

Performance Measures for Dynamic Multi-objective Optimisation Algorithms

Mardé Helbig^{a,b}, Andries P. Engelbrecht^b

^a*Meraka Institute, CSIR, Scientia, Meiring Naudé Road, 0184, Brummeria, Pretoria, South Africa.
Corresponding author. Tel: +27 12 841 3523. Cell: +27 79 590 3803*

^b*Department of Computer Science, University of Pretoria, Pretoria, South Africa*

Abstract

When algorithms solve dynamic multi-objective optimisation problems (DMOOPs), performance measures are required to quantify the performance of the algorithm and to compare one algorithm's performance against that of other algorithms. However, for dynamic multi-objective optimisation (DMOO) there are no standard performance measures. This article provides an overview of the performance measures that have been used so far. In addition, issues with performance measures that are currently being used in the DMOO literature are highlighted.

Keywords: Dynamic multi-objective optimisation, performance measures

1. Introduction

In order to determine whether an algorithm can solve dynamic multi-objective optimisation problems (DMOOPs) efficiently, the algorithm's performance should be quantified with functions referred to as performance measures. The set of performance measures chosen for a comparative study of dynamic multi-objective optimisation algorithms (DMOAs) influence the results and effectiveness of the study. Comprehensive overviews of performance measures used for dynamic single-objective optimisation (DSOO) [2; 50; 53] and static MOO (SMOO) [4; 11; 45] do exist in the literature. However, a lack of standard performance measures is one of the main problems in the

Email addresses: mhelbig@csir.co.za (Mardé Helbig), engel@cs.up.ac.za (Andries P. Engelbrecht)

field of dynamic multi-objective optimisation (DMOO). Furthermore, a comprehensive overview of performance measures used for DMOO does not exist in the literature. Therefore, the selection of appropriate performance measures to use for DMOO is not a trivial task. This article seeks to address this problem by:

- providing a comprehensive overview of performance measures currently used in the DMOO literature, and
- highlighting issues with performance measures that are currently used to evaluate the performance of DMOAs.

It should be noted that this article does not discuss DMOOPs. However, the reader is referred to [32] for a comprehensive overview of DMOOPs.

The rest of the article is outlined as follows: Formal definitions of concepts that are required as background for this article are provided in Section 2. Section 3 discusses performance measures that have been used for multi-objective optimisation (MOO) and that have been adapted for DMOO. Performance measures that are currently used in the DMOO literature are discussed in Section 4. Section 5 highlights issues with current performance measures that are frequently used to measure the performance of DMOAs. This section also makes proposals as to which performance measures can be used for DMOO. Finally, the conclusions are discussed in Section 6.

2. Definitions

This section provides definitions with regards to MOO and DMOO that are required as background for the rest of the article.

2.1. Multi-objective Optimisation

The objectives of a multi-objective optimisation problem (MOOP) are normally in conflict with one another, i.e. improvement in one objective leads to a worse solution for at least one other objective. Therefore, when solving MOOPs the definition of optimality used for single-objective optimisation problems (SOOPs) has to be adjusted. For MOOPs, when one decision vector dominates another, the dominating decision vector is considered as a better decision vector.

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

Using the notation above, decision vector domination is defined as follows:

Definition 1. Decision 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 $\mathbf{x}_1 < \mathbf{x}_2$, iff $f_k(\mathbf{x}_1) \leq f_k(\mathbf{x}_2)$, $\forall k = 1, \dots, n_k$ and $\exists i = 1, \dots, n_k: f_i(\mathbf{x}_1) < f_i(\mathbf{x}_2)$.

The best decision vectors are called Pareto-optimal, defined as follows:

Definition 2. Pareto-optimal: A decision vector \mathbf{x}^* is Pareto-optimal if $\nexists \mathbf{x}: f_k(\mathbf{x}) < f_k(\mathbf{x}^*)$, where $\mathbf{x} \neq \mathbf{x}^* \in F$. If \mathbf{x}^* is Pareto-optimal, the objective vector, $\mathbf{f}(\mathbf{x}^*)$, is also Pareto-optimal.

The set of all the Pareto-optimal decision vectors are referred to as the Pareto-optimal set (POS), defined as:

Definition 3. Pareto-optimal Set: The POS, P^* , is formed by the set of all Pareto-optimal decision vectors, i.e. $P^* = \{\mathbf{x}^* \in F \mid \nexists \mathbf{x} \in F: \mathbf{x} < \mathbf{x}^*\}$.

The POS contains the best trade-off solutions for the MOOP. The set of corresponding objective vectors are called the Pareto-optimal front (POF) or Pareto front, which is defined as follows:

Definition 4. Pareto-optimal Front: For the objective vector $\mathbf{f}(\mathbf{x})$ and the POS P^* , the POF, $PF^* \subseteq O$, is defined as $PF^* = \{\mathbf{f} = (f_1(\mathbf{x}^*), f_2(\mathbf{x}^*), \dots, f_{n_k}(\mathbf{x}^*)) \mid \mathbf{x}^* \in P^*\}$.

2.2. Dynamic Multi-objective Optimisation

Using the notation defined in Section 2.1, an unconstrained DMOOP can be mathematically defined as:

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

where $\mathbf{W}(t)$ is a matrix of time-dependent control parameters of an objective function at time t , $\mathbf{W}(t) = (\mathbf{w}_1(t), \dots, \mathbf{w}_{n_k}(t))$, n_x is the number of decision variables, $\mathbf{x} = (x_1, \dots, x_{n_x}) \in \mathbb{R}^{n_x}$ and $\mathbf{x} \in [\mathbf{x}_{min}, \mathbf{x}_{max}]^{n_x}$ refers to the boundary constraints.

In order to solve a DMOOP the goal of an algorithm is to track the POF over time, i.e. for each time step, find

$$PF^*(t) = \{\mathbf{f}(t) = (f_1(\mathbf{x}^*, \mathbf{w}_1(t)), f_2(\mathbf{x}^*, \mathbf{w}_2(t)), \dots, f_{n_k}(\mathbf{x}^*, \mathbf{w}_{n_k}(t))) \mid \mathbf{x}^* \in P^*(t)\} \quad (2)$$

3. Static MOO Performance Measures

Performance measures enable the quantification of an algorithm's performance with regards to a specific requirement, such as the number of non-dominated solutions found, closeness to the true POF (accuracy), and the diversity or spread of the solutions. According to Zitzler *et al.* [76], a performance measure is defined as follows:

Definition 5. Performance Measure: An m -ary performance measure, P , is a function $P : \Omega^m \rightarrow \mathbb{R}$, that assigns each of the m approximated POFs, $POF_1^*, POF_2^*, \dots, POF_m^*$, a real value $P(POF_1^*, POF_2^*, \dots, POF_m^*)$.

This section discusses static MOO measures that have been adapted in the literature and used in DMOO. The discussion on static MOO performance measures is by no means complete, and the reader is referred to [13; 41; 48; 71] for detailed information on performance measures used for static MOO.

Outperformance relations that are used to evaluate performance measures are discussed in Section 3.1. Section 3.2 discusses performance measures that quantify an

algorithm's performance with regards to accuracy, i.e. the found non-dominated solutions' (POF^*) closeness to the true POF (POF). Performance measures that calculate the diversity or spread of the solutions found are discussed in Section 3.3. Section 3.4 discusses performance measures that calculate the overall quality of the solutions found, taking into account both accuracy and diversity.

3.1. Outperformance Relations

When an algorithm solves a MOOP where the objective functions are in conflict with one another, the algorithm tries to find the best possible set of non-dominated solutions, i.e. a set of solutions that are as close as possible to POF and where the solutions are diverse and evenly spread along POF^* . However, once POF^* is found, a decision maker selects one of these solutions according to his/her own defined preferences.

Hansen and Jaszkiwicz [29] introduced an outperformance relation under the following assumptions:

- The preferences of the decision maker are not known a priori.
- Let POF_A^* and POF_B^* be two approximated POFs. Then, POF_A^* outperforms POF_B^* if the decision maker finds:
 - a better solution in POF_A^* than in POF_B^* for specific preferences, and
 - for another set of preferences the solution selected from POF_A^* is not worse than solutions found in POF_B^* .
- All possible preferences of the decision maker can be modeled with functions, referred to as a set of utility functions, U .

Definition 6. Outperformance Relation (subject to a set of utility functions): Let A and B be two sets representing approximations of the same POF. Let u denote an utility function and $\{U|A > B\} \subseteq U$ denote a subset of utility functions for which A is better than B , i.e. $\{U|A > B\} = \{u \in U|u(A) > u(B)\}$. Then A outperforms B , denoted as $A \mathcal{O}_U B$, if $U(A > B) \neq \emptyset$ and $U(B > A) = \emptyset$.

The weakest assumption about the decision maker's preferences that is generally made when solving MOOPs is that the utility function is compatible with the dominance relation, i.e. the decision maker prefers non-dominated solutions [52]. Therefore, the decision maker can limit his/her selection of the best solution to the set of

non-dominated solutions (ND) in $A \cup B$, i.e. $ND(A \cup B)$. Based on the dominance relation assumption, Hansen and Jaszekiewicz [29] defined three dominance based relations, namely weak, strong and complete outperformance. These three relations are presented in Table 1.

Table 1: Outperformance relations defined by Hansen and Jaszekiewicz [29]

Relation	Symbol	Definition
Weak outperformance	O_W	$A \neq B$ and $ND(A \cup B) = A$
Strong outperformance	O_S	$A \neq B$, $ND(A \cup B) = A$ and $B \setminus ND(A \cup B) \neq \emptyset$
Complete outperformance	O_C	$A \neq B$, $ND(A \cup B) = A$ and $B \cap ND(A \cup B) = \emptyset$

The outperformance relations only identify whether one set is better than another set, but doesn't quantify with how much the one set is better than the other. Therefore, based on the outperformance relations, Hansen and Jaszekiewicz [29] defined compatibility and weak compatibility with an outperformance relation. In addition to the outperformance relations, Knowles [41] introduced the concepts of monotony and relativity that are important when evaluating the efficiency of performance measures. These concepts are presented in Table 2. In Table 2, PM refers to performance measure.

Table 2: Performance measure concepts defined by Hansen and Jaszekiewicz [29] and Knowles [41]

Concept	Definition
Weak compatibility	If $A O B$, PM evaluates A as being not worse than B
Compatibility	If $A O B$, PM evaluates A as being better than B
Monotony	Let C contain a new non-dominated solution. Then, PM evaluates $A \cup C$ as being not worse than A
Relativity	Let D contain solutions of POF . Then, PM evaluates D as being better than any POF^*

Weak compatibility with O_W is sufficient for weak monotony and weak relativity [41].

From the above definitions it should be clear that $O_C \subset O_S \subset O_W$, i.e. complete outperformance is the strongest outperformance relation, and weak outperformance is

the weakest outperformance relation. Therefore, it is most difficult for a performance measure to be compatible with O_W , and the easiest for a performance measure to be compatible with O_C [41]. According to Knowles [41], performance measures that are not compatible with these outperformance relations, cannot be relied on to provide evaluations that are compatible with Pareto dominance.

If a performance measure is compatible with the concept of monotony, it will not decrease a set's evaluation if a new non-dominated point is added, which adheres to the goal of finding a diverse set of solutions. Furthermore, if a performance measure does not adhere to the concept of relativity, it will evaluate some approximation sets as being better than the true POF, which is not accurate.

Knowles [41] evaluated the performance measures frequently used in MOO according to their compatibility with the outperformance relations defined by Hansen and Jaszkiewicz. The performance measures' compatibility with the outperformance relations, as indicated by Knowles, are highlighted below where the performance measures are discussed in more detail.

3.2. Accuracy Performance Measures

Generational Distance

The generational distance (GD) measures the convergence of the approximated set towards the true POF (POF). The GD is defined as:

$$GD = \frac{\sqrt{\sum_{i=1}^{n_{POF^*}} d_i^2}}{n_{POF^*}} \quad (3)$$

where n_{POF^*} is the number of solutions in POF^* and d_i is the Euclidean distance in the objective space between solution i of POF^* and the nearest member of POF' . POF' contains sampled solutions of POF that are used as a reference set. Therefore, GD determines how close POF^* is to the sampled solutions of POF .

GD is easy to calculate and intuitive. However, knowledge about POF is required and a reference set, POF' , has to be available. It is important that the reference set contains a diverse set of solutions, since the selection of the solutions will impact on the results obtained from this performance measure. Furthermore, since the distance metric is used, scaling and normalisation of the objectives are required. GD is not

weakly compatible with O_W , but is compatible with O_S . Unfortunately, this performance measure will rate a POF^* with only one solution that is on POF' better than another POF^* that has one hundred solutions that are very close to POF' . Therefore, GD does not adhere to the property of monotony. Furthermore, GD does adhere to the concept of weak relativity, because any subset of POF' will always equal or improve the GD value obtained by POF^* s found by MOO algorithms.

It should be noted that GD is computationally expensive, especially for large or unlimited archives, or when DMOOPs with a large number of objectives are used.

Inverted Generational Distance

To overcome non-adherence to the concept of monotony by GD, Sierra and Coello Coello [56] introduced the inverse generational distance (IGD). The mathematical definition of IGD is the same as GD in Equation (3), except for the way in which the distance is calculated:

$$IGD = \frac{\sqrt{\sum_{i=1}^{n_{POF'}} d_i^2}}{n_{POF'}} \quad (4)$$

where $n_{POF'}$ is the number of solutions in POF' and d_i is the Euclidean distance in the objective space between solution i of POF' and the nearest member of POF^* .

IGD is compatible with relativity, since POF' always obtains an IGD value of zero and POF^* will only receive an IGD value of zero if $POF^* = POF'$. Furthermore, IGD is compatible with monotony, because it will rate a POF^* with more non-dominated solutions that are close to POF as a better set than another POF^* that only has one solution that falls within POF' . However, IGD is computationally expensive to calculate for a larger POF' or a larger POF^* , since for each point in POF' , the distance between that point and each of the points in POF^* has to be calculated. The usage of the distance function also requires scaling and normalisation of the objective function values, as is the case with GD.

Error Ratio

Van Veldhuizen [62] introduced the error ratio that measures the ratio of non-dominated solutions in POF^* that are not elements of POF' to the non-dominated solutions in POF^* that are elements of POF' . The error ratio is defined as

$$E = \frac{\sum_{i=1}^{n_{POF^*}} e_i}{n_{POF^*}} \quad (5)$$

where $e_i = 0$ if $x_i \in POF'$, $\forall x_i \in POF^*$ and $e_i = 1$ if $x_i \notin POF'$, $\forall x_i \in POF^*$. A small error ratio indicates a good performance.

If POF_A^* has two solutions with one solution in POF' , $E = 0.5$. However, if POF_B^* has one hundred solutions with one solution in POF' and the other solutions very close to POF' , $E = 0.99$. According to E , POF_A^* is a better set of solutions than POF_B^* . However, POF_B^* is more desirable. Therefore, E is only weakly compatible with O_C . E has weak relativity, because any subset of POF' will achieve the lowest E value, namely $E = 0$. It is not compatible with monotony, because if a non-dominated solution is added to POF^* that is not an element of POF' , E will increase.

The compatibility of the accuracy performance measures with the outperformance relations and the concepts of monotony and relativity is summarised in Table 3. In Tables 3 to 5 and Tables 6 to 10, M and R refer to the concepts of monotony and relativity respectively, C and W indicate that the performance measure is compatible or weakly compatible with the relation respectively, and “-” indicates that the performance measure is neither compatible nor weakly compatible with the relation.

Table 3: Compatibility of accuracy performance measures

Performance Measure	O _W	O _S	O _C	M	R
GD	-	C	C	-	W
IGD	W	C	C	W	C
E	-	-	W	-	W

3.3. Diversity Performance Measures

Diversity can be measured either by measuring how evenly the solutions are spread along POF^* or the extent of POF^* .

Number of Solutions

The easiest performance measure to calculate is the number of non-dominated solutions (NS) in POF^* . Van Veldhuizen [62] referred to this metric as the overall nondominated vector generation (ONVG). Even though this measure does not provide any information with regards to the quality of the solutions, it provides additional information when

comparing the performance of various algorithms. For example, one algorithm may have a better GD value, but only half of the NS that have been found by the other algorithm.

NS is not weakly compatible with any of the outperformance relations. According to Knowles [41] weak compatibility with O_W is necessary to ensure weak monotony. However, with NS this is not the case. Adding a non-dominated solution to POF^* increases, and thereby improves, NS. Therefore, NS is compatible with monotony. Furthermore, NS is weakly compatible with relativity only if the size of POF^* is smaller or equal to the size of POF' .

Spacing Metric of Schott

The *Spacing* metric, introduced by Schott [54], measures how evenly the points of POF^* are distributed in the objective space. Spacing is calculated as:

$$S = \sqrt{\frac{1}{n_{POF^*} - 1} \sum_{m=1}^{n_{POF^*}} (d_{avg} - d_m)^2}$$

with

$$d_m = \min_{j=1, \dots, n_{POF^*}} \left\{ \sum_{k=1}^{n_k} |f_{km}(\mathbf{x}) - f_{kj}(\mathbf{x})| \right\} \quad (6)$$

where d_m is the minimum value of the sum of the absolute difference in objective function values between the m -th solution in POF^* and any other solution in POF^* , d_{avg} is the average of all d_m values and n_k is the number of objective functions. If $S = 0$, the non-dominated solutions of POF^* is uniformly spread or spaced [13]. However, this does not mean that the solutions are necessarily good, since they can be uniformly spaced in POF^* , but not necessarily uniformly spaced in POF [41; 18].

The spacing metric of Schott is not even weakly compatible with O_W [41]. Adding a non-dominated solution to POF^* will not necessarily decrease the value of S and POF' does not necessarily have the lowest spacing metric value. Therefore, S does not adhere to the principles of either monotony or relativity.

It should be noted that this performance measure was designed to be used with other performance measures, has a low computational cost, and can provide useful information about the distribution of the solutions found [41]. Since the Euclidean

distance is used in the calculation of the measure, the objectives should be normalised before calculating the measure.

Spacing Metric of Deb

S provides information with regards to how evenly the non-dominated solutions are spaced on POF^* . However, it does not provide any information with regards to the extent or spread of the solutions. To address this shortcoming, Deb [15] introduced a measure of spread, defined as:

$$\Delta = \frac{\sum_{k=1}^{n_k} d_k^e + \sum_{i=1}^{n_{POF^*}} |d_i - d_{avg}|}{\sum_{k=1}^{n_k} d_k^e + n_{POF^*} d_{avg}} \quad (7)$$

with d_i being any distance measure between neighbouring solutions, d_{avg} is the mean of these distance measures, and d_k^e is the distance between the extreme solutions of POF^* and POF' .

Similar to S , Δ is not compatible with O_w and does not adhere to monotony or relativity.

Maximum Spread

Zitzler [71] introduced a measure of maximum spread that measures the length of the diagonal of the hyperbox that is created by the extreme function values of the non-dominated set. The maximum spread is defined as:

$$MS = \sqrt{\sum_{k=1}^{n_k} \left(\overline{POF_k^*} - \underline{POF_k^*} \right)^2} \quad (8)$$

where $\overline{POF_k^*}$ and $\underline{POF_k^*}$ is the maximum and minimum value of the k -th objective in POF^* respectively. A high MS value indicates a good extent (or spread) of solutions.

This measure can be normalised in the following way [13]:

$$MS_{norm} = \sqrt{\frac{1}{n_k} \sum_{k=1}^{n_k} \left(\frac{\overline{POF_k^*} - \underline{POF_k^*}}{\overline{POF_k} - \underline{POF_k}} \right)^2} \quad (9)$$

If POF_A^* outperforms POF_B^* (weakly, strongly or completely), but the non-dominated solutions of POF_B^* have a larger extent than the non-dominated solutions of POF_A^* , then POF_B^* will obtain a higher MS value. Therefore, MS is not weakly compatible with any of the outperformance relations. Adding a non-dominated solution to POF^* will not necessarily lead to a higher MS value. Therefore, MS adheres to weak monotony.

POF' will obtain the maximum MS value, but even a POF^* that only has two non-dominated solutions at the extreme points of POF' will also obtain the maximum MS value. Therefore, MS adheres to weak relativity.

C-Metric

The set coverage metric (C -metric) introduced by Zitzler [71] measures the proportion of solutions in set B that are weakly dominated by solutions in set A . The C -metric is defined as:

$$C(A, B) = \frac{| \{b \in B \mid \exists a \in A : a \leq b\} |}{|B|} \quad (10)$$

If $C(A, B) = 1$, all solutions in set B are weakly dominated by set A and if $C(A, B) = 0$ no solution in set B is weakly dominated by set A . Let POF_A^* and POF_B^* be the approximated POFs found by two algorithms with $POF_A^* \subset POF_B^*$, $ND(POF_B^*) = POF_B^*$. Then $C(POF_A^*, POF_B^*) < 1$ and $C(POF_B^*, POF_A^*) = 1$. Therefore, POF_B^* outperforms POF_A^* . Under the assumption that, if $C(A, B) = 1$ and $C(B, A) < 1$ evaluates set A as being better than set B , the C -metric is compatible with O_W [41].

It is important to note that the domination operator is not a symmetric operator, i.e. $C(A, B)$ is not necessarily equal to $1 - C(B, A)$. Therefore, if many algorithms are compared against each other, this metric would have to be calculated twice for each possible combination of algorithms. However, it should be noted that the C -metric is cycle-inducing, in other words if more than two sets are compared, the sets may not be ordered and in these cases no conclusions can be made [41].

If POF is known, set A can be selected as the set of sampled points of the true POF, POF' , and set B as the POF^* found by the algorithm. Then the C -metric can be calculated separately for each algorithm. Let POF_A^* and POF_B^* be the approximated POFs found by two algorithms as defined above and POF' a reference set with $ND(POF') = POF'$ and $POF_B^* \subseteq POF'$. Then, $C(POF', POF_A^*) = C(POF_B^*) = 1$ and $C(POF_A^*, POF') < C(POF_B^*, POF')$. In order to ensure compatibility with O_W , POF_A^* has to be evaluated by the C -metric as being worse than POF_B^* . Therefore, the following assumption should be made: if $C(R, E) = C(E, R) = 1$ and $C(E, R) < C(D, R)$, then E performs worse than D with regards to the C -metric, where D and E are two sets that are compared with one another using the reference set R [41]. Under this

aforementioned assumption, the C -metric is compatible with O_W when a reference set is used.

The C -metric does not adhere to the concept of monotony, since POF^* can add a non-dominated solution that is weakly dominated by the set that POF^* is compared against. However, the C -metric is weakly compatible with relativity, since $C(POF^*, POF')$ cannot obtain a higher C -metric value than $C(POF', POF^*)$.

U -measure

Leung and Wang [44] introduced the U -measure to measure the diversity of the found non-dominated solutions. Let $R = r_k$ be the set of reference points, where r_k is the extreme point of objective k of the union of all non-dominated solution of all POF^* 's found by the algorithms for the same POF that are compared with one another. Let χ be the set $\{d_i\}$ and $\bar{\chi}$ the set of $\{d_j\}$, where d_i is the distance between two neighbouring solutions and d_j is the distance between a reference point, r_k , and its nearest neighbour. Let \bar{d} be the average of the distances in χ and let $\bar{\chi}^*$ be the set $\{d'_j | d'_j = d_j + \bar{d}\}$. Then, the U -measure is defined as:

$$U = \frac{1}{n_{POF^*}} \sum_{j=1}^{n_{POF^*}} \left| \frac{d'_j}{d_{ideal}} - 1 \right|$$

with

$$d_{ideal} = \sum_{j=1}^{n_{POF^*}} \frac{d'_j}{n_{POF^*}} \quad (11)$$

A smaller U value indicates better uniformity of the non-dominated solutions of POF^* . Since distances are calculated in the U -measure, the objectives have to be normalised. Similar to S and Δ , the U -measure is not weakly compatible with any of the outperformance relations and does not adhere to monotony or relativity.

Table 4 summarises the compatibility of the diversity performance measures with the outperformance relations and the concepts of monotony and relativity. In Tables 4 to 5, C^* and W^* indicate that the performance measure is either compatible or weakly compatible with the relation, but only under certain conditions.

3.4. Combined Performance Measures

Hypervolume

The hypervolume or S -metric (first referred to as “the size of the space covered”) mea-

Table 4: Compatibility of diversity performance measures

Performance Measure	O_W	O_S	O_C	M	R
NS	–	–	–	C	W^*
S	–	–	–	–	–
Δ	–	–	–	–	–
MS	–	–	–	W	W
C	W^*	C	C	–	W
U	–	–	–	–	–

measures how much of the objective space is dominated by a non-dominated set [74; 75]. The definition of a dominated region and the traditional definition of the hypervolume are as follows:

Definition 7. Dominated Region: Let $\mathbf{f}_1 = \{f_{1_1}, f_{1_2}, \dots, f_{1_k}\}$ be a solution in the objective space and \mathbf{f}_{ref} a reference vector dominated by \mathbf{f}_1 . Then the region that is dominated by \mathbf{f}_1 and bounded by \mathbf{f}_{ref} is defined as the set,

$$R(\mathbf{f}_1, \mathbf{f}_{ref}) \triangleq \{\mathbf{f}_b | \mathbf{f}_b < \mathbf{f}_{ref} \text{ and } \mathbf{f}_1 < \mathbf{f}_b, \mathbf{f}_b \in \mathbb{R}^{m_k}\} \quad (12)$$

Let A be a non-dominated set of vectors, \mathbf{f}_i , for $i = 1, \dots, |A|$. Then the region dominated by A and bounded by the reference vector, \mathbf{f}_{ref} , is defined as the set:

$$R(A, \mathbf{f}_{ref}) \triangleq \bigcup_{i=1, \dots, |A|} R(\mathbf{f}_i, \mathbf{f}_{ref}) \quad (13)$$

Definition 8. Hypervolume: The hypervolume (HV) or S -metric of set A with respect to the reference vector \mathbf{f}_{ref} is the hyperarea or Lebesgue integral of the set $R(A, \mathbf{f}_{ref})$.

The reference vector can be any vector outside the feasible objective space, since this will result in a non-negative value for all possible non-dominated sets in the feasible objective space. Usually, the reference vector or reference point that is used in the HV calculation is the vector that consists of the worst value for each objective of the union of all non-dominated solutions of all POF^* that are compared against each other. It should be noted that the selected reference vector will affect the ordering of the non-dominated sets that are compared against each other, since all of the non-dominated sets use the same reference vector [41]. A high HV value indicates a good approximation set.

The HV is compatible with O_W if the upper boundary of the dominated region is set in such a way that all feasible non-dominated sets that are evaluated have a positive

HV value. The HV is therefore compatible with the outperformance relations. The HV is weakly compatible with monotony and weakly compatible with relativity. It is scaling independent and no prior knowledge of the true POF is required. According to Zitzler *et al.* [72] the HV is the only performance measure in the literature that has the following two qualities: if an approximation set A dominates another set B the HV provides a strictly better value for A ; and if a set obtains the maximum possible HV value for a MOOP, it contains all Pareto-optimal objective values.

One flaw of the HV is that it is biased towards convex areas of the POF [72]. Furthermore, it is computationally expensive to calculate, with a computational cost of $O(n^{k+1})$ with k representing the number of objectives [41]. However, recent research developed algorithms that reduce the computational cost of the HV. For example, Fonseca *et al.* [22] proposed an $O(|A|\log|A|)$ algorithm and Beume and Rudolph [5] proposed an algorithm with a complexity of $O(|A|^{k/2})$, where A is the non-dominated set and k is the number of objectives.

Hypervolume Ratio

To overcome the bias of the HV towards convex regions of the POF, Van Veldhuizen [62] proposed the hypervolume ratio (HVR), defined as:

$$HVR = \frac{HV(POF^*)}{HV(POF)} \quad (14)$$

The HVR normalises the HV and, assuming that the maximum HV is obtained by the true POF, the value of the HVR will be between 0 and 1. A high HVR indicates a good approximated POF. It should be noted that, for the HVR calculation, the reference vector is selected as the worst objective values for each objective from the union of the non-dominated solutions of all POF^* that are compared against each other, as well as POF' .

Similar to the HV, the HVR is compatible with O_w if the upper boundary of the dominated region is set in such a way that all feasible non-dominated sets that are evaluated have a positive HV value. Therefore, the HVR is compatible with the outperformance relations. Furthermore, the HVR is weakly compatible with monotony and relativity.

ϵ -metric

Zitzler *et al.* [76] presented the ϵ -metric to compare approximation sets. It measures the factor by which an approximation set is worse than another approximation set with respect to all objectives, i.e. it provides the factor ϵ where for any solution in set B there is at least one solution in set A that is not worse by a factor of ϵ in all objectives. The ϵ -measure uses the concept of ϵ -dominance. The definition of objective vector ϵ -domination is:

Definition 9. Weak Objective Vector ϵ -Domination: An objective vector \mathbf{f}_1 ϵ -dominates another objective vector \mathbf{f}_2 , denoted by $\mathbf{f}_1 <_{\epsilon} \mathbf{f}_2$, iff $f_1(\mathbf{x}_i)/(1 + \epsilon) \leq f_2(\mathbf{x}_i)$, $\forall i = 1, \dots, n_i$, and $\exists j = 1, \dots, n_i: f_1(\mathbf{x}_j)/(1 + \epsilon) < f_2(\mathbf{x}_j)$.

Using the above definition, the ϵ -metric is defined as:

$$I_{\epsilon}(A, B) = \inf_{\epsilon \in \mathbb{R}} \{ \forall \mathbf{f}_2 \in B \exists \mathbf{f}_1 \in A: f_1(\mathbf{x}_i) <_{\epsilon} f_2(\mathbf{x}_i) \} \quad (15)$$

The ϵ -metric is not weakly compatible with O_W , but is compatible with O_S and O_C . Adding a non-dominated solution may lead to a worse I_{ϵ} value. Therefore, the ϵ -metric is not weakly compatible with monotony. $I_{\epsilon}(POF^*, POF') = 1$ if $POF^* = POF'$. In addition, $I_{\epsilon}(POF^*, POF') = 1$ if POF^* contains some solutions from POF' . Therefore, I_{ϵ} is weakly compatible with relativity.

The compatibility of the combined performance measures with the outperformance relations are summarised in Table 5.

Table 5: Compatibility of combined performance measures

Performance Measure	O_W	O_S	O_C	M	R
HV	C*	C*	C*	W*	W*
HVR	C*	C*	C*	W*	W*
I_{ϵ}	-	C	C	-	W

4. Current DMOO Performance Measures

This section discusses performance measures that are currently being used to evaluate the performance of DMOO algorithms. Section 4.1 discusses performance measures that quantify the accuracy of the found POF. Performance measures that are

used to measure the diversity of the non-dominated solutions are discussed in Section 4.2. Combined performance measures that measure accuracy and diversity of the non-dominated solutions are discussed in Section 4.3. Section 4.4 discusses the measurement of an algorithm's robustness when an environment change occurs.

4.1. Accuracy Performance Measures

GD Measure

Mehnen *et al.* [49] used the GD metric to evaluate the performance of algorithms solving DMOOPs. They calculated the GD metric in decision space, since the DMOOPs that were used in the study had POSs that dynamically shifted over time, and named the performance measure G_τ . If GD is calculated in decision space, GD measures the distance of the approximated POS, POS^* , to the true POS, POS . Zhou *et al.* [70] used the GD metric (and the variance of GD) in objective space for DMOO, but referred to the performance measure as the distance indicator D . A number of other researchers have used GD to evaluate DMOO algorithms, as shown in Table 7. Goh and Tan adapted GD for DMOO as follows:

$$VD = \frac{1}{\tau} \sum_{t=1}^{\tau} VD(\tau)$$

with

$$VD(\tau) = \frac{\sqrt{n_{POF^*} \sum_{i=1}^{n_{POF^*}} d_i^2 (\tau \% \tau_i)}}{n_{POF^*}} \quad (16)$$

where τ is the current iteration number, τ_i is the frequency of change and $\%$ is the modulo operator. The performance measure, referred to as variational distance (VD), is calculated in the decision space every iteration just before a change in the environment occurs.

Similar to GD, VD is not weakly compatible with O_W , but is compatible with O_S and O_C . If a new non-dominated solution is added to POF^* that is further away from POF than the other solutions in POF^* , VD will increase. Therefore, VD is not weakly compatible with monotony. However, the GD value of POF will always be less or equal to the GD value of any POF^* . Therefore, GD is weakly compatible with relativity. Since distance is used in the calculation of VD, the objectives have to be normalised.

When solving DMOOPs, similar to VD, the performance measure is calculated every iteration just before an environmental change occurs. Therefore, prior knowledge of when changes occur is required. However, if a performance measure is calculated while the algorithm is running (also referred to as online calculation), prior knowledge about changes in the environment is not required. In this case the performance measure can be calculated on the non-dominated solutions that were obtained at the iteration just before the change occurred. Furthermore, if the performance measure is calculated after the algorithm has completed its runs (also referred to as offline calculation), the algorithm can keep record of the iterations when changes occurred.

Success Ratio

Similar to the error ratio (refer to Section 3.2), Mehnen *et al.* [49] used the success ratio to quantify the ratio of the solutions found that are members of the true POF. The success ratio is defined as

$$SC_{\tau} = \frac{|\{\mathbf{x} | f(\mathbf{x}) \in POF'\}|}{|POF^*|} \quad (17)$$

where a high success ratio, SC_{τ} , indicates good performance.

If an algorithm finds many non-dominated solutions that are not Pareto-optimal but very close to POF' , the POF^* will obtain a lower SC_{τ} value than an algorithm that finds only one Pareto-optimal solution. Therefore, SC_{τ} is only weakly compatible with O_C and is not weakly compatible with either O_W or O_S .

If a non-dominated solution is added to POF^* that is not Pareto-optimal, the value of SC_{τ} decreases and therefore SC_{τ} is not compatible with monotony. Since POF' will obtain the same SC_{τ} value than subsets of POF' , SC_{τ} is weakly compatible with relativity.

The compatibility of the accuracy performance measures with the outperformance relations and the concepts of monotony and relativity is summarised in Table 6.

Table 6: Compatibility of accuracy performance measures

Performance Measure	O_W	O_S	O_C	M	R
GD	–	C	C	–	W
SC_{τ}	–	–	W	–	W

In Tables 7 to 9, x indicates that the performance measure was used, x* indicates

that the performance measure was calculated in decision space and x^* indicates that the variance of the performance measure was used. The usage of the accuracy performance measures in the DMOO literature is summarised in Table 7 (refer to Tables 3 and 6). Table 7 shows that most researchers have used the *GD* or *VD* performance measure to quantify the accuracy of *POF**.

Table 7: Usage of DMOO accuracy performance measures

PM	Usage	Authors
GD	x	[20; 28; 30; 70; 36; 10; 37; 43; 51]
	x^*	[20; 30; 49; 36; 37; 51]
	x^*	[28; 30; 70; 43]
	x^{**}	[30]
IGD	x	[46; 64; 65]
	x^*	[64]
VD	x	[31]
	x^*	[60; 25; 24; 42]
Acc	x	[9; 6; 7; 8; 57; 31]
SC_{τ}	x	[49]

4.2. Diversity Performance Measures

MS' measure

Goh and Tan [25] introduced an adapted version of *MS* (refer to Equation (9) in Section 3.3) to measure how well *POF** covers *POF'*. Contrary to *MS*, the adapted *MS*, *MS'*, takes into account the proximity of *POF** to *POF'*. *MS'* is defined as:

$$MS' = \sqrt{\frac{1}{n_k} \sum_{k=1}^{n_k} \left[\frac{\min[\overline{POF}_k^*, \overline{POF}_k'] - \max[\underline{POF}_k^*, \underline{POF}_k']}{\overline{POF}_k' - \underline{POF}_k'} \right]^2} \quad (18)$$

Similar to *MS*, *MS'* is not weakly compatible with any of the outperformance relations. Adding a non-dominated solution to *POF** will not necessarily lead to a higher *MS'* value. Therefore, *MS'* adheres to weak monotony. *POF'* will obtain the maximum *MS* value, but even a *POF** that only has two non-dominated solutions at the extreme points of *POF'* will also obtain the maximum *MS* value. Therefore, *MS'* adheres to weak relativity.

PL measure

Since many diversity performance measures are based on the Euclidean distance and

therefore do not take the shape of the POF into account, Mehnen *et al.* [49] introduced a performance measure, the *PL* measure, that is based on path lengths or path integrals.

The length of the path between two solutions is defined as:

Definition 10. Length of Path between Two Solutions: Let γ be the path between two solutions in objective space, \mathbf{a} and \mathbf{b} , that is differentiable in $[\mathbf{a}, \mathbf{b}]$. Then the length of a path between $[\mathbf{a}, \mathbf{b}]$ on γ is defined as:

$$L(\gamma, \mathbf{a}, \mathbf{b}) := \int_b^a |\dot{\gamma}| dt = \int_b^a \sqrt{\dot{\gamma}_1^2 + \dots + \dot{\gamma}_m^2} dt \quad (19)$$

where $\dot{\gamma}$ is the derivative of γ and $|\dot{\gamma}|$ is the Euclidean norm of $\dot{\gamma}$.

The *PL* performance measure is the normalised product of the path between sorted neighbouring solutions on *POF*, defined as

$$PL := \frac{\ln\left(\prod_{\mathbf{f}(\mathbf{x}_i) \in POF} \zeta(\mathbf{x})\right)}{\ln e^{L_{POF}}} = \frac{\sum_{\mathbf{f}(\mathbf{x}_i) \in POF} \ln(\zeta(\mathbf{x}))}{L_{POF}} \quad (20)$$

where $\zeta(\mathbf{x}_i) = L(\gamma, \mathbf{f}(\mathbf{x}_i), \mathbf{f}(\mathbf{x}_{i+1})) + 1$ and \mathbf{f} represents the objective functions. For the calculation of *PL*, a solution is considered as being in *POF* if the solution is within an ϵ -region near *POF*.

In order to calculate the *PL* performance measure, an analytic closed description of the true POF is required. However, according to Mehnen *et al.* the calculation of the *PL* measure is complicated when a DMOOP has more than two objectives, or has a discontinuous POF. In these situations Mehnen *et al.* [49] recommend the usage of *S* [54] (refer to Section 3.3).

PL is not weakly compatible with the outperformance relations. However, it is weakly compatible with monotony, since the value of *PL* increases when a new non-dominated solution that is within ϵ -distance of *POF* is added to *POF**.

Set Coverage Metric

Guan *et al.* [28] introduced a set coverage measure that is based on the *S* and *D* metrics introduced by Zitzler [71]. The HV of the objective space that is dominated by *POF** but not by *POF'*, referred to as the *D*-metric, is defined as

$$D(POF^*, POF') = HV(POF^* + POF') - HV(POF') \quad (21)$$

The set coverage metric is then defined as

$$\eta = \frac{D(POF^*, POF')}{HV(POF')} + \frac{D(POF', POF^*)}{HV(POF')} \quad (22)$$

Therefore, the set coverage metric, η , is the normalised sum of the:

- HV of the objective space that is dominated by POF^* and not by POF' , and
- HV of the objective space that is dominated by POF' and not by POF^* .

η is weakly compatible with O_W if the HV is weakly compatible with O_W . Therefore, η is weakly compatible with O_W if the reference vector is selected in such a manner that that all feasible non-dominated sets that are evaluated have a positive HV value. If the reference vector is selected in this manner, η is compatible with all the outperformance relations. Furthermore, η is then weakly compatible with monotony and weakly compatible with relativity.

Pareto Front Extent

Zhang and Qian [68] introduced the coverage scope (CS) measure to quantify the average width or coverage of the non-dominated set. CS is calculated by averaging the maximum distance between each solution in POF^* and the other solutions in POF^* . Therefore, CS is defined as

$$CS = \frac{1}{n_{POF^*}} \sum_{i=1}^{n_{POF^*}} \max\{\|\mathbf{f}(\mathbf{x}_i) - \mathbf{f}(\mathbf{x}_j)\|\} \quad (23)$$

with $\mathbf{x}_i, \mathbf{x}_j \in POF^*$, $i \geq 1$ and $j \leq n_{POF^*}$.

A higher CS value indicates a better performance. CS is similar to S [54] (refer to Section 3.3), but uses the maximum distance where S uses the minimum distance between the non-dominated solutions in POF^* . Similar to S , CS is not weakly compatible with the outperformance relations. Furthermore, CS is not weakly compatible with monotony, since adding a non-dominated solution to POF^* can decrease the CS value. The CS value of POF' can be smaller than the CS value of POF^* . Therefore, CS is not weakly compatible with relativity. A summary of the compatibility of diversity performance measures is shown in Table 8.

Table 8: Compatibility of diversity performance measures

Performance Measure	O_W	O_S	O_C	M	R
MS'	–	–	–	W	W
PL	–	–	–	W	–
η	C^*	C^*	C^*	W^*	W^*
CS	–	–	–	–	–

Table 9 summarises the usage of the diversity performance measures in the DMOO

literature (refer to Tables 4 and 8). In Table 9, Entropy refers to the normalised entropy between the POF^* s, Accum refers to the accumulated genetic diversity that measures the moment of inertia and N_E refers to the variation of the minimum normalised Euclidean distance of the POF^* solutions. Table 9 indicates that most researchers used S and MS to quantify the diversity of the non-dominated solutions in POF^* .

Table 9: Usage of DMOO diversity performance measures

PM	Authors
NS	[28; 27]
C	[63; 1; 68]
η	[28]
CS	[68]
S	[49; 27; 26; 31; 68]
U	[63; 47]
PL	[49]
MS, MS'	[60; 25; 24; 42; 31]
Δ	[67]
Entropy	[10; 3]
Accum	[3]
N_E	[28]

4.3. Combined Performance Measures

Accuracy Measure

A measure of accuracy introduced by Weicker [66] for DSOO was adapted by Cámara *et al.* [57] for DMOO. This measure quantifies the quality of the solutions as a relation between the HV of POF^* and the maximum HV (HV_{max}) that has been found so far. The accuracy measure is defined as

$$acc(t) = \frac{HV(POF^*(t))}{HV_{max}(POF^*(t))} \quad (24)$$

The accuracy measure, acc , is compatible with O_W if the upper boundary of the dominated region is set in such a way that all feasible non-dominated sets that are evaluated have a positive HV value (refer to Section 3.4). Under these conditions, acc is compatible with the outperformance relations and weakly compatible with monotony and relativity.

Hypervolume Difference

Zhou *et al.* [70] suggested to use the hypervolume difference (HVD) to measure the

quality of the found POF. HVD is defined as

$$HVD = HV(POF) - HV(POF^*) \quad (25)$$

However, when the true POF is unknown, the HVD cannot be used. Zheng [69] used the maximum HV to measure the quality of the found POF.

Cámara *et al.* [8] extended the definition of their accuracy measure (Equation (24)) to use the HVD when the true POF is known. The alternative accuracy measure is defined as

$$acc_{alt}(t) = |HV(POF(t)) - HV(POF^*(t))| \quad (26)$$

where $acc_{alt}(t)$ is the absolute HVD at time t . The absolute values ensures that $acc_{alt}(t) \geq 0$, even if $HV(POF^*) > HV(POF')$.

HVD is compatible with the outperformance relations if HV is compatible with the outperformance relations.

Optimal Subpattern Assignment Measure

Recently Tantar *et al.* [61] introduced performance measures that are based on performance measures used in quantifying the tracking quality in multi-object tracking problems. The performance measures are developed based on the optimal subpattern assignment (OSPA) measure that can be used to compare sets with different cardinality [55].

Let $P = (F, X, N)$ define a DMOOP with F and X representing a set of objective functions and a set of decision variables respectively. N represents the neighbourhood function described by a sphere of center c and radius r , defined as

$$N(c, r) = \{\mathbf{x} \in X \mid d(\mathbf{x}, c) < r \text{ and } \exists h \mid \mathbf{x}hc\} \quad (27)$$

where d is the distance between a solution, \mathbf{x} , and the center point of the neighbourhood, c , and $\exists h \mid \mathbf{x}hc$ indicates that the neighbourhood can be reached through a transformation h .

Let A and B represent two approximated POFs, POF_A^* and POF_B^* . Then the following two performance measures are defined:

$$M_{loc}(X, Y) = \left(\frac{1}{n_{POF_B^*}} \min_{j \in P} \sum_{i=1}^{n_{POF_A^*}} d(x_i, y_{j(i)})^p \right)^{\frac{1}{p}} \quad (28)$$

where $d(\mathbf{x}, \mathbf{y}) = \min\{c, d(\mathbf{x}, \mathbf{y})\}$ is the minimum distance between two solutions that are cut off by c . When comparing A and B , the solutions from B that are in the neighbourhood of a given solution from A are determined by considering all permutations of solutions from B , referred to as the set P . M_{loc} quantifies the quality of the coverage of A as compared to B . A drawback of this performance measure is the computational cost, because of the calculation of permutations for each solution under consideration.

The other performance measure is defined as

$$M_{card}(X, Y) = \left(\frac{c^p (n_{POF_B^*} - n_{POF_A^*})}{n_{POF_B^*}} \right)^{\frac{1}{p}} \quad (29)$$

M_{card} is a cardinality penalty function that is used when $|A| \neq |B|$, and is zero if the two sets have the same cardinality. M_{card} measures the influence of the cardinality difference on the overall quality of the larger set, with the cut-off term as the error quantification factor.

The OSPA metric is then defined as:

$$OSPA(X, Y) = M_{loc}(X, Y) + M_{card}(X, Y) \quad (30)$$

$OSPA$ is not weakly compatible with the outperformance relations. However, $OSPA$ is weakly compatible with monotony.

Table 10 summarises the compatibility with the outperformance relations by the combined performance measures.

Table 10: Compatibility of combined performance measures

Performance Measure	O _w	O _s	O _c	M	R
<i>HVD</i>	C*	C*	C*	W*	W*
<i>acc_{alt}</i>	C*	C*	C*	W*	W*
<i>OSPA</i>	–	–	–	W	–

The usage of the combined performance measures in the DMOO literature is summarised in Table 11 (refer to Tables 5 and 10).

4.4. Robustness Performance Measures

The robustness of an algorithm refers to how well the algorithm recovers after an environment change occurs.

Stability Measure

The effect of the changes in the environment on the accuracy (*acc* defined in Equa-

Table 11: Usage of DMOO combined performance measures

PM	Usage	Authors
HV	x	[9; 6; 27; 59; 1; 7; 3; 26; 65]
HVR	x	[17; 46; 69; 39; 8; 57; 14; 31]
HVD	x	[70; 8; 57]
HV_{max}	x	[69; 7]
	x^p	[69]
I_ϵ	x	[40]
OSPA	x	[61]

tion 24) of the algorithm can be quantified by the measure of stability that was introduced by Weicker [66] for DSOO and adapted for DMOO by Cámara *et al.* [57]. Stability is defined as

$$stab(t) = \max\{0, acc(t-1) - acc(t)\} \quad (31)$$

where a low *stab* value indicates good performance.

The compatibility of *stab* depends on the definition of *acc*. If *acc* as defined in Equation (24) is used, *stab* is compatible with O_W if *acc* is compatible with O_W . Under these conditions, *stab* is compatible with the outperformance relations and weakly compatible with monotony and relativity.

Reactivity Measure

Cámara *et al.* [57] presented a measure of reactivity based on the reactivity performance measure introduced by Weicker [66] for DSOO. Reactivity measures how long it takes for an algorithm to recover after a change in the environment, by determining how long it takes for an algorithm to reach a specified accuracy threshold. The reactivity performance measure is defined as

$$react(t, \epsilon) = \min\{t' - t \mid t < t' < \tau_{max}, t' \in \mathbb{N}, \frac{acc(t')}{acc(t)} \geq (1 - \epsilon)\} \quad (32)$$

where τ_{max} is the maximum number of iterations or generations.

Similar to *stab*, *react* is weakly compatible with O_W if *acc* is weakly compatible with O_W . *react*'s compatibility with monotony and relativity also depends on *acc*'s compatibility with monotony and relativity.

The compatibility with the outperformance relations by the robustness performance measures is summarised in Table 12.

Table 12: Compatibility of robustness performance measures

Performance Measure	O _W	O _S	O _C	M	R
<i>stab</i>	C*	C*	C*	W*	W*
<i>react</i>	C*	C*	C*	W*	W*

Table 13 summarises the usage of performance measures that quantifies robustness in the DMOO literature.

Table 13: Usage of DMOO robustness performance measures

PM	Authors
stab	[9; 6; 7; 8; 57; 31]
react	[9; 6; 7; 8; 57]

When algorithms solve DMOOPs, five major issues should be taken into consideration when selecting performance measures to quantify the performance of the algorithms, namely: algorithms losing track of the changing POF, the effect of outlier solutions in the found POF, boundary constraints violations, calculating the performance measures in either the objective or decision space, and comparing the performance of the various algorithms. These issues are discussed in the next section.

5. Issues with Current DMOO Performance Measures

Section 4 discussed a number of performance measures that have been used to quantify the performance of algorithms on DMOOPs. Even though these measures have been used in a number of articles, they suffer from a number of problems mostly related to aspects of dynamic environments [33]. These problems make general applicability of these performance measures to all DMOOPs not possible.

Section 5.1 discusses misleading results that can occur when algorithms lose track of the changing POF. The effect of outlier solutions in POF^* on the quantification of an algorithm's performance is discussed in Section 5.2. Section 5.3 discusses the effect of boundary constraints violations on the performance of DMOO algorithms. Furthermore, performance measures can be calculated in either the objective or decision space as discussed in Section 5.4. Finally, Section 5.5 discusses issues when comparing the performance of the various algorithms.

5.1. Losing Track of the Changing POF

When a DMOO algorithm loses track of the changing POF, and POF changes over time in such a way that its HV value decreases, many of the current performance measures will give misleading results [33]. Figure 1 illustrates example POFs where the POF changes over time in such a way that, if the HV is calculated with the reference vector being the worst values of each objective, the HV will decrease over time. A decrease in the HV will occur if for each example the POF changes from convex to concave. Figure 1 illustrates three such POFs. In Figure 1, the first POF is represented by the bottom line and the last POF by the top line.

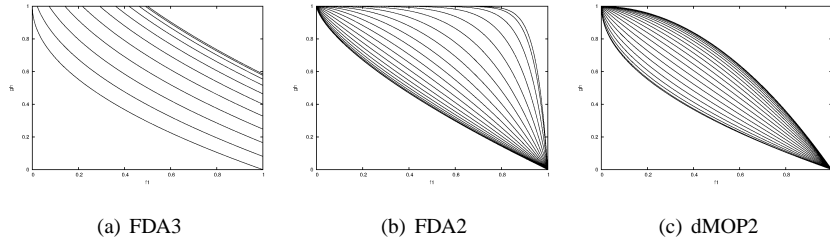


Figure 1: Examples of DMOOPs where the HV value of POF decreases over time for certain time steps

The problem of losing track of the POF was first observed by Helbig and Engelbrecht [34], where five algorithms were used to solve the FDA2 DMOOP. These algorithms included a dynamic VEPSO (DVEPSO) which uses clamping to manage boundary constraint violations (DVEPSO-A) [34]; DVEPSO that uses per element re-initialisation to manage boundary constraint violations (DVEPSO-B) [34]; DNSGA-II-A, a variation of NSGA-II where a percentage of individuals are randomly selected and replaced with newly created individuals if an environment change occurs [17]; DNSGA-II-B, a variation of NSGA-II where, after an environment change, a percentage of individuals are randomly selected and replaced with individuals that are mutated from existing individuals [17]; and dCOEA [25]. Figure 2 illustrates example POFs obtained by these algorithms in comparison with POF (Figure 2(f)). It is clear from these figures that DNSGA-II-A, DNSGA-II-B and dCOEA lost track of the changing POF, with the DVEPSO algorithms being more successful in tracking the POF. It is therefore

expected that the values of the performance measures should be better for the DVEPSO algorithms than for the evolutionary algorithms.

The performance measure values of these algorithms for a change frequency of 10 is presented in Table 14, where *NS* refers to the number of non-dominated solutions found, *S* refers to the spacing measure defined by Schott [54], *HVR* refers to the HV ratio [46], *Acc* and *Stab* refers to measures of accuracy and stability presented by Cámara *et al.* [9], and *VD* and *MS* refer to the adapted generational distance and maximum spread performance measures for dynamic environments proposed by Goh and Tan [25].

As shown in Table 14, performance measures *VD* and *MS* indicate the DVEPSO algorithms to be better than the evolutionary algorithms. However, the measures that make use of the *HV*, namely *HVR*, *Acc* and *Stab*, rank the evolutionary algorithms as being better than the DVEPSO algorithms. This occurs since the *HV* value of *POF* decreases over time and therefore from the time where an algorithm loses track of the changing *POF*, its *HV* value is higher than that of *POF* and therefore higher than that of algorithms that are tracking the changing *POF*. Since the *HV* value of *POF* decreases over time, *HVR* (which divides the *HV* of *POF** by the *HV* of *POF*) still does not address this problem.

The following papers used the *HV* or *HVR* without using any accuracy measure that requires knowledge of the true *POF*: [1; 3; 6; 7; 8; 9; 14; 17; 26; 27; 39; 57; 59; 69]. Therefore, if any of the algorithms that were evaluated in these studies lost track of the changing *POF*, the performance measure values that were obtained may be misleading.

Table 14: Performance Measure Values for FDA2

τ_t	Algorithm	NS	S	HVR	Acc	Stab	VD	MS	R
10	DVEPSO-A	73.4	0.00118	0.99533	0.97848	0.00049	0.45824	0.90878	4
10	DVEPSO-B	63	0.00391	0.99905	0.98157	0.00029	0.43234	0.88916	3
10	DNSGAI-A	39.4	0.00044	1.0044	0.98681	9.565×10^{-06}	0.71581	0.77096	2
10	DNSGAI-B	39.6	0.00042	1.00441	0.98683	9.206×10^{-06}	0.71681	0.77866	1
10	dCOEA	38.4	0.00051	1.00209	0.98454	0.00122	0.70453	0.61923	5

The issue of an algorithm losing track of the changing *POF* is unique to DMOO. In order to overcome this problem, acc_{alt} proposed by Cámara *et al.* (refer to Equation (26) in Section 4.3) should be used when the *POF* is known. Furthermore, if acc_{alt}

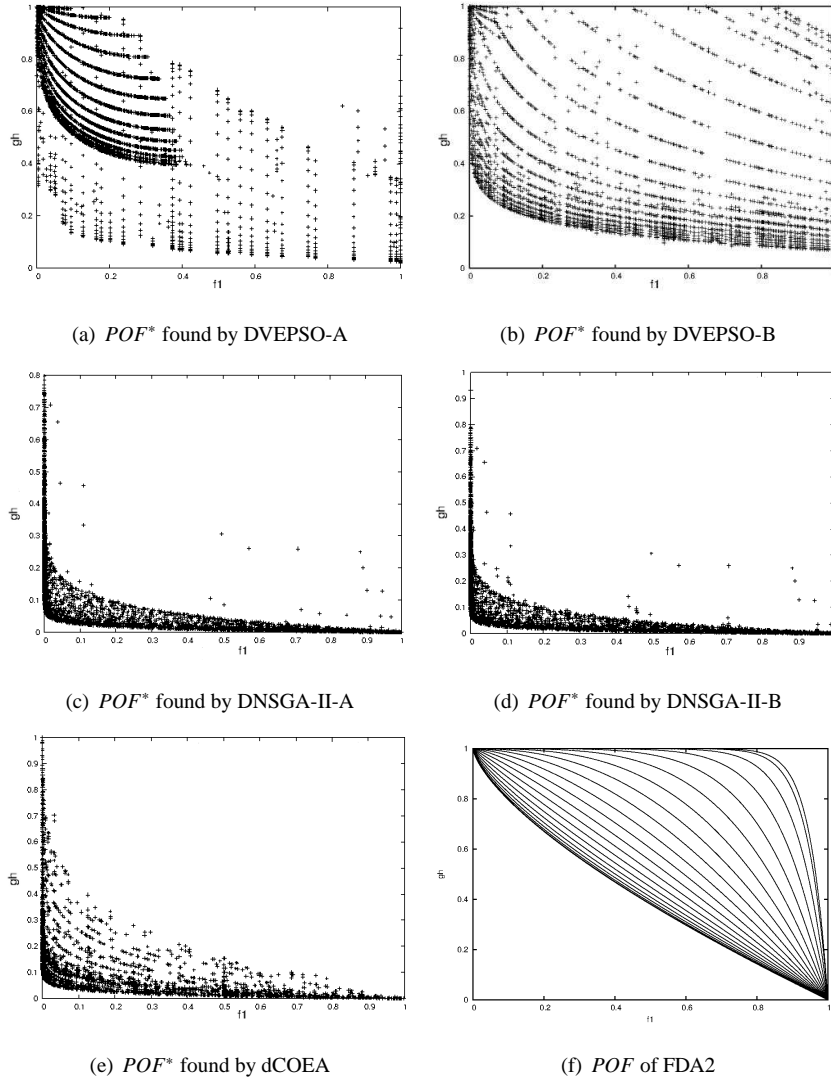


Figure 2: POF and POF^* found by various algorithms for FDA2 with $n_t = 10$, $\tau_t = 10$ and 1000 iterations

is used for acc , $Stab$ will also be reliable even if an algorithm loses track of POF .

If POF is unknown, as is the case with most real-world problems, the deviation of the performance measures that use the HV measure should also be calculated. If the performance measure's deviation varies much more for certain algorithms, it may indicate that one or more of the algorithms have lost track of the changing POF and that

the performance measure can not reliably be used to compare the performance of the different algorithms. Therefore, the graphs of POF^* s should be plotted and checked to determine whether any algorithm has lost track of the changing POF.

Even though various performance measures were used, the misleading performance measures can play a large enough role to influence the overall ranking of the algorithms. This is shown in Table 14. Even though the DVEPSO algorithms ranked the highest with regards to NS , VD and MS , the measures that make use of the HV value affected the ranking in such a way that the DVEPSO algorithms ranked as number 3 and 4 respectively and were outranked by the DNSGA-II algorithms - portraying incorrect ordering.

It should be noted that the stability measure, $Stab$, proposed by Cámara *et al.* [9] measures the change in the values of acc at two consecutive time steps (refer to Section 4.4). Under normal circumstances a low $Stab$ value indicates that the performance of the algorithm is not severely affected by the change in the environment. However, in situations where one or more algorithm(s) lost track of the changing POF, the lowest $Stab$ value will be obtained by the algorithms that lost track of the changing POF. Therefore, the results obtained with the $Stab$ performance measure will be misleading. Table 14 shows that the NSGA-II algorithms obtained a better $Stab$ value than the DVEPSO algorithms. Clearly, as indicated by the POFs shown in Figure 2, this is not the case.

Even though $Stab$ has been proposed to provide additional information and not to be used on its own, it should be noted that the choice of the performance measure used to measure acc influences the reliability of $Stab$.

5.2. Outliers in the POF

When algorithms solve DMOOPs and the environment changes frequently, the POF^* that has been found by the algorithm for a specific time step may contain outliers [33]. This occurs, because in the number of iterations or generations available to the algorithm to solve the specific POF, the algorithm found non-dominated solutions that are further away from the true POF. In the time available before the environment changes, the algorithm has not found any solutions closer to the true POF that dominates these

outlier solutions. Figure 3 illustrates an example POF^* that contains outliers.

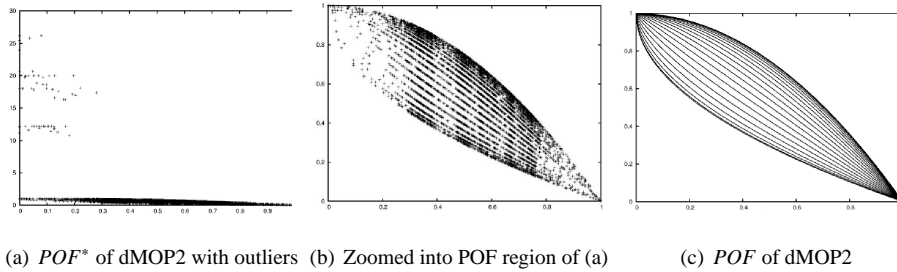


Figure 3: Example of a POF^* that contains outlier solutions.

Outliers will skew results obtained using:

- distance-based performance measures, such as GD , VD , PL , CS and M_{loc} ,
- performance measures that measure the spread of the solutions, such as MS , and
- the HV performance measure.

The influence of outlier solutions on the calculation of GD and VD is illustrated in Table 15. Due to the large distance between the outliers and POF as shown in Figures 3 and 4, the resulting GD and VD is much larger with the outliers present compared to when the outliers are not present. However, it should be noted that the severity of the influence of outliers on distance calculations depends on the number of outliers and their distance from POF .

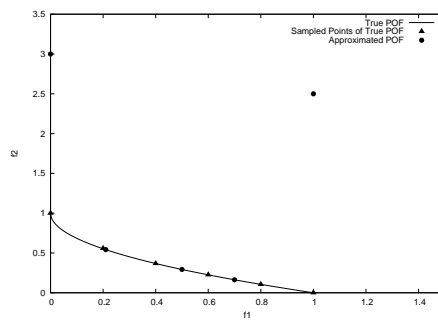


Figure 4: POF^* of FDA1 with outlier solutions

Table 15: GD, VD and MS values for FDA1

Outliers	GD	VD	MS
Yes	2.05565	4.596574	0.91833
No	0.00942	0.016311	0.4342

Furthermore, when a performance measure, such as MS of Cámara *et al.* [9], measure the extent or spread of the approximated POF, these outlier solutions may cause the performance measure to be misleading with regards to the performance of the algorithm. In Figure 4, the outlier solutions' f_1 and f_2 values will become the \overline{PF}_i^* and \underline{PF}_i^* in the f_2 and f_1 objective in Equation (9) respectively. In Figure 4, POF^* only contains non-dominated solutions with f_1 values in the range of [0.2, 0.7] and f_2 values in the range of [0, 0.5] without the outlier solutions. However, with the outlier solutions the f_1 values will be calculated as being in the range of [0, 1.0] and f_2 values in the range of [0.2, 3]. This will result in the maximum MS value and will not give a true reflection of the diversity of solutions that has been found by the algorithm. The influence of the outliers on the value of MS is shown in Table 15.

When solving DMOOPs, many researchers use the HV performance measure, especially when POF is unknown. When comparing various algorithms' POF^* s, the same reference vector is used. HV values that are calculated with different reference vectors cannot be compared against each other. Furthermore, outlier solutions influence the reference vector values that are used to calculate the HV. Typically, the reference vector is chosen as the worst values obtained for each objective. Therefore, for POF^* in Figure 4 the reference vector for the HV is [1.1, 3.1] and [1.1, 1.1] with and without the outlier values respectively. This results in much larger HV values when outliers are present, as shown in Table 16. From Table 16 it is clear that HVR and acc_{alt} provide a more accurate representation of the performance of the algorithm, resulting in a better HVR value without outliers than with the outliers. However, when the HV is used, the POF^* with outliers obtain a better HV value than the POF^* without the outliers. Therefore, if POF is unknown and the HV is used, outlier solutions may lead to misleading results and algorithms being ranked incorrectly.

One approach to manage outliers in POF^* is to remove the outliers from POF^* .

Table 16: HV, HVR and HVD values for FDA1

Outliers	HV	HVR	acc _{alt}
Yes	2.49898	0.84461	0.45974
No	0.69798	0.91994	0.06074

However, no consensus exists on the approach that should be followed to decide which non-dominated solutions in POF^* should be classified as outliers.

It should be noted that, as the number of objectives increases, more outlier solutions may become present in POF^* . This is the case, since as the number of objectives increases, more solutions that are found by the algorithm will be non-dominated with regards to the other solutions in POF^* . Furthermore, outliers in POF^* will cause the same problems when solving static MOOPs. However, since algorithms generally have longer time to converge towards POF with static MOOPs than with DMOOPs where the environment changes, the possibility of the occurrence of outliers increases when solving DMOOPs.

5.3. Boundary Constraint Violations

The candidate solutions of certain computational intelligence algorithms tend to move outside the boundary constraints of an optimisation problem while searching for solutions. For example, it has been shown theoretically that most particles in a particle swarm optimisation (PSO) algorithm [38] leave the bounds within the first few iterations [18; 23]. If a particle finds a better solution outside the bounds, its personal best position is set to this new position. Should this position be better than the current neighbourhood or global best, other particles are also pulled outside of the bounds. Consequently, particles may converge on a solution outside the bounds of the search space. This behaviour of particles is empirically analyzed by Engelbrecht [19].

If a genetic algorithm (GA) [35] uses blending cross-over, such as parent-centric cross-over [16], offspring may be generated outside the boundaries of the search space due to the stochastic component of the blending process.

Most evolutionary programming [21] algorithms sample mutational step sizes from zero-mean distributions with tails that asymptotically approach infinity and negative infinity. Consequently, large mutational step sizes can potentially be added to parent in-

dividuals, moving offspring outside of the bounds. If such offspring have better fitness than parent individuals, these offspring survive to the next generation with a chance of obtaining a solution that does not lie within the bounds of the search space.

With differential evolution's [58] mutation operator, a weighted difference of two vectors are added to the parameter vector. If this weighted difference is large, it may cause the trial vector to move outside the boundary constraints of the optimisation problem.

Most unconstrained DMOOPs have boundary constraints that limit the search space. However, if an algorithm does not manage boundary constraint violations, infeasible solutions may be added to POF^* . These infeasible solutions may dominate feasible solutions in POF^* , which will cause the feasible solutions to be removed from POF^* . Furthermore, the infeasible solutions may cause misleading results with regards to an algorithm's performance.

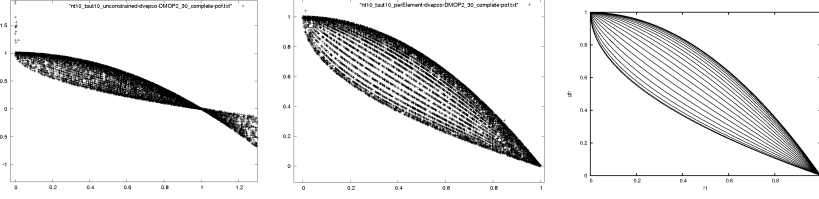
Figure 5(a) illustrates a POF^* that was found by dynamic VEPSO (DVEPSO) that did not manage boundary constraint violations (DVEPSO_u) when solving dMOP2, DVEPSO that manages boundary constraint violations (DVEPSO_c), and the true POF (POF). From Figure 5 it is clear that POF^* of DVEPSO_u has a larger HV value than both POF^* of DVEPSO_c (refer to Figure 5(b)) and POF (refer to Figure 5(c)). This is confirmed in Table 17. This incorrectly indicates that the POF^* that contains solutions that are outside the bounds of the search space to be better. Therefore, when comparing various algorithms with one another, it is important to check that all algorithms manage boundary constraint violations to ensure a fair comparison. It should be noted that the issue of boundary constraint violations is applicable to both SMOO and DMOO.

Table 17: HVR values for dMOP2

Algorithm	HVR
DVEPSO _u	1.00181
DVEPSO _c	0.99978

5.4. Objective Space versus Decision Space

Accuracy measures, such as VD or GD , can be calculated with respect to either the decision or the objective space. Using objective space, VD measures the distance be-



(a) POF^* of dMOP2 with feasible and infeasible solutions (b) POF^* of dMOP2 with only feasible solutions (c) POF of dMOP2

Figure 5: Example of a POF^* that contains infeasible solutions due to boundary constraint violations

tween the non-dominated solutions of POF^* and POF' . Therefore, VD measures the closeness of POF^* to POF . Since one of the goals of solving a DMOOP is to track the changing POF, the accuracy should be measured in the objective space. If the VD measure is calculated in the decision space, the distance between POS^* and POS is calculated. Calculating the VD measure in the decision space may be useful to determine how close POS^* is from POS . However, if for a DMOOP a small change in the POS causes a big change in the POF, it may occur that even though the algorithm's POS^* is very close to POS , POF^* is quite far from POF . This is illustrated with an example DMOOP defined as:

$$DMOOP_1 = \begin{cases} \text{Minimize : } f(\mathbf{x}, t) = (f_1(\mathbf{x}_I), g(\mathbf{x}_{II}) \cdot h(\mathbf{x}_{III}, f_1(\mathbf{x}_I), g(\mathbf{x}_{II}), t)) \\ f_1(\mathbf{x}_I) = x_1 \\ g(\mathbf{x}_{II}) = 1 - \sum_{x_i \in \mathbf{x}_{II}} \sqrt{x_i - G(t)} - \sum_{x_j \in \mathbf{x}_{III}} (x_j - G(t))^2 \\ h(\mathbf{x}_{III}, f_1, g, t) = 1 - \left(\frac{f_1}{g}\right)^{H(t)} \\ \text{where :} \\ G(t) = \sin(0.5\pi t), \quad t = \frac{1}{n_t} \lfloor \frac{\tau}{\tau_t} \rfloor \\ H(t) = 1.5 + G(t) \\ \mathbf{x}_I \in [0, 1]; \quad \mathbf{x}_{II}, \mathbf{x}_{III} \in [-1, 1] \end{cases} \quad (33)$$

where \mathbf{x}_I , \mathbf{x}_{II} and \mathbf{x}_{III} are subsets of the decision variables, f_1 affects the spread of solutions of POF , g affects the level of difficulty an algorithm experiences when converging to POF and h influences the discontinuity or convexity of POF [73].

For DMOOP₁, both its POS and POF changes over time, defined as:

$$POS : x_i = G(t), \quad \forall x_i \in \mathbf{x}_{II}, \mathbf{x}_{III}$$

$$POF : y = 1 - f_1^{H(t)}$$

Let $\mathbf{x}_{II} = \{x_1, x_2, x_3\}$, $\mathbf{x}_{III} = \{x_4, x_5, x_6\}$, $t = 0.1$, $G(t) = 0.156$, $\mathbf{x}_1 = \{0.14, 0.16, 0.16, 0.16, 0.16, 0.16\}$ and $\mathbf{x}_2 = \{0.16, 0.16, 0.16, 0.16, 0.14, 0.16\}$. Then, measuring the distance between the solution and the true POS (i.e. in decision space), $d(\mathbf{x})_{dec}$, \mathbf{x}_1 and \mathbf{x}_2 obtains the same d_{dec} value. However, \mathbf{x}_1 and \mathbf{x}_2 produces the following gh values respectively: 0.937512 and 0.93183. The true POF value for \mathbf{x}_1 and \mathbf{x}_2 are 0.961453 and 0.951914 respectively. The difference between the gh values found by \mathbf{x}_1 and \mathbf{x}_2 and the true POF values, d_{obj} , are 0.023941 and 0.020084 respectively. Therefore, even though in the decision space the difference between \mathbf{x}_1 and \mathbf{x}_2 and the true POS produces the same d_{dec} value, their difference in objective space, d_{obj} , is different, with \mathbf{x}_2 being closer to the true POF than \mathbf{x}_1 .

Measuring VD in the decision space will indicate how close the decision variable values are from POS . However, the VD value measured in decision space will not give a true reflection of the accuracy of POF^* with regards to POF . Therefore, measuring VD in decision space to determine the accuracy of the algorithm's solutions found, is only appropriate for DMOOPs of Type I where the POS changes over time, but the POF remains static. However, for DMOOPs of Type II and III, measuring VD in the decision space will not provide any information with regards to how well the algorithm has tracked the changing POF and therefore for these type of DMOOPs VD should be measured in objective space.

The following papers measured either GD or VD in only the decision space: [24; 25; 42; 49; 60]. In [60] only FDA1, which is a Type I DMOOP, was used and therefore measuring in the decision variable space is appropriate. In [42], three DMOOPs of Type I (FDA1, DIMP1 and DIMP2) were used and one DMOOP of Type II (FDA3). For the Type II DMOOP, calculating in the decision space will only provide information with regards to tracking of the changing POS, but not with regards to the tracking of the changing POF. In [24; 25; 49], DMOOPs of Types I, II and III were used. For the DMOOP of Type III, measuring in the decision space only indicates whether POS^* remains close to POS , which remains static over time. Therefore, it provides no information with regards to how well the algorithm has tracked the changing POF . The issue of calculating performance measures in either decision or objective space is unique to DMOO, since with SMOO both the POS and POF remain static.

5.5. Comparing Performance of Algorithms

When the performance of various algorithms are compared against one another, typically various performance measures are used. Some algorithms will perform very well with regards to certain performance measures and not so well with regards to other performance measures. Therefore, typically each algorithm will be ranked according to its performance with regards to each performance measure. Then, for each algorithm its average rank is calculated. These averaged ranks are then used to determine how well each algorithm performed with regards to the other algorithms. Examples can be found in [12; 33]. However, it should be noted that the performance measures that are used for comparing various algorithms should be chosen with care. If the wrong performance measures are selected, it may lead to incorrect ordering as discussed in Section 5.1 and illustrated in Table 14 and [34]. Therefore, if *POF* is known, the usage of *acc_{alt}* is suggested. However, more research is required to determine the best performance measure(s) for cases where *POF* is unknown.

6. Conclusion

This article provided an overview of the performance measures that are currently being used to measure the performance of DMOO algorithms. Problems with current performance measures were discussed, showing that algorithms that lose track of the POF and outliers in the found POF can produce misleading results when performance measures based on distance calculations, performance measures measuring the extend or spread of the non-dominated solutions, the hypervolume or the hypervolume ratio are used to measure the performance of DMOO algorithms. The importance of managing boundary constraint violations were highlighted. Furthermore, the difference between calculating accuracy performance measures in decision variable space and objective space was discussed.

Taking into consideration the issues with current performance measures, the importance of performance measure selection when comparing various DMOO algorithms' performance and the influence that this selection of performance measures can have on the ordering of the algorithms, were presented.

Future work includes testing the performance of DMOO algorithms using the acc_{alt} performance measure, i.e. the absolute value of the difference in HV values of the approximated POF and the true POF, to quantify the performance of the algorithms. Furthermore, more research is required with regards to performance measures that are not vulnerable to the issues discussed in this article and that can be used when the true POF is unknown.

References

- [1] Z. Avdagić, S. Konjicija, and S. Omanović, *Evolutionary approach to solving non-stationary dynamic multi-objective problems*, Foundations of Computational Intelligence Volume 3 (A. Abraham, A-E. Hassanien, P. Siarry, and A. Engelbrecht, eds.), Studies in Computational Intelligence, vol. 203, Springer Berlin/Heidelberg, 2009, pp. 267–289.
- [2] D. Ayvaz, H., and F. Gurgun, *Performance evaluation of evolutionary heuristics in dynamic environments*, Applied Intelligence **37** (2012), no. 1, 130–144.
- [3] C.R.B. Azevedo and A.F.R. Araujo, *Generalized immigration schemes for dynamic evolutionary multiobjective optimization*, Proceedings of Congress on Evolutionary Computation, June 2011, pp. 2033–2040.
- [4] E. Besada-Portas, L. de la Torre, A. Moreno, and J.L. Risco-Martín, *On the performance comparison of multi-objective evolutionary {UAV} path planners*, Information Sciences **238** (2013), no. 0, 111–125.
- [5] N. Beume and G. Rudolph, *Faster s-metric calculation by considering dominated hypervolume as klee’s measure problem*, Proceedings of Computational Intelligence, IASTED/ACTA Press, 2007, pp. 233–238.
- [6] M. Cámara, J. Ortega, and F. de Toro, *The parallel single front genetic algorithm (psfga) in dynamic multi-objective optimization*, Computational and Ambient Intelligence (F. Sandoval, A. Prieto, J. Cabestany, and M. Graña, eds.), Lecture Notes in Computer Science, vol. 4507, Springer Berlin/Heidelberg, 2007, pp. 300–307.
- [7] M. Cámara, J. Ortega, and F. de Toro, *A single front genetic algorithm for parallel multi-objective optimization in dynamic environments*, Neurocomputing **72** (2009), no. 16-18, 3570–3579, Financial Engineering; Computational and Ambient Intelligence (IWANN 2007).
- [8] M. Cámara, J. Ortega, and F. de Toro, *Approaching dynamic multi-objective optimization*

- problems by using parallel evolutionary algorithms*, Advances in Multi-Objective Nature Inspired Computing (C. Coello Coello, C. Dhaenens, and L. Jourdan, eds.), Studies in Computational Intelligence, vol. 272, Springer Berlin/Heidelberg, 2010, pp. 63–86.
- [9] M. Cámara, J. Ortega, and F.J. de Toro, *Parallel processing for multi-objective optimization in dynamic environments*, International Parallel and Distributed Processing Symposium **0** (2007), 243–250.
- [10] H. Chen, M. Li, and X. Chen, *Using diversity as an additional-objective in dynamic multi-objective optimization algorithms*, Electronic Commerce and Security, International Symposium **1** (2009), 484–487.
- [11] S. Cheng, Y. Shi, and Q. Qin, *On the performance metrics of multiobjective optimization*, Advances in Swarm Intelligence (Ying Tan, Yuhui Shi, and Zhen Ji, eds.), Lecture Notes in Computer Science, vol. 7331, Springer Berlin Heidelberg, 2012, pp. 504–512.
- [12] Pinar Civicioglu, *Artificial cooperative search algorithm for numerical optimization problems*, Information Sciences **229** (2013), no. 0, 58–76.
- [13] K. Deb, *Multi-objective optimization using evolutionary algorithms*, John Wiley & Sons, Ltd, 2004.
- [14] K. Deb, *Single and multi-objective dynamic optimization: two tales from an evolutionary perspective*, Tech. Report 2011004, Kalyanmoy Deb Kanpur Genetic Algorithms Laboratory (KanGAL), February 2011.
- [15] K. Deb, S. Agarwal, A. Pratap, and T. Meyarivan, *A fast and elitist multiobjective genetic algorithm: Nsga-ii*, Tech. Report 200001, Kanpur, India, 2000.
- [16] K. Deb, A. Anand, and D. Joshi, *A computationally efficient evolutionary algorithm for real-parameter optimization*, Evolutionary Computation **10** (2002), no. 4, 371–395.
- [17] K. Deb, U.B. Rao N., and S. Karthik, *Dynamic multi-objective optimization and decision-making using modied nsga-ii: a case study on hydro-thermal power scheduling*, Proceedings of International Conference on Evolutionary Multi-criterion optimization (Matsushima, Japan), 2007, pp. 803–817.
- [18] A.P. Engelbrecht, *Fundamentals of computational swarm intelligence*, John Wiley and Sons, Ltd, 2005.
- [19] A.P. Engelbrecht, *Fruitless wandering: Where are my particles?*, 2012, Proceedings of World Congress on Computational Intelligence: Congress on Evolutionary Computation. To be published.
- [20] M. Farina, K. Deb, and P. Amato, *Dynamic multiobjective optimization problems: test*

- cases, approximations, and applications*, IEEE Transactions on Evolutionary Computation **8** (2004), no. 5, 425–442.
- [21] L.J. Fogel, *On the organization of intellect*, Ph.D. thesis, University of California, 1964.
- [22] C.M. Fonseca, L. Paquete, and M. Lopez-Ibanez, *An improved dimension-sweep algorithm for the hypervolume indicator*, Proceedings of Congress on Evolutionary Computation, July 2006, pp. 1157–1163.
- [23] F.v.d.Bergh, *An analysis of particle swarm optimizers*, Ph.D. thesis, Department of Computer Science University of Pretoria, 2002.
- [24] C-K. Goh and K. Tan, *A coevolutionary paradigm for dynamic multi-objective optimization*, Evolutionary Multi-objective Optimization in Uncertain Environments, Studies in Computational Intelligence, vol. 186, Springer Berlin/Heidelberg, 2009, pp. 153–185.
- [25] C-K. Goh and K.C. Tan, *A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization*, IEEE Transactions on Evolutionary Computation **13** (2009), no. 1, 103–127.
- [26] M. Greeff and A. Engelbrecht, *Dynamic multi-objective optimisation using pso*, Multi-Objective Swarm Intelligent Systems (Nadia Nedjah, Leandro dos Santos Coelho, and Luiza de Macedo Mourelle, eds.), Studies in Computational Intelligence, vol. 261, Springer Berlin/Heidelberg, 2010, pp. 105–123.
- [27] M. Greeff and A.P. Engelbrecht, *Solving dynamic multi-objective problems with vector evaluated particle swarm optimisation*, Proceedings of World Congress on Computational Intelligence (WCCI): Congress on Evolutionary Computation (Hong Kong), June 2008, pp. 2917–2924.
- [28] S-U. Guan, Q. Chen, and W. Mo, *Evolving dynamic multi-objective optimization problems with objective replacement*, Artificial Intelligence Review **23** (2005), 267–293.
- [29] M.P. Hansen and A. Jaskiewicz, *Evaluating the quality of approximations to the non-dominated set*, Tech. Report IMM-REP-1998-7, Technical University of Denmark, 22 March 1998.
- [30] I. Hatzakis and D. Wallace, *Dynamic multi-objective optimization with evolutionary algorithms: a forward-looking approach*, Proceedings of the Conference on Genetic and Evolutionary Computation (New York, NY, USA), ACM, 2006, pp. 1201–1208.
- [31] M. Helbig and A.P. Engelbrecht, *Archive management for dynamic multi-objective optimisation problems using vector evaluated particle swarm optimisation*, Proceedings of Congress on Evolutionary Computation (New Orleans, U.S.A.), June 2011, pp. 2047–2054.

- [32] M. Helbig and A.P. Engelbrecht, *Dynamic multi-objective optimisation problems*, Tech. report, University of Pretoria, Pretoria, South Africa, 2013.
- [33] M. Helbig and A.P. Engelbrecht, *Issues with performance measures for dynamic multi-objective optimisation*, Proceedings of IEEE Symposium Series on Computational Intelligence (Singapore), apr 2013, pp. 17–24.
- [34] M. Helbig and A.P. Engelbrecht, *Metaheuristics for dynamic optimization*, Studies in Computational Intelligence, ch. Dynamic multi-objective optimization using PSO, pp. 147–188, Springer Verlag Berlin/Heidelberg, 2013, Available online at: http://link.springer.com/chapter/10.1007/978-3-642-30665-5_8. Last accessed on: 5 December 2012.
- [35] J.H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.*, University of Michigan Press, Oxford, UK, 1975.
- [36] A. Isaacs, V. Puttige, T. Ray, W. Smith, and S. Anavatti, *Development of a memetic algorithm for dynamic multi-objective optimization and its applications for online neural network modeling of uavs*, Proceedings of World Congress on Computational Intelligence: International Joint Conference on Neural Networks, June 2008, pp. 548 –554.
- [37] A. Isaacs, T. Ray, and W. Smith, *Memetic algorithm for dynamic bi-objective optimization problems*, Proceedings of Congress on Evolutionary Computation (Trondheim, Norway), may 2009, pp. 1707–1713.
- [38] J. Kennedy and R.C. Eberhart, *Particle swarm optimization*, Proceedings of International Conference on Neural Networks, vol. IV, 1995, pp. 1942–1948.
- [39] A.K.M. Khaled, A. Talukder, and M. Kirley, *A pareto following variation operator for fast-converging multiobjective evolutionary algorithms*, Proceedings of World Congress on Computational Intelligence: Congress on Evolutionary Computation, June 2008, pp. 2270–2277.
- [40] K. Kim, R.I. McKay, and B-R. Moon, *Multiobjective evolutionary algorithms for dynamic social network clustering*, Proceedings of the Conference on Genetic and Evolutionary Computation, 2010.
- [41] J.D. Knowles, *Local-search and hybrid evolutionary algorithms for pareto optimisation*, Ph.D. thesis, Department of Computer Science The University of Reading, 2002.
- [42] W. Koo, C. Goh, and K. Tan, *A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment*, Memetic Computing **2** (2010), no. 2, 87–110.
- [43] M.S. Lechuga, *Multi-objective optimisation using sharing in swarm optimisation algo-*

- rithms*, Ph.D. thesis, University of Birmingham, July 2009.
- [44] Y-W. Leung and Y. Wang, *U-measure: A quality measure for multiobjective programming*, IEEE Transactions on Systems, Man, and Cybernetics **33** (2003), no. 3, 337–343.
- [45] K. Li, S. Kwong, J. Cao, M. Li, J. Zheng, and R. Shen, *Achieving balance between proximity and diversity in multi-objective evolutionary algorithm*, Information Sciences **182** (2012), no. 1, 220–242, *Nature-Inspired Collective Intelligence in Theory and Practice*.
- [46] X. Li, J. Branke, and M. Kirley, *On performance metrics and particle swarm methods for dynamic multiobjective optimization problems*, Proceedings of Congress on Evolutionary Computation, sept. 2007, pp. 576–583.
- [47] C-A. Liu, *New dynamic multiobjective evolutionary algorithm with core estimation of distribution*, Electrical and Control Engineering, International Conference on **0** (2010), 1345–1348.
- [48] G. Lizárraga Lizárraga, *On the evaluation of the quality of non-dominated sets*, Ph.D. thesis, Centro de Investigación en Matemáticas, A.C. (CIMAT), April 2009.
- [49] J. Mehnen, G. Rudolph, and T. Wagner, *Evolutionary optimization of dynamic multiobjective functions*, Tech. Report CI-204/06, Universität Dortmund, Universität Dortmund, Fachbereich Informatik/XI, 44221, Dortmund, Germany, May 2006.
- [50] A. Nakib and P. Siarry, *Performance analysis of dynamic optimization algorithms*, Metaheuristics for Dynamic Optimization (Enrique Alba, Amir Nakib, and Patrick Siarry, eds.), Studies in Computational Intelligence, vol. 433, Springer Berlin Heidelberg, 2013, pp. 1–16.
- [51] T. Ray, A. Isaacs, and W. Smith, *A memetic algorithm for dynamic multiobjective optimization*, Multi-Objective memetic algorithms (C-K. Goh, Y-S. Ong, and K. Tan, eds.), Studies in Computational Intelligence, vol. 171, Springer Berlin/Heidelberg, 2009, pp. 353–367.
- [52] R.E. Rosenthal, *Principles of multiobjective optimization*, Decision Sciences **16** (1985), 132–152.
- [53] B. Sarasola and E. Alba, *Quantitative performance measures for dynamic optimization problems*, Metaheuristics for Dynamic Optimization (Enrique Alba, Amir Nakib, and Patrick Siarry, eds.), Studies in Computational Intelligence, vol. 433, Springer Berlin Heidelberg, 2013, pp. 17–33.
- [54] J.R. Schott, *Fault tolerance design using single and multi-criteria genetic algorithms*, Master’s thesis, Department of Aeronautics and Astronautics Massachusetts Institute of Tech-

nology, 1995.

- [55] D. Schuhmacher, B-T. Vo, and B-N. Vo, *A consistent metric for performance evaluation of multi-object filters*, IEEE Transactions on Signal Processing **56** (2008), no. 8, 3447–3457.
- [56] M. Sierra and C. Coello Coello, *Improving pso-based multi-objective optimization using crowding, mutation and ϵ -dominance*, Evolutionary multi-criterion optimization (Carlos Coello Coello, Arturo Hernandez Aguirre, and Eckart Zitzler, eds.), Lecture Notes in Computer Science, vol. 3410, Springer Berlin Heidelberg, 2005, pp. 505–519.
- [57] M. Cámara Sola, *Parallel processing for dynamic multi-objective optimization*, Ph.D. thesis, Universidad de Granada, Dept. of Computer Architecture and Computer Technology, Universidad de Granada, Spain, April 2010.
- [58] R. Storn and K. Price, *Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces*, Journal of Global Optimization **11** (1997), no. 4, 341–359.
- [59] A.K.M. Talukder and A. Khaled, *Towards high speed multiobjective evolutionary optimizers*, Proceedings of the Genetic and Evolutionary Computation Conference (New York, NY, USA), ACM, 2008, pp. 1791–1794.
- [60] K. Tan and C. Goh, *Handling uncertainties in evolutionary multi-objective optimization*, Computational Intelligence: Research Frontiers (Jacek Zurada, Gary Yen, and Jun Wang, eds.), Lecture Notes in Computer Science, vol. 5050, Springer Berlin/Heidelberg, 2008, pp. 262–292.
- [61] E. Tantar, A-A. Tantar, and P. Bouvry, *On dynamic multi-objective optimization, classification and performance measures*, Proceedings of Congress on Evolutionary Computation, June 2011, pp. 2759–2766.
- [62] D.A. van Veldhuizen, *Multiobjective evolutionary algorithms: classification, analyses, and new innovations*, Ph.D. thesis, Graduate School of Engineering Air University, 1999.
- [63] Y. Wang and C. Dang, *An evolutionary algorithm for dynamic multi-objective optimization*, Applied Mathematics and Computation **25** (2008), 6–18.
- [64] Y. Wang and B. Li, *Investigation of memory-based multi-objective optimization evolutionary algorithm in dynamic environment*, Proceedings of Congress on Evolutionary Computation (Trondheim, Norway), may 2009, pp. 630–637.
- [65] Y. Wang and B. Li, *Multi-strategy ensemble evolutionary algorithm for dynamic multi-objective optimization*, Memetic Computing **2** (2010), no. 1, 3–24.
- [66] K. Weicker, *Performance measures for dynamic environments*, Parallel Problem Solv-

- ing from Nature (J. Guervós, P. Adamidis, H-G. Beyer, H-P. Schwefel, and J-L. Fernández-Villaca nas, eds.), Lecture Notes in Computer Science, vol. 2439, Springer Berlin/Heidelberg, 2002, pp. 64–73.
- [67] S-Y. Zeng, G. Chen, L. Zheng, H. Shi, H. de Garis, L. Ding, and L. Kang, *A dynamic multi-objective evolutionary algorithm based on an orthogonal design*, Proceedings of Congress on Evolutionary Computation (Vancouver, Canada), 16-21 July 2006, pp. 573–580.
- [68] Z. Zhang and S. Qian, *Artificial immune system in dynamic environments solving time-varying non-linear constrained multi-objective problems*, Soft Computing - A Fusion of Foundations, Methodologies and Applications **15** (2011), 1333–1349.
- [69] B. Zheng, *A new dynamic multi-objective optimization evolutionary algorithm*, Proceedings of Third International Conference on Natural Computation, vol. 5, August. 2007, pp. 565–570.
- [70] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, *Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization*, Evolutionary Multi-Criterion Optimization (Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, eds.), Lecture Notes in Computer Science, vol. 4403, Springer Berlin/Heidelberg, 2007, pp. 832–846.
- [71] E. Zitzler, *Evolutionary algorithms for multiobjective optimization: methods and applications*, Ph.D. thesis, Swiss Federal Institute of Technology (ETH) Zurich Switzerland, 1999.
- [72] E. Zitzler, D. Brockhoff, and L. Thiele, *The hypervolume indicator revisited: on the design of pareto-compliant indicators via weighted integration*, Evolutionary Multi-Criterion Optimization (S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, eds.), Lecture Notes in Computer Science, vol. 4403, Springer Berlin/Heidelberg, 2007, pp. 862–876.
- [73] E. Zitzler, K. Deb, and L. Thiele, *Comparison of multiobjective evolutionary algorithms: empirical results*, Evolutionary Computation **8** (2000), no. 2, 173–195.
- [74] E. Zitzler and L. Thiele, *Multiobjective optimization using evolutionary algorithms a comparative case study*, **1498** (1998), 292–301.
- [75] E. Zitzler and L. Thiele, *Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach*, IEEE Transactions on Evolutionary Computation **3** (1999), no. 4, 257–271.
- [76] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, and V. Grunert da Fonseca, *Performance assessment of multiobjective optimizers: an analysis and review*, IEEE Transactions on

Evolutionary Computation 7 (2003), no. 2, 117–132.



Mardé Helbig is a senior researcher at the Meraka Institute of the Council for Scientific and Industrial Research in South Africa. She obtained her Ph.D. from the University of Pretoria in South Africa. She is the vice-chair of the South African Chapter of the IEEE Computational Intelligence Society (CIS) and a member of the IEEE CIS Task Force on Multi-objective Optimisation. She annually serves as reviewer for four journals and three conferences. Her research interests include dynamic multi-objective optimisation, multi-objective optimisation and computational intelligence algorithms.



Andries Engelbrecht is a Professor and the head of department of computer science at the University of Pretoria. He holds the position of South African Research Chair in Artificial Intelligence, and leads the Computational Intelligence Research Group at the University of Pretoria. He has published over 200 papers in journals and international conference proceedings, and authored two books. Annually, he serves as reviewer for over 30 journals and 10 conferences, and is an Associate Editor for three top journals of the field. In addition, he was the founding chair of the South African chapter of the IEEE Computational Intelligence Society.