

# An Assignment Problem Formulation for Dominance Move Indicator


\*

1<sup>st</sup>Claudio Lucio do Val Lopes 2<sup>nd</sup> Flávio Vinícius Cruzeiro Martins

PPGMMC<sup>1</sup>

CEFET-MG


BH, Brazil

claudiolucio@gmail.com 

PPGMMC<sup>1</sup>

CEFET-MG


BH, Brazil

flaviocruzeiro@cefetmg.br 

3<sup>rd</sup> Elizabeth F. Wanner

PPGMMC<sup>1</sup> - CEFET-MG

Computer Science Group, Aston University,  
Birmingham, UK

efwanner@decom.cefetmg.br 

**Abstract**—Dominance move (DoM) is a binary quality indicator to compare solution sets in multiobjective optimization. The indicator allows a more natural and intuitive relation when comparing solution sets. Like the  $\epsilon$ -indicators, it is Pareto compliant and does not demand any parameters or reference sets. In spite of its advantages, the combinatorial calculation nature is a limitation. The original formulation presents an efficient method to calculate it in a bi-objective case only. This work presents an assignment formulation to calculate DoM in problems with three objectives or more. Some initial experiments, in the bi-objective space, were done to show that DoM has a similar interpretation as  $\epsilon$ -indicators, and to show that our model formulation is correct. Next, other experiments, using three dimensions, were also done to show how DoM could be compared with other indicators: inverted generational distance (IGD) and hypervolume (HV). The assignment formulation for DoM is valid not only for three objectives but for more. Finally, there are some strengths and weaknesses, which are discussed and detailed.

**Index Terms**—multiobjective optimization, quality indicator, performance assessment, exact method, evolutionary algorithms

## I. INTRODUCTION

Many real-world optimization problems are composed of multiple and conflicting objectives. Although traditional approaches can be used to combine the objectives into a single one and solve the resulting problem, several multi and many-objective optimization techniques have proven to be efficient techniques dealing with the true multiobjective nature of such problems. [1].

The solution sets are formed in such a way that each solution represents a trade-off among objectives. If a comparison of different solution sets is needed, many performance measures can be applied [2]. Graphical techniques represent an alternative way to help examine the solution sets visually. Those techniques are quite useful when the problems have two or three objectives only. However, when the number of objectives is higher than three, this task is challenging (if not impractical), needing proper visualization techniques that can exhibit solution set features like location, shape, and distribution [3].

<sup>1</sup> PPGMMC - Programa de pós-graduação em modelagem matemática e computacional

When it is necessary to summarize the solution sets, taking into account their characteristics and features, quality indicators are widely applied [4]. These indicators have been used to compare the outcomes of multiobjective optimizers quantitatively. Ideally, a quality indicator should be able not only to state whether an outcome is better than others but also in what aspects. In a recent paper [2], 100 quality indicators were discussed including some that are considered state-of-the-art. The quality aspects of these indicators, as well as their strengths and weaknesses, are examined and compared.

A unary quality indicator is a mapping that assigns an approximation set to a real number [5], and it is used to compare approximation sets generated by an optimizer. Inverted generational distance (IGD) [6], hypervolume (HV) [7], [8], [9], and  $\epsilon$ -additive/multiplicative indicator [4] are some examples, to name a few. Despite their applicability, some indicators require a pre-defined reference point or the knowledge of the true Pareto front.

While unary indicators are able to summarize only one solution set, binary indicators take into account two approximation sets and return a real value, which can be used to say whether an approximation is better than others. For two sets  $P$  and  $Q$ , for example, if  $P$  weakly dominates  $Q$ , then  $I(P, Q) = 0$ . If  $P$  dominates some points of  $Q$ , and  $Q$  does not dominate any point of  $P$ , it is fair to expect that the indicator supports  $P$  to  $Q$ .

In [10], a new quality measure, called dominance move (DoM) is proposed. DoM measures the minimum ‘effort’ that one solution set has to make in trying to dominate another set, more specifically the sum of the movement needed to make a set dominant. It has the same  $\epsilon$ -indicators’ interpretation and it can capture all quality aspects of solution sets, such as Pareto convergence and spread. The authors propose an exact algorithm to calculate DoM for the bi-objective case that can be computed in low computational cost. However, as stated in [10], it can not be used or extended to three or more dimensions due to the indicator combinatorial nature.

In an attempt to overcome this difficulty, this work focuses on a DoM formulation as an assignment problem, and its

solution using a mixed-integer programming method [11] is proposed. Assignment problems have some variants; however, it is common for the problems to involve a form to optimally match the elements of two or more sets, in which the assignment's complexity refers to the number of items to be matched [12].

The paper is organized as follows. In Section II, some definitions and related work are presented. Section III introduces the DoM indicator and our formulation to treat it as an assignment problem. Next, a mixed-integer programming formulation is also presented as an assignment implementation. Its strengths and weaknesses to solve the assignment problem are also discussed. Some experiments are presented in Section IV, firstly in the bi-objective case, showing that DoM overcomes some  $\epsilon$ -indicators weaknesses, and it is still in agreement with the DoM calculation algorithm developed in [2]. Moreover, in the experiments section, some solution sets were generated by IBEA, NSGAI, and SPEA2 and used to assess and compare the DoM indicator with other common indicators (HV and IGD). In Section V, some observations and future research considerations are finally described.

## II. DEFINITIONS AND RELATED WORK

In general, a multiobjective optimization problem (MOP) includes  $\mathbf{x}$  decision variable vector from a decision space  $\Omega \subseteq R^N$ , and a set of  $M$  objective functions. Without loss of generality, a minimization MOP can be simply defined as [13]:

$$\text{Min } F(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_{|M|}(\mathbf{x})]^T, \quad \mathbf{x} \in \Omega \quad (1)$$

The  $F : \Omega \rightarrow \Theta \subseteq R^M$  is formed by a set of  $M$  objective functions, which is a mapping from decision space  $\Omega$  to vectors in  $M$ -dimensional objective space  $\Theta$ . We are interested in the evaluation of these objective vector (solution) sets, and the comparison relation among these vectors.

Considering two solutions  $\mathbf{p}, \mathbf{q} \in \Theta$ , it is possible to establish a relation in which  $\mathbf{p}$  is said to *weakly dominate*  $\mathbf{q}$  if  $p_i \leq q_i$  for  $1 \leq i \leq M$ , and is denoted as  $\mathbf{p} \preceq \mathbf{q}$ . In addition, if there is at least one objective  $i$  in which  $p_i < q_i$  then it is said that  $\mathbf{p}$  *dominates*  $\mathbf{q}$ , and is denoted as  $\mathbf{p} \prec \mathbf{q}$ . A solution  $\mathbf{p} \in \Theta$  is called Pareto optimal if there is no  $\mathbf{q} \in \Theta$  that dominates  $\mathbf{p}$ . The set of all Pareto optimal solutions of an MOP is called Pareto optimal frontier. In the same way, the weak dominance relation can be defined to solution sets:

*Weak Dominance*: The set  $P$  weakly dominates  $Q$ , denoted as  $P \preceq Q$ , if every solution  $\mathbf{q} \in Q$  is weakly dominated by at least one solution  $\mathbf{p} \in P$ .

The goal of a multiobjective algorithm is to generate approximation sets representing the Pareto front of a MOP. In the last years, the growth of multiobjective algorithms lead to a key issue: the evaluation and comparison of approximation sets generated by these algorithms. To assess the quality of sets in MOP, one must take into account several aspects, such as convergence to the true Pareto front, spread of the solution, etc. Quality indicators represent a way to quantitatively evaluate the approximation sets generated by different algorithms.

Ideally, a quality indicator should not only be able to say which algorithm is better than the other but also to identify in what aspects. The following definition formalizes a quality indicator [4]:

*Quality indicator*: An  $k$ -ary quality indicator  $I$  is a function  $I: \Theta^k \rightarrow \mathbb{R}$ , which assigns each vector of  $k$  solutions sets  $(P_1, P_2, \dots, P_k)$  a real value  $I(P_1, P_2, \dots, P_k)$ .

Quality indicators can be unary, binary, or  $k$ -ary, defining a real value to one solution set, two solution sets, or  $k$  solution sets, respectively. For a comprehensive review of quality indicators, in [2], some indicators are defined and discussed using their quality facets as: *convergence*, *spread*, *uniformity*, and *cardinality*. Issues such strengths, weaknesses, and evaluation are also analyzed.

Many indicators have been used in multiple situations in the literature [2]. Hypervolume (HV), used in [8], [9], inverted generational distance (IGD) used in [6], and  $\epsilon$ -indicator are some examples. DoM and these indicators will be used in this paper and are defined below:

- **Hypervolume (HV)**: Let  $r' = (r_1, \dots, r_m)$  be reference points in the objective space that is dominated by all approximation sets. Let  $P$  be one approximation set. The HV value of  $P$  with regard to  $r'$  represents the volume of the region which is dominated by  $P$  and dominates  $r'$ . Generally, the computational cost is exponential regarding to the number of objectives.
- **Inverted generational distance (IGD)**: Let  $R^* = (r^*_1, \dots, r^*_m)$  be a reference set of uniformly distributed points on the Pareto front. Considering  $P$  as an approximation set to the Pareto front, the inverted generational distance between  $R^*$  and  $P$  is defined as:

$$IGD(R^*, P) = \frac{\sum_{r \in R^*} d(r, P)}{|R^*|} \quad (2)$$

$d(r, P)$  is the minimum Euclidean distance from point  $r$  to approximation set  $P$ . The IGD metric is able to measure both diversity and convergence of  $P$  if  $|R^*|$  is large enough [14]. The computational cost is  $O(|M| \times |R^*| \times |P|)$ , where  $|M|$  is the number of objectives.

- **$\epsilon$ -additive/multiplicative indicator**: it is a extension to the evaluation of approximation schemes in operational research and theory [4]. For two solution sets  $P$  and  $Q$ , the additive  $\epsilon$ -indicator,  $I_\epsilon(P, Q)$  is the minimum value that can be added to each solution in  $Q$ , such that they become weakly dominated by at least one solution in  $P$ . Formally, the additive  $\epsilon$ -indicator is calculated as:

$$I_{\epsilon+}(P, Q) = \max_{q \in Q} \min_{p \in P} \max_{i \in \{1..|M|\}} p^i - q^i \quad (3)$$

in which  $p^i$  denotes the objective value of solution  $p$  in the  $i$ th objective, and  $|M|$  is the number of objectives. For the multiplicative  $\epsilon$ -indicator, the  $p^i - q^i$  is replaced by  $\frac{p^i}{q^i}$ . A value of  $I_{\epsilon+}(P, Q) \leq 0$  or  $I_{\epsilon \times}(P, Q) \leq 1$  implies that  $P$  weakly dominates  $Q$ . The computational cost is  $O(|M| \times |P| \times |Q|)$ .

- **Dominance move (DoM):** it is a measure for comparing two sets of multidimensional points being classified as a binary indicator. It considers the movement of points in one set to make this set weakly dominated by the other set. DoM can be defined as follows, [10]: Let  $P$  be a set of points in  $\{p_1, p_2, \dots, p_{NP}\}$  and  $Q$  be a set of points in  $\{q_1, q_2, \dots, q_{NQ}\}$ . The dominance move of  $P$  to  $Q$ ,  $D(P, Q)$ , is the minimum total distance of moving points of  $P$ , such that any point in  $Q$  is weakly dominated by at least one point in  $P$ . In fact, the problem is to find  $\{p'_1, p'_2, \dots, p'_{NP}\}$  from  $\{p_1, p_2, \dots, p_{NP}\}$  positions, such that  $P'$  weakly dominates  $Q$ , and the total move from  $\{p_1, p_2, \dots, p_{NP}\}$  to  $\{p'_1, p'_2, \dots, p'_{NP}\}$ , denoted as  $d(p_i, p'_i)$ , must be minimum. The formal definition of DoM can be expressed as:

$$D(P, Q) = \underset{P' \leq Q}{\text{minimize}} \sum_{i=1}^{NP} d(p_i, p'_i) \quad (4)$$

The number of possibilities to find  $P'$  is numerous. Any combination of some  $P'$  can dominate  $Q$ , considering (4). The authors of [10], proposed an exact solution for calculating DoM in a bi-objective case [2]. The algorithm can be outlined as:

*Step 1:* Remove the dominated points in both  $P$  and  $Q$ , separately. Remove the points of  $Q$  that are dominated by at least one point in  $P$ .

*Step 2:* Denote  $R = P \cup Q$  and start the process. Each point of  $Q$  in  $R$  is considered as a group. For each group of  $Q$ , find its inward neighbor  $r = n_R(q)$  in  $R$ . If the point  $r \in P$ , then merge  $r$  into the group of  $q$ , otherwise  $r \in Q$ . If  $r$  is not assigned to one group, merge the two groups of  $q$  and  $r$  into one group.

*Step 3:* If there exists no point  $q \in Q$  such that  $q = n_R(n_R(q))$  (i.e., there is a loop between the points) in any group, then the procedure ends and there is an optimal solution to the case.

*Step 4:* There is a loop in one or some groups. The procedure replaces these loops with the ideal point. The ideal point is formed by the best of each objective in each point inside the loop or group. Return to step 3 until convergence.

The definitions, theorems, and corollaries to prove that this algorithm is correct in the bi-objective case are presented in [2]. Furthermore, DoM is Pareto dominant compliant and any prior problem knowledge and pre-defined parameter are not necessary. However, due to the combinatorial nature of the problem, the authors stated that there is no solution for three or more objectives.

### III. THE DOMINANCE MOVE CALCULATION AS AN ASSIGNMENT PROBLEM

Our proposal concept of DoM calculation is based on the observation that the problem is, in fact, a particular case of an assignment problem with two levels and some constraints. To deal with the problem, we have to find an assignment of

$P$  to  $Q$  with the restrictions that each  $q$  must be assigned to one  $p$  with the minimum distance. Nevertheless, in classic assignment problems,  $P$  does not change its features, and this aspect must be considered for the DoM calculation.

A simple and hypothetical example to clarify the situation can be given as follows: consider  $P$  as  $\{(1.5, 1.3, 1.1), (1.4, 2.1, 1.8)\}$  and  $Q$  as  $\{(1.4, 1.2, 1.0), (1.3, 2.0, 2.0)\}$ . The possible inward neighbor  $r = n_R(q)$  of points  $q_1$ , and  $q_2$  can be, respectively,  $p_1$  and  $p_2$ . This creates an assignment of  $P$  to  $Q$  with the minimum  $D(P, Q)$ , considering that  $P$  is fixed:  $D(P, Q) = d(p_1, q_1) + d(p_2, q_2) = 0.5$ . However, if we considered a movement from  $P$  to  $P'$ , then  $p_1$  would be transformed into  $p'_1 = \{(1.4, 1.3, 1.1)\}$ . In this sense, we can find a better assignment and lower value of  $D(P, Q) = d(p_1, p'_1) + d(p'_1, q_1) + d(p'_1, q_2) = 0.4$ . Clearly, other assignments from  $P$  to  $P'$  and to  $Q$  are capable to generate the same value.

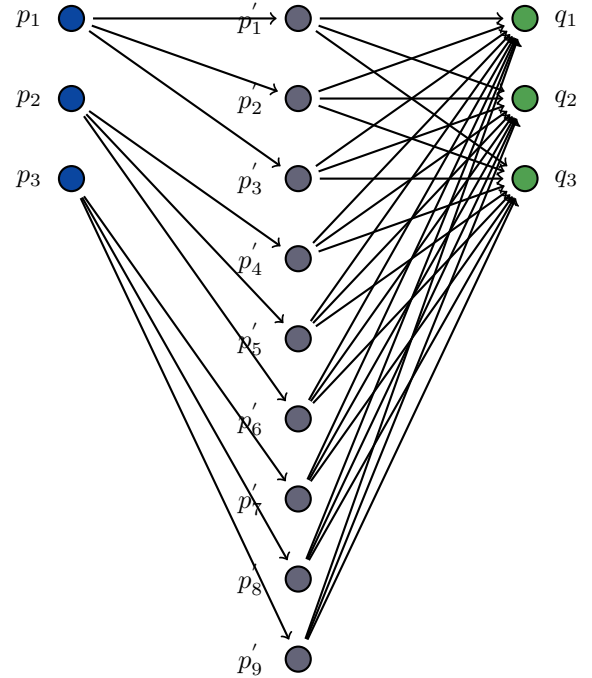


Figure 1. One possible example of assignment between  $P$ , with  $NP = 3$ , and  $Q$  with  $NQ = 3$ . Considering improvements in  $P$ ,  $P'$  is generated, and in this example,  $NP' = 9$ . The distances between  $P$  to  $P'$  and  $P'$  to  $Q$  are, respectively,  $d(p_i, p'_k)$  and  $d(p'_k, q_j)$  corresponding to edges.

Inspired by an assignment problem approach, the proposed DoM calculation is explained in detail. Let  $P$  and  $Q$  be two sets of points, with  $p_i$  points  $i \in \{1, \dots, NP\}$  and  $q_j$  points  $j \in \{1, \dots, NQ\}$ .  $P$  and  $Q$  are given in the problem.  $P'$  is a set of points containing the candidates to dominate  $Q$  with some update in one or many objectives. This update generates a better distance such as expressed in (4). It should be noted that  $p'_k$  with  $k \in \{1, \dots, NP'\}$  can be generated from  $p$ .

To better understand our approach, this concept is illustrated in Figure(1). Suppose the solution sets  $P$  and  $Q$  have both  $NP = NQ = 3$ .  $P'$  is generated with 3 possibilities for each

$p_i$ . In total, there are 9  $p'_k$  generated. The first assortment of edges from  $p_i$  to  $p'_k$  represents the distance  $d(p_i, p'_k)$  as a way to improve  $p_i$  generating  $p'_k$  candidate. The second assortment is from  $p'_k$  to  $q_j$ , and represents the distance  $d(p'_k, q_j)$ , which can be seen as the distance from some  $p'_k$  to weakly dominate some  $q_j$  or a  $g$  group formed by more than one  $q_j$ .

In a typical assignment problem, the goal is to find a one-to-one match between  $n$  tasks and  $o$  agents, for example. The objective function minimizes the total cost of the assignments as  $c_{(i,j)}$ , from task  $j$  to agent  $i$ . At most, one agent must do a task, and every task must be done, as proposed in [12]. The mathematical model for the classic assignment problem is given as in (5):

$$\begin{aligned}
& \text{minimize } \sum_{i=1}^N \sum_{j=1}^O c_{(i,j)} x_{(i,j)} \\
& \text{subject to} \\
& \sum_{i=1}^N x_{(i,j)} = 1, \quad \forall j \in O \\
& \sum_{j=1}^O x_{(i,j)} = 1, \quad \forall i \in N \\
& x_{(i,j)} \in \{0, 1\} \quad \forall i \in N, \forall j \in O
\end{aligned} \tag{5}$$

The DoM assignment model is detailed in (6). The objective function searches for a valid path from  $p_i$  to  $q_j$  through  $p'_k$  with the minimum distance between pairs. The first set of constraints guarantee that, for each  $q_j$ , there is a valid path. The next constraints involving  $xc_{(k,j)}$ , a binary variable, guarantee that the path from  $p_i$  to  $q_j$  is valid. When  $xc_{(k,j)}$  is 1, then there is a valid path in the assignment graph. On the other hand, if  $xc_{(k,j)}$  is 0, the path is infeasible. Essentially, it is necessary that  $p'_k$ , generated from  $p_i$ , must be shared between  $p_i$  and  $q_j$ . Again, the problem can be viewed as a bipartite graph with two layers, such as the example in Figure 1.

$$\begin{aligned}
& \text{minimize } \sum_{i=1}^{NP} \sum_{k=1}^{NP'} d(p_i, p'_k) x_{(i,k)} + \sum_{k=1}^{NP'} \sum_{j=1}^{NQ} d(p'_k, q_j) x_{(k,j)} \\
& \text{subject to} \\
& \sum_{k=1}^{NP'} xc_{(k,j)} = 1, \quad j = (1, \dots, NQ) \\
& xc_{(k,j)} \leq x_{(i,k)}, \quad i = (1, \dots, NP), k = (1, \dots, NP'), \\
& \quad j = (1, \dots, NQ) \\
& xc_{(k,j)} \leq x_{(k,j)}, \quad k = (1, \dots, NP'), j = (1, \dots, NQ) \\
& xc_{(k,j)} \geq x_{(i,k)} + x_{(k,j)} - 1, \quad i = (1, \dots, NP), \\
& \quad k = (1, \dots, NP'), j = (1, \dots, NQ) \\
& x_{(i,k)} \in \{0, 1\}, \quad i = (1, \dots, NP), k = (1, \dots, NP') \\
& x_{(k,j)} \in \{0, 1\}, \quad k = (1, \dots, NP'), j = (1, \dots, NQ) \\
& xc_{(k,j)} \in \{0, 1\} \quad k = (1, \dots, NP'), j = (1, \dots, NQ)
\end{aligned} \tag{6}$$

In (6), the  $d(p_i, p'_k)$  and  $d(p'_k, q_j)$  must be computed beforehand. Two distance matrices can represent these two parameters. Consider that  $|L|$  is the number of solutions in an arbitrary set, for example, and  $-M-$  the number of objectives. The total number of pairwise comparisons to calculate the distance matrix is  $\lceil |L|(|L| - 1)/2 \rceil$ . Each comparison can be a vector operation with  $-M-$  summations to obtain a pairwise distance element. Some works deal with the task of how to calculate the distance matrices efficiently such as in [15]. However, these computations can become prohibitive when either  $|L|$  or  $|M|$  are large (thousands of magnitude). For the solution sets context, the number of elements represents a critical value to be chosen.

It is important to note that model (6) does not deal with the problem of finding the  $p'_k$  candidates. Proposing the  $p'_k$  candidates is a hard task given its combinatorial nature. Still, it is possible to use the full combinatorial approach, which represents all the combinations selecting one  $p_i$  and all the possible  $g$  groups in  $Q$ . Another exploratory possibility could try to learn ‘good’ candidate features. A machine learning approach could use a limited number of generated candidates and find such characteristics in candidates using a loss function as in (6). Considering  $g$  as a group with one or many  $q_j$ , and assuming that  $p_i$  will be used as a base to be updated, one could generate  $p'_k$  candidates which weakly dominate all  $g$  group while minimizing (6).

## IV. EXPERIMENTS

### A. Bi-objective experiments

The first experiment was done to show that DoM and  $\epsilon$ -indicators have a similar interpretation. We used the same simple bi-objective problem proposed in [4]. There are four solution sets as can be viewed in Figure 2. P is the Pareto front and there is a dominance relation among  $A_1, A_2$  and  $A_3$ :  $A_1 \succeq A_2, A_1 \succeq A_3, A_2 \succeq A_3$ .

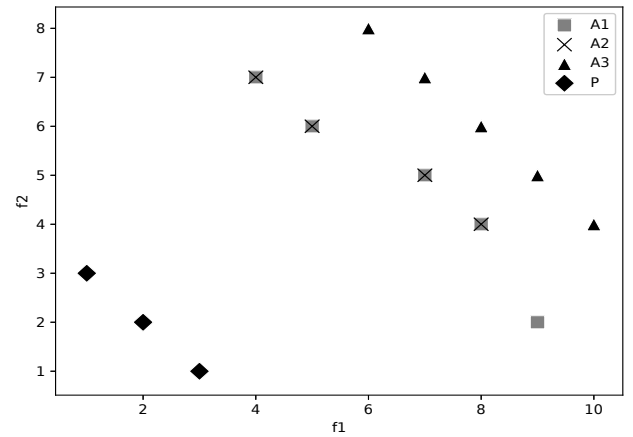


Figure 2. Experiment using four solution sets proposed in [4] to show how  $\epsilon$ -indicators assess the sets characteristics. There are four solutions sets. P is the Pareto front and there are explicit relations among the other solutions sets:  $A_1 \succeq A_2, A_1 \succeq A_3, A_2 \succeq A_3$

It is expected that an indicator should reflect all the solution set features. In this sense, Table I presents the values for all combinations among  $A_1$ ,  $A_2$ ,  $A_3$ , and  $P$ . It can be observed that DoM and  $\epsilon$ -indicators have the same interpretation, and the comparisons lead to the same conclusions among the solution sets. Nonetheless, it is relevant to observe some differences:

- $\epsilon$ -indicators are only related to one particular solution and only one objective in whole solution set. There is an information loss, because the indicator ignores the difference in other objectives. It can be viewed, for example, in comparison with  $\epsilon$ -indicator( $A_1, P$ ) and DoM( $A_1, P$ ). Considering  $\epsilon$ -indicator( $A_1, P$ ), it was obtained using the first solution from  $A_1$  and  $P$  on  $f_2$  objective; otherwise, the DoM( $A_1, P$ ) has explored  $f_1$  and  $f_2$  objectives, the distance value was generated using the optimal problem resolution that was obtained from the first  $A_1$  solution  $a_1 = (4, 7)$ , generate a surrogate point  $a'_1 = (2, 2)$  that dominate  $p_2$  and have a dominance move distance of one for each  $p_1$  and  $p_3$ , summing the whole dominance move equals to nine;
- $\epsilon$ -additive is not able to capture differences concerning cardinality of solution sets (observe  $\epsilon$ -additive( $A_3, A_1$ ) and  $\epsilon$ -additive( $A_3, A_2$ )). At the same time,  $\epsilon$ -multiplicative presents the same proportion related as DoM;
- DoM presents greater values than  $\epsilon$ -indicators (observe DoM( $A_1, P$ ) versus  $\epsilon$ -indicator( $A_1, P$ ), or DoM( $A_3, P$ ) versus  $\epsilon$ -indicator( $A_3, P$ )). This fact can be explained since DoM takes into account information from all objectives.

The  $\epsilon$ -indicators also measure the minimum value added to one solution set to make it be weakly dominated by another set. However, as it can be observed in Table I, there is an information loss. This information loss is critical, considering many objectives scenarios. One simple example, proposed in [10], can be easily observed: consider two 10-objective solutions, such as  $p_1 = \{0, 0, 0, \dots, 1\}$  and  $q_1 = \{1, 1, 1, \dots, 0\}$ . In this case,  $\epsilon$ -additive( $p_1, q_1$ ) =  $\epsilon$ -additive( $q_1, p_1$ ) = 1.

The second experiment was done to show the correctness of DoM assignment calculation, and how it addresses the quality indicator facets: convergence, spread, uniformity, and cardinality [2]. The same guidelines proposed in [10] to solve DoM in the bi-objective case were applied. The data was provided by Dr Miqing Li. Our method presented the same results, which were found in the original work. This concordance showed that the proposed DoM assignment model was not only correct, but in agreement with the DoM concept and with the exact algorithm for the bi-objective case presented in [10].

### B. Multiobjective experiments

After using some artificial test sets, the next experiment aimed to (i) validate the DoM assignment model using problems with three objectives and (ii) assess the comparison results with other state-of-the-art quality indicators, such as HV and IGD. Visualization of approximation sets was also

Table I  
COMPARISONS AMONGST  $\epsilon$ -ADDITIVE,  $\epsilon$ -MULTIPLICATIVE, AND DoM INDICATORS. A VALUE OF  $\epsilon$ -ADDITIVE  $\leq 0$ ,  $\epsilon$ -MULTIPLICATIVE  $\leq 1$  OR DoM  $\leq 0$  IMPLIES THAT P WEAKLY DOMINATES Q. THE SOLUTIONS SETS ARE PRESENTED GRAPHICALLY IN FIGURE 2

Quality indicator	P solution sets	Q solution sets			
		$A_1$	$A_2$	$A_3$	$P$
$\epsilon$ -additive	$A_1$	0.000	0.000	-1.000	4.000
	$A_2$	2.000	0.000	0.000	4.000
	$A_3$	2.000	2.000	0.000	5.000
	$P$	-1.000	-3.000	-3.000	0.000
$\epsilon$ -multiplicative	$A_1$	1.000	1.000	0.900	4.000
	$A_2$	2.000	1.000	1.000	4.000
	$A_3$	2.000	1.500	1.000	6.000
	$P$	0.500	0.428	0.333	1.000
DoM	$A_1$	0.000	0.000	0.000	9.000
	$A_2$	2.000	0.000	0.000	9.000
	$A_3$	8.000	6.000	0.000	12.000
	$P$	0.000	0.000	0.000	0.000

applied to provide an important insight into the properties of the approximation sets while validating the conclusions.

In all tests, algorithms such as IBEA, NSGAI, and SPEA2 were used to generate the solution sets. It is important to note that any other algorithm could have been applied to generate the solution sets. Our main goal was to validate the effectiveness of the proposed DoM assignment formulation and not perform an algorithm ranking.

In each experiment, and for our purpose, an important parameter had to be chosen beforehand: the definition of the population size (i.e., others parameters were kept default in each software used). The question is closely related to the  $p'_k$  candidates and the solution set cardinality (one of the quality indicator facet). Generally speaking, in order to have a good approximation set of the Pareto front, in terms of convergence, spread, and uniformity, the number of non dominated solutions grows exponentially concerning the problem dimension.

Using the model in (6), the selection of the  $p'_k$  candidates was done using the full combinatorial approach: all possible combinations selecting one  $p_i$  and all the possible  $g$  groups in  $Q$ . The number of such candidates is detailed in (7), in which  $g$  is a group with one or many  $q_j$ , and assuming that  $p_i$  will be used as a base to be updated, generating  $p'_k$ , which can weakly dominate all  $g$  group.

$$NP \sum_{g=1}^{NQ} \binom{NQ}{g} = NP \left( \binom{NQ}{1} + \binom{NQ}{2} + \dots + \binom{NQ}{NQ} \right) = NP(2^{NQ} - 1) \quad (7)$$

Based on (6) and (7), the population size for the algorithms was set to 20. It is relevant to note that the number of objectives does not change the model parameters, since the matrices with  $d(p_i, p'_k)$  and  $d(p'_k, q_j)$  do not suffer structural impact (i.e., the number of row and columns remains the same, regardless of the number of objectives).

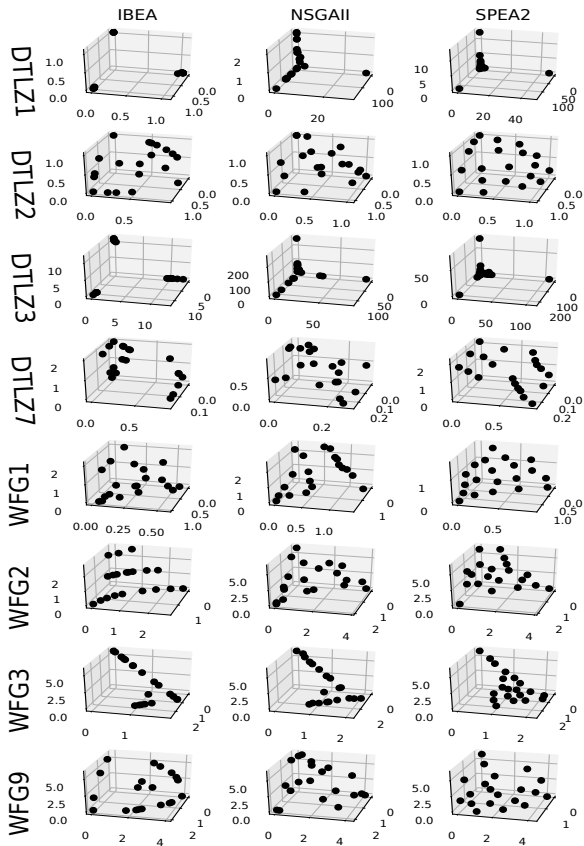


Figure 3. Solution sets with  $NP = NQ = 20$  solutions and three objectives,  $M = 3$ , generated by IBEA, NSGAII, and SPEA2 algorithms applied to DTLZ1, DTLZ2, DTLZ3, WFG1, WFG2, WFG3 and WFG9 problem sets.

Figure 3 presents all the solution sets. The *DTLZ* and *WFG* families were chosen. These problems present many different characteristics as convexity/concavity, disconnection, multimodality, and degeneracy. In this way, DTLZ1, DTLZ2, DTLZ3, and DTLZ7, WFG1, WFG2, WFG3, and WFG9 were selected. Some algorithms were also used to test and compare the DoM with other indicators: IBEA [16], SPEA2 [17], and NSGAII [9]. The algorithms were run 21 times using each problem set, and the final combined Pareto front is presented in 3. The number of function evaluations to each algorithm run was set to 10000 in each experiment.

Tables II and III show two unary quality indicators: the inverted generational distance (IGD) and hypervolume (HV) for the problem sets depicted in 3. It is mandatory to have a reference set and point, respectively, to calculate these indicators, and this task is not only a challenging one [2], but also sometimes provided by the user [8]. For HV, we decided to use a dominated point chosen amongst all algorithm solutions and objectives. Concerning IGD, we built a reference set using the following strategy: we took all the solutions from all algorithms, and extracted the non dominated points, thus creating the reference set.

The DoM quality measure is a binary indicator that takes as input, two approximation sets. In Table IV, for example, it

is possible to see all the comparisons among the solution sets generated by the algorithms for the *DTLZ* and *WFG* families.  $P$  was the solution set generated by the algorithm which it was trying to dominate, and  $Q$  was the solution set which was being dominated.

Table II  
IGD QUALITY INDICATOR FOR PROBLEM SETS: DTLZ1, DTLZ2, DTLZ3, DTLZ7, WFG1, WFG2, WFG3 AND WFG9 GENERATED USING IBEA, NSGAII AND SPEA.

Problem set	IGD		
	IBEA	NSGAII	SPEA2
DTLZ1	0.175	0.024	0.056
DTLZ2	0.091	0.091	0.078
DTLZ3	0.132	0.052	0.049
DTLZ7	0.114	0.253	0.082
WFG1	0.144	0.112	0.096
WFG2	0.133	0.090	0.086
WFG3	0.061	0.055	0.053
WFG9	0.099	0.178	0.103

For the DTLZ1 test set, detailed in Tables II and III, the algorithm which presented the best IGD was NSGAII. For the HV indicator, it was difficult to compare the algorithms due to the inflated values. There was a tie between IBEA and NSGAII; however, SPEA2 showed a better value. Using DoM approach and the comparison among algorithms, the sets generated by IBEA and SPEA2 were both indicated as the best solutions. The results are presented in Table IV. It is clear that DoM indicated IBEA as the best choice when compared with SPEA2,  $\text{DoM}(\text{IBEA}, \text{SPEA2}) = 0.769$  against  $\text{DoM}(\text{SPEA2}, \text{IBEA}) = 1.085$ . The solution sets from other algorithms easily dominated NSGAII. Taking a closer look at the DTLZ1 problem set presented in Figure 3, IBEA showed the smallest scale in all graph axis. DoM is sensible to all objectives, and the other algorithms had points near the IBEA solution set. However, the ‘effort’ to dominate the solution sets was smaller, favouring IBEA.

In the DTLZ2 case, IGD did not indicate differences between IBEA and NSGAII, and, in the end the best solution set was generated by SPEA2 (see Table II). Considering HV, the best algorithms were SPEA2 and IBEA (observe Table III). Looking at Table IV, the best values pointed to NSGAII and SPEA2. Observing NSGAII and SPEA2 in Figure 3, it is possible to note that there is a similar graph scale, but SPEA2 and NSGAII presented a better uniformity among the points in each solution set.

The results for DTLZ3 were presented in Tables II and III: for IGD, the best algorithm was SPEA2; and for HV, SPEA2 had the best value. However, it is relevant to note that these problem sets showed inflated solutions in the same way as DTLZ1. The DoM values among all algorithms (as shown in Table IV) favoured IBEA and SPEA2, but in a two by two comparison, SPEA2 had a better value when compared to IBEA,  $\text{DoM}(\text{SPEA2}, \text{IBEA}) = 7.645$ .

In the DTLZ7 problem set, the best HV values were given by IBEA and SPEA2. Considering IGD, the best one was for

SPEA2. Using Table IV, SPEA2 generated the best candidate solutions. Again, the values were smaller, when compared to DoM values from SPEA2 to dominate all other sets.

Table III  
HYPERVOLUME QUALITY INDICATOR FOR PROBLEM SETS: DTLZ1, DTLZ2, DTLZ3, DTLZ7, WFG1, WFG2, WFG3 AND WFG9 GENERATED USING IBEA, NSGAII AND SPEA.

Problem set	HV		
	IBEA	NSGAII	SPEA2
DTLZ1	1.048e+05	1.048e+05	1.049e+05
DTLZ2	0.352	0.319	0.356
DTLZ3	5.976e+06	5.969e+06	5.977e+06
DTLZ7	0.138	0.108	0.124
WFG1	3.832	3.255	3.541
WFG2	39.543	37.168	38.626
WFG3	17.952	16.757	16.492
WFG9	25.798	13.992	20.052

In the WFG family, results are presented in Tables II and III. The best algorithms were SPEA2 and NSGAII for IGD; however, for HV indicator, the best one was IBEA. In Table IV, the best algorithm was IBEA. Comparing IBEA to SPEA2, for example, IBEA had a lower value of DoM,  $\text{DoM}(\text{IBEA}, \text{SPEA2}) = 0.493$ , in contrast with  $\text{DoM}(\text{SPEA2}, \text{IBEA}) = 0.655$ . Observe that the values were close to each other.

In WFG2, the best values for IGD, SPEA2 and NSGAII, were close, presenting a little difference (see Table II). Considering HV, the best one was the IBEA (Table III), but the values were once again subtle. Using DoM, detailed in Table IV, there is an indication that IBEA was the best one when comparing the algorithms in a two-by-two manner. Something that should be noted is that the values were close to each other in DoM; the same phenomenon could be observed in IGD and HV, as well.

Using WFG3, for the IGD indicator, SPEA2 was the best one (subtle difference related to NSGAII), and IBEA was the best solution set considering HV. Assessing DoM in Table IV, there was an indication that IBEA also had better values.

Finally, for the WFG9 problem set, Tables II and III showed that for IGD, IBEA had lower value. For HV, IBEA algorithm had a better value. Considering DoM, presented in Table IV, IBEA was clearly the most competitive algorithm presenting the best values.

All the experiments were done using *Platypus* [18] and *PyGMO* [19] to generate the problem sets and to calculate the indicators (HV and IGD). The model in (6) was implemented using *Python* and *GUROBI* [20] version 9.0.0 build v9.0.0rc2 running on a Linux 64 bits operational system with 12 CPU's and 16Gb of RAM. The gap solver parameter was kept as *GUROBI* default value  $1e - 4$ .

The method proposed had two stages. The first one was to calculate matrices involving the distances  $d(p_i, p_k)$  and  $d(p_k, q_j)$ . The distance calculation matrices were implemented in  $O(n^2)$ , and, as discussed before, there is room for improvement in this implementation. The second step was to generate and solve model 6, which was implemented using

Table IV  
DOM VALUES FOR SOME MEMBERS OF THE PROBLEM SET FAMILIES DTLZ AND WFG. THE APPROXIMATION SETS WERE GENERATED BY IBEA, NSGAII, AND SPEA2 ALGORITHMS. IT MUST BE NOTED THAT  $P$  WAS THE SOLUTION SET GENERATED BY THE ALGORITHM THAT IT WAS TRYING TO DOMINATE, AND  $Q$  WAS THE SOLUTION SET GENERATED BY THE ALGORITHM BEING DOMINATED. EACH SOLUTION SET HAD  $NP = NQ = 20$  SOLUTIONS AND  $M = 3$  OBJECTIVES.

Problem set	DoM(P,Q)			
	P	Q		
		IBEA	NSGAII	SPEA2
DTLZ1	IBEA	0.000	0.121	0.769
	NSGAII	1.535	0.000	1.535
	SPEA2	1.085	0.176	0.000
DTLZ2	IBEA	0.000	0.929	0.966
	NSGAII	0.757	0.000	0.778
	SPEA2	0.726	0.908	0.000
DTLZ3	IBEA	0.000	0.044	8.137
	NSGAII	16.468	0.000	16.474
	SPEA2	7.645	2.121	0.000
DTLZ7	IBEA	0.000	0.417	0.721
	NSGAII	1.559	0.000	1.736
	SPEA2	0.667	0.420	0.000
WFG1	IBEA	0.000	0.839	0.493
	NSGAII	1.092	0.000	1.015
	SPEA2	0.655	1.506	0.000
WFG2	IBEA	0.000	1.393	1.401
	NSGAII	1.514	0.000	1.608
	SPEA2	1.556	1.731	0.000
WFG3	IBEA	0.000	1.602	1.440
	NSGAII	2.226	0.000	1.923
	SPEA2	2.732	1.954	0.000
WFG9	IBEA	0.000	1.491	1.442
	NSGAII	2.643	0.000	2.372
	SPEA2	3.210	2.394	0.000

Table V  
EXPERIMENT EXECUTION METRICS: SIMPLEX ITERATIONS (FROM BRANCH AND BOUND EXECUTION), AND THE TIME SPENT IN SECONDS TO SOLVE THE MODEL. DESCRIPTIVE STATISTICS ARE MINIMUM, MAXIMUM, AND QUANTILES.

Metric	Descriptive Statistics				
	Min	Q1	Q2	Q3	Max
simplex iterations	26212	43822	60457	75103	163869
time(seconds)	2.000	7.475	17.595	42.255	98.550

mixed-integer programming capabilities, such as branch and bound.

Descriptive statistics from the tests are presented in Table V. There are two metrics: simplex iterations from the branch and bound algorithm, and the time spent to solve the model. The median time spent by the model was  $\sim 17$  seconds, with 60457 simplex iterations. In some cases, the model was solved in two seconds; however, in the worst case, the model spent  $\sim 98$  seconds to solve (i.e., this case happened in the WFG9 problem set when NSGAII was trying to dominate IBEA).

In this section, the goal was to verify if the DoM assignment formulation was a feasible approach for dealing with problems that have three objective functions. It is worthy to note that the maximum number of points was established to 20 (more solutions in each set increase the computational complexity



and time). Additionally, it is relevant to observe that the assignment problem formulation is not affected by the problem set dimensionality. The distance matrices, which are parameters of the model, are not altered with the problem dimensionality. Moreover, the proposed method has two stages, and just the first one, distance matrices calculation, is affected by the number of objectives/dimensions, which remains viable, at least in some hundreds of objectives/dimensions.

## V. CONCLUSION

DoM is a binary indicator that considers the minimum move of one set to dominate the other set weakly. The indicator is Pareto compliant and does not demand any parameters or reference sets. Besides, it treats some weaknesses which come from the  $\epsilon$ -indicators but offers a similar interpretation. In this sense, it represents a natural and intuitive relation when comparing solutions, providing a valid measure to infer Pareto dominance relations, mainly in high dimensions. The great question about DoM is its calculation concerning its computational complexity.

We explored a new formulation to calculate DoM and dealt with it as an assignment problem. The idea used  $P$  and  $Q$ , for example, as solution sets that have to be the solutions assigned to each other. Comparisons with artificial bi-dimensional examples were made, detailing that DoM has the same interpretation as the  $\epsilon$ -indicators, and that our formulation presented the same results provided by the original DoM formulation. Additionally, some problem sets in three dimensions were also tested and showed that DoM assignment results obtained were in agreement and compliant when compared with other common indicators used in literature (IGD and HV).

DoM formulations as an assignment problem brought some particular constraints and questions, as it was discussed in model formulation. Two calculation stages were presented: i) the matrices distance calculation, which is smoothly affected by the number of objectives (e.g., for some thousands of dimensions), and ii) the model, as an assignment formulation, implemented using mixed-integer programming, which is affected by the number of the elements in each solution sets. To the best of our knowledge, even with these limitations, an exact method to calculate DoM in three or more dimensions is not known until now.

As a future research, the assignment formulation could be extended. One possible idea is to introduce the distance calculation inside the mixed-integer programming model. Possibly, it could deal with a greater number of solutions in each set. Yet, another possibility is to not use a full combinatorial approach. Otherwise, a machine learning approach could be applied to learn a function that describes features that good  $p$ ' solutions should have to dominate some  $q$ 's being generated by  $p$  set.

Finally, DoM is an indicator that is capable of expressing many quality indicators characteristics. An indicator with such a feature could improve not only the comparison among algorithms, but also it the strategies used by the algorithms which are indicator based, for example.

## REFERENCES

- [1] S. Chand and M. Wagner, "Evolutionary many-objective optimization: A quick-start guide," *Surveys in Operations Research and Management Science*, vol. 20, pp. 35–42, 12 2015.
- [2] M. Li and X. Yao, "Quality evaluation of solution sets in multiobjective optimisation: A survey," *ACM Comput. Surv.*, vol. 52, no. 2, pp. 26:1–26:38, Mar. 2019. [Online]. Available: <http://doi.acm.org/10.1145/3300148>
- [3] A. Ibrahim, S. Rahnamayan, M. V. Martin, and K. Deb, "3d-radviz antenna: Visualization and performance measure for many-objective optimization," *Swarm and Evolutionary Computation*, vol. 39, pp. 157–176, 2018. [Online]. Available: <https://doi.org/10.1016/j.swevo.2017.09.011>
- [4] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, April 2003.
- [5] M. T. M. Emmerich and A. H. Deutz, "A tutorial on multiobjective optimization: fundamentals and evolutionary methods," *Natural Computing*, vol. 17, no. 3, pp. 585–609, Sep 2018. [Online]. Available: <https://doi.org/10.1007/s11047-018-9685-y>
- [6] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Modified distance calculation in generational distance and inverted generational distance," in *Evolutionary Multi-Criterion Optimization*, A. Gaspar-Cunha, C. Henggeler Antunes, and C. C. Coello, Eds. Cham: Springer International Publishing, 2015, pp. 110–125.
- [7] J. Deng and Q. Zhang, "Approximating hypervolume and hypervolume contributions using polar coordinate," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 913–918, Oct 2019.
- [8] K. Yang, M. Emmerich, A. Deutz, and T. Bäck, "Efficient computation of expected hypervolume improvement using box decomposition algorithms," *Journal of Global Optimization*, vol. 75, no. 1, pp. 3–34, Sep 2019. [Online]. Available: <https://doi.org/10.1007/s10898-019-00798-7>
- [9] E. Bradford, A. Schweidtmann, and A. Lapkin, "Efficient multiobjective optimization employing gaussian processes, spectral sampling and a genetic algorithm," *Journal of Global Optimization*, vol. 71, 02 2018.
- [10] M. Li and X. Yao, "Dominance move: A measure of comparing solution sets in multiobjective optimization," *CoRR*, vol. abs/1702.00477, 2017. [Online]. Available: <http://arxiv.org/abs/1702.00477>
- [11] M. Jünger, T. M. Lieblich, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, Eds., *50 Years of Integer Programming 1958-2008 - From the Early Years to the State-of-the-Art*. Springer, 2010. [Online]. Available: <https://doi.org/10.1007/978-3-540-68279-0>
- [12] D. Pentic, "Assignment problems: A golden anniversary survey," *European Journal of Operational Research*, vol. 176, pp. 774–793, 01 2007.
- [13] Y. Yuan, Y. S. Ong, A. Gupta, and H. Xu, "Objective Reduction in Many-Objective Optimization: Evolutionary Multiobjective Approaches and Comprehensive Analysis," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 189–210, 2018.
- [14] R. Cheng, M. Li, Y. Tian, X. Xiang, X. Zhang, S. Yang, Y. Jin, and X. Yao, "Benchmark functions for the cec'2018 competition on many-objective optimization," Tech. Rep., 2018.
- [15] M. Al-Neama, N. Reda, and F. Ghaleb, "An improved distance matrix computation algorithm for multicore clusters," *BioMed research international*, vol. 2014, p. 406178, 06 2014.
- [16] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *in Proc. 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII)*. Springer, 2004, pp. 832–842.
- [17] E. Zitzler, M. Laumanns, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm," Tech. Rep., 2001.
- [18] D. Brockhoff and T. Tušar, "Benchmarking algorithms from the platypus framework on the biobjective bbob-biobj testbed," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, ser. GECCO '19. New York, NY, USA: ACM, 2019, pp. 1905–1911. [Online]. Available: <http://doi.acm.org/10.1145/3319619.3326896>
- [19] D. Izzo, "Pygmo and pykep: Open source tools for massively parallel optimization in astrodynamics (the case of interplanetary trajectory optimization)," 01 2012, pp. –.
- [20] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2020. [Online]. Available: <http://www.gurobi.com>