

A novel framework to construct quality indicators using the linear assignment problem

Diana Cristina Valencia-Rodríguez^{1*} and Carlos A. Coello Coello¹

^{1*}Departamento de Computación, CINVESTAV-IPN, Av. IPN 2508,
San Pedro Zacatenco, 07360, Mexico City, Mexico .

*Corresponding author(s). E-mail(s): diana.valencia@cinvestav.mx;
Contributing authors: carlos.coellocoello@cinvestav.mx;

Abstract

In Multi-objective optimization problems (MOPs), we aim to simultaneously find the maximum or minimum values of two or more (often conflicting) objective functions. MOPs are often found in real-world applications and, consequently, many methods have been proposed to solve them. Multi-Objective Evolutionary Algorithms (MOEAs) are popular techniques to solve MOPs due to their ease of use and their generality. Assessing the performance of MOEAs has become increasingly important due to the high number of MOEAs that have been developed in recent years. This work presents a novel framework for creating quality indicators using the linear assignment problem. In addition, we introduce two novel quality indicators generated using this framework and present examples to validate their performance. Furthermore, we present a novel algorithm called MOEA-kAP that can incorporate any indicator generated using our proposed framework as a density estimator. Our experimental results show that MOEA-kAP outperforms state-of-the-art algorithms.

Keywords: Quality indicator, Multi-objective optimization, Linear assignment problem, Evolutionary algorithm

1 Introduction

For the purposes of this paper, we are interested in solving multi-objective optimization problems (MOPs) where the objective functions are in conflict with each other (i.e., improving one objective causes the deterioration of at least another one). Because of the conflicting nature of their objectives, MOPs do not have a single solution, but a

set of trade-off solutions that we call the *Pareto optimal set*. The objective function values corresponding to the solutions contained in the Pareto optimal set form the so-called *Pareto Front*.

MOPs can be found in many areas, including civil engineering, finance, physics, or water resource management. Because of that, several techniques have been proposed to solve MOPs since the 1970s. In recent years, Multi-Objective Evolutionary Algorithms (MOEAs) have gained popularity in this area, because they present several advantages, from which the main ones are that they do not require any gradient information, they are relatively easy to use and have a wide applicability (Deb, 2001; Coello Coello et al, 2007).

Performance assessment of MOEAs has become increasingly important, mainly because of their popularity. Therefore, many performance measures (also called *quality indicators*) have been introduced over the years. Such performance indicators typically focus on one or more of the following aspects of the resulting set of solutions in objective function space (also known as approximation set) (Audet et al, 2021):

1. *Convergence*: closeness to the optimal solutions.
2. *Extent (or spread)*: coverage of a wide range of solutions
3. *Uniformity*: distribution of the solutions.

It is worth noting that the combination of extent and uniformity is referred to as diversity in evolutionary multi-objective optimization.

As mentioned before, a variety of performance indicators have been proposed in the literature over the years (Li and Yao, 2019). One of the most popular performance indicators is the hypervolume (Zitzler, 1999), which measures the size of the objective space that the solutions cover collectively. One advantage of the hypervolume indicator is that it has been mathematically proved that its maximization leads to Pareto optimal solutions (Fleischer, 2003). However, its main disadvantage is that its computational cost increases exponentially with the number of objectives (Beume et al, 2009), which makes it unsuitable for problems having a relatively low dimensionality (normally, it is unsuitable for problems having more than six objectives).

Another popular performance indicator is the Inverted Generational Distance (IGD) (Coello Coello and Reyes Sierra, 2004), which measures the average distance in objective space between each solution in a set of Pareto optimal solutions and its nearest solution in an approximation set. This indicator allows us to assess convergence, spread, and uniformity. Nevertheless, the main drawback of this indicator is that it requires the Pareto optimal set or at least a high-quality approximation of it.

In 2023, we proposed a novel quality indicator that tries to overcome the drawbacks of the existing performance indicators: the I_{LAP} (Valencia-Rodríguez and Coello Coello, 2023). This indicator employs the Linear Assignment Problem (LAP) to assess the convergence and diversity of an approximation set. The LAP consists of a set of agents and tasks, where assigning an agent to a task involves a cost. Therefore, its aim is to minimize the overall assignment cost.

In contrast, when assessing a multi-objective optimizer, we have a set of solutions and reference vectors representing different regions of the Pareto Front. Hence, the

minimum cost of assigning the solutions to the reference vectors is used as a performance indicator. Our experimental study showed that this new indicator correctly ranks approximation sets in problems with more than four objectives. Moreover, the I_{LAP} has shown a superior performance over other popular indicators.

We realized that modifying the cost function of the I_{LAP} , it is possible to produce a quality indicator with different features. Thus, we propose here the use of the LAP as a general framework for creating quality indicators, which we call I_{FLAP} . Additionally, we also introduce two quality indicators generated using the new framework: P_{LAP} and D_{LAP} . Furthermore, we have developed a novel MOEA, called MOEA-kAP, that uses any indicator created using the I_{FLAP} as a density estimator.

The remainder of this paper is organized as follows. Section 2 introduces the necessary background related to our proposal. Section 3 presents our proposed framework, its properties, and two novel quality indicators generated using this framework. Section 4 describes our proposed algorithm called MOEA-kAP. Finally, Section 5 presents our conclusions and some possible paths for future research.

2 Background

In multi-objective optimization, the aim is to solve problems of the type¹:

$$\text{minimize } \mathbf{f}(\mathbf{x}) := [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})] \quad (1)$$

subject to:

$$g_i(\mathbf{x}) \leq 0 \quad i = 1, 2, \dots, m \quad (2)$$

$$h_i(\mathbf{x}) = 0 \quad i = 1, 2, \dots, p \quad (3)$$

where $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ is the vector of decision variables, $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, k$ are the objective functions and $g_i, h_j : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, m$, $j = 1, \dots, p$ are the constraint functions of the problem.

A few additional definitions are required to introduce the notion of optimality used in multiobjective optimization:

Definition 1. Given two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^k$, we say that \mathbf{x} **dominates** \mathbf{y} (denoted by $\mathbf{x} \prec \mathbf{y}$), if $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$ for all $i = 1, \dots, k$, and $f_j(\mathbf{x}) < f_j(\mathbf{y})$ in at least one j .

Definition 2. We say that a vector of decision variables $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n$ is **nondominated** with respect to \mathcal{X} , if there does not exist another $\mathbf{x}' \in \mathcal{X}$ such that $\mathbf{x}' \prec \mathbf{x}$.

Definition 3. We say that a vector of decision variables $\mathbf{x}^* \in \mathcal{F} \subset \mathbb{R}^n$ (\mathcal{F} is the feasible region) is **Pareto-optimal** if it is nondominated with respect to \mathcal{F} .

Definition 4. The **Pareto Optimal Set** \mathcal{P}^* is defined by:

$$\mathcal{P}^* = \{\mathbf{x} \in \mathcal{F} | \mathbf{x} \text{ is Pareto-optimal}\}$$

¹Without loss of generality, we will assume only minimization problems.

Definition 5. The **Pareto Front** \mathcal{PF}^* is defined by:

$$\mathcal{PF}^* = \{\mathbf{f}(\mathbf{x}) \in \mathbb{R}^k | \mathbf{x} \in \mathcal{P}^*\}$$

Definition 6. A vector $\mathbf{x} \in \mathcal{X}$ is said to *weakly dominate* another vector $\mathbf{y} \in \mathcal{F}$ (denoted as $\mathbf{x} \preceq \mathbf{y}$), if $f_i(\mathbf{x}) \leq f_j(\mathbf{y})$ for all $i = 1, \dots, m$.

Therefore, our aim is to obtain the Pareto optimal set from the set \mathcal{F} of all the decision variable vectors that satisfy (2) and (3). Note however that in practice, not all the Pareto optimal set is normally desirable or achievable, and decision makers tend to prefer certain types of solutions or regions of the Pareto front (Branke and Deb, 2005).

The outcome of the multi-objective optimizers (such as MOEAs) is called an *approximation set* (Zitzler et al, 2003):

Definition 7. Let $A \subseteq \mathcal{Z}$ be a set of objective vectors ($\mathcal{Z} \subseteq \mathbb{R}$). A is called an *approximation set* if any element of A does not weakly dominate any other objective in A . The set of all approximation sets is denoted as Ω .

The approximation sets are usually evaluated through quality indicators that assess different aspects of the set (Zitzler et al, 2003):

Definition 8. An m -ary *quality indicator* I is a function $I : \Omega^m \rightarrow \mathbb{R}$, which assigns each vector (A_1, A_2, \dots, A_m) of m approximation sets a real value $I(A_1, \dots, A_m)$

2.1 Linear assignment problem

The Linear Assignment Problem (LAP) involves assigning n agents to n tasks, where each assignment incurs a cost. Therefore, the aim is to minimize the overall assignment cost. In simpler terms, we are given an $n \times n$ cost matrix $C = (c_{ij})$, and we need to assign each row to a different column to minimize the sum of the corresponding entries.

By introducing a binary matrix $X = (x_{ij})$ such that

$$x_{ij} = \begin{cases} 1 & \text{if row } i \text{ is assigned to column } j, \\ 0 & \text{otherwise,} \end{cases}$$

The LAP can be modeled as follows (Burkard et al, 2012):

$$\min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \tag{4}$$

$$s.t. \quad \sum_{j=1}^n x_{ij} = 1 \quad (i = 1, 2, \dots, n), \tag{5}$$

$$\sum_{i=1}^n x_{ij} = 1 \quad (j = 1, 2, \dots, n), \tag{6}$$

$$x_{ij} \in \{0, 1\} \quad (i, j = 1, 2, \dots, n). \quad (7)$$

The LAP can be solved using the so-called Kuhn-Munkres algorithm, which Harold Kuhn introduced in 1955 (Kuhn, 1955) and was enhanced by J. Munkres in 1957 (Munkres, 1957). The most efficient implementation of this algorithm is $O(n^3)$.

3 Our proposed framework

In this work, we propose using the LAP as a framework to build quality indicators in the following way. Let W be a set of reference vectors representing different regions of the Pareto Front, and let A be an approximation set. Moreover, let c be a function that scores the relation between a reference vector and a solution. Then, we propose to use the minimum cost of assigning the vectors in W to the solutions in A as a performance indicator. Hence, we can estimate to what extent the approximation set covers the different regions of the Pareto front. Formally, this framework, called I_{FLAP} , is defined as follows:

Definition 3.1 (I_{FLAP}). *Given a set of reference vectors $W = \{\mathbf{w}_1, \dots, \mathbf{w}_n\}$, an approximation set $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$, and a cost matrix C such that $C_{ij} = c(\mathbf{w}_i, \mathbf{a}_j)$ where $c : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}^+$. Then, the I_{FLAP} is defined as:*

$$I_{FLAP} = \frac{1}{n} \min_{x \in \mathcal{X}} \left\{ \sum_{i=1}^n \sum_{j=1}^n C_{ij} x_{ij} \right\} \quad (8)$$

where \mathcal{X} is the set of permutation matrices.

To calculate the I_{FLAP} , we evaluate each solution and reference vector on function c to construct matrix C . Next, we solve the LAP by applying the Kuhn-Munkres algorithm to C . Finally, we use the minimum assignment cost as the indicator value. The matrix computation is performed in $O(mn^2)$ where m is the number of objectives, and the LAP problem is solved in $O(n^3)$. Therefore, the computational complexity of computing the I_{FLAP} is $O(mn^2 + n^3)$.

3.1 Framework properties

We can identify three crucial properties of the I_{FLAP} , which are described next.

Property 1. *Given a reference set W , the minimum possible overall assignment cost is obtained by summing up the minimum values from the cost functions defined by each reference vector in W . It is assumed that these solutions are different.*

Proof. Let $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ be an approximation set and $W = \{\mathbf{w}_1, \dots, \mathbf{w}_n\}$ be a reference set. Without loss of generality, suppose that $f(\mathbf{a}_i, \mathbf{w}_i)$ is the minimum value of function f with respect to the reference vector \mathbf{w}_i for all $i = 1, \dots, n$. Therefore, we want to prove that the minimum possible overall assignment cost is determined by:

$$\sum_{i=1}^n f(\mathbf{a}_i, \mathbf{w}_i).$$

For the sake of contradiction, let us assume that $\sum_{i=1}^n f(\mathbf{a}_i, \mathbf{w}_i)$ is not the minimum possible overall assignment cost. Hence, there exists another approximation set $B = \{\mathbf{b}_1, \dots, \mathbf{b}_n\}$ such that

$$\sum_{i=1}^n f(\mathbf{a}_i, \mathbf{w}_i) > \sum_{i=1}^n f(\mathbf{b}_i, \mathbf{w}_i)$$

Therefore, there is at least one $j \in \{1, \dots, n\}$ where the following holds:

$$f(\mathbf{a}_j, \mathbf{w}_j) > f(\mathbf{b}_j, \mathbf{w}_j).$$

However, this is a contradiction since $f(\mathbf{a}_j, \mathbf{w}_j)$ is the minimum value of function f given vector \mathbf{w}_j . \square

Property 2. *The I_{FLAP} considers all solutions of the approximation set to evaluate it.*

The I_{FLAP} assigns each element of the approximation set to each reference vector while minimizing the sum of the assignment costs. Hence, the contribution of each solution is considered to obtain the indicator's value. This property allows us to differentiate the performance between approximation sets fully.

Property 3. *Using the I_{FLAP} , we can create several quality indicators by changing the scoring function.*

Changing the function that scores the relation between a reference vector and a solution allows us to measure different characteristics of the Pareto Front. The upcoming sections will present two quality indicators produced using the I_{FLAP} , which have different properties.

3.2 Designing a convergence-diversity indicator

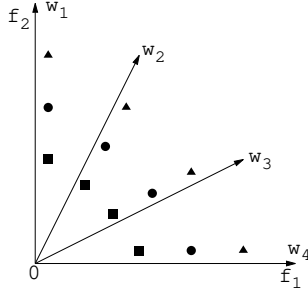
As mentioned in the previous sections, we must select a scoring function to create a performance indicator using the I_{FLAP} . In this section, we propose using the Achievement Scalarizing Function (ASF) (Miettinen, 1999), which is defined as follows:

$$u^{ASF}(\mathbf{w}, \mathbf{f}'(\mathbf{x})) := \max_{i \in \{1, \dots, m\}} \left\{ \frac{f'_i}{w_i} \right\} \quad (9)$$

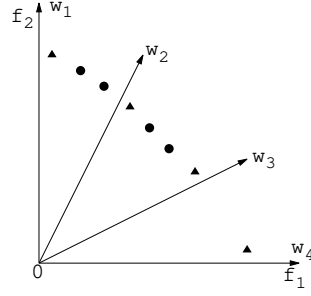
where $\mathbf{w} \in \mathbb{R}^m$, $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^m$ is the objective vector, $\mathbf{f}'(\mathbf{x}) = \mathbf{f}(\mathbf{x}) - \mathbf{z}^*$, and $\mathbf{z}^* \in \mathbb{R}^m$ is a reference point. The ASF transforms a multi-objective problem into a single-optimization problem by using a reference vector. Moreover its minimization leads to a weakly Pareto optimal solution (Miettinen, 1999). A detailed explanation of this can be found in (Tutun and Deb, 2015).

In addition, we propose employing Uniform Design with the Hammersley method (UDH) (Molinet Berenguer and Coello Coello, 2015) to create the reference set. The formal definition of the new indicator (which we called P_{LAP}) is the following:

Definition 3.2 (P_{LAP}). *Given a set of reference vectors $W = \{\mathbf{w}_1, \dots, \mathbf{w}_n\}$, an approximation set $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$, and a cost matrix C such that $C_{ij} =$*



(a) **Measuring convergence.**
 $P_{LAP}(\text{triangles}) = 25000.4375$,
 $P_{LAP}(\text{circles}) = 25000.3375$,
and $P_{LAP}(\text{squares}) = 25000.2375$.



(b) **Measuring diversity.**
 $P_{LAP}(\text{triangles}) = 25000.4375$, and
 $P_{LAP}(\text{circles}) = 150000.5375$.

Fig. 1: Examples where the P_{LAP} assesses convergence and diversity of approximation sets of a problem with two objectives. A lower value of the indicator is preferred.

$u^{ASF}(\mathbf{w}_i, \mathbf{a}'_j)$ where $a'_{j,k} = \frac{a_{j,k} - z_k^{min}}{z_k^{max} - z_k^{min}}$, where $\mathbf{z}^{min}, \mathbf{z}^{max} \in \mathbb{R}^m$ are the minimum and maximum possible values for each objective, and $\mathbf{z}^* = \mathbf{0}$. Then, the P_{LAP} is defined as:

$$P_{LAP} = \frac{1}{n} \min_{x \in \mathcal{X}} \left\{ \sum_{i=1}^n \sum_{j=1}^n C_{ij} x_{ij} \right\} \quad (10)$$

where \mathcal{X} is the set of permutation matrices.

The P_{LAP} measures convergence because it always considers the best values of the ASF. Moreover, we can infer from Property 1 that the minimum possible value of P_{LAP} is the sum of the minimum values of the ASF associated with each reference vector. Consequently, the approximation set that achieves the minimum value of P_{LAP} comprises at least weakly Pareto optimal solutions. Figure 1a shows an example where P_{LAP} correctly ranks three approximation sets, where a lower value implies a better approximation.

On the other hand, the P_{LAP} measures the diversity of a set since it quantifies how much each solution covers the Pareto Front regions given by the reference vectors. Figure 1b shows an example where P_{LAP} assesses diversity of two approximation sets.

3.3 Designing a diversity indicator

In this section, we introduce a new performance indicator called D_{LAP} that assesses the diversity of an approximation set. To accomplish this, we propose using the minimum Euclidean distance between a point and a reference vector. Given a solution $\mathbf{a} = [a_1, \dots, a_m]$ and a reference vector $\mathbf{w} = [w_1, \dots, w_m]$, this distance can be computed

using the following expression:

$$d(\mathbf{w}, \mathbf{a}) = \left\| \mathbf{a} - \frac{\mathbf{a} \cdot \mathbf{w}}{\mathbf{w} \cdot \mathbf{w}} \mathbf{w} \right\| \quad (11)$$

where $\|\cdot\|$ is the l^2 -norm.

The minimum Euclidean distance allows us to measure how much a region of the Pareto Front is covered. Therefore, we also propose using the UDH method to generate uniformly distributed reference vectors in objective space. The formal definition of the D_{LAP} is the following:

Definition 3.3 (D_{LAP}). *Given a set of reference vectors $W = \{\mathbf{w}_1, \dots, \mathbf{w}_n\}$, an approximation set $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$, and a cost matrix C such that $C_{ij} = d(\mathbf{w}_i, \mathbf{a}'_j)$ where $\mathbf{a}'_{j,k} = \frac{a_{j,k} - z_k^{min}}{z_k^{max} - z_k^{min}}$, where $\mathbf{z}^{min}, \mathbf{z}^{max} \in \mathbb{R}^m$ are the minimum and maximum possible values for each objective. Then, the D_{LAP} is defined as:*

$$D_{LAP} = \frac{1}{n} \min_{x \in \mathcal{X}} \left\{ \sum_{i=1}^n \sum_{j=1}^n C_{ij} x_{ij} \right\} \quad (12)$$

where \mathcal{X} is the set of permutation matrices.

We conducted three experiments to validate the D_{LAP} . The first experiment involved generating three approximation sets on the hyperplane $f_1 + f_2 + f_3 = 1$. The first set concentrates the solutions in a corner of the hyperplane (see Figure 2a), the second set randomly places the solutions on the hyperplane (see Figure 2b), and the third set distributes the solutions uniformly (see Figure 2c). We observed that as the indicator value decreases, the distribution of the sets improves.

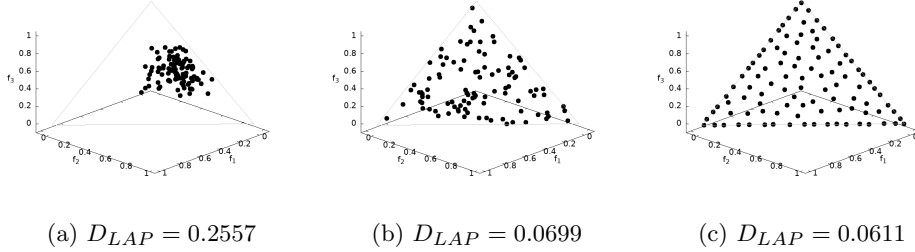


Fig. 2: Artificially generated Pareto Fronts with different distributions

In the second experiment, we took the approximation set of Figure 2c and scaled it by 75% and 50% (see Figure 3). This experiment aimed to test the indicator's ability to measure a set's coverage. The experimental results indicate that a lower indicator value corresponds to a better coverage of the set.

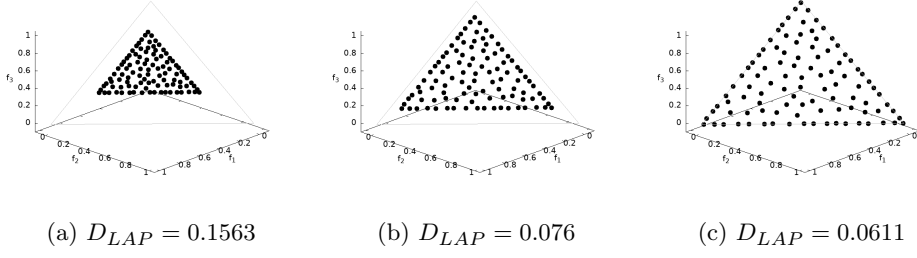


Fig. 3: Artificially generated Pareto Fronts with different coverage

In the third experiment, we inverted the approximation sets in Figure 2. This experiment aims to evaluate if this indicator is sensitive to the rotation of Pareto Fronts. Figure 4 shows that it still holds that the lower the value of D_{LAP} , the better the distribution.

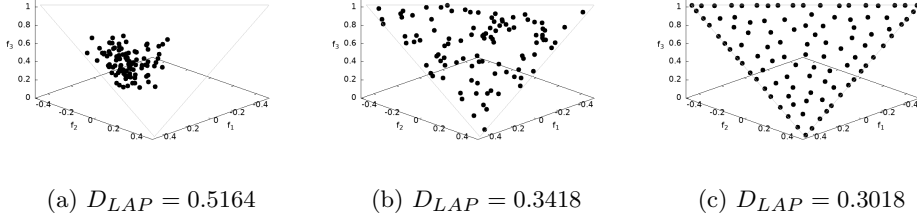


Fig. 4: Artificially generated Pareto fronts with different distributions located on an inverted unit simplex.

4 Our proposed algorithm

This section presents a novel algorithm in which you can incorporate any indicator generated using the I_{FLAP} as a density estimator. We called this algorithm a Multi-Objective Evolutionary Algorithm based on the k -cardinality Assignment Problem (MOEA-kAP). Algorithm 1 shows the operation of MOEA-kAP. First, a population is randomly generated and evaluated. Then, in the main loop, new individuals are generated using $DE/rand/1/bin$ (Storn and Price, 1997), and the resulting solutions are evaluated. After that, the best individuals are selected for the next generation using the kAP selection. This process continues until it reaches a maximum number of generations.

The kAP selection is the core process of MOEA-kAP and is displayed in Algorithm 2. This selection scheme first applies non-dominated sorting over a populations

Algorithm 1 Multi-Objective Evolutionary Algorithm based on the k -cardinality Assignment Problem (MOEA-kAP)

Require: Maximum number of generations (g_{max}), Differential evolution parameters (C_r, D), Reference vectors (W), MOP

- 1: P_1 = Generate initial population randomly
- 2: Evaluate each individual in P_1
- 3: **for** $t = 1, g_{max}$ **do**
- 4: Q_t = Generate offspring using P_t and $DE/rand/1/bin$
- 5: Evaluate each individual in Q_t
- 6: P_{t+1} = kAP_selection(P_t, Q_t, W)
- 7: **end for**

that combines parents and offspring (see lines 1-2). Then, the next generation's population (P_{t+1}) is created by selecting the best non-dominated sets until the sum of P_{t+1} and the current non-dominated set exceeds N (see lines 3-7).

We must take the top $k = N - |P_{t+1}|$ solutions from the last non-dominated set to have a population of N individuals. However, it is essential to note that the previous non-dominated sets have already covered certain regions defined by the weight vectors. Thus, it is preferable to pick solutions in areas that have not yet been covered. For this purpose, we identify which weight vectors are covered. We assume that a weight vector is covered when at least one solution in the population obtains its minimum value using the weight vector. Algorithm 3 summarizes the process of finding the non-covered weight vectors.

On the other hand, it is crucial to note that the number of non-covered weight vectors may be larger than the number of solutions to pick. Hence, we cannot consider it a LAP since both numbers must be identical. Instead, this is known as a k -cardinality Assignment Problem (k -AP) (Dell'Amico and Martello, 1997). In this type of problem, there are n agents, m tasks, and a positive integer k . The objective is to assign k agents to k tasks to minimize the sum of the assignment costs.

Given an $m \times n$ cost matrix $C = (c_{ij})$, and a positive integer $k \leq \min(m, n)$. The k -AP can be modeled as follows:

$$\min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (13)$$

$$s.t. \sum_{j=1}^n x_{ij} \leq 1 \quad (1 \leq i \leq m), \quad (14)$$

$$\sum_{i=1}^m x_{ij} \leq 1 \quad (1 \leq j \leq n), \quad (15)$$

$$\sum_{i=1}^m \sum_{j=1}^n x_{ij} = k \quad (1 \leq i \leq m, 1 \leq j \leq n) \quad (16)$$

$$x_{ij} \in \{0, 1\} \quad (1 \leq i \leq m, 1 \leq j \leq n). \quad (17)$$

Guo-Zhong Bai proposed transforming the k -AP into a LAP by modifying the cost matrix and then solving the problem using the Kuhn-Munkres algorithm (Bai, 2009). The modified cost matrix (\tilde{C}) is created by adding $n - k$ rows and $m - k$ columns to the original cost matrix (C). Moreover, each element of matrix \tilde{C} is defined as follows:

$$\tilde{C}_{ij} = \begin{cases} C_{i,j} & \text{if } (1 \leq i \leq m) \text{ and } (1 \leq j \leq n) \\ M & \text{if } (m < i \leq m + n - k) \text{ and } (n < j \leq m + n - k) \\ 0 & \text{if } (1 \leq i \leq m) \text{ and } (n < j \leq m + n - k) \\ 0 & \text{if } (1 \leq j \leq n) \text{ and } (m < i \leq m + n - k) \end{cases} \quad (18)$$

where M is an arbitrarily large positive number and $i, j = 1, \dots, m + n - k$. Algorithm 4 illustrates the creation of the modified cost matrix.

In summary, in order to select the best k individuals, we first normalized the P_{t+1} population and the last non-dominated set (see lines 8-10 of Algorithm 2). Then, we obtain the non-covered vectors (see line 11), construct the new modified matrix (see lines 12-13) using the resulting vectors, and solve the linear assignment problem. Finally, the previous non-dominated sets and the assigned solutions are selected for the next generation.

Algorithm 2 kAP_selection

Require: Parents population (P_t), Offspring population (Q_t), Reference vectors (W)

Ensure: Next population (P_{t+1})

```

1:  $R_t = P_t \cup Q_t$ 
2:  $\mathcal{F} =$  Perform fast non-dominated sorting using  $R_t$ 
3:  $P_{t+1} = \emptyset$  and  $i = 1$ 
4: while  $|P_{t+1}| + |\mathcal{F}_i| \leq N$  do
5:    $P_{t+1} = P_{t+1} \cup \mathcal{F}_i$ 
6:    $i = i + 1$ 
7: end while
8:  $\mathbf{z}_{max}, \mathbf{z}_{min} =$  Obtain the limits of  $P_{t+1} \cup \mathcal{F}_i$ 
9:  $\tilde{P}_{t+1} =$  Normalize  $P_{t+1}$  using  $\mathbf{z}_{max}$  and  $\mathbf{z}_{min}$ 
10:  $\tilde{\mathcal{F}}_i =$  Normalize  $\mathcal{F}_i$  using  $\mathbf{z}_{max}$  and  $\mathbf{z}_{min}$ 
11:  $\tilde{W} =$  obtain_non_covered_vectors( $\tilde{P}_{t+1}, W$ )
12:  $k = N - |P_{t+1}|$ 
13:  $\tilde{C} =$  construct_k_matrix( $\tilde{\mathcal{F}}_i, \tilde{W}, k$ )
14:  $I \leftarrow$  Obtain the best assignment in  $\tilde{C}$  using the Kuhn-Munkres algorithm
15: for all  $i \in I$  do
16:    $P_{t+1} = P_{t+1} \cup \{\mathcal{F}_{i,j}\}$ 
17: end for
18: return  $P_{t+1}$ 

```

Algorithm 3 obtain_non_covered_vectors

Require: Population (P), Reference vectors (W)

Ensure: List of non covered vectors (non_cover)

```
1:  $non\_cover$  = Initialize list of size  $N$  with ones
2: for all  $\mathbf{a} \in P$  do
3:    $idx = \arg \min_{i \in \{1, \dots, |W|\}} c(\mathbf{w}_i, \mathbf{a})$ 
4:    $non\_cover[idx] = 0$ 
5: end for
6: return  $non\_cover$ 
```

Algorithm 4 construct_k_matrix

Require: Normalized front population (\tilde{P}), Non covered vectors (\tilde{W}), Number of individuals to select (k)

Ensure: Cost matrix (C)

```
1:  $m = |\tilde{W}|$ 
2:  $n = |\tilde{P}|$ 
3:  $M$  = arbitrarily large positive number
4:  $C$  = Create matrix of size  $(n + m - k) \times (n + m - k)$ 
5: for  $i = 1, m$  do
6:   for  $j = 1, n$  do
7:      $C_{i,j} = c(\tilde{W}_i, \tilde{P}_j)$ 
8:   end for
9: end for
10: for  $i = m, n + m - k$  do
11:   for  $j = n, n + m - k$  do
12:      $C_{i,j} = M$ 
13:   end for
14: end for
15: return  $C$ 
```

4.1 Experimental analysis

We compared the performance of MOEA-kAP with respect to that of the NSGA-III (Deb and Jain, 2014) and with respect to a version of MOEA/D (Qingfu Zhang and Hui Li, 2007) that uses Differential Evolution. For this purpose, we adopted the DTLZ1-DTLZ7 test problems from the Deb-Thiele-Laumanns-Zitzler (DTLZ) (Deb et al, 2005) test suite using three objectives. The number of variables is defined by $n = 3 + k - 1$ where $k = 5$ for DTLZ1, $k = 10$ for DTLZ2-DTLZ6, and $k = 20$ for DTLZ7.

We adopted three indicators to assess performance: hypervolume (Zitzler, 1999), IGD^+ (Ishibuchi et al, 2015), and s-energy (Hardin and Saff, 2004). The hypervolume indicator measures the size of the objective space covered by the approximation set. The IGD^+ indicator measures the average distance between a reference set and

Table 1: Average and standard deviation of the IGD⁺ indicator over 30 executions. Darker cells imply better values. Moreover, the numbers below each value represent the identifiers of the algorithms with the worst performance according to the Wilcoxon rank sum test.

	MOEA/D-DE (0)	MOEA-kAP (1)	NSGA-III (2)
DTLZ1	6.120e+00 (1.10e+01)	2.3386e-02 (5.09e-02) 0,2	5.4093e-02 (8.45e-02) 0
DTLZ2	3.767e-02 (5.47e-04)	2.5659e-02 (5.61e-05) 0	2.2453e-02 (2.33e-06) 0,1
DTLZ3	2.386e+01 (3.29e+01)	4.8558e-01 (7.96e-01) 0,2	9.0021e-02 (1.81e-01) 0
DTLZ4	4.5382e-02 (1.15e-02)	3.1976e-02 (1.20e-02) 0	5.669e-02 (7.65e-02) 0,1
DTLZ5	5.8124e-03 (1.58e-04) 2	5.1957e-03 (3.13e-04) 0,2	6.974e-03 (8.22e-04)
DTLZ6	5.4551e-03 (3.19e-06) 2	2.9014e-03 (9.46e-06) 0,2	9.083e-03 (7.21e-04)
DTLZ7	9.858e-02 (2.78e-02)	4.8975e-02 (4.24e-03) 0	4.2030e-02 (2.30e-03) 0,1

a Pareto Front approximation. Furthermore, the s-energy indicator measures the uniform distribution of a set in a d -dimensional manifold. Thus, these indicators allow us to assess both diversity and convergence of the approximation sets generated by each algorithm.

Regarding the algorithms' parameters, we set $F = 1.0$ and $C_r = 0.4$ for Differential Evolution. Moreover, we set the crossover probability to $p_c = 1.0$, the mutation probability to $p_m = 1/n$, the crossover distribution index to $\eta_c = 20$, and the mutation distribution index to $\eta_m = 20$ for the crossover and mutation operators of NSGA-III. In addition, MOEA/D-DE used a neighborhood size $T = 20$, a neighborhood selection probability $\delta = 0.9$, and the PBI function with $\theta = 5$. We also used the D_{LAP} indicator as the density estimator of MOEA-kAP.

Additionally, we generated the reference points for NSGA-III and MOEA/DE using the Das and Dennis's systematic approach (Das and Dennis, 1998), which places the points on an $(k-1)$ -dimensional unit simplex. The number of reference points (H) generated by this method is given by $H = \binom{k+p-1}{p}$, where p is the number of divisions along each objective, and k is the number of objectives. In this experiment, we set $p = 12$ as suggested in (Deb and Jain, 2014), resulting in 91 reference points. However, we used a population size of 92 to meet NSGA-III's requirement for an even number of individuals. Moreover, we set the maximum number of generations to 1000.

Table 2: Average and standard deviation of the hypervolume indicator over 30 executions. Darker cells imply better values. Moreover, the numbers below each value represent the identifiers of the algorithms with the worst performance according to the Wilcoxon rank sum test.

	MOEA/D-DE (0)	MOEA-kAP (1)	NSGA-III (2)
DTLZ1	1.326e+00 (1.64e-02)	1.3310e+00 (2.48e-06)	1.3310e+00 (4.44e-16) 0,1
DTLZ2	7.003e-01 (2.07e-03)	7.2946e-01 (4.67e-04) 0	7.4663e-01 (5.58e-06) 0,1
DTLZ3	1.285e+00 (9.86e-02)	1.3310e+00 (2.47e-06) 0	1.3310e+00 (1.80e-07) 0,1
DTLZ4	7.197e-01 (3.33e-02)	7.5094e-01 (3.22e-02) 0	7.3752e-01 (1.03e-01) 0,1
DTLZ5	2.7113e-01 (1.25e-04) 2	2.7433e-01 (3.56e-04) 0,2	2.696e-01 (1.43e-03)
DTLZ6	2.5915e-01 (6.70e-06) 2	2.6408e-01 (1.45e-05) 0,2	2.535e-01 (1.41e-03)
DTLZ7	7.432e-01 (1.42e-02)	7.8776e-01 (3.70e-03) 0	7.9389e-01 (3.39e-03) 0,1

Tables 1, 2, and 3 show the average and standard deviation of the indicators' values of each algorithm. Moreover, the numbers below each value represent the statistically worst algorithms according to the Wilcoxon rank sum test. The identifier for MOEA/D-DE is 0, for MOEA-kAP is 1, and for NSGA-III is 2.

Regarding the IGD^+ indicator, the best algorithm is the MOEA-kAP because it ranked first in 4 out of 7 problems. The second-best algorithm is the NSGA-III, which ranked first in 3 problems. Hence, it is shown that the approximations obtained by the MOEA-kAP are usually near the Pareto Front in comparison with the other algorithms.

In terms of the hypervolume indicator, the NSGA-III algorithm performed the best as it ranked first in 4 out of 7 problems. Additionally, the MOEA-kAP algorithm was the second-best performer, as it ranked first in 3 problems. Therefore, MOEA-kAP is a competitive algorithm regarding the area of the approximation set that is able to cover.

On the other hand, if we examine the results concerning the s-energy indicator, we can observe that the MOEA-kAP outperforms the other algorithms in 6 out of 7 problems. Thus, it is shown that the MOEA-kAP outperforms the other algorithms in terms of the distribution of the approximation set.

In addition, Table 4 presents the average runtime over 30 executions of each algorithm. The MOEA/D-DE exhibits the best execution time, while the MOEA-kAP

Table 3: Average and standard deviation of the s-energy indicator over 30 executions. Darker cells imply better values. Moreover, the numbers below each value represent the identifiers of the algorithms with the worst performance according to the Wilcoxon rank sum test.

	MOEA/D-DE (0)	MOEA-kAP (1)	NSGA-III (2)
DTLZ1	1.291e+13 (2.39e+13)	5.2593e+09 (6.96e+08) 0	1.4442e+11 (7.52e+11) 0,1
DTLZ2	1.555e+13 (9.12e+12)	5.6136e+04 (1.80e+02) 0	4.6289e+04 (2.01e+00) 0,1
DTLZ3	2.700e+13 (3.67e+13)	1.2287e+09 (8.98e+08) 0	8.4395e+09 (3.88e+10) 0,1
DTLZ4	5.411e+13 (2.49e+13)	1.7817e+06 (6.08e+06) 0	1.7945e+11 (5.48e+11) 0,1
DTLZ5	2.694e+13 (2.41e+13)	1.0282e+08 (1.66e+08) 0,2	6.6723e+11 (1.11e+12) 0
DTLZ6	6.788e+12 (8.80e+12)	5.3466e+09 (1.69e+10) 0,2	1.8199e+12 (5.55e+12) 0
DTLZ7	2.067e+13 (2.01e+13)	5.6285e+07 (2.40e+08) 0,2	2.6027e+09 (1.01e+10) 0

Table 4: Average execution time in seconds for each algorithm.

	MOEA/D-DE	MOEA-kAP	NSGA-III
DTLZ1	15.864992	86.342058	19.805620
DTLZ2	16.565173	85.256798	19.199497
DTLZ3	16.438486	81.106055	20.314408
DTLZ4	16.742942	83.959776	19.587641
DTLZ5	16.798523	75.291789	20.219775
DTLZ6	16.380325	96.555691	19.893042
DTLZ7	17.254832	87.401130	20.675136

has the worst. However, the latter time is not prohibitively high and, as mentioned in Section 3, it increases linearly with respect to the number of objectives.

5 Conclusions and future work

In this work, we presented a novel framework for creating quality indicators using the linear assignment problem (I_{FLAP}). This framework allows us to build indicators with a polynomial computational cost that consider every element of the approximation set to assess its performance. Using this framework, we created two novel indicators: P_{LAP} and D_{LAP} . Our exploratory analysis suggests that the D_{LAP} indicator can

correctly determine the diversity of approximation sets, and that our proposed P_{LAP} indicator can determine both convergence and diversity. In addition, we introduced an algorithm that can incorporate any quality indicator generated using the I_{FLAP} as a density indicator: the MOEA-kAP. Our experimental results showed that our proposed MOEA-kAP outperforms state-of-the-art multi-objective evolutionary algorithms in problems with different characteristics.

As part of our future work, we would like to conduct a more in-depth exploration of the mathematical properties of the framework. Furthermore, we aim to test the P_{LAP} with several scalarizing functions to determine the most effective way to evaluate the quality of an approximation set. Regarding the D_{LAP} , we intend to substitute the Euclidean distance with another function, such as an angle, to measure the proximity between a reference vector and a point. Finally, we also plan to design new strategies that allow a reduction in the computational time required by our approach.

Statements and Declarations

The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

- Audet C, Bignon J, Cartier D, et al (2021) Performance indicators in multiobjective optimization. *European Journal of Operational Research* 292(2):397–422. <https://doi.org/10.1016/j.ejor.2020.11.016>
- Bai Gz (2009) A new algorithm for k-cardinality assignment problem. In: 2009 International Conference on Computational Intelligence and Software Engineering, pp 1–4, <https://doi.org/10.1109/CISE.2009.5363717>
- Beume N, Fonseca CM, Lopez-Ibanez M, et al (2009) On the Complexity of Computing the Hypervolume Indicator. *IEEE Transactions on Evolutionary Computation* 13(5):1075–1082
- Branke J, Deb K (2005) Integrating User Preferences into Evolutionary Multi-Objective Optimization. In: Jin Y (ed) *Knowledge Incorporation in Evolutionary Computation*. Springer, Berlin Heidelberg, p 461–477, ISBN 3-540-22902-7
- Burkard RE, Dell’Amico M, Martello S (2012) *Assignment Problems, Revised Reprint*. Other Titles in Applied Mathematics, Society for Industrial and Applied Mathematics
- Coello Coello CA, Reyes Sierra M (2004) A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In: Monroy R, Arroyo-Figueroa G, Sucar LE, et al (eds) *MICAI 2004: Advances in Artificial Intelligence*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 688–697

- Coello Coello CA, Lamont GB, Van Veldhuizen DA (2007) *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd edn. Springer, New York, ISBN 978-0-387-33254-3
- Das I, Dennis JE (1998) Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization* 8(3):631–657. <https://doi.org/10.1137/S1052623496307510>
- Deb K (2001) *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, ISBN 0-471-87339-X
- Deb K, Jain H (2014) An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation* 18(4):577–601
- Deb K, Thiele L, Laumanns M, et al (2005) Scalable Test Problems for Evolutionary Multiobjective Optimization. In: Abraham A, Jain L, Goldberg R (eds) *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*. Springer, USA, p 105–145
- Dell’Amico M, Martello S (1997) The k-cardinality assignment problem. *Discrete Applied Mathematics* 76(1):103–121. [https://doi.org/10.1016/S0166-218X\(97\)00120-0](https://doi.org/10.1016/S0166-218X(97)00120-0), second International Colloquium on Graphs and Optimization
- Fleischer M (2003) The Measure of Pareto Optima. Applications to Multi-objective Metaheuristics. In: Fonseca CM, Fleming PJ, Zitzler E, et al (eds) *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*. Springer. Lecture Notes in Computer Science. Volume 2632, Faro, Portugal, pp 519–533
- Hardin D, Saff E (2004) Discretizing Manifolds via Minimum Energy Points. *Notices of the American Mathematical Society* 51(10):1186–1194
- Ishibuchi H, Masuda H, Tanigaki Y, et al (2015) Modified Distance Calculation in Generational Distance and Inverted Generational Distance. In: Gaspar-Cunha A, Antunes CH, Coello CC (eds) *Evolutionary Multi-Criterion Optimization, 8th International Conference, EMO 2015*. Springer. Lecture Notes in Computer Science Vol. 9019, Guimarães, Portugal, p 110–125
- Kuhn HW (1955) The hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2(1–2):83–97. <https://doi.org/10.1002/nav.3800020109>
- Li M, Yao X (2019) Quality Evaluation of Solution Sets in Multiobjective Optimisation: A Survey. *ACM Computing Surveys* 52(2):1–38. Article Number: 26

- Miettinen K (1999) Nonlinear Multiobjective Optimization, International Series in Operations Research & Management Science, vol 12. Kluwer Academic Publishers
- Molinet Berenguer JA, Coello Coello CA (2015) Evolutionary Many-Objective Optimization Based on Kuhn-Munkres' Algorithm. In: Gaspar-Cunha A, Antunes CH, Coello CC (eds) Evolutionary Multi-Criterion Optimization, 8th International Conference, EMO 2015. Springer. Lecture Notes in Computer Science Vol. 9019, Guimarães, Portugal, p 3–17
- Munkres J (1957) Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics* 5(1):32–38. <https://doi.org/10.1137/0105003>
- Qingfu Zhang, Hui Li (2007) MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation* 11(6):712–731. <https://doi.org/10.1109/TEVC.2007.892759>
- Storn R, Price K (1997) Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11(4):341–359. <https://doi.org/10.1023/A:1008202821328>
- Tutum CC, Deb K (2015) A multimodal approach for evolutionary multi-objective optimization (memo): Proof-of-principle results. In: Gaspar-Cunha A, Henggeler Antunes C, Coello CC (eds) Evolutionary Multi-Criterion Optimization: 8th International Conference, EMO 2015. Springer International Publishing, pp 3–18
- Valencia-Rodríguez DC, Coello Coello CA (2023) A novel performance indicator based on the linear assignment problem. In: Emmerich M, Deutz A, Wang H, et al (eds) Evolutionary Multi-Criterion Optimization: 12th International Conference, EMO 2023. Springer Nature Switzerland, pp 348–360
- Zitzler E (1999) Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland
- Zitzler E, Thiele L, Laumanns M, et al (2003) Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation* 7(2):117–132