limited the diversity of the sub-swarm. Therefore, this paper investigates four padding approaches, namely using:

- the sub-swarm's global best
- the personal best (pbest) of another particle in the same sub-swarm
- the gbest of another sub-swarm
- the resulting particle obtained from performing parent-centric crossover (PCX) on another particle's position, pbest, and its sub-swarm's gbest.

The rest of the paper's layout is as follows: Section II provides background information on dynamic multi-objective optimisation (DMOO). The DVEPSO algorithm and padding required for knowledge transfer is discussed in Section III. Section IV discusses the experimental setup, such as the algorithm configurations, bechmark functions, performance measures and statistical analysis used for this study. The results obtained from the experiments are discussed in Section V. Finally, conclusions are drawn in Section VI.

II. Background

This sections provides information on DMOO that is required for the rest of the paper.

Let the n_x-dimensional decision space be represented by $S \subseteq \mathbb{R}^{n_x}$. Let the feasible space be represented by $F \subseteq S$, where $F = S$ for boundary constrained optimisation problems. Let $\mathbf{x} = (x_1, x_2, \dots, x_{n_x}) \in S$ represent a decision vector and let $f_k\colon \mathbb{R}^{n_x} \to \mathbb{R}$ define a single objective function. Let an objective vector containing n_k objective function evaluations be represented by $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_{n_k}(\mathbf{x})) \in O \subseteq \mathbb{R}^{n_k}$, with O representing the objective space. Then a boundary constrained DMOOP is defined as:

$$
\begin{array}{ll}
minimise & \mathbf{f}(\mathbf{x}, \mathbf{W}(t)) \\
subject\ to & \mathbf{x} \in [\mathbf{x}_{min}\,, \mathbf{x}_{max}]^{n_x}
\end{array} \tag{1}
$$

where $\mathbf{W}(t)$ is a matrix of time-dependent control parameters of an objective function at time t and n_x is the number of decision variables. The boundary constraints are referred to as $\mathbf{x} = (x_1, \ldots, x_{n_x}) \in \mathbb{R}^{n_x}$ and $[\mathbf{x}_{min}\,, \mathbf{x}_{max}]^{n_x}$.

The quality of two DMOO solutions is compared using vector domination, defined as:

Definition 1. Vector Domination: Let f_k be an objective function. Then, a decision vector $\mathbf{x}_1$ dominates another decision vector $\mathbf{x}_2$, denoted by $\mathbf{x}_1 \prec \mathbf{x}_2$, if and only if

- $f_k(\mathbf{x}_1) \leq f_k(\mathbf{x}_2),\ \forall k = 1, \ldots, n_k$; and
- $\exists i = 1, \ldots, n_k \colon f_i(\mathbf{x}_1) < f_i(\mathbf{x}_2)$.

The best decision vectors is referred to as being Pareto-optimal, defined as follows:

Definition 2. Pareto-optimal: A decision vector $\mathbf{x}^*$ is Pareto-optimal if

$$
\nexists k \colon f_k(\mathbf{x}) \prec f_k(\mathbf{x}^*) \tag{2}
$$

The set of all Pareto-optimal decision vectors is referred to as the Pareto-optimal set (POS). If the decision vector $\mathbf{x}^*$ is Pareto-optimal, the objective vector, $\mathbf{f}(\mathbf{x}^*)$, is also Pareto-optimal. The set of objective vectors that correspond to the decision vectors in the POS is referred to as the Pareto-optimal front (POF). When solving a DMOOP, the goal of a DMOA is to track the POF over time and to find for each environment a diverse set of solutions.

III. Padding for Knowledge Sharing

This section discusses the DVEPSO algorithm and the padding of DVEPSO's particle's for knowledge sharing.

A. Dynamic Vector Evaluated Particle Swarm Optimisation Algorithm

Parsopoulos *et al.* [16] introduced a co-operative PSO-based algorithm called the vector evaluated particle swarm optimisation (VEPSO) algorithm. It was extended for DMOO by Greeff and Engelbrecht [6], referred to as DVEPSO. The number of DVEPSO's sub-swarms is equal to the number of the DMOOP's objectives that the algorithm is optimising. Each sub-swarm optimises only one objective function and knowledge of its best solutions is shared with the other sub-swarms. This shared knowledge is contained in the global guide and is used to update the velocity of the particles. A knowledge sharing strategy is used to select a sub-swarm, as well as which particle of the selected swarm will be used for knowledge sharing, i.e. in the velocity update of the particles.

The following default configuration of DVEPSO is used in this study [8], [14]:

- Each objective function is optimised by a global best PSO.
- A particle's new position is selected as the particle's new personal best (pbest) if the particle's new position leads to a better objective function value than its current pbest. Only the objective function being optimised by the sub-swarm is taken into consideration when these two positions are compared with one anther, i.e. Pareto-dominance is not used.
- A particle's position is selected as the new gbest of the sub-swarm if the particle's new position dominates the current gbest. However, if a particle's new position is non-dominated with regards to the sub-swarm's current gbest, one of these two positions is randomly selected as the sub-swarm's new gbest.
- A specified number of particles, referred to as sentry particles [2], are randomly selected and re-evaluated after the algorithm performed the specific iteration, but before the next iteration starts. If after re-evaluation any sentry particle's fitness value differs with more than a specificed value, the sub-swarm is notified that a change in the environment has occurred.
- If a change in the environment has occurred, 30% of the particles of the sub-swarm whose objective function has changed is randomly re-initialised [8]. The archive's non-dominated solutions are re-evaluated and solutions that have become dominated are removed from the archive.
- A random knowledge sharing topology is used [7], where the sub-swarm selected for knowledge sharing can be another sub-swarm or the sub-swarm itself. Therefore, for some iterations a sub-swarm may end up using its own global guide to update its particles' velocity. The global guide of the selected swarm is chosen using tournament selection.
- If the archive is full, a solution is removed from a crowded region in the archive.

B. Padding Additional Dimensions

Knowledge is shared amongst the sub-swarms of DVEPSO when the velocities of a sub-swarm's particles are updated. The random knowledge sharing strategy may result in the global guide being selected from a sub-swarm whose particles have a different dimension. In these cases, the global guide can only provide information about the decision variables that are applicable to the objective function that is being optimised by the selected global guide's sub-swarm. Therefore, padding is required for the other decision variables (dimensions).

This process of padding is illustrated in Figure 1. In Figure 1, DVEPSO is used to solve a two-objective DMOOP, with objectives f_1 and f_2. f_1 has a dimension of one and f_2 has a dimension of n. Since each sub-swarm only solves one objective, the particles of S_1 will have a dimension of one, and the particles of S_2 will have a dimension of n. If a particle of S_1 is selected to share knowledge with S_2, the particle will be used to update the velocity of the particles in S_2. However, only one dimension of the particles of S_1's position is in common with particles of S_2's position. Knowledge about additional dimensions (the uncommon dimensions) are required before the selected particle (global guide) can be used to update the velocity of S_2's particles. Therefore, padding is required to obtain the missing dimensions.

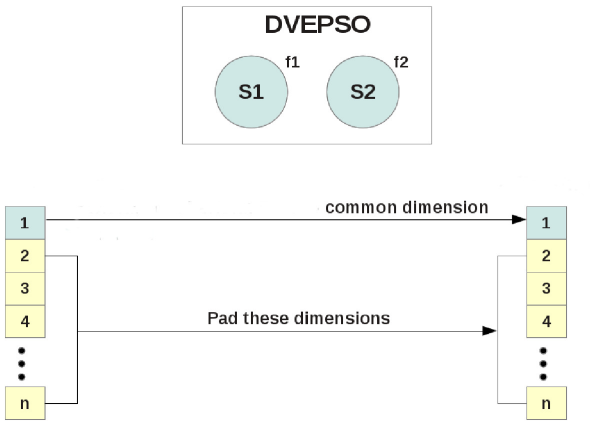


Fig. 1: Applying padding to share knowledge between the swarms of DVEPSO

In this paper, the following four approaches are investigated for the padding (explained with reference to Figure 1):

- Using the gbest of the same sub-swarm ($S2$) to pad the missing dimensions (referred to as P_g in the rest of the paper).
- Firstly selecting another sub-swarm according to the knowledge transfer strategy. Once a sub-swarm is found that contains particles with more dimensions:
  - A pbest of the selected sub-swarm is randomly chosen to pad the missing dimensions (referred to as P_{rp} in the rest of the paper).
  - The gbest of the selected sub-swarm is chosen to pad the missing dimensions (referred to as P_{rg} in the rest of the paper).
  - A particle of the chosen sub-swarm is randomly chosen. Parent-centric crossover (PBX) is then performed on the selected particle's position, its pbest and the selected sub-swarm's gbest (referred to as P_c in the rest of the paper).

IV. Experimental Setup

This section discusses the experimental setup used for this study. Section IV-A discusses the algorithms that were used for this study. The benchmark functions and performance measures are discussed in Sections IV-B and IV-C respectively. The statistical analysis that were performed on the obtained data is discussed in Section IV-D.

A. Algorithms

The default DVEPSO configuration as discussed in Section III-A is used in this study. In addition, the padding approaches discussed in Section III-B are used to pad the missing dimensions when required. Therefore, the performance of four DVEPSO configurations are compared in this paper, namely P_g, P_{rg}, P_{rp} and P_c (refer to Section III-B).

For this study, each algorithm configuration was executed for 30 independent runs on each benchmark function and for each environment. Each run had 20 environment changes, i.e. if a change occurred every 10 iterations, each run had 200 iterations.

B. Benchmark Functions

According to a comprehensive analysis of DMOOPs in [13], seven benchmark functions with various characteristics were selected to evaluate the DVEPSO configurations on. These functions are DIMP2 [15], dMOP2 [5], dMOP2$_{dec}$ [11], HE2 [9], HE7 and HE9 [13], [8], FDA4 [3], FDA5 [3] and FDA5$_{dec}$ [11]. For each benchmark function the following severity of change (n_t) and frequency of change (τ_t) combinations were used: $n_t = 10$ and $\tau_t = 10$ (a fast changing environment), $n_t = 10$ and $\tau_t = 25$ (a slow changing environment), $n_t = 10$ and $\tau_t = 50$ (a very slow changing environment), $n_t = 1$ and $\tau_t = 10$ (a severely changing environment) and $n_t = 20$ and $\tau_t = 10$ (a gradually changing environment).

A modified version of DIMP2 with a concave POF (referred to as DIMP2 in the rest of the paper) is used in this study. Each decision variable of DIMP2 has its own rate of change. dMOP2's POF changes from convex to concave and vice versa over time. However, with dMOP2$_{dec}$, in addition to the characteristics of dMOP2's POF, its POF is deceptive, i.e. there is at least one local POF and the search space favours the local POF and not the global POF. HE2 have a disconnected POF, i.e. the POF consists of disconnected continuous pieces. HE7 has a POF that changes from convex to concave and vice versa over time, a non-linear POS, and each decision variable has a different POS. All of the functions discussed so far are 2-objective functions. FDA4, FDA5 and FDA5$_{dec}$ are 3-objective functions with a non-linear POF. In addition, the spread of solutions in the POF of FDA5 and FDA5$_{dec}$ changes over time and FDA5$_{dec}$'s POF is deceptive.

C. Performance Measures

Based on an analysis of performance measures in [12], two performance measures were selected for this study. The first measure is the alternative accuracy measure (acc_{alt}) [1], referred to in this article as acc (a low acc value indicates good performance). acc is measured as the absolute difference between the hypervolume (HV) of the true or optimal POF and the hypervolume of the approximated (found) POF at a specific time t. The second measure, stability ($stab$) [1], quantifies the effect that changes in the environment have on acc of the DMOA. It measures the difference in acc values for consecutive environments and a low $stab$ value indicates good performance.

These two measures require the hypervolume value, which was calculated with the source code of Fonseca *et al* [4]. For the hypervolume calculations, the reference vector was calculated for each benchmark function, consisting of the worst objective function value for each dimension.

D. Statistical Analysis

For each DMOOP, environment, and performance measure wins and losses were calculated as proposed in [10]. For each time step just before a change in the environment occurred, the average performance measure value over 30 runs was calculated. A Kruskal-Wallis test was performed on these average values obtained by the DVEPSO configurations. A pair-wise Mann-Whitney U test was performed for each pair of DVEPSO configurations if the Kruskal-Wallis test indicated a statistical significant difference between these performance measure values. If the Mann-Whitney U test indicated a statistical significant difference, wins and losses were awarded as follows:

1) At each time step just before a change in the environment occurred, the average performance measure values of the two DMOAs were compared.
2) For each environment, the DMOA with the best performance measure value was awarded a win and the other DMOA was awarded a loss.
3) In order to ensure that a DMOA's performance on a specific DMOOP did not lead to skewed results, the number of wins and losses were normalised for each DMOOP.

For all statistical tests a confidence level of 95% was used.

V. Results

This section discusses the results of the experiments. Section V-A discusses the performance of the algorithms with regards to the two measures. The performance of the algorithms in each environment is discussed in Section V-B. A discussion of the results, as well as observations that were made, are presented in Section V-D.

A. Performance Per Measure

The wins and losses that were obtained per measure, over all benchmarks and all environments, are discussed in this section. Table I presents the wins and losses obtained for acc. For acc, P_{rg} performed the best, obtaining 6.9 more wins than losses. The second best performing configuration was P_{rp}, being awarded 1.3 more wins than losses. Both P_c and P_g performed much worse, obtaining more losses than wins. No statistical significant difference was observed for $stab$.

TABLE I: Overall wins and losses for acc obtained by DVEPSO using various padding approaches

PM	**Results**	**Padding Approaches**			
		P_c	P_g	P_{rg}	P_{rp}
acc	Wins	7.95	9.45	13.45	13.15
acc	Losses	13.05	12.55	6.55	11.85
acc	Diff	-5.1	-3.1	6.9	1.3
acc	Rank	4.0	3.0	**1.0**	2.0

B. Performance Per Environment

Table II presents the wins and losses over all benchmark functions and all performance measures for each of the environments. No padding approach won in more than one type of environment and each padding approach came last in at least one environment. P_g obtained the most wins in two environments, but also the least wins (worst rank) in two other environments. Both random appraoches, P_{rg} and P_{rp} performed well, obtaining a top three rank in 4 environments and obtaining the worst rank in only one environment. These two appraoches performed well in fast changing environments and P_g performed well in slower changing environments.

TABLE II: Overall wins and losses per environment for DVEPSO using various padding approaches

n_t-τ_t	**Results**	**Padding Approaches**			
		P_c	P_g	P_{rg}	P_{rp}
10-10	Wins	1.1	0.2	6.9	2.8
10-10	Losses	3.9	5.8	0.1	1.2
10-10	Diff	-2.8	-5.6	6.8	1.6
10-10	Rank	3.0	4.0	**1.0**	2.0
10-25	Wins	1.95	3.0	1.0	1.05
10-25	Losses	3.05	0.0	1.0	2.95
10-25	Diff	-1.1	3.0	0.0	-1.9
10-25	Rank	3.0	**1.0**	2.0	4.0
10-50	Wins	0.3	2.5	0.55	3.65
10-50	Losses	1.7	1.5	2.45	1.35
10-50	Diff	-1.4	1.0	-1.9	2.3
10-50	Rank	3.0	2.0	4.0	**1.0**
20-10	Wins	5.4	0.9	3.45	3.1
20-10	Losses	2.3	4.1	1.95	4.5
20-10	Diff	3.1	-3.2	1.5	-1.4
20-10	Rank	**1.0**	4.0	2.0	3.0
1-10	Wins	1.9	2.85	1.95	3.15
1-10	Losses	4.8	1.15	1.45	2.45
1-10	Diff	-2.9	1.7	0.5	0.7
1-10	Rank	4.0	**1.0**	3.0	2.0

C. Overall Performance

The wins and losses over all benchmark functions, all performance measures and all environments are presented in Table III. P_{rg} performed the best, obtaining 6.9 more wins than losses. Even though P_{rp} obtained only slightly less wins than P_{rg}, it obtained more losses than P_{rg}. These two random approaches outperformed the other two approaches by being the only two approaches that were awarded more wins than losses.

TABLE III: Overall wins and losses for DVEPSO using various padding approaches

n_t-τ_t	**Results**	**Padding Approaches**			
		P_c	P_g	P_{rg}	P_{rp}
all	Wins	10.65	9.45	13.85	13.75
all	Losses	15.75	12.55	6.95	12.45
all	Diff	-5.1	-3.1	6.9	1.3
all	Rank	4.0	3.0	**1.0**	2.0

D. Discussion of Results

A statistical significant difference in results was only observed for DIMP2 and HEF5. Figures 2 and 3 present the approximated POFs found by the padding approaches for HE7 and DIMP2 with n_t = 1-τ_t = 10 over all 30 runs. Figure 3 illustrates the existance of outliers in the approximated POFs. In fast changing environments outliers will exist in the approximated POFs, since the algorithm does not find a solution that dominates the outlier in number of iterations that are available before the environment changes. However, it can be seen that the outliers only exist in the first few environments. Since DIMP2 is a type I DMOOP, its POS changes over time but the POF remains static. Therefore, the outliers are eliminated in later environments.

The wins and losses for DIMP2 and HE7 are presented in Tables IV and V respectively. P_{rg} performed the best for DIMP2 and was also the only approach that was awarded more wins than losses. For HE7, P_{rp} outperformed the other approaches, obtaining 6.6 more wins than losses. The average acc values obtained for DIMP2 and HE7 (with n_t = 1-τ_t = 10) by the padding approaches for each environment are presented in Figure 4. From Figure 4 it can be seen that P_g consistently obtained the lowest acc values for DIMP2. Similarly, P_{rp} consistently obtained the best acc values for HE7.

TABLE IV: Overall wins and losses for DVEPSO using various padding approaches for DIMP2

n_t-τ_t	**Results**	**Padding Approaches**			
		P_c	P_g	P_{rg}	P_{rp}
all	Wins	0.1	0.9	2.0	0.0
all	Losses	1.9	1.1	0.0	0.0
all	Diff	-1.8	-0.2	2.0	0.0
all	Rank	4.0	3.0	**1.0**	2.0

TABLE V: Overall wins and losses for DVEPSO using various padding approaches for HEF5

n_t-τ_t	**Results**	**Padding Approaches**			
		P_c	P_g	P_{rg}	P_{rp}
all	Wins	6.05	1.2	3.75	10.7
all	Losses	5.65	6.8	5.15	4.1
all	Diff	0.4	-5.6	-1.4	6.6
all	Rank	2.0	4.0	3.0	**1.0**

Even though no statistical significant difference were observed for the other functions, some observations can be made from the results. The acc values for each environment obtained by the various padding appraoches for FDA4, FDA5$_{dec}$ and HE2 are presented in Figure 5. For FDA4 all approaches obtain a better acc value over time. FDA4 is also a type I problem (similar to DIMP2) and therefore even though the POS changes over time, the POF remains static. For FDA5$_{dec}$ the acc values increase and decrease over time. Furthermore, the acc values vary over a big range. This occurs since the spread of the solutions change over time and the POF is deceptive. Therefore, the algorithms (padding approaches) struggle to converge to good solutions. Similar to FDA5$_{dec}$, the acc values for HE2 increases and decrease over time. However, over time the acc increase more than what they decrease. Therefore, as time goes by, all padding approaches struggle even more to converge to good solutions.

VI. Conclusions

When solving dynamic multi-objective optimisation problems (DMOOPs) with the dynamic vector evaluated particle swarm optimisation (DVEPSO) algorithm, each sub-swarm optimises only one objective and knowledge is then shared between the sub-swarms. The knowledge is shared by using the global guide of either the sub-swarm or from another sub-swarm when updating the particle's velocity. However, it may occur that the particles of the various sub-swarms have different dimensions, since they only contain the decision variables applicable to their sub-swarm's objective function. Therefore, if a particle with a smaller dimension is chosen for knowledge sharing, it does not have values for all of the decision variables that are required for the velocity update. In these cases, padding is required to obtain values for these missing dimensions.

This paper investigated four approaches to pad the missing dimensions, namely using the sub-swarm's own gbest, using a randomly selected pbest or gbest from another swarm and using parent-centric crossover (PBX) on a randomly selected particle's position, pbest and gbest. The results indicated that with regards to accuracy (acc) the random approaches (random gbest and random pbest) performed the best in fast changing environments, and the approach that uses the sub-swarm's own gbest performed the best in slowly changing environments. Therefore, an approach that combines the usage of these two approaches may perform well in both slowly changing and fast changing environments.

In addition, all approaches struggled to converge to the Pareto-optimal front (POF) of HE2, HE7 and FDA5$_{dec}$ in fast and severely changing environments. Outliers also occurred when solving DIMP2, especially in fast and severely changing environments. Therefore, more research is required to improve the convergence of DVEPSO in these type of environments.

References

[1] M. Cámara, J. Ortega, and F. de Toro. A single front genetic algorithm for parallel multi-objective optimization in dynamic environments. *Neurocomputing*, 72(16–18):3570–3579, 2007.

[2] A. Carlisle and G. Dozier. Tracking changing extrema with adaptive particle swarm optimizer. In *Proceedings of World Automation Congress*, volume 13, pages 265–270, Orlando, Florida, U.S.A., June 2002.

[3] M. Farina, K. Deb, and P. Amato. Dynamic multiobjective optimization problems: test cases, approximations, and applications. *IEEE Transactions on Evolutionary Computation*, 8(5):425–442, October 2004.

[4] C. Fonseca, L. Paquete, and M. López-Ibáñez. An improved dimension - sweep algorithm for the hypervolume indicator. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 1157–1163, Vancouver, Canada, 16-21 July 2006.

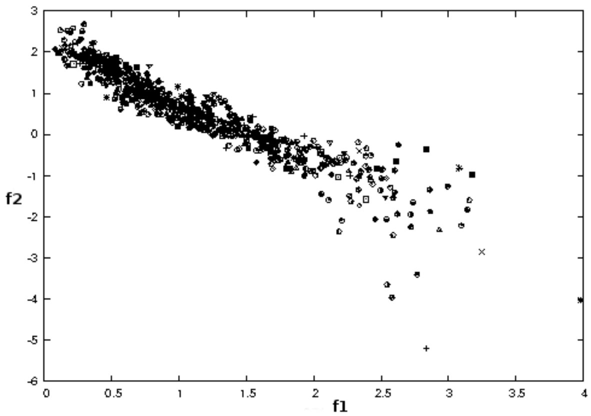
[5] C.-K. Goh and K. Tan. A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 13(1):103–127, February 2009.

[6] M. Greeff and A. Engelbrecht. Solving dynamic multi-objective problems with vector evaluated particle swarm optimisation. In *Proceedings of World Congress on Computational Intelligence (WCCI): Congress on Evoluationary Computation*, pages 2917–2924, Hong Kong, June 2008.

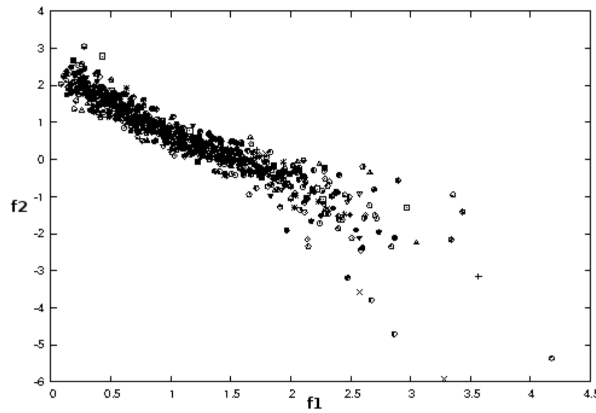
[7] M. Greeff and A. Engelbrecht. Dynamic multi-objective optimisation using PSO. In N. Nedjah, L. dos Santos Coelho, and L. de Macedo Mourelle, editors, *Multi-Objective Swarm Intelligent Systems*, volume 261 of *Studies in Computational Intelligence*, pages 105–123. Springer Berlin/Heidelberg, 2010.

[8] M. Helbig. *Solving dynamic multi-objective optimisation problems using vector evaluated particle swarm optimisation*. PhD thesis, University of Pretoria, 2012.

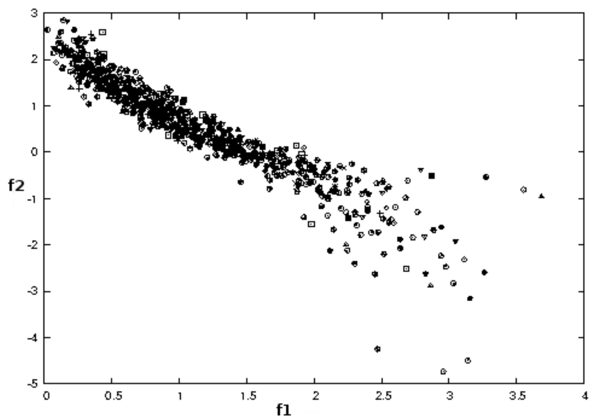
[9] M. Helbig and A. Engelbrecht. Archive management for dynamic multi-objective optimisation problems using vector evaluated particle swarm optimisation. In *Proceedings of Congress on Evolutionary Computation*, pages 2047–2054, New Orleans, U.S.A., June 2011.



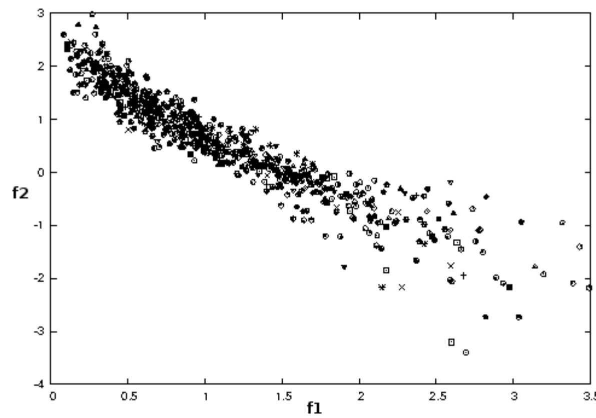
(a) P_c



(b) P_g



(c) P_{rg}



(d) P_{pg}

Fig. 2: POF found for HE7 with n_t = 1-$\tau_t = 10$ by the various padding approaches over 30 runs

[10] M. Helbig and A. Engelbrecht. Analysing the performance of dynamic multi-objective optimisation algorithms. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 1531–1539, Cancún, Mexico, June 2013.

[11] M. Helbig and A. Engelbrecht. Benchmarks for dynamic multi-objective optimisation. In *Proc. of IEEE Symposium Series on Computational Intelligence*, pages 84–91, Singapore, April 2013.

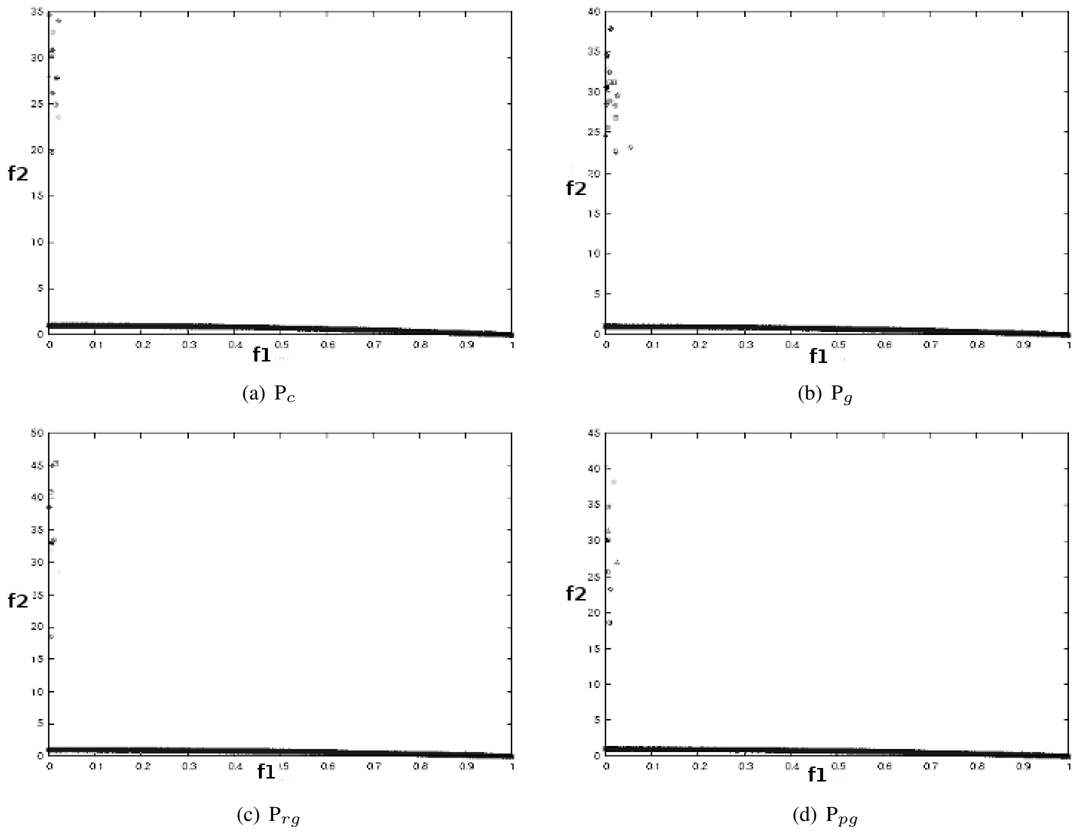
[12] M. Helbig and A. Engelbrecht. Performance measures for dynamic multi-objective optimisation algorithms. *Information Sciences*, 250:61–81, November 2013.

[13] M. Helbig and A. Engelbrecht. Dynamic multi-objective optimisation problems. *ACM Computing Surveys*, 2014. In Press.

[14] M. Helbig and A. Engelbrecht. Heterogeneous dynamic vector evaluated particle swarm optimisation for dynamic multi-objective optimisation. In *IEEE Congress on Evoluationary Computation*, pages 3151–3159, July 2014.

[15] W. Koo, C. Goh, and K. Tan. A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment. *Memetic Computing*, 2(2):87–110, 2010.

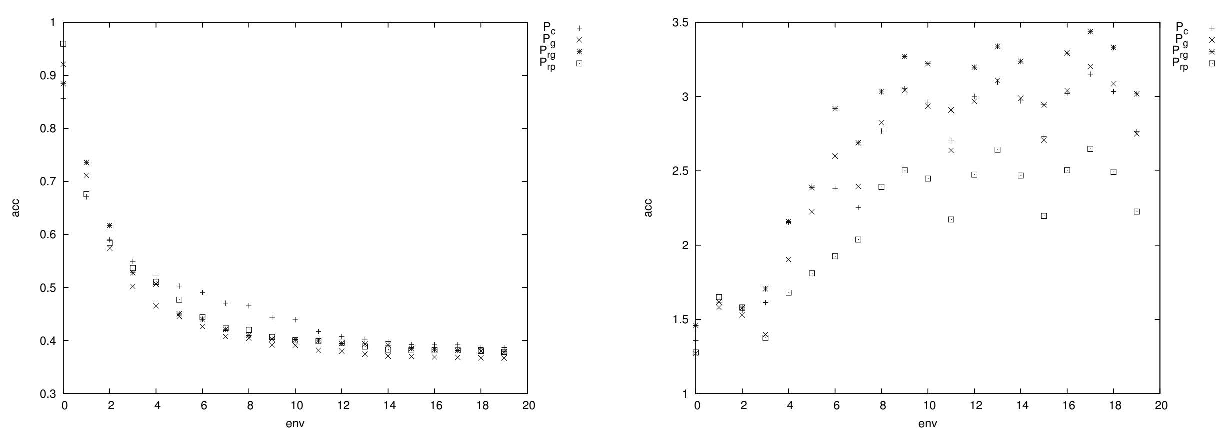
[16] K. Parsopoulos and M. Vrahatis. Recent approaches to global optimization problems through particle swarm optimization. *Natural Computing*, 1(2-3):235–306, 2002.



(a) P_c (b) P_g

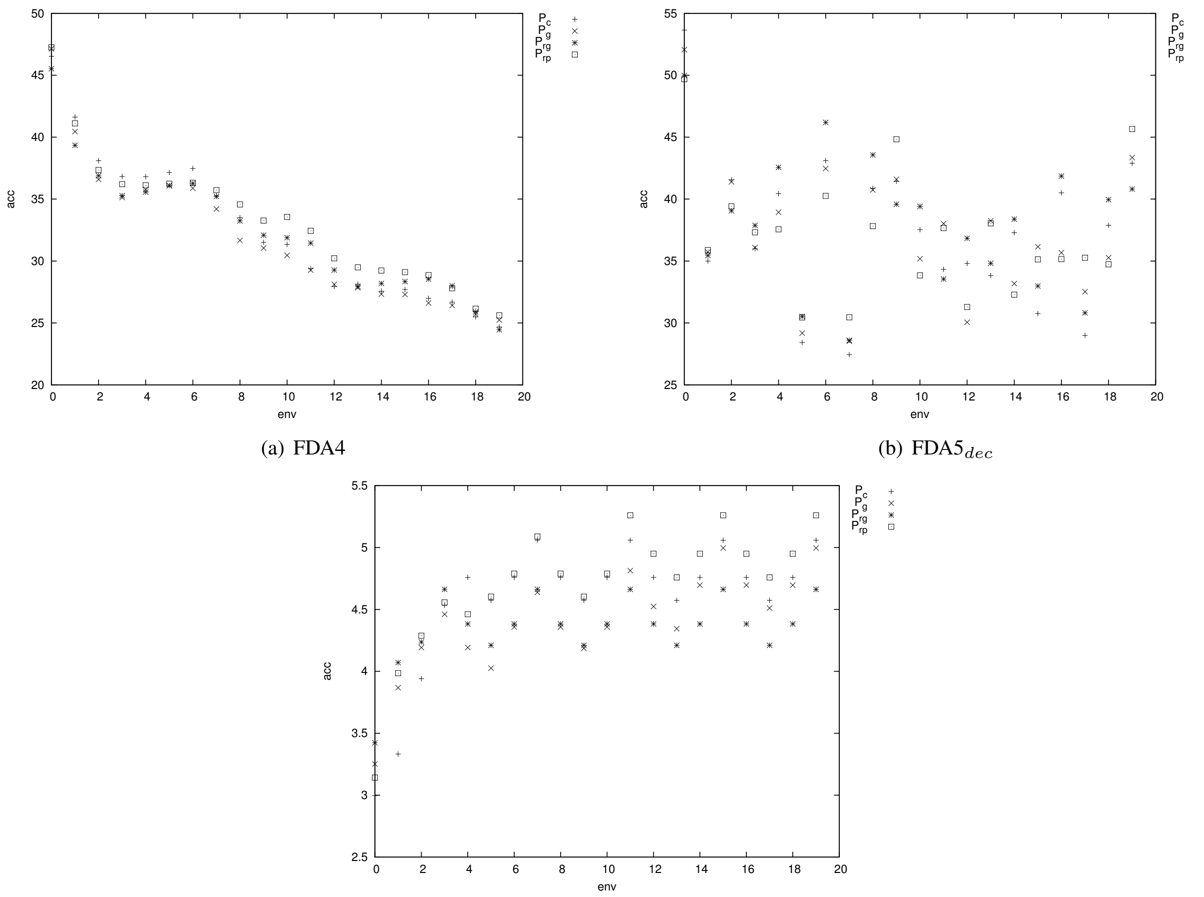
(c) P_{rg} (d) P_{pg}

Fig. 3: POF found for DIMP2 with $n_t = 1\text{-}\tau_t = 10$ by the various padding approaches over 30 runs



(a) DIMP2 (b) HE7

Fig. 4: acc values for DIMP2 and HE7 with $n_t = 1\text{-}\tau_t = 10$ for the various padding approaches for each of the 20 environments



(a) FDA4

(b) FDA5$_{dec}$

(c) HE2

Fig. 5: acc values for FDA4, FDA5$_{dec}$ and HE2 with $n_t = 1$-$\tau_t = 10$ for the various padding approaches for each of the 20 environments