

# Analyses of Guide Update Approaches for Vector Evaluated Particle Swarm Optimisation on Dynamic Multi-Objective Optimisation Problems

Mardé Helbig  
CSIR: Meraka Institute  
Pretoria, South Africa, 0184, and  
Department of Computer Science  
University of Pretoria  
Pretoria, South Africa, 0184  
Email: mhelbig@csir.co.za

Andries P. Engelbrecht  
Department of Computer Science  
University of Pretoria  
Pretoria, South Africa, 0184  
Email: engel@cs.up.ac.za

**Abstract**—The vector evaluated particle swarm optimisation (VEPSO) algorithm is a multi-swarm variation of particle swarm optimisation (PSO) used to solve static multi-objective optimisation problems (SMOOPs). Recently, VEPSO was extended to the dynamic VEPSO (DVEPSO) algorithm to solve dynamic multi-objective optimisation problems (DMOOPs) that have at least one objective that changes over time. The search process of DVEPSO is driven through local and global guides that can be updated in various ways. This paper investigates the influence of various guide update approaches on the performance of DVEPSO. DVEPSO is also compared against a competitive-cooperative evolutionary algorithm. The results indicate that DVEPSO performs well in fast changing environments, but struggles to converge to discontinuous Pareto-optimal fronts (POFs).

## I. INTRODUCTION

Real-world multi-objective optimisation problems (MOOPs) are mostly dynamic in nature with at least one of the conflicting objectives changing over time. These kind of problems are referred to as dynamic multi-objective optimisation problems (DMOOPs). DMOOPs, in contrast with dynamic single-objective optimisation problems (DSOOPs), do not have a single solution but a set of trade-off solutions referred to as the Pareto-optimal front (POF). Therefore, an algorithm that solves a DMOOP aims to track the POF over time and to find a diverse set of solutions.

The vector evaluated particle swarm optimisation (VEPSO) algorithm, which is a multi-swarm variation of PSO, was introduced by Parsopoulos *et al.* [1], [2] to solve static multi-objective optimisation problems (SMOOPs). Each swarm optimises only one objective function and the search process is driven through local and global guides. The local guides, also referred to as the personal bests, contain information about the particles' own experience with regards to a single objective. On the other hand, the global guides, also referred to as the global bests, contain information obtained by a pre-defined neighbourhood of particles, with regards to another objective. A knowledge sharing topology determines which objective's

gbest is used [3].

Recently, VEPSO was extended to dynamic VEPSO (DVEPSO) to solve DMOOPs [4]. This paper investigates the influence of knowledge sharing through guide update approaches on the performance of the DVEPSO algorithm. Various combinations of local and global guide update approaches are investigated. The update approaches include approaches that do not use Pareto-dominance, as well as newly proposed approaches that do incorporate Pareto dominance. It should be noted that this paper focusses on guide update approaches, and not on guide selection approaches. Guide selection approaches focus on the selection of solutions from the archive to guide the optimisation process to ensure a diverse set of solutions. The guides that are selected from the archive are then used as the local (personal best) and global guides (global best) of the particle swarm optimisation (PSO) algorithm. The guide update approaches discussed in this paper focus on methods that are used to update the swarm's local (personal best) and global (global best) guides using the solutions found by the particles.

The rest of the paper is organised as follows: Section II provides background information with regards to the VEPSO algorithm and dynamic MOO (DMOO). The DVEPSO algorithm, as well as guide update approaches, are discussed in Section III. Section IV provides information with regards to the experiments that are conducted to evaluate the performance of the algorithms. Information is provided with regards to the benchmark functions and performance measures that have been used, as well as the statistical analysis that was performed on the obtained results. The results of these experiments are provided in Section V. Finally, conclusions are presented in Section VI.

## II. VECTOR EVALUATED PSO

Inspired by the vector evaluated genetic algorithm (VEGA) [5], Parsopoulos *et al.* [1], [2] introduced a multi-swarm variation of PSO to solve SMOOPs. This multi-

objective algorithm is referred to as the VEPSO algorithm, where each swarm (PSO) optimises only one objective function. Knowledge of best solutions is then shared with the other swarms. This shared knowledge, contained in the gbest of another swarm, is then used to update the velocity of the particles:

$$\begin{aligned} S_k.v_{ij}(t+1) &= wS_k.v_{ij}(t) + c_1r_{1j}(t)(S_k.y_{ij}(t) - S_k.x_{ij}(t)) \\ &+ c_2r_{2j}(t)(S_s.\hat{y}_i(t) - S_k.x_{ij}(t)) \end{aligned} \quad (1)$$

where  $k = 1, \dots, m$  represents the index of the respective swarm,  $v_{ij}(t)$  and  $x_{ij}(t)$  represent the  $j$ -th dimension of the velocity and position of particle  $i$  at time  $t$  respectively,  $w$  is the inertia weight,  $S_s.\hat{y}_i$  is the global best of the  $s$ -th swarm,  $c_1$  and  $c_2$  are respectively the cognitive and social coefficients,  $\mathbf{r}_1, \mathbf{r}_2 \in [0, 1]^n$  and  $n$  is the dimension of the search space. The index,  $s$ , of the swarm from which knowledge is obtained, is selected based on a knowledge sharing topology [3]. If a random topology is used,  $s$  is selected randomly from  $[1, m]$ . If a ring topology is used,  $s$  is selected according to Equation (2), where

$$s = \begin{cases} m & \text{for } j = 1 \\ j - 1 & \text{for } j = 2, \dots, m \end{cases} \quad (2)$$

The best positions, pbest and gbest, are respectively referred to as the local and global guides, and guide the search by influencing the position of each particle. The original VEPSO articles [1], [2] do not indicate whether Pareto-dominance is used to update guides. Therefore, it is assumed that the original version of VEPSO updates the local and global guides according to the particles' fitness with regards to only one objective, i.e. the objective that the specific swarm is optimising.

When solving DMOOPs, an algorithm has to be able to detect a change in the environment and respond to the change in an appropriate way. One approach frequently used to detect changes is the sentry particle approach [6], where a random number of sentry particles are selected after each iteration and re-evaluated before the next iteration. If the current fitness value of the sentry particle differs more than a specified value from its fitness value just after the previous iteration, the swarm is alerted that a change has occurred.

After a change in the environment has occurred, the local and global guides are outdated and should be updated according to the particles' fitness with regards to the new DMOOP. The next section discusses approaches to update local and global guides.

### III. GUIDE UPDATE APPROACHES

The DVEPSO algorithm was introduced in [4]. Knowledge sharing is implemented via pbest and gbest positions, according to a knowledge sharing topology [3]. The following new guide update strategies are proposed for the local and global guides:

- The dominant approach, where each particle's fitness is measured with respect to all objectives of the DMOOP. If the particle's position dominates the current local guide, the particle's current position is selected as the new local

guide. This strategy is referred to as  $p_d$ . If this approach is used to update a global guide, it is referred to as  $g_d$ .

- The non-dominated approach, where a guide is updated only if its fitness can be improved and if the new position is non-dominated with respect to the guide.  $p_n$  refers to a local guide update using non-dominance, and  $g_n$  refers to a global guide update.
- The random approach, where a guide is updated if the new position is non-dominated with respect to the guide, by randomly selecting either the particle position or the corresponding guide.  $p_r$  refers to a local guide update and  $g_r$  to a global guide update using the random strategy.

In addition, the newly proposed guide update approaches are compared against the standard VEPSO guide update, where the particle's fitness is measured with regards to only the objective function that the specific swarm optimises. Only if an improvement in the fitness of the current guide can be obtained, the guide is updated. No Pareto-dominance information is used. With reference to a local guide, this approach is referred to as  $p_s$  and with reference to a global guide,  $g_s$ . All combinations of the above local and global guide update strategies are evaluated in this paper.

The following default configuration of DVEPSO is used for this research:

- Each swarm has 20 particles and a random swarm topology is used.
- The non-dominated solutions found so far is stored in an archive, with size set to 100. If the archive is full, a solution from a crowded region in the found POF is removed.
- Sentry particles is used for change detection. If a change has been detected, 30% of the particles of the swarm(s) whose objective function changed is re-initialised. The non-dominated solutions in the archive is re-evaluated and the solutions that have become dominated are removed from the archive.

### IV. EXPERIMENTAL SETUP

This section describes experiments that were conducted, using benchmark functions and performance measures discussed in Sections IV-A and IV-B respectively. Section IV-C discusses the statistical analysis that was performed on the obtained data. The comparison of DVEPSO and dCOEA is discussed in Section IV-D.

All experiments consisted of 30 independent runs and each run continued for 1000 iterations. For all benchmark functions, the severity of change ( $n_t$ ) was set to 10 and the frequency of change ( $\tau_t$ ) was set to either 10 or 50 to evaluate the performance of DMOO algorithms in both a fast and slowly changing environment. The PSO parameters were set to values that lead to convergent behaviour [7], namely  $w = 0.72$  and  $c_1 = c_2 = 1.49$ .

All code was implemented in the Computational Intelligence library (CIlib) [8]. All simulations were run on the Sun Hybrid System's Harpertown and Nehalem Systems of the Center for High Performance Computing [9].

## A. Benchmark Functions

This section discusses the benchmark functions that were used to study the influence of guide update approaches on the performance of DVEPSO. Five benchmark functions were used of various DMOO Types, namely DIMP2 [10], FDA3<sub>Camara</sub> [11], dMOP2 [12], dMOP3 [12] and HE2 [13]. Various types of DMOOPs exist. Farina *et al.* [11] classified DMOOPs into four types, namely:

- **Type I** where the POF (corresponding objective function values) does not change over time, but the Pareto-optimal set (POS) (optimal set of decision variables) changes.
- **Type II** where both the POF and the POS change.
- **Type III** where the POF changes, but the POS remains unchanged.
- **Type IV** where both the POF and the POS remain unchanged, even though the problem changes over time.

DIMP2 is a Type I problem where each decision variable has its own rate of change, except the variable  $x_1$  that controls the spread of solutions. FDA3<sub>Camara</sub> is a Type II DMOOP where the density of solutions in the POF changes over time. dMOP2 is a Type II DMOOP with a POF that changes from convex to concave. dMOP3 is a Type I DMOOP where the spread of the POF solutions changes over time. HE2 is a Type III DMOOP with a discontinuous POF that consists of various disconnected continuous sub-regions.

## B. Performance Measures

This section discusses the performance measures that were used to determine the performance of DVEPSO for the different guide update approaches.

The first performance measure is the number of non-dominated solutions (NS) in the found POF,  $POF^*$ .

The second performance measure is the hypervolume distance (HVD) or alternative accuracy measure [14]:

$$acc(t) = |HV(POF(t)) - HV(POF^*(t))| \quad (3)$$

where  $acc(t)$  is the absolute HVD at time  $t$ . Using the absolute values ensure that  $acc(t) \geq 0$ , even if the HV of the approximated POF is larger than the HV of the true POF. A low  $acc$  value indicates a good performance.

The effect of the changes in the environment on the accuracy ( $acc$  is defined in Equation 3) of the algorithm is quantified by the third measure, namely stability [15]:

$$stab(t) = \max\{0, acc(t-1) - acc(t)\} \quad (4)$$

where a low  $stab$  value indicates good performance.

## C. Statistical Analysis of Obtained Data

This section discusses the statistical analysis procedure performed on the obtained data. For each function and for each  $\tau_t$  value, a Kruskal-Wallis test was performed over the obtained data to determine whether there is a statistically significant difference in performance. If this test indicated that there was a difference, pairwise Mann-Whitney U tests were performed between the obtained data for the various guide update approaches.

For each guide update approach, if the pairwise Mann-Whitney U test indicated a statistically significant difference, a win was recorded for the winning algorithm and a loss for the losing algorithm. These wins and losses are reported in Tables I to III.

According to these results, the best DVEPSO guide update combinations were selected and compared against dCOEA. The same process was then followed to calculate the wins and losses for the selected DVEPSO guide update combinations and dCOEA. These results are reported in Tables IV to VII.

All statistical tests were performed for a confidence level of 95%. The null hypothesis for the guide update approaches was that there are no statistical significant difference between the performance of the various guide update approaches. When comparing DVEPSO configurations with dCOEA, the null hypothesis was that there are no statistical significant difference between the performance of the various DMOO algorithms. For both cases, the alternative hypothesis was that there is a difference in mean performance.

## D. Comparison of DMOO Algorithms

The performance of DVEPSO was compared against a dynamic competitive-cooperative coevolutionary algorithm (dCOEA) proposed by Goh and Tan [12]. The source code of dCOEA was obtained from the authors of [12]. While DVEPSO uses one swarm per objective, dCOEA uses one sub-population to optimise one decision variable. The parameters of dCOEA were set according to [12].

## V. RESULTS

This section discusses the results that were obtained. Section V-A discusses the results obtained for the various guide update approaches. The results of experiments that were run to compare the performance of DVEPSO with dCOEA are discussed in Section V-B.

### A. Guide Update Approaches

This section discusses the effect of various guide update approaches on the performance of DVEPSO. The results are presented in Tables I to III; PM refers to the performance measure being analysed and  $s$ ,  $n$ ,  $r$  and  $d$  of the pbest-gbest combinations refer to  $p_s$ ,  $p_n$ ,  $p_r$ ,  $p_d$  or  $g_s$ ,  $g_n$ ,  $g_r$  and  $g_d$  respectively. Only results with statistical significant differences are presented in the tables.

When solving FDA3<sub>Camara</sub>, the various guide update approaches had no statistical significant influence on the performance of DVEPSO with regards to  $stab$  (refer to Table I) for  $\tau_t = 10$  and  $\tau_t = 50$ , as well as with regards to  $acc$  when  $\tau_t = 50$ . However, with regards to  $acc$  for  $\tau_t = 10$  all pbest-gbest combinations performed well, except for  $p_s$ - $g_n$ . With regards to NS, all pbest-gbest combinations performed well, except for all  $p_s$  combinations that performed poorly.

Table I indicates that, when solving DIMP2, the various guide update approaches had no statistical significant influence on the performance of DVEPSO with regards to NS. However, with regards to  $acc$  and  $stab$ ,  $p_d$ - $g_r$  obtained the best performance for  $\tau_t = 10$  and  $p_r$ - $g_d$  for  $\tau_t = 50$ .

TABLE I: Wins and Losses of FDA3<sub>Camara</sub>, DIMP2, dMOP2 and dMOP3

DMOOP	n <sub>t</sub>	τ <sub>t</sub>	PM	Results	pbest-gbest combination															
					s-s	s-n	s-d	s-r	n-s	n-n	n-d	n-r	r-s	r-n	r-d	r-r	d-s	d-n	d-d	d-r
FDA3	10	10	acc	Wins	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
FDA3	10	10	acc	Losses	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FDA3	10	10	acc	Diff	0	-2	0	0	0	0	1	0	1	0	0	0	0	0	0	0
FDA3	10	10	acc	Rank	3	16	3	3	3	3	1	3	1	3	3	3	3	3	3	3
FDA3	10	10	NS	Wins	0	0	0	0	4	4	4	4	4	4	4	4	4	4	4	4
FDA3	10	10	NS	Losses	12	12	12	12	0	0	0	0	0	0	0	0	0	0	0	0
FDA3	10	10	NS	Diff	-12	-12	-12	-12	4	4	4	4	4	4	4	4	4	4	4	4
FDA3	10	10	NS	Rank	13	13	13	13	1	1	1	1	1	1	1	1	1	1	1	1
FDA3	10	50	NS	Wins	0	0	0	0	4	4	4	4	4	4	4	4	4	4	4	4
FDA3	10	50	NS	Losses	12	12	12	12	0	0	0	0	0	0	0	0	0	0	0	0
FDA3	10	50	NS	Diff	-12	-12	-12	-12	4	4	4	4	4	4	4	4	4	4	4	4
FDA3	10	50	NS	Rank	13	13	13	13	1	1	1	1	1	1	1	1	1	1	1	1
DIMP2	10	10	acc	Wins	0	0	0	0	1	1	0	0	2	2	1	1	1	1	5	6
DIMP2	10	10	acc	Losses	2	10	4	2	0	0	1	2	0	0	0	0	0	0	0	0
DIMP2	10	10	acc	Diff	-2	-10	-4	-2	1	1	-1	-2	2	2	1	1	1	1	5	6
DIMP2	10	10	acc	Rank	12	16	15	12	5	5	11	12	3	3	5	5	5	5	2	1
DIMP2	10	50	acc	Wins	0	0	0	0	0	0	0	0	0	0	4	0	2	1	0	0
DIMP2	10	50	acc	Losses	0	0	0	1	1	3	0	0	0	0	0	2	0	0	0	0
DIMP2	10	50	acc	Diff	0	0	0	-1	-1	-3	0	0	0	0	4	-2	2	1	0	0
DIMP2	10	50	acc	Rank	4	4	4	13	13	16	4	4	4	4	1	15	2	3	4	4
DIMP2	10	10	stab	Wins	0	0	0	0	1	1	0	0	2	1	1	1	1	2	2	6
DIMP2	10	10	stab	Losses	1	10	4	1	0	0	1	1	0	0	0	0	0	0	0	0
DIMP2	10	10	stab	Diff	-1	-10	-4	-1	1	1	-1	-1	2	1	1	1	1	2	2	6
DIMP2	10	10	stab	Rank	11	16	15	11	5	5	11	11	2	5	5	5	5	2	2	1
DIMP2	10	50	stab	Wins	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0
DIMP2	10	50	stab	Losses	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0
DIMP2	10	50	stab	Diff	0	0	0	0	0	-2	0	0	0	0	1	0	0	1	0	0
DIMP2	10	50	stab	Rank	3	3	3	3	3	16	3	3	3	3	1	3	3	1	3	3
dMOP2	10	10	acc	Wins	7	1	1	1	4	0	4	3	4	3	3	3	3	3	4	3
dMOP2	10	10	acc	Losses	0	12	12	12	1	7	1	0	1	0	0	0	0	1	0	0
dMOP2	10	10	acc	Diff	7	-11	-11	-11	3	-7	3	3	3	3	3	3	3	3	3	3
dMOP2	10	10	acc	Rank	1	14	14	14	2	13	2	2	2	2	2	2	2	2	2	2
dMOP2	10	50	acc	Wins	0	2	0	1	4	4	4	5	5	4	5	5	5	5	4	4
dMOP2	10	50	acc	Losses	13	12	14	12	0	6	0	0	0	0	0	0	0	0	0	0
dMOP2	10	50	acc	Diff	-13	-10	-14	-11	4	-2	4	5	5	4	5	5	5	5	4	4
dMOP2	10	50	acc	Rank	15	13	16	14	7	12	7	1	1	7	1	1	1	1	7	7
dMOP2	10	10	stab	Wins	6	4	3	2	1	0	1	0	1	0	0	0	0	0	1	0
dMOP2	10	10	stab	Losses	0	1	1	1	1	5	3	0	2	0	1	0	0	0	3	1
dMOP2	10	10	stab	Diff	6	3	2	1	0	-5	-2	0	-1	0	-1	0	0	0	-2	-1
dMOP2	10	10	stab	Rank	1	2	3	4	5	16	14	5	11	5	11	5	5	5	14	11
dMOP2	10	50	stab	Wins	0	0	0	0	4	4	4	4	5	4	4	4	5	5	4	4
dMOP2	10	50	stab	Losses	12	12	12	12	0	3	0	0	0	0	0	0	0	0	0	0
dMOP2	10	50	stab	Diff	-12	-12	-12	-12	4	1	4	4	5	4	4	4	5	5	4	4
dMOP2	10	50	stab	Rank	13	13	13	13	4	12	4	4	1	4	4	4	1	1	4	4
dMOP3	10	10	acc	Wins	0	0	0	0	4	4	4	4	4	4	4	4	4	4	4	4
dMOP3	10	10	acc	Losses	12	12	12	12	0	0	0	0	0	0	0	0	0	0	0	0
dMOP3	10	10	acc	Diff	-12	-12	-12	-12	4	4	4	4	4	4	4	4	4	4	4	4
dMOP3	10	10	acc	Rank	13	13	13	13	1	1	1	1	1	1	1	1	1	1	1	1
dMOP3	10	50	acc	Wins	0	0	0	0	4	4	4	4	4	4	4	5	4	4	4	4
dMOP3	10	50	acc	Losses	12	12	12	12	0	1	0	0	0	0	0	0	0	0	0	0
dMOP3	10	50	acc	Diff	-12	-12	-12	-12	4	3	4	4	4	4	4	5	4	4	4	4
dMOP3	10	50	acc	Rank	13	13	13	13	2	12	2	2	2	2	2	1	2	2	2	2
dMOP3	10	50	stab	Wins	0	0	0	0	2	1	3	2	2	2	2	2	1	1	2	2
dMOP3	10	50	stab	Losses	9	12	1	0	0	0	0	0	0	0	0	0	0	0	0	0
dMOP3	10	50	stab	Diff	-9	-12	-1	0	2	1	3	2	2	2	2	2	1	1	2	2
dMOP3	10	50	stab	Rank	15	16	14	13	2	10	1	2	2	2	2	2	10	10	2	2
dMOP3	10	10	NS	Wins	0	0	0	0	4	4	4	4	4	4	4	4	4	4	4	4
dMOP3	10	10	NS	Losses	12	12	12	12	0	0	0	0	0	0	0	0	0	0	0	0
dMOP3	10	10	NS	Diff	-12	-12	-12	-12	4	4	4	4	4	4	4	4	4	4	4	4
dMOP3	10	10	NS	Rank	13	13	13	13	1	1	1	1	1	1	1	1	1	1	1	1
dMOP3	10	50	NS	Wins	0	11	0	0	0	0	0	0	0	0	0	1	0	0	0	0
dMOP3	10	50	NS	Losses	0	0	0	0	1	1	1	1	1	1	2	0	1	1	1	1
dMOP3	10	50	NS	Diff	0	11	0	0	-1	-1	-1	-1	-1	-1	-2	1	-1	-1	-1	-1
dMOP3	10	50	NS	Rank	3	1	3	3	6	6	6	6	6	6	16	2	6	6	6	6

Furthermore,  $p_s-g_n$  and  $p_s-g_d$  obtained the worst performance for  $\tau_t = 10$  and  $p_n-g_n$  for  $\tau_t = 50$ .

Similar to DIMP2, when solving dMOP2 the various guide update approaches had no statistical significant influence on the performance of DVEPSO with regards to NS (refer to Table I). However, with regards to *acc* and *stab*, for  $\tau_t = 50$  all combinations performed well, except for  $p_n-g_n$  and all  $p_s$  combinations.  $p_s-g_s$  obtained the best performance with regards to *acc* and *stab* for  $\tau_t = 10$ . Furthermore, all  $p_s$  combinations performed very bad with regards to *acc* and *stab* for  $\tau_t = 50$ . For  $\tau_t = 10$ ,  $p_n-g_n$  and all  $p_s$  combinations, except for  $p_s-g_s$ , performed bad with regards to *acc* and with regards to *stab*,  $p_n-g_n$  and  $p_n-g_d$  performed poorly.

Table I indicates that, when solving DMOP3, the various guide update approaches had no statistical significant influence on the performance of DVEPSO with regards to *stab* for  $\tau_t = 10$ . With regards to *stab* for  $\tau_t = 50$ , all pbest-gbest combinations performed well, except for the  $p_s$  combinations. With regards to *acc*, all combinations performed well, except for the  $p_s$  combinations. For  $\tau_t = 10$ , all pbest-gbest combinations, except the  $p_s$  combinations, performed well with regards to NS. However, for  $\tau_t = 50$  the  $p_s$  combinations outperformed the other pbest-gbest combinations and  $p_s-g_n$  completely outperformed all other pbest-gbest combinations with regards to NS.

In contrast to dMOP3, Table II indicates that, when solving HE, all  $p_s$  combinations completely outperformed all other pbest-gbest combinations with regards to *acc* and NS. Furthermore, with regards to *stab*, three of the four  $p_s$  combinations completely outperformed the other pbest-gbest combinations, namely  $p_s-g_s$ ,  $p_s-g_d$  and  $p_s-g_r$ . With regards to NS when  $\tau_t = 50$ ,  $p_r-g_s$  and  $p_d-g_s$  performed very bad against the other pbest-gbest combinations.

The overall wins and losses of the guide update approaches are presented in Table III. For  $\tau_t = 10$ ,  $p_d-g_r$  obtained the best overall performance and for  $\tau_t = 50$ ,  $p_d-g_s$  performed the best. The guide update approach of the original version of VEPSO,  $p_s-g_s$ , performed well with regards to accuracy and stability for  $\tau_t = 10$  and with regards to NS for  $\tau_t = 50$ . However,  $p_s-g_s$  lead to really poor performance with regards to accuracy and stability for  $\tau_t = 50$  and with regards to NS for  $\tau_t = 10$ . It obtained the best overall rank of all guide update approaches with regards to stability for  $\tau_t = 10$ . However, Pareto-dominance-based guide update approaches outperformed  $p_s-g_s$  with regards to accuracy and NS for both  $\tau_t = 10$  and  $\tau_t = 50$  and with regards to stability for  $\tau_t = 50$ . Therefore, the results indicate that Pareto-dominance guide update approaches enhances the performance of DVEPSO in both fast and slower changing environments.

Based on the obtained results, the following DVEPSO configurations were selected for comparison with dCOEA:  $p_d-g_s$  and  $p_d-g_r$  that obtained the best overall performance for  $\tau_t = 10$  and  $\tau_t = 50$  respectively, and  $p_s-g_n$  that performed the best with regards to NS for  $\tau_t = 50$ .

## B. Comparison of DMOO Algorithms

This section compares the performance of DVEPSO, using the best guide update combinations, and dCOEA. The results of this comparison are presented in Tables IV to VII.

The results in Table IV indicate that, when solving FDA3<sub>Camara</sub>, DVEPSO outperformed dCOEA with regards to *stab* for both  $\tau_t = 10$  and  $\tau_t = 50$  and *acc* for  $\tau_t = 10$ . However, dCOEA outperformed DVEPSO with regards to *acc* for  $\tau_t = 50$  and NS for both  $\tau_t = 10$  and  $\tau_t = 50$ .

On DIMP2, DVEPSO outperformed dCOEA with regards to both *acc* and *stab* for both  $\tau_t = 10$  and  $\tau_t = 50$ . However, dCOEA outperformed DVEPSO with regards to NS for both  $\tau_t = 10$  and  $\tau_t = 50$ .

When solving dMOP2, DVEPSO outperformed dCOEA with regards to *acc* and *stab* for  $\tau_t = 10$ . However, for  $\tau_t = 50$  there was no statistical significant difference between the performance of DVEPSO and dCOEA with regards to *acc* and *stab*. Furthermore, with regards to NS, dCOEA outperformed DVEPSO.

In contrast to dMOP2, dCOEA outperformed DVEPSO on dMOP3 with regards to *acc* and *stab* for both  $\tau_t = 10$  and  $\tau_t = 50$ . However, DVEPSO outperformed dCOEA with regards to NS for both  $\tau_t = 10$  and  $\tau_t = 50$ . The results are presented in Table V.

Similar to dMOP3, dCOEA outperformed DVEPSO on HE2 with regards to *acc* and *stab* for both  $\tau_t = 10$  and  $\tau_t = 50$ . However, DVEPSO outperformed dCOEA with regards to NS for both  $\tau_t = 10$  and  $\tau_t = 50$  (refer to Table VI).

The overall wins and losses of DVEPSO and dCOEA are presented in Table VII. Note that DVEPSO outperformed dCOEA with regards to *acc* and *stab* for  $\tau_t = 10$ , whereas dCOEA outperformed DVEPSO with regards to *acc* and *stab* for  $\tau_t = 50$ . These results indicate that DVEPSO obtained a good accuracy and stability in fast changing environments. While dCOEA struggled when the environment changed frequently, it obtained a good accuracy and stability when it had more time to converge towards the true POF. Furthermore, dCOEA found on average more solutions than DVEPSO. It should also be noted that DVEPSO struggled to converge towards a discontinuous POF, which can clearly be seen in its poor performance with HE2.

Figure 1 illustrates  $POF^*$ s that have been found at all time steps just before an environmental change for  $\tau_t = 10$  by DVEPSO (using the  $p_d-g_r$  update combination) and dCOEA. DVEPSO converged with a high accuracy towards the POFs of DIMP2 and dMOP2, but dCOEA struggled to converge towards these POFs. On HE2, DVEPSO found solutions in the correct shape of the POF, but its solutions were quite far from the true POF. On dMOP3, dCOEA found solutions closer to the true POF, but both algorithms struggled to consistently converge to the POF.

## VI. CONCLUSION

This paper investigated the influence of guide update approaches on the performance of DVEPSO. Various pbest-gbest combinations were used to determine which combination

TABLE II: Wins and Losses of HE2 for DVEPSO

$n_t$	$\tau_t$	PM	Results	pbest-gbest combination																
				s-s	s-n	s-d	s-r	n-s	n-n	n-d	n-r	r-s	r-n	r-d	r-r	d-s	d-n	d-d	d-r	
10	10	<i>acc</i>	Wins	13	12	14	15	0	0	0	0	0	0	0	0	0	0	0	0	0
10	10	<i>acc</i>	Losses	2	3	1	0	4	4	4	4	4	4	4	4	4	4	4	4	4
10	10	<i>acc</i>	Diff	11	9	13	15	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
10	10	<i>acc</i>	Rank	3	4	2	1	5	5	5	5	5	5	5	5	5	5	5	5	5
10	50	<i>acc</i>	Wins	13	12	14	15	0	1	4	0	1	5	3	1	8	0	0	0	6
10	50	<i>acc</i>	Losses	2	3	1	0	9	7	4	9	5	4	5	6	4	8	12	4	2
10	50	<i>acc</i>	Diff	11	9	13	15	-9	-6	0	-9	-4	1	-2	-5	4	-8	-12	2	4
10	50	<i>acc</i>	Rank	3	4	2	1	14	12	8	14	10	7	9	11	5	13	16	6	6
10	10	<i>stab</i>	Wins	13	0	13	13	0	0	0	0	0	0	0	0	0	0	0	0	0
10	10	<i>stab</i>	Losses	0	3	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3
10	10	<i>stab</i>	Diff	13	-3	13	13	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3
10	10	<i>stab</i>	Rank	1	4	1	1	4	4	4	4	4	4	4	4	4	4	4	4	4
10	50	<i>stab</i>	Wins	13	4	13	12	0	0	0	0	0	0	0	0	0	0	0	0	0
10	50	<i>stab</i>	Losses	0	2	0	0	3	4	3	4	3	3	3	3	3	3	4	4	4
10	50	<i>stab</i>	Diff	13	2	13	12	-3	-4	-3	-4	-3	-3	-3	-3	-3	-3	-4	-4	-4
10	50	<i>stab</i>	Rank	1	4	1	3	5	13	5	13	5	5	5	5	5	5	13	13	13
10	10	<i>NS</i>	Wins	12	15	13	14	0	2	0	0	0	0	2	0	0	0	0	0	0
10	10	<i>NS</i>	Losses	3	0	2	1	4	4	4	4	6	4	4	4	4	4	6	6	4
10	10	<i>NS</i>	Diff	9	15	11	13	-4	-2	-4	-4	-6	-4	-2	-4	-4	-4	-6	-4	-4
10	10	<i>NS</i>	Rank	4	1	3	2	7	5	7	7	15	7	5	7	7	7	15	7	7
10	50	<i>NS</i>	Wins	13	14	12	14	3	6	2	6	0	10	0	2	0	2	3	2	2
10	50	<i>NS</i>	Losses	2	0	3	0	5	5	7	4	13	4	9	5	13	7	5	7	7
10	50	<i>NS</i>	Diff	11	14	9	14	-2	1	-5	2	-13	6	-9	-3	-13	-5	-2	-5	-5
10	50	<i>NS</i>	Rank	3	1	4	1	8	7	11	6	15	5	14	10	15	11	8	11	11

TABLE III: Overall Wins and Losses for Guide Update Approaches

$n_t$	$\tau_t$	PM	Results	pbest-gbest combination																
				s-s	s-n	s-d	s-r	n-s	n-n	n-d	n-r	r-s	r-n	r-d	r-r	d-s	d-n	d-d	d-r	
10	10	<i>acc</i>	Wins	20	13	15	16	9	5	9	7	11	9	8	8	8	8	13	13	6
10	10	<i>acc</i>	Losses	16	39	29	26	5	11	6	6	5	4	4	4	4	4	5	4	4
10	10	<i>acc</i>	Diff	4	-26	-14	-10	4	-6	3	1	6	5	4	4	4	4	8	9	9
10	10	<i>acc</i>	Rank	5	16	15	14	5	13	11	12	3	4	5	5	5	5	2	1	1
10	50	<i>acc</i>	Wins	13	14	14	16	8	9	12	9	10	13	16	11	19	10	8	14	14
10	50	<i>acc</i>	Losses	27	27	27	25	10	17	4	9	5	4	5	8	4	8	12	4	4
10	50	<i>acc</i>	Diff	-14	-13	-13	-9	-2	-8	8	0	5	9	11	3	15	2	-4	10	10
10	50	<i>acc</i>	Rank	16	14	14	13	10	12	5	9	6	4	2	7	1	8	11	3	3
10	10	<i>stab</i>	Wins	19	4	16	15	2	1	1	0	3	1	1	1	1	2	3	6	6
10	10	<i>stab</i>	Losses	1	14	5	2	4	8	7	4	5	3	4	3	3	3	6	4	4
10	10	<i>stab</i>	Diff	18	-10	11	13	-2	-7	-6	-4	-2	-2	-3	-2	-2	-1	-3	2	2
10	10	<i>stab</i>	Rank	1	16	3	2	6	15	14	13	6	6	11	6	6	5	11	4	4
10	50	<i>stab</i>	Wins	13	4	13	12	6	5	7	6	7	6	7	6	6	7	6	6	6
10	50	<i>stab</i>	Losses	21	26	13	12	3	9	3	4	3	3	3	3	3	3	4	4	4
10	50	<i>stab</i>	Diff	-8	-22	0	0	3	-4	4	2	4	3	4	3	3	4	2	2	2
10	50	<i>stab</i>	Rank	15	16	12	12	5	14	1	9	1	5	1	5	1	9	9	9	9
10	10	<i>NS</i>	Wins	12	15	13	14	8	10	8	8	8	8	10	8	8	8	8	8	8
10	10	<i>NS</i>	Losses	27	24	26	25	4	4	4	4	6	4	4	4	4	4	6	4	4
10	10	<i>NS</i>	Diff	-15	-9	-13	-11	4	6	4	4	2	4	6	4	4	4	2	4	4
10	10	<i>NS</i>	Rank	16	13	15	14	3	1	3	3	11	3	1	3	3	3	11	3	3
10	50	<i>NS</i>	Wins	13	25	12	14	7	10	6	10	4	14	4	7	4	6	7	6	6
10	50	<i>NS</i>	Losses	13	14	12	14	8	11	7	11	5	15	6	6	5	7	8	7	7
10	50	<i>NS</i>	Diff	0	11	0	0	-1	-1	-1	-1	-1	-1	-2	1	-1	-1	-1	-1	-1
10	50	<i>NS</i>	Rank	3	1	3	3	6	6	6	6	6	6	6	6	6	6	6	6	6

produces the best results. It has been found that better results were achieved when the pbest is updated only if a new solution dominates the previous pbest. Furthermore, two gbest update approaches lead to better accuracy, namely either using the standard VEPSO gbest update without Pareto-dominance, or if a pbest is non-dominated with regards to the current gbest, randomly select the pbest or the current gbest as the new gbest.

The performance of the three best DVEPSO guide update combinations were compared against a cooperative-competitive evolutionary algorithm (dCOEA). The results indicated that DVEPSO performed well with a good accuracy and stability in fast changing environments. However, DVEPSO struggled to converge towards a discontinuous POF.

Future work includes a more in-depth analysis of the influence of various parameters on the performance of DVEPSO and comparing DVEPSO against more state-of-the-art DMOO algorithms.

ACKNOWLEDGMENT

The authors would like to thank the Centre for High Performance Computing (CHPC) for the use of their infrastructure

for this research. Furthermore, they would like to thank C.-K. Goh for sharing his source code of the dCOEA algorithm.

REFERENCES

- [1] K. Parsopoulos and M. Vrahatis, "Recent approaches to global optimization problems through particle swarm optimization," *Natural Computing*, vol. 1, no. 2-3, pp. 235-306, 2002.
- [2] K. Parsopoulos, D. Tasoulis, and M. Vrahatis, "Multiobjective optimization using parallel vector evaluated particle swarm optimization," in *Proc. of IASTED International Conference on Artificial Intelligence and Applications*, Innsbruck Austria, 2004.
- [3] M. Greeff and A. Engelbrecht, "Dynamic multi-objective optimisation using pso," in *Multi-objective Swarm Intelligent Systems*, ser. Studies in Computational Intelligence, 2010, vol. 261, pp. 105-123.
- [4] —, "Solving dynamic multi-objective problems with vector evaluated particle swarm optimisation," in *Proc. of IEEE World Congress on Evolutionary Computation: IEEE Congress on Evolutionary Computation*, Hong Kong, June 2008, pp. 2917-2924.
- [5] J. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proc. of the 1st International Conference on Genetic Algorithms*, 1985, pp. 93-100.
- [6] A. Carlisle and G. Dozier, "Adapting particle swarm optimization to dynamic environments," in *Proc. of International Conference on Artificial Intelligence (ICAI 2000)*, 2000, pp. 429-434.
- [7] F. van den Bergh, "An analysis of particle swarm optimizers," Ph.D. dissertation, University of Pretoria, 2002.

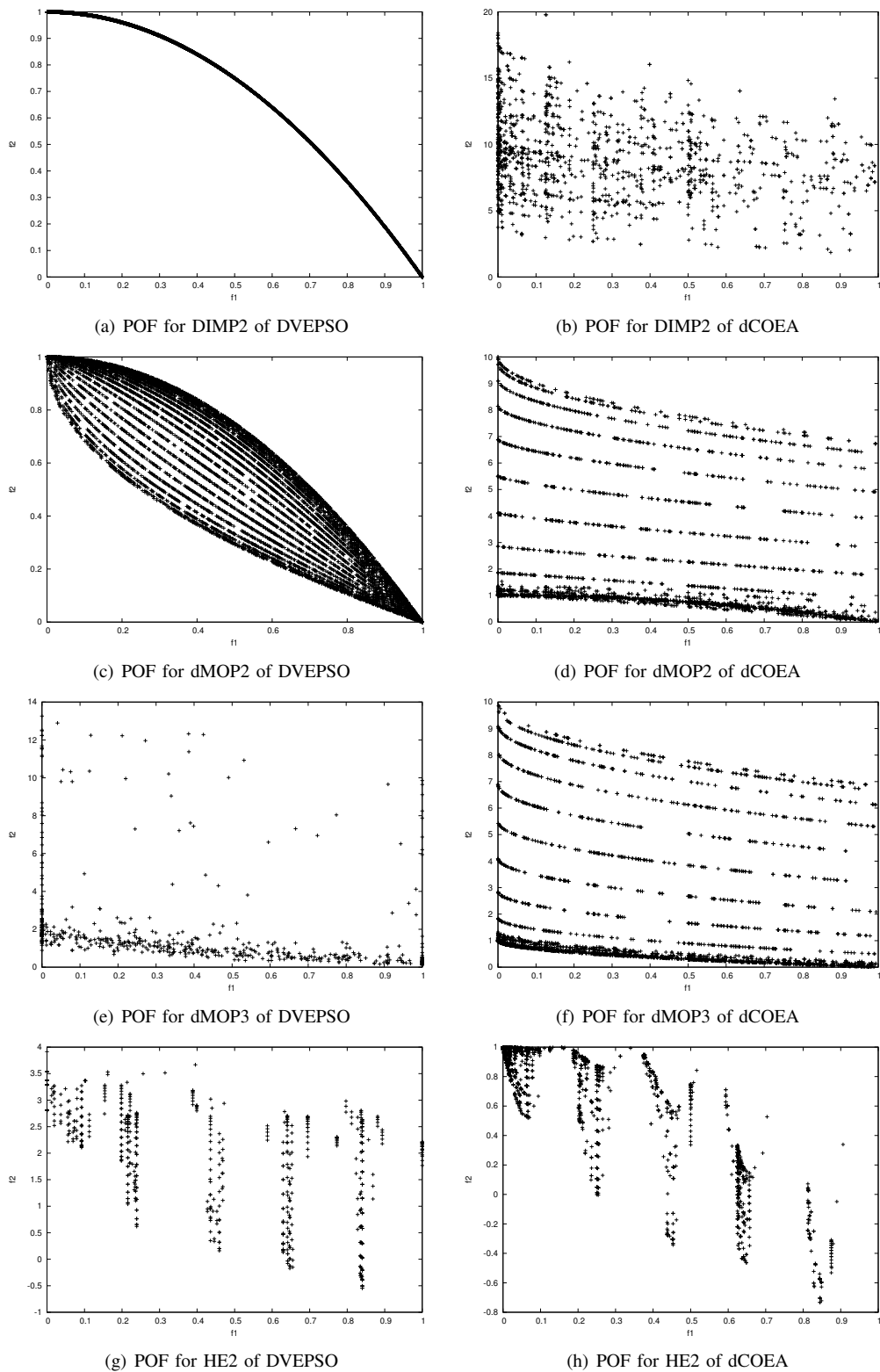


Fig. 1: POFs found by DVEPSO  $p_d$ - $g_r$  and dCOEA for various DMOOPs with  $\tau_t = 10$

[8] G. Pampara, A. Engelbrecht, and T. Cloete, "Cilib: A collaborative framework for computational intelligence algorithms - part i," in *In Proc. of IEEE World Congress on Computational Intelligence (WCCI)*, Hong Kong, 1-8 June 2008, pp. 1750-1757, source code available at: <http://www.cilib.net>. Last accessed on: 6 March 2011.

[9] CHPC, "Sun hybrid system," <http://www.chpc.ac.za/sun>. Last accessed online on: 20 December 2011.

[10] W. Koo, C. Goh, and K. Tan, "A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment," *Memetic Computing*, vol. 2, no. 2, pp. 87-110, 2010.

TABLE IV: Wins and Losses of FDA3<sub>Camara</sub>, DIMP2 and dMOP2 for DVEPSO and dCOEA

DMOOP	n <sub>t</sub>	τ <sub>t</sub>	PM	Results	Algorithm			
					DVEPSO			dCOEA
					d-r	d-s	s-n	
FDA3	10	10	acc	Wins	1	1	1	0
FDA3	10	10	acc	Losses	0	0	0	3
FDA3	10	10	acc	Diff	1	1	1	-3
FDA3	10	10	acc	Rank	<b>1</b>	<b>1</b>	<b>1</b>	<b>4</b>
FDA3	10	50	acc	Wins	0	0	0	3
FDA3	10	50	acc	Losses	1	1	1	0
FDA3	10	50	acc	Diff	-1	-1	-1	3
FDA3	10	50	acc	Rank	<b>2</b>	<b>2</b>	<b>2</b>	<b>1</b>
FDA3	10	10	stab	Wins	1	1	1	0
FDA3	10	10	stab	Losses	0	0	0	3
FDA3	10	10	stab	Diff	1	1	1	-3
FDA3	10	10	stab	Rank	<b>1</b>	<b>1</b>	<b>1</b>	<b>4</b>
FDA3	10	10	NS	Wins	0	0	0	3
FDA3	10	10	NS	Losses	1	1	1	0
FDA3	10	10	NS	Diff	-1	-1	-1	3
FDA3	10	10	NS	Rank	<b>2</b>	<b>2</b>	<b>2</b>	<b>1</b>
FDA3	10	50	NS	Wins	0	0	0	3
FDA3	10	50	NS	Losses	1	1	1	0
FDA3	10	50	NS	Diff	-1	-1	-1	3
FDA3	10	50	NS	Rank	<b>2</b>	<b>2</b>	<b>2</b>	<b>1</b>
DIMP2	10	10	acc	Wins	2	2	1	0
DIMP2	10	10	acc	Losses	0	0	2	3
DIMP2	10	10	acc	Diff	2	2	-1	-3
DIMP2	10	10	acc	Rank	<b>1</b>	<b>1</b>	3	4
DIMP2	10	50	acc	Wins	1	1	1	0
DIMP2	10	50	acc	Losses	0	0	0	3
DIMP2	10	50	acc	Diff	1	1	1	-3
DIMP2	10	50	acc	Rank	<b>1</b>	<b>1</b>	<b>1</b>	4
DIMP2	10	10	stab	Wins	2	2	1	0
DIMP2	10	10	stab	Losses	0	0	2	3
DIMP2	10	10	stab	Diff	2	2	-1	-3
DIMP2	10	10	stab	Rank	<b>1</b>	<b>1</b>	3	4
DIMP2	10	50	stab	Wins	0	0	0	0
DIMP2	10	50	stab	Losses	0	0	0	0
DIMP2	10	50	stab	Diff	0	0	0	0
DIMP2	10	50	stab	Rank	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
DIMP2	10	10	NS	Wins	0	0	0	3
DIMP2	10	10	NS	Losses	1	1	1	0
DIMP2	10	10	NS	Diff	-1	-1	-1	3
DIMP2	10	10	NS	Rank	<b>2</b>	<b>2</b>	<b>2</b>	<b>1</b>
DIMP2	10	50	NS	Wins	0	0	0	3
DIMP2	10	50	NS	Losses	1	1	1	0
DIMP2	10	50	NS	Diff	-1	-1	-1	3
DIMP2	10	50	NS	Rank	<b>2</b>	<b>2</b>	<b>2</b>	<b>1</b>

TABLE V: Wins and Losses of dMOP3 for DVEPSO and dCOEA

n <sub>t</sub>	τ <sub>t</sub>	PM	Results	Algorithm			
				DVEPSO			dCOEA
				d-r	d-s	s-n	
10	10	acc	Wins	0	0	0	3
10	10	acc	Losses	1	1	1	0
10	10	acc	Diff	-1	-1	-1	3
10	10	acc	Rank	<b>2</b>	<b>2</b>	<b>2</b>	<b>1</b>
10	50	acc	Wins	0	0	0	3
10	50	acc	Losses	1	1	1	0
10	50	acc	Diff	-1	-1	-1	3
10	50	acc	Rank	<b>2</b>	<b>2</b>	<b>2</b>	<b>1</b>
10	10	stab	Wins	0	0	0	3
10	10	stab	Losses	1	1	1	0
10	10	stab	Diff	-1	-1	-1	3
10	10	stab	Rank	<b>2</b>	<b>2</b>	<b>2</b>	<b>1</b>
10	50	stab	Wins	0	0	0	3
10	50	stab	Losses	1	1	1	0
10	50	stab	Diff	-1	-1	-1	3
10	50	stab	Rank	<b>2</b>	<b>2</b>	<b>2</b>	<b>1</b>
10	10	NS	Wins	1	1	1	0
10	10	NS	Losses	0	0	0	3
10	10	NS	Diff	1	1	1	-3
10	10	NS	Rank	<b>1</b>	<b>1</b>	<b>1</b>	4
10	50	NS	Wins	1	1	1	0
10	50	NS	Losses	0	0	0	3
10	50	NS	Diff	1	1	1	-3
10	50	NS	Rank	<b>1</b>	<b>1</b>	<b>1</b>	4

TABLE VI: Wins and Losses of HE2 for DVEPSO and dCOEA

n <sub>t</sub>	τ <sub>t</sub>	PM	Results	Algorithm			
				DVEPSO			dCOEA
				d-r	d-s	s-n	
10	10	acc	Wins	0	0	2	3
10	10	acc	Losses	2	2	1	0
10	10	acc	Diff	-2	-2	1	3
10	10	acc	Rank	<b>3</b>	<b>3</b>	<b>2</b>	<b>1</b>
10	50	acc	Wins	0	0	2	3
10	50	acc	Losses	2	2	1	0
10	50	acc	Diff	-2	-2	1	3
10	50	acc	Rank	<b>3</b>	<b>3</b>	<b>2</b>	<b>1</b>
10	10	stab	Wins	0	0	0	1
10	10	stab	Losses	0	0	1	0
10	10	stab	Diff	0	0	-1	1
10	10	stab	Rank	<b>2</b>	<b>2</b>	<b>4</b>	<b>1</b>
10	50	stab	Wins	0	0	1	2
10	50	stab	Losses	2	1	0	0
10	50	stab	Diff	-2	-1	1	2
10	50	stab	Rank	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>
10	10	NS	Wins	2	2	1	0
10	10	NS	Losses	0	0	2	3
10	10	NS	Diff	2	2	-1	-3
10	10	NS	Rank	<b>1</b>	<b>1</b>	3	4
10	50	NS	Wins	2	3	1	0
10	50	NS	Losses	1	0	2	3
10	50	NS	Diff	1	3	-1	-3
10	50	NS	Rank	<b>2</b>	<b>1</b>	3	4

TABLE VII: Overall Wins and Losses for DVEPSO and dCOEA

n <sub>t</sub>	τ <sub>t</sub>	PM	Results	Algorithm			
				DVEPSO			dCOEA
				d-r	d-s	s-n	
10	10	acc	Wins	4	4	5	6
10	10	acc	Losses	3	3	4	9
10	10	acc	Diff	1	1	1	-3
10	10	acc	Rank	<b>1</b>	<b>1</b>	<b>1</b>	4
10	50	acc	Wins	1	1	3	9
10	50	acc	Losses	4	4	3	3
10	50	acc	Diff	-3	-3	0	6
10	50	acc	Rank	<b>3</b>	<b>3</b>	<b>2</b>	<b>1</b>
10	10	stab	Wins	4	4	3	4
10	10	stab	Losses	1	1	4	9
10	10	stab	Diff	3	3	-1	-5
10	10	stab	Rank	<b>1</b>	<b>1</b>	3	4
10	50	stab	Wins	0	0	1	5
10	50	stab	Losses	3	2	1	0
10	50	stab	Diff	-3	-2	0	5
10	50	stab	Rank	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>
10	10	NS	Wins	3	3	2	9
10	10	NS	Losses	3	3	5	6
10	10	NS	Diff	0	0	-3	3
10	10	NS	Rank	<b>2</b>	<b>2</b>	<b>4</b>	<b>1</b>
10	50	NS	Wins	3	4	2	9
10	50	NS	Losses	4	5	3	6
10	50	NS	Diff	-1	-1	-1	3
all	all	NS	Rank	<b>2</b>	<b>2</b>	<b>2</b>	<b>1</b>

[11] M. Farina, K. Deb, and P. Amato, "Dynamic multiobjective optimization problems: test cases, approximations, and applications," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 5, pp. 425–442, Oct. 2004.

[12] C.-K. Goh and K. Tan, "A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 103–127, February 2009.

[13] M. Helbig and A. P. Engelbrecht, "Archive management for dynamic multi-objective optimisation problems using vector evaluated particle swarm optimisation," in *Evolutionary Computation (CEC), 2011 IEEE Congress on*, New Orleans, U.S.A., June 2011, pp. 2047–2054.

[14] M. Cámara, J. Ortega, and F. de Toro, "Approaching dynamic multi-objective optimization problems by using parallel evolutionary algorithms," in *Advances in Multi-Objective Nature Inspired Computing*, ser. Studies in Computational Intelligence. Springer Berlin / Heidelberg, 2010, vol. 272, pp. 63–86.

[15] M. C. Sola, "Parallel processing for dynamic multi-objective optimization," Ph.D. dissertation, Universidad de Granada, Dept. of Computer Architecture and Computer Technology, Universidad de Granada, Spain, April 2010.