# Analysing the Performance of Dynamic Multi-objective Optimisation Algorithms

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

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

***Abstract*—Dynamic multi-objective optimisation problems (DMOOPs) have more than one objective, with at least one objective changing over time. Since at least two of the objectives are normally in conflict with one another, a single solution does not exist and the goal of the algorithm is to track a set of trade-off solutions over time. Analysing the performance of a dynamic multi-objective optimisation algorithm (DMOA) is not a trivial task. For each environment (before a change occurs) the DMOA has to find a set of solutions that are both diverse and as close as possible to the optimal trade-off solution set. In addition, the DMOA has to track the changing set of trade-off solutions over time. Approaches used to analyse the performance of dynamic single-objective optimisation algorithms (DSOAs) and DMOAs do not provide any information about the ability of the algorithms to track the changing optimum. Therefore, this paper introduces a new approach to analyse the performance of DMOAs and applies this approach to the results obtained by five DMOAs. In addition, it compares the new analysis approach to another approach that does not take the tracking ability of the DMOAs into account. The results indicate that the new analysis approach provide additional information, measuring the ability of the algorithm to find good performance measure values while tracking the changing optima.**

## I. Introduction

Many real-world problems have more than one objective, with at least two objectives that are in conflict with one another and at least one objective that changes over time. These kind of problems are referred to as dynamic multi-objective optimisation problems (DMOOPs).

When solving DMOOPs, a dynamic multi-objective optimisation algorithm (DMOA) has to find a set of solutions that are both diverse and as close as possible to the optimal set of solutions. In addition, the DMOA has to track the optimal set of solutions over time, i.e. be able to find a new set of solutions when the environment changes. Therefore, when analysing the performance of a DMOA, the approach should enable the analysis of the ability of the DMOA to obtain good performance measure values while tracking the optimal set of solutions over time.

However, approaches used to analyse the performance of dynamic single-objective optimisation algorithms (DSOAs) and DMOAs do not take into account the ability of the algorithm to track the changing optima over time. This paper proposes a new approach to analyse the performance of a DMOA in such a way that the results provide information with regards to how well the algorithm performs over time, while coping with a changing environment. The goal of this new approach is to provide a first step towards a uniform comparison of DMOAs.

The rest of the paper is outlined as follows: Section II provides definitions that are required for the rest of the paper and discusses various approaches that have been proposed to analyse the performance of DMOAs and DSOAs. The experiments conducted for this study are discussed in Section III. Section IV discusses the results obtained by the DMOAs. In addition, the resuls are analysed using two wins-losses approaches. Section V discusses whether this new approach does provide additional information with regards to the tracking ability of a DMOA. Finally, the research is concluded in Section VI.

## II. Background

Section II-A provides definitions that are required for the rest of the paper. Approaches proposed in the literature to analyse the performance of DSOAs and DMOAs are discussed in Section II-B.

### A. Definitions

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

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

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

Two solutions are compared using vector domination, defined as follows:

**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

- $\mathbf{x}_1$ is at least as good as $\mathbf{x}_2$ for all the objectives, i.e. $f_k(\mathbf{x}_1) \leq f_k(\mathbf{x}_2)$, $\forall k = 1, \ldots, n_k$; and
- $\mathbf{x}_1$ is strictly better than $\mathbf{x}_2$ for at least one objective, i.e. $\exists i = 1, \ldots, n_k \colon f_i(\mathbf{x_1}) < f_i(\mathbf{x_2})$ .

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

**Definition 2. Pareto-optimal**: A decision vector $\mathbf{x}^*$ is Pareto-optimal if there does not exist a decision vector $\mathbf{x} \neq \mathbf{x}^* \in F$ that dominates it, i.e. $\nexists k \colon f_k(\mathbf{x}) \prec f_k(\mathbf{x}^*)$. If $\mathbf{x}^*$ is Pareto-optimal, the objective vector, $\mathbf{f}(\mathbf{x}^*)$, is also Pareto-optimal.

The Pareto-optimal set (POS) is the set that contains all the Pareto-optimal decision vectors, defined as:

**Definition 3. Pareto-optimal Set**: The POS is the set of all Pareto-optimal decision vectors, i.e.

$$POS = \{\mathbf{x}^* \in F \,|\, \nexists \mathbf{x} \in F \colon \mathbf{x} \prec \mathbf{x}^*\} \quad (2)$$

The set of corresponding objective vectors is referred to as the Pareto-optimal front (POF), defined as follows:

**Definition 4. Pareto-optimal Front**: For the objective vector $\mathbf{f}(\mathbf{x})$ and the POS, the POF, $POF \subseteq O$, is defined as

$$POF = \{\mathbf{f} = (f_1(\mathbf{x}^*), \ldots, f_{n_k m}(\mathbf{x}^*)) \,|\, \mathbf{x}^* \in POS\} \quad (3)$$

When solving a DMOOP, the goal of a DMOA is to track the POF over time, i.e. for each time step, to find

$$POF(t) = \{\mathbf{f}(t) = (f_1(\mathbf{x}^*, \mathbf{w}_1(t)), f_2(\mathbf{x}^*, \mathbf{w}_2(t)), \ldots, f_{n_k}(\mathbf{x}^*, \mathbf{w}_{n_k}(t))) \,|\, \mathbf{x}^* \in POS(t)\} \quad (4)$$

### *B. Performance Evaluation of DMOAs*

Two main approaches are used to analyse the performance of DSOAs and DMOAs, namely:

- calculating the performance measure at each time step just before a change in the environment occurred and then calculating the average of these performance measure values. Algorithms' performance is then analysed based on the average performance measure value and the standard deviation of the performance measure value [1], [2], [3], [4], [5], [6], [7].
- calculating wins and losses based on the average performance measure value (refer to wins-losses$_A$ in Section III-D). If two algorithms' performance are compared, for each function a statistical test (such as Kruskal-Wallis) is performed on the performance measure values for each time step to determine whether there is a statistical significant difference between the two algorithms' performance. If there is a statistical significant difference, their average performance measure values are compared and the algorithm with the best average value is awarded a win and the other algorithm a loss [8], [9], [10]. This approach has also been applied to static multi-objective optimisation (MOO) [11].

However, both of these approaches do not provide information with regards to how well or poorly an algorithm has tracked the changing optima. Both approaches only use the average performance measure value calculated over all time steps (just before a change in the environment occurred) and do not analyse how the algorithms' performance compare at each time step. Even though the wins and losses approach do take the performance measure values at each time step into account when the statistical test is performed, it does not take these values into account when assigning the wins or losses to the algorithms - the wins and losses are awarded based on the average performance measure values of the two algorithms that are compared with one another.

Using the average value may indicate whether an algorithm has a good or poor overall performance. For dynamic single-objective optimisation problem (DSOOP) the offline error and the best-of-generations or collective-mean-fitness are used to measure the performance of DSOAs [12]. However, even though a low offline error may indicate that an algorithm performed reasonably well over all time steps, it does not provide information with regards to how well the algorithm performed with regards to the other algorithms at the various time steps or environments during the run.

One approach proposed in the literature to analyse an algorithm's tracking behaviour over time is plotting the performance measure value over time [1], [13], [14], [15], [16], [17], [18], [19], [20]. However, it is diffcult to compare the tracking ability of multiple algorithms when using this aproach, since the graph can become cluttered. In addition, this approach is time consuming, since a graph would have to be drawn for each performance measure and for each problem. It will be difficult to use this approach to determine how well an algorithm tracks the changing optima over time for a specific group of problems, such as multi-modal problems or problems where the optima changes in a certain way (e.g. the DMOOP Types as defined by Farina *et al.* [21]).

## III. Experimental Setup

This section discusses the experimental setup of the experiments conducted for this study. The DMOAs are discussed in Section III-A. The benchmark functions and performance measures used to evaluate the performance of the DMOAs are discussed in Sections III-B and III-C respectively. Section III-D discusses the two approaches that are used to analyse the performance of the DMOAs.

### *A. DMOO Algorithms*

Five DMOAs were selected for this study, namely:

- Two non-dominated sorting genetic algorithm II (NSGA-II) [24], [25] versions adapted by Deb *et al.* [26] for dynamic multi-objective optimisation (DMOO):

**TABLE I:** Parameter values of the DMOEAs

| DMOEA | $n_i$ | $p_c$ | $p_m$ | Other Parameters |
|---|---|---|---|---|
| DNSGA-II-A | 100 | 0.9 | $\frac{4}{x_k}$ | polynomial mutation [22], simulated binary crossover [23], $\zeta = 30$ |
| DNSGA-II-B | 100 | 0.8 | $\frac{5}{x_k}$ | polynomial mutation [22], simulated binary crossover [23], $\zeta = 30$ |
| dCOEA | $100n_x$ | 0.5 | $\frac{9}{L}$ | $SC_{ratio} = 0.7$, $R_{size} = 5$ |

- the dynamic non-dominated sorting genetic algorithm II-A (DNSGA-II-A), where after a change in the environment occurred a percentage of the population is replaced with randomly created individuals.
- DNSGA-II-B (DNSGA-II-B), where after a change in the environment occurred a percentage of the population is replaced with mutated solutions of randomly selected existing solutions.

- the dynamic cooperative-competitive evolutionary algorithm (dCOEA) proposed by Goh and Tan [6].
- the multi-objective particle swarm optimisation (MOPSO) algorithm [27] extended by Lechuga [28] for DMOO, referred to as dynamic MOPSO (DMOPSO) in this paper.
- the dynamic vector evaluated particle swarm optimisation (DVEPSO) algorithm, a cooperative particle swarm optimisation (PSO)-based algorithm proposed by Greeff[1] and Engelbrecht [29].

The source code of the dCOEA algorithm was obtained from the authors of [6]. The source code of the static NSGA-II algorithm was obtained from [30] and was adapted for DMOO according to [26]. The source code of MOPSO was obtained from the authors of [28], adapted for DMOO according to [28] and extended using sentry particles to detect a change in the environment. The source code of DVEPSO was developed by Helbig and Engelbrecht and is available in the opensource library, CIlib [31].

Each algorithm was optimised for the benchmark functions, as discussed in [32]. For each algorithm, the algorithm's performance for various parameter values was evaluated on the 2-objective benchmark functions discussed in Section III-B using the performance measures discussed in Section III-C. The optimised parameters for each DMOA are presented in Tables I and II. In Tables I and II, $n_i$ represents the number of individuals or particles. In Table I, $p_c$ and $p_m$ represent the probability for crossover and mutation respectively, $\zeta$ is the % of individuals that are replaced by either mutated or randomly created individuals, $SC_{ratio}$ represents the ratio between the stochastic solutions and the competitors, and $R_{size}$ is the number of archive solutions stored in temporal memory. In Table II, $c_1$ and $c_2$ are positive acceleration coefficients, $w$ is the inertia weight and $\sigma_{share}$ represents the distance that the particles should remain from each other when using fitness sharing. $p_s$-$g_r$ is a standard pbest update where only the objective that the sub-swarm is optimising is taken into account and a gbest (the best position found so far by all particles of the swarm) update where Pareto-dominance information is used and if two solutions are non-dominated with regards to each other, one solution is randomly selected as the gbest. $cl$ represents the clamping boundary violation management approach that places any particle that violates a specific boundary of the search space on or close to the violated boundary. $ra$-$t$ is the random knowledge sharing topology that uses tournament selection to select the gbest and $ac$ represents an archive management strategy that clears all solutions from the archive after a change in the environment has occurred. $ri$-30-$c$ is the approach that is used to respond to a change in the environment, namely re-initialising 30% of the particles of the sub-swarm whose objective function has changed.

[1]Greeff is the maiden name of Helbig

**TABLE II:** Parameter values of the PSO-based DMOAs

| DMOA | $n_i$ | $c_1$ | $c_2$ | $w$ | Other Parameters |
|---|---|---|---|---|---|
| DMOPSO | 100 | 1.49 | 1.49 | 0.72 | self-adapting $\sigma_{share}$ |
| DVEPSO | 100 | 1.49 | 1.49 | 0.72 | guide update: $p_s$-$g_r$,<br>boundary violation management: $cl$,<br>knowledge sharing: $ra$-$t$,<br>response applied to archive: $ac$,<br>response applied to particles: $ri$-30-$c$ |

## B. Benchmark Functions

Based on the analysis of DMOOPs in [32], [33], [34], eighteen benchmark functions were selected of various DMOOP Types [21] to compare the performance of the five DMOAs. These functions include a modified version of DIMP2 [7] with a concave POF (referred to as DIMP2 in the rest of the article), ZJZ [35], FDA2 [21], FDA2$_{Camara}$ [36], FDA3 [21], FDA3$_{Camara}$ [36], FDA5 [21], FDA5$_{iso}$ [34], FDA5$_{dec}$ [34], dMOP2 [6], dMOP3 [6], dMOP2$_{iso}$ [34], dMOP2$_{dec}$ [34], HE1 and HE2 [8], and HE6, HE7 and HE9 [33], [32]. For each benchmark function the following severity of change ($n_t$) and frequency of change ($\tau_t$) combinations were used: $n_t = 10$ and $\tau_t = 10$, $n_t = 10$ and $\tau_t = 25$, $n_t = 10$ and $\tau_t = 50$, $n_t = 1$ and $\tau_t = 10$, and $n_t = 20$ and $\tau_t = 10$.

## C. Performance Measures

Based on an analysis of performance measures in [32], [37], [38], three performance measures were selected for this study, namely:

- the number of non-dominated solutions (NS) in the found POF.
- the alternative accuracy measure ($acc_{alt}$) [5], referred to in this article as $acc$. A low $acc$ value indicates good performance.
- stability ($stab$) [5] that quantifies the effect of changes in the environment on $acc$ of the DMOA. A low $stab$ value indicates good performance.

### D. Statistical Analyses

This section discusses the two approaches that are used to analyse the performance of the DMOAs.

The general calculation of wins and losses that is performed for each performance measure (refer to Section III-C) is presented in Algorithm 1. For the Kruskal-Wallis test the average performance measure values of each time step just before a change in the environment occurred was used. In Algorithm 1, $Diff = \#wins - \#losses$, where $Diff$ is the difference between the number of wins and number of losses assigned to the DMOA and $pm$ refers to performance measures values. Two approaches to assign wins and losses are discussed below. The first approach, wins-losses$_A$, was proposed for DMOO by Helbig and Engelbrecht [8], [9], [37] and used by various authors for static optimisation [10], [11]. The second approach, wins-losses$_B$, is the approach that is proposed in this paper.

**Algorithm 1** Calculation of wins and losses

```
for each DMOOP do
    for each n_t-τ_t combination do
        perform Kruskal-Wallis tests on pm
        if statistical significant difference then
            for each DMOA-pair do
                perform Mann-Whitney U test on pm
                if statistical significant difference then
                    assign wins and losses
                end if
            end for
        end if
    end for
    calculate Diff for the n_t-τ_t combination
end for
calculate Diff for the DMOOP
```

**Approach A** (wins-losses$_A$)
This approach does not take into account the tracking ability of the DMOA. If the Mann-Whitney U test indicates that there is a significant difference between the performance measure values of the DMOAs, the overall average value of the performance measure is used to determine the winning DMOA. The DMOA with the best performance measure average over all time steps (just before a change in the environment) is awarded a win and the losing DMOA is awarded a loss.

**Approach B** (wins-losses$_B$)
This approach takes into account the tracking ability of a DMOA. If the Mann-Whitney U test indicates that there is a significant difference, the average performance measure value of each time step just before a change in the environment occurred are used to award wins and losses. At each time step just before a change in the environment occurred, the average performance measure values of the two DMOAs are compared. The DMOA with the best performance measure value is awarded a win and the other DMOA is awarded a loss. In order to ensure that a DMOA that tracks the changing POF very well for a DMOOP does not lead to skewed results, the number of wins and losses are normalised as follows:

$$\#wins_{norm} = \frac{\#wins}{\#changes}$$
$$\#losses_{norm} = \frac{\#losses}{\#changes} \tag{5}$$

where $\#changes$ represents the number of changes that occurred during the entire run. In situations where the time steps at which a change in the environment occurs are unknown, the algorithm should log detected changes during the run.

## IV. Results

This section discusses the results obtained by the five DMOAs. The overall performance of the DMOAs is discussed in Section IV-A. The DMOAs performance with regards to the various Types of DMOOPs is discussed in Section IV-B.

### A. Overall Performance

The overall performance of the DMOAs measured over all performance measures, environments and DMOOPs is presented in Table III. In all tables presented in this section, w-l refers to the wins-losses approach, A refers to wins-losses$_A$, B refers to wins-losses$_B$, DN-II-A and DN-II-B refers to DNSGA-II-A and DNSGA-II-B respectively, dC refers to dCOEA, DM refers to DMOPSO and DV refers to DVEPSO.

**TABLE III:** Overall Wins and Losses by the various DMOOAs using wins-losses$_A$ and wins-losses$_B$

| w-l | Results | DMOO Algorithm | | | | |
|---|---|---|---|---|---|---|
| | | **DN-II-A** | **DN-II-B** | **dC** | **DM** | **DV** |
| A | Wins | 346 | 390 | 284 | 288 | 326 |
| A | Losses | 292 | 248 | 566 | 320 | 274 |
| A | Diff | 54 | 142 | -282 | -32 | 52 |
| A | Rank | 2 | **1** | 5 | 4 | 3 |
| B | Wins | 326.11 | 354.105 | 296.495 | 364.075 | 416.7 |
| B | Losses | 257.43 | 233.845 | 522.025 | 377.87 | 367.315 |
| B | Diff | 68.68 | 120.26 | -225.53 | -13.795 | 49.385 |
| B | Rank | 2 | **1** | 5 | 4 | 3 |

Considering the overall performance of the DMOAs, the two wins-losses approaches did not produce different results. The ranking of the DMOAs are similar for both wins-losses approaches. However, using wins-losses$_B$ leads to all DMOAs, except DNSGA-II-A, obtaining a smaller difference between the number of wins and losses. For both wins-losses approaches dCOEA and DMOPSO obtained more losses than wins, indicating a poor performance. All other DMOAs were awarded more wins than losses.

Table IV presents the overall performance of the DMOAs with regards to each performance measure. The DMOAs' performance is measured over all environments and DMOOPs.

The data in Table IV for wins-losses$_A$ was presented in [37]. The two wins-losses approaches lead to a different ranking of the algorithms for both $acc$ and $stab$. Using wins-losses$_B$, DVEPSO and DMOPSO switched rank, with DVEPSO obtaining the best rank. Therefore, even though DMOPSO obtained the best average performance measure values, DVEPSO obtained the best performance measure values for the various time steps, i.e. DVEPSO tracked the changing

**TABLE IV:** Overall Wins and Losses for Various Performance Measures using wins-losses$_A$ and wins-losses$_B$

| w-l | PM | Results | DMOO Algorithm | | | | |
|---|---|---|---|---|---|---|---|
| | | | DN-II-A | DN-II-B | dC | DM | DV |
| A | $acc$ | Wins | 94 | 109 | 129 | 118 | 119 |
| A | $acc$ | Losses | 120 | 104 | 164 | 86 | 95 |
| A | $acc$ | Diff | -26 | 5 | -35 | 32 | 24 |
| A | $acc$ | Rank | 4 | 3 | 5 | **1** | 2 |
| B | $acc$ | Wins | 89.67 | 102.4 | 141.57 | 140.62 | 171.2 |
| B | $acc$ | Losses | 129.12 | 118.55 | 148.99 | 126.19 | 120.61 |
| B | $acc$ | Diff | -39.33 | -15.6 | -6.43 | 14.62 | 51.2 |
| B | $acc$ | Rank | 5 | 4 | 3 | 2 | **1** |
| A | $stab$ | Wins | 67 | 94 | 39 | 117 | 96 |
| A | $stab$ | Losses | 89 | 66 | 200 | 39 | 50 |
| A | $stab$ | Diff | -22 | 28 | -161 | 78 | 46 |
| A | $stab$ | Rank | 4 | 3 | 5 | **1** | 2 |
| B | $stab$ | Wins | 71.91 | 81.15 | 53.24 | 138.31 | 91.88 |
| B | $stab$ | Losses | 65.39 | 60.87 | 165.84 | 43.32 | 103.07 |
| B | $stab$ | Diff | 6.91 | 21.15 | -111.76 | 95.31 | -11.12 |
| B | $stab$ | Rank | 3 | 2 | 5 | **1** | 4 |
| A | $NS$ | Wins | 185 | 187 | 116 | 53 | 111 |
| A | $NS$ | Losses | 83 | 78 | 202 | 195 | 129 |
| A | $NS$ | Diff | 102 | 109 | -86 | -142 | -18 |
| A | $NS$ | Rank | 2 | **1** | 4 | 5 | 3 |
| B | $NS$ | Wins | 164.94 | 169.52 | 100.5 | 84.14 | 152.32 |
| B | $NS$ | Losses | 61.07 | 53.69 | 206.51 | 206.06 | 142.09 |
| B | $NS$ | Diff | 103.94 | 116.52 | -105.5 | -121.86 | 10.32 |
| B | $NS$ | Rank | 2 | **1** | 4 | 5 | 3 |

POF better than DMOPSO. Furthermore, using wins-losses$_A$, dCOEA ranked last for $acc$, but with wins-losses$_B$ dCOEA ranked third. Therefore, dCOEA tracked the changing POF reasonably well, even though it obtained not such a good overall performance measure average. DNSGA-II-B obtained more wins than losses with wins-losses$_A$, but was awarded more losses than wins with wins-losses$_B$. Therefore, DNSGA-II-B obtained a reasonably good $acc$ average over all time steps, but not such a good $acc$ average for each of the different time steps. For $stab$, DVEPSO performed well using wins-losses$_A$, obtaining more wins than losses and being ranked second. However, using wins-losses$_B$ DVEPSO obtained more losses than wins and was awarded the second worst rank. Therefore, DVEPSO did not obtain good $stab$ values while tracking the changing POF. In contrast, DNSGA-II-A obtained 22 more losses than wins using wins-losses$_A$, but 6.91 more wins than losses using wins-losses$_B$, with a rank of 4 and 3 for wins-losses$_A$ and wins-losses$_B$ respectively. For $NS$, DVEPSO performed much better using wins-losses$_B$ than using wins-losses$_A$. Using wins-losses$_A$ DVEPSO was awarded 18 more losses than wins, but using wins-losses$_B$ DVEPSO was awarded 10.32 more wins than losses.

The overall performance of the DMOAs with regards to each environment type are presented in Table V. For Table V, the DMOAs' performance was measured over all performance measures and DMOOPs.

The number of wins and losses obtained by each algorithm varies depending on which wins-losses approach was used, causing a few minor changes in the ranks obtained by the DMOAs for the various types of environments. Furthermore, for $n_t = 10$ and $\tau_t = 10$, using wins-losses$_A$ dCOEA obtained the third best rank with 22 more losses than wins, and using wins-losses$_B$ dCOEA was awarded 25.57 more wins than losses and obtained the second best rank. These results indicate that for $n_t = 10$ and $\tau_t = 10$ dCOEA obtained good performance measure values for the various time steps, i.e. obtaining good performance measure values while tracking the changing POF.

**TABLE V:** Overall Wins and Losses for Various Frequencies and Severities of Change using wins-losses$_A$ and wins-losses$_B$

| w-l | $\mathbf{n_t}$ | $\tau_t$ | PM | Results | DMOO Algorithm | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | DN-II-A | DN-II-B | dC | DM | DV |
| A | 10 | 10 | all | Wins | 65 | 77 | 91 | 62 | 89 |
| A | 10 | 10 | all | Losses | 97 | 84 | 113 | 97 | 59 |
| A | 10 | 10 | all | Diff | -32 | -7 | -22 | -35 | 30 |
| A | 10 | 10 | all | Rank | 4 | 2 | 3 | 5 | **1** |
| B | 10 | 10 | all | Wins | 61.67 | 67.4 | 118.57 | 77.62 | 136.2 |
| B | 10 | 10 | all | Losses | 88.12 | 79.55 | 93.99 | 114.19 | 86.61 |
| B | 10 | 10 | all | Diff | -26.33 | -11.6 | 25.57 | -36.38 | 50.2 |
| B | 10 | 10 | all | Rank | 4 | 3 | 2 | 5 | **1** |
| A | 10 | 25 | all | Wins | 72 | 80 | 40 | 46 | 55 |
| A | 10 | 25 | all | Losses | 37 | 31 | 114 | 60 | 51 |
| A | 10 | 25 | all | Diff | 35 | 49 | -74 | -14 | 4 |
| A | 10 | 25 | all | Rank | 2 | **1** | 5 | 4 | 3 |
| B | 10 | 25 | all | Wins | 65.55 | 72.475 | 39.375 | 62.825 | 62.9 |
| B | 10 | 25 | all | Losses | 33.45 | 30.525 | 104.625 | 68.3 | 68.225 |
| B | 10 | 25 | all | Diff | 32.55 | 42.475 | -64.625 | -5.175 | -5.1 |
| B | 10 | 25 | all | Rank | 2 | **1** | 5 | 4 | 3 |
| A | 10 | 50 | all | Wins | 69 | 76 | 47 | 35 | 49 |
| A | 10 | 50 | all | Losses | 36 | 32 | 93 | 63 | 52 |
| A | 10 | 50 | all | Diff | 33 | 44 | -46 | -28 | -3 |
| A | 10 | 50 | all | Rank | 2 | **1** | 5 | 4 | 3 |
| B | 10 | 50 | all | Wins | 66.5 | 71.25 | 39.25 | 56.0 | 58.75 |
| B | 10 | 50 | all | Losses | 30.25 | 29.75 | 91.75 | 66.85 | 71.15 |
| B | 10 | 50 | all | Diff | 36.5 | 42.25 | -51.75 | -10.0 | -12.25 |
| B | 10 | 50 | all | Rank | 2 | **1** | 5 | 3 | 4 |
| A | 1 | 10 | all | Wins | 66 | 72 | 49 | 87 | 73 |
| A | 1 | 10 | all | Losses | 72 | 62 | 115 | 42 | 56 |
| A | 1 | 10 | all | Diff | -6 | 10 | -66 | 45 | 17 |
| A | 1 | 10 | all | Rank | 4 | 3 | 5 | **1** | 2 |
| B | 1 | 10 | all | Wins | 60.37 | 63.62 | 45.36 | 94.36 | 77.44 |
| B | 1 | 10 | all | Losses | 57.63 | 53.38 | 110.6 | 54.8 | 64.74 |
| B | 1 | 10 | all | Diff | 3.37 | 10.62 | -64.64 | 40.36 | 13.44 |
| B | 1 | 10 | all | Rank | 4 | 3 | 5 | **1** | 2 |
| A | 20 | 10 | all | Wins | 74 | 85 | 57 | 58 | 60 |
| A | 20 | 10 | all | Losses | 50 | 39 | 131 | 58 | 56 |
| A | 20 | 10 | all | Diff | 24 | 46 | -74 | 0 | 4 |
| A | 20 | 10 | all | Rank | 2 | **1** | 5 | 4 | 3 |
| B | 20 | 10 | all | Wins | 72.02 | 79.36 | 53.94 | 73.27 | 81.41 |
| B | 20 | 10 | all | Losses | 47.98 | 40.64 | 121.06 | 73.73 | 76.59 |
| B | 20 | 10 | all | Diff | 25.02 | 39.36 | -67.06 | 0.27 | 5.41 |
| B | 20 | 10 | all | Rank | 2 | **1** | 5 | 4 | 3 |

### B. *Performance per DMOOP Type*

This section presents the obtained results for the various DMOOP Types as defined by Farina *et al.* [21].

*1) Type II DMOOPs:* Table VI presents the performance of the DMOAs for Type II DMOOPs with regards to each performance measure. With Type II DMOOPs, both the POF and POS change over time [21].

Using wins-losses$_A$, DVEPSO performed really well with regards to $acc$ and $stab$, obtaining the best rank for these two performance measures. Furthermore, DVEPSO obtained more wins than losses for both measures. However, using wins-losses$_B$ DVEPSO obtained much more wins than losses for $acc$ than with wins-losses$_A$. Therefore, DVEPSO obtained good $acc$ average values for the various time steps and a good overall $acc$ average. Wins-losses$_A$ indicates that both DNSGA-II algorithms were awarded more wins than losses for $acc$. However, using wins-losses$_B$, both algorithms obtained more losses than wins. Therefore, wins-losses$_B$ indicates that both DNSGA-II algorithms did not perform as well with regards to $stab$ over time while tracking the changing POF. For $stab$, using wins-losses$_B$, it can be seen that DVEPSO struggled to obtain a good $stab$ value while tracking the changing POF. In addition, using wins-losses$_A$ it seems as though there was a difference in performance between the DNSGA-II algorithms for $stab$. However, wins-losses$_B$ indicates that both DNSGA-II algorithms obtained a good $stab$ value while tracking the changing POF.

**TABLE VI:** Overall Wins and Losses solving Type II DMOOPs for Various Performance Measures using wins-losses$_A$ and wins-losses$_B$

| w-l | $n_t$ | $\tau_t$ | PM | Results | DMOO Algorithm | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | DN-II-A | DN-II-B | dC | DM | DV |
| A | all | all | $acc$ | Wins | 55 | 55 | 36 | 56 | 63 |
| A | all | all | $acc$ | Losses | 43 | 41 | 106 | 41 | 34 |
| A | all | all | $acc$ | Diff | 12 | 14 | -70 | 15 | 29 |
| A | all | all | $acc$ | Rank | 4 | 3 | 5 | 2 | **1** |
| B | all | all | $acc$ | Wins | 48.53 | 45.8 | 45.525 | 66.58 | 106.05 |
| B | all | all | $acc$ | Losses | 56.01 | 59.15 | 98.995 | 63.365 | 34.965 |
| B | all | all | $acc$ | Diff | -7.48 | -13.35 | -53.47 | 3.215 | 71.085 |
| B | all | all | $acc$ | Rank | 3 | 4 | 5 | 2 | **1** |
| A | all | all | $stab$ | Wins | 36 | 45 | 18 | 53 | 59 |
| A | all | all | $stab$ | Losses | 43 | 30 | 104 | 20 | 14 |
| A | all | all | $stab$ | Diff | -7 | 15 | -86 | 33 | 45 |
| A | all | all | $stab$ | Rank | 4 | 3 | 5 | 2 | **1** |
| B | all | all | $stab$ | Wins | 46.91 | 46.375 | 21.42 | 64.235 | 41.275 |
| B | all | all | $stab$ | Losses | 28.915 | 29.685 | 82.305 | 20.5 | 58.81 |
| B | all | all | $stab$ | Diff | 17.995 | 16.69 | -60.885 | 43.735 | -17.535 |
| B | all | all | $stab$ | Rank | 2 | 3 | 5 | **1** | 4 |
| A | all | all | $NS$ | Wins | 96 | 92 | 60 | 31 | 90 |
| A | all | all | $NS$ | Losses | 48 | 50 | 116 | 105 | 50 |
| A | all | all | $NS$ | Diff | 48 | 42 | -56 | -74 | 40 |
| A | all | all | $NS$ | Rank | **1** | 2 | 4 | 5 | 3 |
| B | all | all | $NS$ | Wins | 89.49 | 89.735 | 56.15 | 48.06 | 101.37 |
| B | all | all | $NS$ | Losses | 39.99 | 37.475 | 110.585 | 129.315 | 67.44 |
| B | all | all | $NS$ | Diff | 49.5 | 52.26 | -54.435 | -81.255 | 33.93 |
| B | all | all | $NS$ | Rank | 2 | **1** | 4 | 5 | 3 |

The DMOAs' performance for Type II DMOOPs with regards to each environment is presented in Table VII.

Using wins-losses$_B$, DVEPSO obtained the top rank for three environments. For $n_t = 20$ and $\tau_t = 10$, DVEPSO obtained the second best rank with wins-losses$_A$, but the best rank using wins-losses$_B$. In addition, using wins-losses$_A$ DNSGA-II-B outranked DNSGA-II-A in each environment. However, using wins-losses$_B$ DNSGA-II-A outranked DNSGA-II-B in each environment. Therefore, even though DNSGA-II-B obtained a beter overall average performance measure value than DNSGA-II-A, DNSGA-II-A obtained better performance measure values over time and therefore performed better than DNSGA-II-B when tracking the POF over time.

*2) Type III DMOOPs:* The performance of the DMOAs for Type III DMOOPs with regards to each performance measure is presented in Table VIII. With Type III DMOOPs, the POF changes over time while the POS remains unchanged [21].

**TABLE VII:** Overall Wins and Losses solving Type II DMOOPs for Various Frequencies and Severities of Change using wins-losses$_A$ and wins-losses$_B$

| w-l | $n_t$ | $\tau_t$ | PM | Results | DMOO Algorithm | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | DN-II-A | DN-II-B | dC | DM | DV |
| A | 10 | 10 | all | Wins | 39 | 39 | 29 | 31 | 52 |
| A | 10 | 10 | all | Losses | 32 | 29 | 69 | 41 | 19 |
| A | 10 | 10 | all | Diff | 7 | 10 | -40 | -10 | 33 |
| A | 10 | 10 | all | Rank | 3 | 2 | 5 | 4 | **1** |
| B | 10 | 10 | all | Wins | 38.91 | 38.04 | 32.21 | 40.93 | 58.77 |
| B | 10 | 10 | all | Losses | 29.47 | 29.97 | 57.7 | 54.4 | 37.32 |
| B | 10 | 10 | all | Diff | 9.44 | 8.07 | -25.49 | -13.47 | 21.45 |
| B | 10 | 10 | all | Rank | 2 | 3 | 5 | 4 | **1** |
| A | 25 | 10 | all | Wins | 39 | 40 | 13 | 22 | 38 |
| A | 25 | 10 | all | Losses | 16 | 16 | 66 | 35 | 19 |
| A | 25 | 10 | all | Diff | 23 | 24 | -53 | -13 | 19 |
| A | 25 | 10 | all | Rank | 2 | **1** | 5 | 4 | 3 |
| B | 10 | 25 | all | Wins | 35.8 | 36.15 | 16.475 | 31.725 | 43.325 |
| B | 10 | 25 | all | Losses | 17.325 | 18.5 | 55.775 | 43.45 | 28.425 |
| B | 10 | 25 | all | Diff | 18.475 | 17.65 | -39.3 | -11.725 | 14.9 |
| B | 10 | 25 | all | Rank | **1** | 2 | 5 | 4 | 3 |
| A | 50 | 10 | all | Wins | 37 | 39 | 15 | 11 | 28 |
| A | 50 | 10 | all | Losses | 12 | 12 | 52 | 33 | 21 |
| A | 50 | 10 | all | Diff | 25 | 27 | -37 | -22 | 7 |
| A | 50 | 10 | all | Rank | 2 | **1** | 5 | 4 | 3 |
| B | 10 | 50 | all | Wins | 37.9 | 37.9 | 17.5 | 19.65 | 37.4 |
| B | 10 | 50 | all | Losses | 14.35 | 15.5 | 48.8 | 36.95 | 34.75 |
| B | 10 | 50 | all | Diff | 23.55 | 22.4 | -31.3 | -17.3 | 2.65 |
| B | 10 | 50 | all | Rank | **1** | 2 | 5 | 4 | 3 |
| A | 1 | 10 | all | Wins | 33 | 34 | 23 | 47 | 55 |
| A | 1 | 10 | all | Losses | 47 | 40 | 67 | 23 | 15 |
| A | 1 | 10 | all | Diff | -14 | -6 | -44 | 24 | 40 |
| A | 1 | 10 | all | Rank | 4 | 3 | 5 | 2 | **1** |
| B | 1 | 10 | all | Wins | 32.2 | 30.94 | 23.67 | 50.39 | 55.93 |
| B | 1 | 10 | all | Losses | 37.62 | 36.9 | 63.68 | 31.59 | 23.34 |
| B | 1 | 10 | all | Diff | -5.42 | -5.96 | -40.01 | 18.8 | 32.59 |
| B | 1 | 10 | all | Rank | 3 | 4 | 5 | 2 | **1** |
| A | 20 | 10 | all | Wins | 39 | 40 | 34 | 29 | 39 |
| A | 20 | 10 | all | Losses | 27 | 24 | 72 | 34 | 24 |
| A | 20 | 10 | all | Diff | 12 | 16 | -38 | -5 | 15 |
| A | 20 | 10 | all | Rank | 3 | **1** | 5 | 4 | 2 |
| B | 20 | 10 | all | Wins | 40.12 | 38.88 | 33.24 | 36.18 | 53.27 |
| B | 20 | 10 | all | Losses | 26.15 | 25.44 | 65.93 | 46.79 | 37.38 |
| B | 20 | 10 | all | Diff | 13.97 | 13.44 | -32.69 | -10.61 | 15.89 |
| B | 20 | 10 | all | Rank | 2 | 3 | 5 | 4 | **1** |

DVEPSO performed worse for $stab$ when using wins-losses$_B$. Therefore, DVEPSO struggled to obtain a good $stab$ value while tracking the changing POF. Furthermore, even though DNSGA-II-A performed much worse than DNSGA-II-B when comparing the average $stab$ value (using wins-losses$_A$), both algorithms performed similar with regards to $stab$ when their tracking ability was taken into account (using wins-losses$_B$). For $NS$, DVEPSO and DMOPSO obtained more losses than wins for both wins-losses$_A$ and wins-losses$_B$. However, using wins-losses$_B$ both DMOAs obtained a better $Diff$ value. In contrast, using wins-losses$_B$ dCOEA obtained a much worse $Diff$ value than with wins-losses$_A$.

Table IX presents the DMOAs' performance for Type III DMOOPs with regards to each environment.

Using wins-losses$_A$, DMOPSO was awarded more losses than wins in two environments, namely $n_t = 10$ and $\tau_t = 25$, and $n_t = 10$ and $\tau_t = 50$. However, with wins-losses$_B$

**TABLE VIII:** Overall Wins and Losses solving Type III DMOOPs for Various Performance Measures using wins-losses$_A$ and wins-losses$_B$

| w-l | $\mathbf{n_t}$ | $\tau_t$ | PM | Results | DMOO Algorithm | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | **DN-II-A** | **DN-II-B** | **dC** | **DM** | **DV** |
| A | all | all | $acc$ | Wins | 32 | 43 | 70 | 53 | 36 |
| A | all | all | $acc$ | Losses | 60 | 50 | 43 | 32 | 49 |
| A | all | all | $acc$ | Diff | -28 | -7 | 27 | 21 | -13 |
| A | all | all | $acc$ | Rank | 5 | 3 | **1** | 2 | 4 |
| B | all | all | $acc$ | Wins | 34.67 | 45.96 | 73.22 | 62.77 | 47.38 |
| B | all | all | $acc$ | Losses | 58.33 | 48.04 | 38.78 | 51.23 | 67.62 |
| B | all | all | $acc$ | Diff | -23.66 | -2.08 | 34.44 | 11.54 | -20.24 |
| B | all | all | $acc$ | Rank | 5 | 3 | **1** | 2 | 4 |
| A | all | all | $stab$ | Wins | 25 | 34 | 15 | 51 | 36 |
| A | all | all | $stab$ | Losses | 28 | 22 | 82 | 7 | 22 |
| A | all | all | $stab$ | Diff | -3 | 12 | -67 | 44 | 14 |
| A | all | all | $stab$ | Rank | 4 | 3 | 5 | **1** | 2 |
| B | all | all | $stab$ | Wins | 24.04 | 30.62 | 16.05 | 61.66 | 31.415 |
| B | all | all | $stab$ | Losses | 22.155 | 18.315 | 72.425 | 14.135 | 36.755 |
| B | all | all | $stab$ | Diff | 1.885 | 12.305 | -56.375 | 47.525 | -5.34 |
| B | all | all | $stab$ | Rank | 3 | 2 | 5 | **1** | 4 |
| A | all | all | $NS$ | Wins | 75 | 79 | 48 | 16 | 21 |
| A | all | all | $NS$ | Losses | 20 | 15 | 69 | 71 | 64 |
| A | all | all | $NS$ | Diff | 55 | 64 | -21 | -55 | -43 |
| A | all | all | $NS$ | Rank | 2 | **1** | 3 | 5 | 4 |
| B | all | all | $NS$ | Wins | 63.92 | 67.82 | 22.51 | 26.57 | 32.445 |
| B | all | all | $NS$ | Losses | 9.44 | 5.735 | 81.35 | 58.49 | 58.25 |
| B | all | all | $NS$ | Diff | 54.48 | 62.085 | -58.84 | -31.92 | -25.805 |
| B | all | all | $NS$ | Rank | 2 | **1** | 5 | 4 | 3 |

DMOPSO was awarded more wins than losses for all environments. Therefore, DMOPSO performed reasonably well in all environments while tracking the changing POF.

Due to space limitations, figures that indicate the relationship between the DMOAs' performance and the frequency and severity of change are not included in this paper. However, the figures are presented in [39].

## V. Discussion

The results in Section IV indicate that wins-losses$_B$ does provide more information with regards to the DMOAs' performance than wins-losses$_A$. Therefore, in this section the performance of the five DMOAs solving FDA2$_{Camara}$ is analysed with the two wins-losses approaches to investigate whether wins-losses$_B$ does provide additional information with regards to the tracking ability of a DMOA.

The true POF of FDA2$_{Camara}$ changes over time and is illustrated in Figure 1 for $n_t = 10$ and $\tau_t = 10$. The POF of FDA2$_{Camara}$ changes in a cyclic manner over time, moving from the top line to bottom line for certain time steps and from the bottom line to the top line for other time steps.

The $acc$ values obtained by the DMOAs for each time step are presented in Figure 2. From Figure 2 it can be seen that:

1) dCOEA performed the best, obtaining a low $acc$ value throughout all time steps.
2) The DNSGA-II algorithms performed not so well during the first 10 time steps but then obtained a good $acc$ value throughout the rest of the time steps. In adition, even though the DNSGA-II algorithms performed quite similar, DNSGA-II-A did outperform DNSGA-II-B during time steps 10 to 29 and 60 to 80. Therefore, DNSGA-II-A did track the changing POF better than DNSGA-II-B.
3) DVEPSO obtained a good $acc$ value for the first 10 time steps, but then its $acc$ value started to increase. However, DVEPSO outperformed DMOPSO during the entire time period.

**TABLE IX:** Overall Wins and Losses solving Type III DMOOPs for Various Frequencies and Severities of Change using wins-losses$_A$ and wins-losses$_B$

| w-l | $\mathbf{n_t}$ | $\tau_t$ | PM | Results | DMOO Algorithm | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | **DN-II-A** | **DN-II-B** | **dC** | **DM** | **DV** |
| A | 10 | 10 | all | Wins | 22 | 27 | 38 | 25 | 17 |
| A | 10 | 10 | all | Losses | 27 | 21 | 27 | 23 | 31 |
| A | 10 | 10 | all | Diff | -5 | 6 | 11 | 2 | -14 |
| A | 10 | 10 | all | Rank | 4 | 2 | **1** | 3 | 5 |
| B | 10 | 10 | all | Wins | 21.22 | 24.35 | 33.77 | 30.15 | 18.55 |
| B | 10 | 10 | all | Losses | 22.14 | 17.34 | 21.48 | 26.16 | 40.92 |
| B | 10 | 10 | all | Diff | -0.92 | 7.01 | 12.29 | 3.99 | -22.37 |
| B | 10 | 10 | all | Rank | 4 | 2 | **1** | 3 | 5 |
| A | 10 | 25 | all | Wins | 28 | 33 | 24 | 19 | 17 |
| A | 10 | 25 | all | Losses | 19 | 14 | 40 | 23 | 25 |
| A | 10 | 25 | all | Diff | 9 | 19 | -16 | -4 | -8 |
| A | 10 | 25 | all | Rank | 2 | **1** | 5 | 3 | 4 |
| B | 10 | 25 | all | Wins | 25.55 | 31.55 | 20.45 | 25.75 | 20.9 |
| B | 10 | 25 | all | Losses | 15.425 | 11.35 | 42.975 | 23.225 | 31.225 |
| B | 10 | 25 | all | Diff | 10.125 | 20.2 | -22.525 | 2.525 | -10.325 |
| B | 10 | 25 | all | Rank | 2 | **1** | 5 | 3 | 4 |
| A | 10 | 50 | all | Wins | 23 | 26 | 28 | 22 | 21 |
| A | 10 | 50 | all | Losses | 22 | 20 | 33 | 23 | 22 |
| A | 10 | 50 | all | Diff | 1 | 6 | -5 | -1 | -1 |
| A | 10 | 50 | all | Rank | 2 | **1** | 5 | 3 | 3 |
| B | 10 | 50 | all | Wins | 22.25 | 25.35 | 18.55 | 30.4 | 21.95 |
| B | 10 | 50 | all | Losses | 15.6 | 14.25 | 35.9 | 26.55 | 26.2 |
| B | 10 | 50 | all | Diff | 6.65 | 11.1 | -17.35 | 3.85 | -4.25 |
| B | 10 | 50 | all | Rank | 2 | **1** | 5 | 3 | 4 |
| A | 1 | 10 | all | Wins | 28 | 33 | 23 | 31 | 18 |
| A | 1 | 10 | all | Losses | 22 | 18 | 42 | 19 | 32 |
| A | 1 | 10 | all | Diff | 6 | 15 | -19 | 12 | -14 |
| A | 1 | 10 | all | Rank | 3 | **1** | 5 | 2 | 4 |
| B | 1 | 10 | all | Wins | 25.09 | 28.93 | 19.89 | 34.86 | 22.0 |
| B | 1 | 10 | all | Losses | 17.85 | 14.05 | 42.57 | 23.19 | 33.11 |
| B | 1 | 10 | all | Diff | 7.24 | 14.88 | -22.68 | 11.67 | -11.11 |
| B | 1 | 10 | all | Rank | 3 | **1** | 5 | 2 | 4 |
| A | 20 | 10 | all | Wins | 31 | 37 | 20 | 23 | 20 |
| A | 20 | 10 | all | Losses | 18 | 14 | 52 | 22 | 25 |
| A | 20 | 10 | all | Diff | 13 | 23 | -32 | 1 | -5 |
| A | 20 | 10 | all | Rank | 2 | **1** | 5 | 3 | 4 |
| B | 20 | 10 | all | Wins | 28.52 | 34.22 | 19.12 | 29.84 | 27.84 |
| B | 20 | 10 | all | Losses | 18.91 | 15.1 | 49.63 | 24.73 | 31.17 |
| B | 20 | 10 | all | Diff | 9.61 | 19.12 | -30.51 | 5.11 | -3.33 |
| B | 20 | 10 | all | Rank | 2 | **1** | 5 | 3 | 4 |

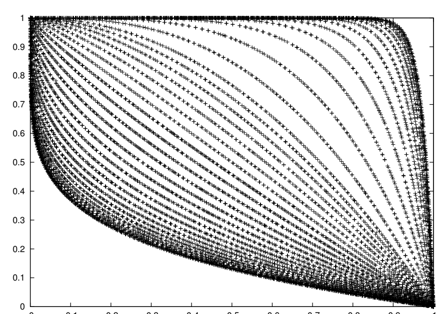


**Fig. 1:** True POF of FDA2$_{Camara}$ for $n_t = 10$ and $\tau_t = 10$

**TABLE X:** $acc$, wins-losses$_A$ and wins-losses$_B$ obtained by the DMOAs for FDA2$_{Camara}$

| w-l | Results | DMOO Algorithm | | | | |
|---|---|---|---|---|---|---|
| | | **DN-II-A** | **DN-II-B** | **dC** | **DM** | **DV** |
| - | $acc$ | 0.1832 | 0.2222 | 0.0474 | 1.1996 | 0.5069 |
| A | Wins | 3 | 2 | 4 | 0 | 0 |
| A | Losses | 1 | 2 | 0 | 3 | 3 |
| A | Diff | 2 | 0 | 4 | -3 | -3 |
| A | Rank | 2 | 3 | **1** | 4 | 4 |
| B | Wins | 2.82 | 1.92 | 4 | 0 | 1.26 |
| B | Losses | 1.18 | 2.08 | 0 | 4 | 2.74 |
| B | Diff | 1.64 | -0.16 | 4 | -4 | -1.48 |
| B | Rank | 2 | 3 | **1** | 5 | 4 |

4) DMOPSO performed the worst, obtaining the worst $acc$ value for all time steps.

The average $acc$ values for FDA2$_{Camara}$ obtained by the DMOAs and the results obtained using wins-losses$_A$ and wins-losses$_B$ for $acc$ are presented in Table X. The following are observed:

- According to the average $acc$ values, dCOEA performed the best, DNSGA-II-A obtained the second best value, DNSGA-II-B the third best value, DVEPSO the second worst value and DMOPSO the worst value. Even though the ranks obtained by algorithms using wins-losses$_B$ correspond to the ranks for the average $acc$ values, this will not always be the case. If an algorithm performs really well for certain time steps and really poor for other time steps, the ranking according to the average $acc$ values and the ranking according to wins-losses$_B$ may differ.
- dCOEA obtained the top rank using both wins-losses$_A$ and wins-losses$_B$. This result corresponds with item 1 listed above.
- DNSGA-II-A outperformed DNSGA-II-B using both wins-losses$_A$ and wins-losses$_B$. This result corresponds with item 2 listed above.
- According to wins-losses$_A$ DMOPSO and DVEPSO performed similar, with each DMOA obtaining 3 losses and no wins and both being ranked last. However, using wins-losses$_B$, DMOPSO obtained 4 losses and no wins, while DVEPSO obtained 2.74 losses and 1.26 wins. In addition, using wins-losses$_B$, DVEPSO obtained the second worst rank and DMOPSO the worst rank. Therefore, wins-losses$_B$ indicates that DVEPSO performed much better than DMOPSO while tracking the changing POF. Results obtained with wins-losses$_A$ do not correspond with items 3 and 4 listed above. However, results obtained using wins-losses$_B$ do correspond with items 3 and 4 listed above.

These observations do confirm that wins-losses$_B$ does provide additional information with regards to how well an algorithm performs over time. Therefore, it does provide information about the tracking ability of a DMOA.

## VI. Conclusion

When solving dynamic optimisation problems, one of the goals of an algorithm is to track the changing optima over time. Therefore, when analysing the performance of an algorithm, the algorithm's tracking ability (i.e. performance while tracking the changing optima) should be taken into account.

This paper proposed a new approach that takes the tracking ability of an algorithm into consideration. It awards wins and losses for each time step just before a change occurs in the environment. Therefore, the results will indicate how well an algorithm performed with regards to a specific performance measure while tracking the changing optima over time. This new approach was compared against another approach that only awards wins and losses based on the overall average performance measure value. The results indicated that the new approach provides additional information and that it is more respresentative of an algorithm's performance over time. Therefore, this approach is applicable to analyse the performance of both DSOAs and DMOAs.

## References


[1] X. Hu and R. Eberhart, "Adaptive particle swarm optimization: detection and response to dynamic systems," in *Proc. of Congress on Evolutionary Computation*, Honolulu, Hawai, U.S.A., May 2002, pp. 1666–1670.

[2] L. Bui, J. Branke, and H. Abbass, "Multiobjective optimization for dynamic environments," in *Proc. of Congress on Evoluationary Computation*, vol. 3, Canberra, Australia, 2005, pp. 2349–2356.

[3] X. Li, J. Branke, and T. Blackwell, "Particle swarm with speciation and adaptation in a dynamic environment," in *Proc. of the annual conference on Genetic and evolutionary computation*, ser. GECCO '06, 2006, pp. 51–58.

[4] T. Blackwell and J. Branke, "Multiswarms, exclusion, and anti-convergence in dynamic environments," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 459–472, August 2006.

[5] M. Cámara, J. Ortega, and F. de Toro, "A single front genetic algorithm for parallel multi-objective optimization in dynamic environments," *Neurocomputing*, vol. 72, no. 16–18, pp. 3570–3579, 2007.

[6] 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.

[7] 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.

[8] M. Helbig and A. Engelbrecht, "Analyses of guide update approaches for vector evaluated particle swarm optimisation on dynamic multi-objective optimisation problems," in *Proc. of the World Congress on Computational Intelligence: Congress on Evolutionary Computation*, Brisbane, Australia, june 2012, pp. 1–8.

[9] ——, "Archive management for dynamic multi-objective optimisation problems using vector evaluated particle swarm optimisation," in *Proc. of Congress on Evolutionary Computation*, New Orleans, U.S.A., June 2011, pp. 2047–2054.

[10] K. Harrison, B. Ombuki-Berman, and A. Engelbrecht, "Knowledge transfer strategies for vector evaluated particle swarm optimization," Brock University, Department of Computer Science, Tech. Rep. CS-12-07, 2012.

[11] J. Grobler and A. Engelbrecht, "Hybridizing pso and de for improved vector evaluated multi-objective optimization," in *Proc. of Congress on Evolutionary Computation*, May 2009, pp. 1255–1262.

[12] T. Nguyen, S. Yang, and J. Branke, "Evolutionary dynamic optimization: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 6, pp. 1–24, October 2012.

[13] R. Eberhart and Y. Shi, "Tracking and optimizing dynamic systems with particle swarms," in *Proc. of Congress on Evolutionary Computation*, May 2001, pp. 94–100.

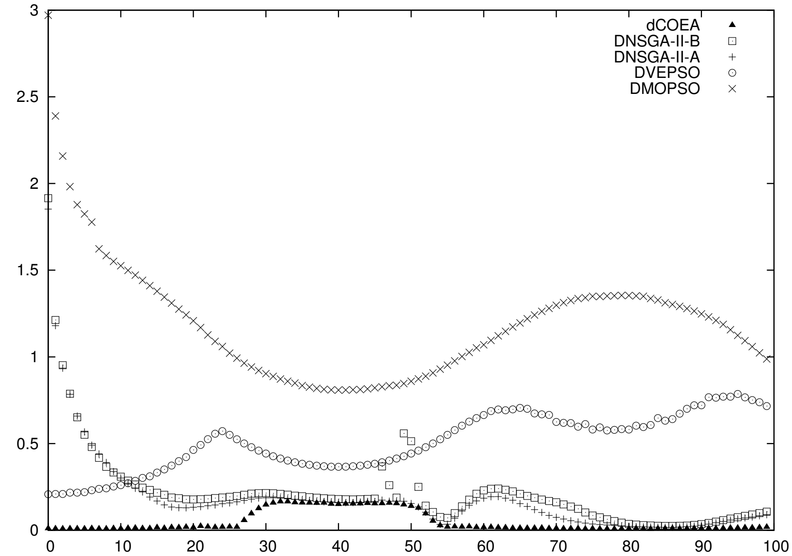


**Fig. 2:** $acc$ values obtained by the DMOAs for FDA2$_{Camara}$, with $n_t = 10$ and $\tau_t = 10$ for 1000 iterations


[14] X. Li and K. Dam, "Comparing particle swarms for tracking extrema in dynamic environments," in *Proc. of Congress on Evoluationary Computation*, vol. 3, Melbourne, Australia, 2003, pp. 1772–1779.

[15] X. Cui, T. Hardin, R. Ragade, T. Potok, and A. Elmaghraby, "Tracking non-stationary optimal solution by particle swarm optimizer," in *Proc. of International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and ACIS International Workshop on Self-assembling Wireless Networks*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 133–138.

[16] S. Yang, "Explicit memory schemes for evolutionary algorithms in dynamic environments," in *Evolutionary Computation in Dynamic and Uncertain Environments*, ser. Studies in Computational Intelligence, S. Yang, Y.-S. Ong, and Y. Jin, Eds. Springer Berlin/Heidelberg, 2007, vol. 51, pp. 3–28.

[17] Z. Avdagić, S. Konjicija, and S. Omanović, "Evolutionary approach to solving non-stationary dynamic multi-objective problems," in *Foundations of Computational Intelligence Volume 3*, ser. Studies in Computational Intelligence, A. Abraham, A.-E. Hassanien, P. Siarry, and A. Engelbrecht, Eds. Springer Berlin/Heidelberg, 2009, vol. 203, pp. 267–289.

[18] C. Azevedo and A. Araujo, "Generalized immigration schemes for dynamic evolutionary multiobjective optimization," in *Proc. of Congress on Evolutionary Computation*, June 2011, pp. 2033–2040.

[19] I. Hatzakis and D. Wallace, "Dynamic multi-objective optimization with evolutionary algorithms: a forward-looking approach," in *Proc. of the Conference on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2006, pp. 1201–1208.

[20] A. Isaacs, T. Ray, and W. Smith, "Memetic algorithm for dynamic bi-objective optimization problems," in *Proc. of Congress on Evolutionary Computation*, Trondheim, Norway, May 2009, pp. 1707–1713.

[21] 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, October 2004. [Online]. Available: http://www.cs.colostate.edu/ genitor/GECCO-2000/gecco2000mainpage.htm

[22] K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, Ltd, 2001.

[23] K. Deb and R. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, pp. 115–148, 1995.

[24] 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, April 2002.

[25] K. Deb, S. Agarwal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," Indian Institute of Technology Kanpur Kanpur Genetic Algorithms Laboratory (KanGAL), Kanpur, India, Tech. Rep. 200001, 2000.

[26] K. Deb, U. R. N., and S. Karthik, "Dynamic multi-objective optimization and decision-making using modied nsga-ii: a case study on hydro-thermal power scheduling," in *Proc. of International Conference on Evolutionary Multi-criterion optimization*, Matsushima, Japan, 2007, pp. 803–817.

[27] C. C. Coello and M. Lechuga, "MOPSO: a proposal for multiple objective particle swarm optimization," in *Proc. of Congress on Evolutionary Computation*, vol. 2, 2002, pp. 1051–1056.

[28] M. Lechuga, "Multi-objective optimisation using sharing in swarm optimisation algorithms," Ph.D. dissertation, University of Birmingham, July 2009. [Online]. Available: http://etheses.bham.ac.uk/303/

[29] M. Greeff and A. Engelbrecht, "Solving dynamic multi-objective problems with vector evaluated particle swarm optimisation," in *Proc. of World Congress on Computational Intelligence (WCCI): Congress on Evolutionary Computation*, Hong Kong, June 2008, pp. 2917–2924.

[30] K. G. A. Library, "Software developed at KanGAL." [Online]. Available: http://www.iitk.ac.in/kangal/codes.shtml

[31] G. Pampará, A. Engelbrecht, and T. Cloete, "Cilib: A collaborative framework for computational intelligence algorithms - part i," in *Proc. of World Congress on Computational Intelligence*, Hong Kong, 1-8 June 2008, pp. 1750–1757, source code available at: http://www.cilib.net.

[32] M. Helbig, "Solving dynamic multi-objective optimisation problems using vector evaluated particle swarm optimisation," Ph.D. dissertation, University of Pretoria, 2012.

[33] M. Helbig and A. Engelbrecht, "Dynamic multi-objective optimisation problems," *Submitted for review*, 2013.

[34] ——, "Benchmarks for dynamic multi-objective optimisation," in *Proc. of IEEE Symposium Series on Computational Intelligence*, Singapore, April 2013, pp. 84–91.

[35] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, "Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, Eds. Springer Berlin/Heidelberg, 2007, vol. 4403, pp. 832–846.

[36] 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, C. C. Coello, C. Dhaenens, and L. Jourdan, Eds. Springer Berlin/Heidelberg, 2010, vol. 272, pp. 63–86.

[37] M. Helbig and A. Engelbrecht, "Issues with performance measures for dynamic multi-objective optimisation," in *Proc. of IEEE Symposium Series on Computational Intelligence*, Singapore, April 2013, pp. 17–24.

[38] ——, "Performance measures for dynamic multi-objective optimisation algorithms," *Submitted for review*, 2013.

[39] ——, "Analysing the performance of DMOAs: Appendix," 2013. [Online]. Available: http://csir-za.academia.edu/MardeHelbig