Swarm and Evolutionary Computation 76 (2023) 101211

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

An ACO-based Hyper-heuristic for Sequencing Many-objective Evolutionary Algorithms that Consider Different Ways to Incorporate the DM's Preferences

Gilberto Rivera [a,*], Laura Cruz-Reyes [b], Eduardo Fernandez [c], Claudia Gomez-Santillan [b], Nelson Rangel-Valdez [b], Carlos A. Coello Coello [d,e]

[a] *División Multidisciplinaria de Ciudad Universitaria, Universidad Autónoma de Ciudad Juárez, 32579, Cd. Juárez, Chihuahua, Mexico*
[b] *Postgraduate and Research Division, National Mexican Institute of Technology/Madero Institute of Technology, 89440 Ciudad Madero, Tamaulipas, Mexico*
[c] *Facultad de Contaduría y Administración, Universidad Autónoma de Coahuila, 27000, Torreón, Coahuila, México*
[d] *CINVESTAV-IPN, Evolutionary Computation Group, 07300, CDMX, Mexico*
[e] *Basque Center for Applied Mathematics (BCAM) & Ikerbasque, 48009 Bilbo, Bizkaia, Spain*

ARTICLE INFO

Keywords:
Ant Colony Optimization
Many-objective evolutionary algorithms
Preference incorporation
Outranking approach
Interval numbers

ABSTRACT

Many-objective optimization is an area of interest common to researchers, professionals, and practitioners because of its real-world implications. Preference incorporation into Multi-Objective Evolutionary Algorithms (MOEAs) is one of the current approaches to treat Many-Objective Optimization Problems (MaOPs). Some recent studies have focused on the advantages of embedding preference models based on interval outranking into MOEAs; several models have been proposed to achieve it. Since there are many factors influencing the choice of the best outranking model, there is no clear notion of which is the best model to incorporate the preferences of the decision maker into a particular problem. This paper proposes a hyper-heuristic algorithm—named HyperACO—that searches for the best combination of several interval outranking models embedded into MOEAs to solve MaOPs. HyperACO is able not only to select the most appropriate model but also to combine the already existing models to solve a specific MaOP correctly. The results obtained on the DTLZ and WFG test suites corroborate that HyperACO can hybridize MOEAs with a combined preference model that is suitable to the problem being solved. Performance comparisons with other state-of-the-art MOEAs and tests for statistical significance validate this conclusion.

1. Introduction

The complexity of real-world problems often requires the simultaneous optimization of several conflicting objective functions. Multi-objective Evolutionary Algorithms (MOEAs) have become powerful tools to solve such problems, mainly when only a few objective functions are involved. However, real-world problems frequently involve more than three objectives: the so-called Many-objective Optimization Problems (MaOPs). Most MOEAs have limitations in addressing MaOPs [1], being more severe for algorithms based on Pareto ranking [2] and less severe for decomposition-based algorithms like MOEA/D [3]. Unfortunately, swarm-based metaheuristics have not been explored so widely in this context [4]. Still, some studies give evidence of the great potential of variants of Ant Colony Optimization [5] and Particle Swarm Optimization [2] to solve MaOPs.

Most MOEAs focus on approximating the Pareto optimal set (abbreviated as *PF*), but identifying this frontier is insufficient. The Decision Maker (DM) needs to find the so-called Region of Interest (RoI)—the privileged zone in the Pareto frontier most in agreement with the DM's preferences—and, finally, identify the best compromise within that RoI. The best compromise is indeed the final solution to the problem. Hence, the problem cannot be solved without articulating the DM's preferences. Such information can be incorporated at different stages of the decision-making process: *a priori*, interactively, or *a posteriori*.

The *a posteriori* incorporation of preferences is widely used by traditional MOEAs; this way assumes that (*i*) the metaheuristic algorithm obtains an approximate Pareto front containing the RoI, and (*ii*) the DM is able to choose the best compromise on this portion of the

* Corresponding author.
E-mail address: gilberto.rivera@uacj.mx (G. Rivera).

https://doi.org/10.1016/j.swevo.2022.101211
Received 30 March 2022; Received in revised form 17 October 2022; Accepted 10 November 2022
Available online 19 November 2022

Pareto set. Contrarily, the *a priori* incorporation of preference requires that the DM defines the preference information before the search process; lastly, in the interactive way, the preference information is articulated progressively (e.g., [6]). Both *a priori* and interactive ways increase the selective pressure toward solutions closer to the RoI, reducing the search space [7].

To deal with preferences, there has been a growing interest in combining MOEAs and Multi-Criteria Decision-Making techniques (MCDM) [7]. In the scientific literature, many proposals articulate the preference information in the evolutionary search process to improve the selective pressure toward the RoI. Two comprehensive reviews on this topic were published by Li et al. [8] and Bechikh et al. [9], summarizing the preference information most commonly used by researchers. The *a priori* and progressive incorporations of preferences on appropriate frameworks could bring closer results to the RoI than the *a posteriori* way [8,10,11]. Nevertheless, the experimental results of Li et al. [8] showed that incorporating preferences into the evolutionary search does not always lead to a better approximation of the RoI than traditional MOEAs, especially when the number of objectives is small. According to Li et al. [8], incorporating preferences becomes more critical when the number of objective functions increases.

One of the approaches that Bechikh et al. [9] listed is the so-called *outranking parameters*, which refers to those methods where outranking relations model the credibility degree of the predicate "solution x is at least as good as solution y." This calculation is based on a non-compensatory approach using criterion weights and veto thresholds (cf. [12]). Notably, the outranking approach is advantageous when the DM is compatible with non-compensatory preferences and should handle non-transitive preferences, incomparability, and veto effects. The outranking approach was recently extended by Fernández et al. [13,14] to deal with imprecise model parameters and criterion scores by using interval numbers. In an *a priori* articulation of preferences, the DM does not know with sufficient precision the appropriate values of their preference model parameters; so, an approach that copes with such imprecise information is a significant advance.

The interval outranking approach [13] has been used as an *a priori* preference articulation in two recent studies [15,16]. Fernández et al. [15] incorporated preferences into MOEA/D's update phase, using six different binary preference relations derived from the credibility degree of the interval outranking. Rivera et al. [16] incorporated interval outranking-based preferences into an ACO algorithm; the outranking model was used for sorting the solutions in the pheromone trail to guide the search toward the RoI. The obtained results in both proposals are compared to state-of-the-art *a posteriori* many-objective metaheuristics.

These papers provide interesting results when DTLZ and WFG test problems are solved: although most of the time the *a priori* articulation of preferences brought better results than *a posteriori* metaheuristics, its effectiveness depends on the particular problem, the number of objective functions, the metric used to evaluate distances to the RoI, and the way to define the binary preference relation derived from the outranking credibility degree. As stated by Li et al. [8], the results confirm that the relative effectiveness of a preference articulation increases with the number of objectives. Nevertheless, these results [15,16] provide evidence against a universal best method of *a priori* incorporation of preferences—even within the outranking paradigm—since there was no single way that could be recommended for all problems and metrics. Then, in the presence of a new problem, the question about the "best" way remains unanswered.

Considering the difficulties above, the motivation of this paper is to propose a more general approach that combines a given set of *a priori* ways of incorporating outranking-based preference models. Such a general approach will be able to manage satisfactorily diverse performances of the different preference incorporation ways. Hyper-heuristic algorithms can integrate several heuristics or metaheuristics, allowing combining and sequencing preference incorporation methods.

The rest of this paper is structured as follows. Section 2 overviews the related literature. Section 3 presents the theoretical foundations needed to introduce our proposal. Section 4 introduces the algorithm used in HyperACO and describes each step in detail. Section 5 presents the numerical results that support the advantages of HyperACO. Lastly, Section 6 discusses pertinent conclusions and provides directions for future research.

2. An overview of the related literature

Hyper-heuristics were created to solve complex search problems of one or multiple objectives. A hyper-heuristic is a search method that includes learning mechanisms that operate on a fixed set of heuristics, monitoring and combining their strengths. Its main aim is to create general search methodologies that automate the design of heuristic methods [17]. The distinctive feature of hyper-heuristics is that they operate in a heuristic search space rather than directly in the search space of the underlying problem, as is the case with most metaheuristic approaches. The two main classes of hyper-heuristics are selection hyper-heuristics and generation hyper-heuristics [18].

Selection hyper-heuristics use a set of previously defined heuristics to solve a problem; the task is to discover a sequence of using those heuristics to improve the quality of solutions. To our knowledge, state-of-the-art hyper-heuristics with preference incorporation fall into this category. This class of hyper-heuristics comprises a High-Level Heuristic (HLH) that controls a set of Low-Level Heuristics (LLHs). The HLH searches the space of the sequences of the LLHs, instead of directly searching the space of solutions of the underlying problem.

Rivera et al. [19] presented a hyper-heuristic algorithm for solving the social project portfolio problem. The proposed method used a genetic algorithm as the HLH and basic operations (e.g., adding, removing, or exchanging projects) as the LLHs. The DM‘s preferences are incorporated *a priori* into the search process using the outranking model by Fernandez et al. [20].

Raghavjee and Pillay [21] proposed a hyper-heuristic to solve the school timetabling problem. The proposed algorithm used permutations of LLHs (exchange heuristics) and incorporated the teachers' preferences as soft constraints.

Muklason et al. [22] addressed the problem of generating examination timetables. They introduced a fairness objective function containing penalties from soft constraint violations; such a fairness function is minimized with the other objectives. The students' preferences were expressed as a difficulty index for each exam to define penalties. The HLH was a well-known hyper-heuristic framework referred to as HyFlex, and the LLHs were 13 common perturbation heuristics and one of movement.

Jakubovski-Filho et al. [23,24] proposed two hyper-heuristics based on reference points for software product line testing. These studies used a variant of NSGA-II as HLH and a set of 12 operators of crossover and mutation as LLHs. The DM's preferences were incorporated by assigning a reference point provided by the tester.

Macias-Escobar et al. [25] proposed a hyper-heuristic to solve dynamic multi-objective optimization problems, using a plane-separation method as HLH and dynamic versions of NSGA-II and GDE3 as LLHs. The preference articulation was based on reference points. The performance of that proposal was evaluated over a set of benchmark problems.

Although there are hyper-heuristics in the scientific literature that incorporate preferences, HyperACO (the hyper-heuristic proposed in this paper) offers MOEAs composed of sequences that indicate how to use different ways of preference incorporation. This feature is relevant because, as far as we know, there is no single model to incorporate preferences into MOEAs that always results in a better performance. Consequently, the advantage of HyperACO is that its resulting MOEAs are *ad hoc* designed to face the challenges that would arise in solving the specifically addressed MaOP. The quality of the results is statistically tested on the DTLZ and WFG test suites (two of the most widely accepted benchmarks for multi-objective optimization) and simultaneously

Table 1
Definition of elements used for the computation of $\sigma(x,y)$.

Element	Description
x, y	A solution x has an image $f(x) = \langle f_1(x), f_2(x), f_3(x), \ldots, f_m(x)\rangle$, where $f_k(x)$ is the value of the kth objective of x. Analogously, $f(y) = \langle f_1(y), f_2(y), f_3(y), \ldots, f_m(y)\rangle$.
Ω	$\Omega = \{\delta_k(x,y) : \delta_k(x,y) > 0\ \forall k \in \{1, 2, 3, \ldots, m\}\}$, where m is the number of objectives and $\delta_k(x,y)$ is the credibility degree of the statement "x outranks y with respect to objective k." This index is calculated using $\delta_k(x, y) = Poss(f_k(y) \geq f_k(x))$.
σ_γ	$\sigma_\gamma = \min\left\{\gamma,\ Poss(\boldsymbol{c}(x,y,\gamma) \geq \boldsymbol{\lambda}),\ 1 - \max_{k\in D(xS_\gamma y)}\{d_k(x,y)\}\right\}$.
$\boldsymbol{w}$, $\boldsymbol{v}$, $\boldsymbol{\lambda}$, $\boldsymbol{\beta}$	Preference model parameters. The DM's value system, denoted as $\boldsymbol{w}, \boldsymbol{v}, \boldsymbol{\lambda}, \boldsymbol{\beta}$, consists of the weight vector $\boldsymbol{w}$, the veto threshold vector $\boldsymbol{v}$, the interval number $\boldsymbol{\lambda}$ that reflects a majority threshold, and the overall credibility threshold $\boldsymbol{\beta}$. All these parameters are interval numbers. Note that $\sum_{k=1}^{m}\underline{w_k} \leq 1$ and $\sum_{k=1}^{m}\overline{w_k} \geq 1$ in a feasible preference system, as well as $\underline{\beta} > 0.5$ and $\underline{\lambda} > 0.5$.
$\boldsymbol{c}(x,y,\gamma)$	The concordance index according to γ, defined as $\boldsymbol{c}(x,y,\gamma) = \left[\underline{c(x,y)}, \overline{c(x,y)}\right]$, where: $\underline{c(x,y)} = \begin{cases} \sum_{k\in C(xS_\gamma y)} \underline{w_k} & \text{if } \sum_{k\in C(xS_\gamma y)} \underline{w_k} + \sum_{k\in D(xS_\gamma y)} \overline{w_k} \geq 1, \\ 1 - \sum_{k\in D(xS_\gamma y)} \overline{w_k} & \text{otherwise.} \end{cases}$ $\overline{c(x,y)} = \begin{cases} \sum_{k\in C(xS_\gamma y)} \overline{w_k} & \text{if } \sum_{k\in C(xS_\gamma y)} \overline{w_k} + \sum_{k\in D(xS_\gamma y)} \underline{w_k} \leq 1, \\ 1 - \sum_{k\in D(xS_\gamma y)} \underline{w_k} & \text{otherwise.} \end{cases}$
$C(xS_\gamma y)$	$C(xS_\gamma y) = \{k \in \{1, 2, 3, \ldots, m\} : \delta_k(x,y) \geq \gamma\}$ is the set of the objectives in the concordance coalition.
$D(xS_\gamma y)$	$D(xS_\gamma y) = \{1, 2, 3, \ldots, m\} \backslash C(xS_\gamma y)$ is the set of the objectives in the discordance coalition.
$d_k(x,y)$	It is the credibility degree of the assertion "the kth objective alone vetoes the assertion x *outranks* y"; $d_k(x,y) = Poss(f_k(x) \geq f_k(y) + \boldsymbol{v}_k)$, where $\boldsymbol{v}_k$ is the veto threshold associated with the kth objective.

Table 2
Variants of the scalarizing functions in MOEA/D/O based on outranking relations.

Variant	*Scalarizing function*	*Preference Relation* $xR_i y$	*Description*
V_1	$xR_1y \wedge g^{te}(x\|\lambda,z) \leq g^{te}(y\|\lambda,z)$	$R_1 : \boldsymbol{\sigma}(x,y) > \boldsymbol{\sigma}(y,x)$	Preference in favor of x, although its credibility may be below.
V_2	$xR_2y \wedge g^{te}(x\|\lambda,z) \leq g^{te}(y\|\lambda,z)$	$R_2 : \boldsymbol{\sigma}(x,y) \geq \boldsymbol{\beta}$	x is at least as good as y.
V_3	$xR_3y \wedge g^{te}(x\|\lambda,z) \leq g^{te}(y\|\lambda,z)$	$R_3 : \boldsymbol{\sigma}(x,y) \geq \boldsymbol{\beta} \wedge \boldsymbol{\sigma}(y,x) \leq \boldsymbol{\beta}$	Asymmetric preference in favor of x.
V_4	$xR_4y \wedge g^{te}(x\|\lambda,z) \leq g^{te}(y\|\lambda,z)$	$R_4 : \boldsymbol{\sigma}(x,y) > \boldsymbol{\sigma}(y,x) \wedge \boldsymbol{\sigma}(x,y) > 0.5$	Preference in favor of x.
V_5	$xR_5y \wedge g^{te}(x\|\lambda,z) \leq g^{te}(y\|\lambda,z)$	$R_5 : \boldsymbol{\sigma}(x,y) \geq \boldsymbol{\beta} \wedge \boldsymbol{\sigma}(y,x) < 0.5$	Strict preference in favor of x.
V_6	$xR_6y \wedge g^{te}(x\|\lambda,z) \leq g^{te}(y\|\lambda,z)$	$R_6 : \vee^{5}_{i=1} xR_i y$	Disjunction of the first five outranking relations.

compared with: (*i*) six different *a priori* ways for incorporating interval outranking models into evolutionary computation, and (*ii*) two state-of-the-art *a posteriori* MOEAs for many-objective optimization.

3. Background

This section overviews the following selected topics: interval mathematics (Subsection 3.1), the outranking approach extended by including interval numbers (Subsection 3.2), different ways to incorporate the interval outranking approach in MOEA/D (Subsection 3.3), and the definition of the RoI in the framework of interval outranking (Subsection 3.4).

3.1. Interval numbers

An interval number is a continuous subset of real numbers bounded by two specific given numbers, the lower and upper limits. Interval mathematics is commonly used to model the imprecision generated by inaccuracies and fluctuations in measurements, beliefs, and judgments [26]. In this paper, interval numbers are represented by letters in boldface and italic; e.g., $\boldsymbol{A} = [\underline{A}, \overline{A}]$, where the lower and upper limits are $\underline{A}$ and $\overline{A}$. All the basic arithmetic operations are also defined on interval numbers. Below, we only exemplify two relevant operations with the interval numbers $\boldsymbol{A}$ and $\boldsymbol{B}$. The following equation defines addition:

$$\boldsymbol{A} + \boldsymbol{B} = [\underline{A} + \underline{B}, \overline{A} + \overline{B}].$$

The possibility function, $Poss(\boldsymbol{A} \geq \boldsymbol{B})$, defines two relations on interval numbers ($\geq$ and $>$) [27–29] as follows:

$$Poss(\boldsymbol{A} \geq \boldsymbol{B}) = \begin{cases} 1 & \text{if } p_{AB} > 1, \\ 0 & \text{if if } p_{AB} < 0, \\ p_{AB} & \text{otherwise,} \end{cases} \tag{1}$$

where $p_{AB} = \frac{\overline{A} - \underline{B}}{(\overline{A} - \underline{A}) + (\overline{B} - \underline{B})}$.

For the case of degenerate intervals, when the intervals numbers are real numbers A and B, $Poss(\boldsymbol{A} \geq \boldsymbol{B}) = 1$ if and only if $A \geq B$; otherwise, $Poss(\boldsymbol{A} \geq \boldsymbol{B}) = 0$.

A realization of $\boldsymbol{A}$ is a real number a in the interval $[\underline{A}, \overline{A}]$ [30]. Fernández et al. [13] reinterpreted the possibility function as the degree of credibility of the statement "once both realizations, a and b, are given from $\boldsymbol{A}$ and $\boldsymbol{B}$, a will be greater than or equal to b." The relation $\boldsymbol{A} \geq \boldsymbol{B}$ is defined by $Poss(\boldsymbol{A} \geq \boldsymbol{B}) \geq 0.5$, and the relation $\boldsymbol{A} > \boldsymbol{B}$ is defined by $Poss(\boldsymbol{A} \geq \boldsymbol{B}) > 0.5$. These relations are transitive and can also be used to compare real numbers.

3.2. The interval outranking approach

Fernández et al. [13] proposed the Interval Outranking Approach (IOA), which can model imprecision and uncertainty in DM's judgments and beliefs. Their proposal simultaneously addresses multi-criteria non-compensatory preferences and poor information in model parameters and objective scores. Additionally, it helps approximate the Region of Interest (RoI) of a particular Multi-objective Optimization Problem (MOP). The IOA is an extension of the ELECTRE methods, and the rest of this section formalizes it.

Let x and y be a pair of actions of a decision set (for example, solutions of a MOP). Let us denote by $\sigma(x,y)$ the credibility index of the assertion "x is at least as good as y." The credibility index is calculated by Eq. (2), whose elements are defined in Table 1.

$$\sigma(x, y) = \max_{\gamma\in\Omega}\{\sigma_\gamma\} \tag{2}$$

The application of the interval outranking approach assumes the DMs can set the model parameter values as interval numbers. In this paper, the parameters are the following: objective weights, veto thresholds, majority thresholds, and credibility thresholds. This setting action does not imply requirements of "rational" behavior from the DM. More information about the IOA and its applications can be found in [13, 31]; additionally, Appendix A provides a numeric example of how to calculate $\sigma(x,y)$.

3.3. Some different ways to incorporate the IOA in MOEA/D

MOEA/D is an evolutionary algorithm proposed by Zhang and Li [3] that solves a MOP by decomposing it into various scalar optimization problems using weight vectors with uniform distribution. During the evolutionary process, if a solution x (offspring of y) is better than y, the parent is substituted by the offspring solution during the update phase. These solutions are compared with each other through a scalarizing

Algorithm 1
Run of sequences.

```
Algorithm name: run_sequence
Input: Sequence to be run (x_i), number of generations for each component of the sequence (g), number of runs (r)
Output: Solution sets resulting by running r times x_i (O^i)
O^i_j ← ∅  ∀j ∈ {1, 2, 3, …, r}
For each j ∈ {1, 2, 3, …, r}
    For each l ∈ {1, 2, 3, …, ħ}
        ▷ Run g generations of the variant x_{i,l} of MOEA/D/O using O^i_j as initial population
        O^i_j ← MOEA_D_O_{x_{i,l}}(O^i_j, g)
Return O^i
```

function. The weighted Chebychev distance is a well-known scalarizing function with a reference point $z = \langle z_1, z_2, z_3, \ldots, z_m \rangle$ (that is the set of the best objective values) and the weight vector $\lambda = \langle \lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_m \rangle$ (associated with the current solution y). The update of y is applied to those new solutions x that satisfy the Chebyshev condition in Eq. (3).

$$g^{te}(x|\lambda, z) \le g^{te}(y|\lambda, z), \text{ where } g^{te}(x|\lambda, z) = \max_{k \in \{1,2,3,\ldots,m\}} \{\lambda_k |f_k(x) - z_k|\} \quad (3)$$

Fernández et al. [15] presented an improved MOEA/D called MOEA/D/O. That paper proposes six different ways of incorporating preferences in the update phase. These six ways are based on different preference relations derived from the IOA; these ways and the associated binary outranking relations are described in Table 2. Each way implies a particular preference relation R_ι between x and y (Column 3). In the MOEA/D/O update phase, the Chebychev distance (Eq. (3)) is combined with an outranking preference relation in Column 3. Thus, the solution update is performed if it satisfies the Chebyshev distance and the related preference condition. Each of those ways constitutes a variant explored in that paper; there is even a variant that incorporates the disjunction of all preference relations and the Chebyshev condition.

In the remainder of this paper, $xR_\iota y$ denotes the preference relation R_ι between solutions x and y under the DM's value system, $DM = (\boldsymbol{w}, \boldsymbol{v}, \boldsymbol{\lambda}, \boldsymbol{\beta})$. As a consequence of veto effects and the Condorcet's Paradox[1], these relations are not transitive. That is, $xR_\iota y$ and $yR_\iota z$ do not necessarily imply $xR_\iota z$. Incomparability $(\neg xR_\iota y \wedge \neg yR_\iota x)$ is also possible. Except for R_2, the remaining preference relations are asymmetric.

Fernández et al. [15] measured the closeness to the RoI using Euclidean and Chebyshev distances. This study revealed that, in many instances, MOEA/D with preference incorporation achieves results closer to the RoI than the original MOEA/D; this effect is more relevant when the number of objectives increases. In general, the six ways to incorporate the interval outranking-based preferences outperformed MOEA/D regarding the Euclidean distance. Contrastingly, MOEA/D tended to perform better when the Chebychev distance was taken. However, no variant was always better than the others. It should be underlined that the results of the comparison of MOEA/D with MOEA/D/O depended on the preference relation used by MOEA/D/O, the type of optimization problem, the number of objective functions, the performance indicator considered as the most relevant by the DM, and even the specific DM's preferences. Such relativeness arises as a serious obstacle when the DM faces a new optimization problem whose characteristics are either unknown or different from those studied by Fernández et al. [15]. In such a case, an algorithm that automatically identifies an appropriate combination of the variants listed in Table 2 would be very promising for treating real-world MaOPs.

3.4. Identifying the Region of Interest

The ROI is closely related to the concept of the best compromise solution of a MOP: the most satisfactory solution for the DM. If the DM's preferences can be represented by a value function U, the best compromise would correspond to the global maximum of function U. In the framework of outranking methods, a formal definition of the best compromise solution was proposed by Fernandez et al. [20] and enhanced by Balderas et al. [27]. According to these papers, $x^* \in PF$ is the best compromise solution only if it fulfills two conditions:

(a) There is no $y \in PF$ such that y is preferred to x^* by the DM.
(b) There are good arguments to support the assertion "x^* is as at least as good as the other solutions that satisfy (a)."

Let us consider the strict preference relation P as the relation R_5 in Table 2, that is:

$$xPy \Leftrightarrow \boldsymbol{\sigma}(x, y) \ge \boldsymbol{\beta} \ \wedge \ \boldsymbol{\sigma}(y, x) < 0.5. \quad (4)$$

Two concepts should be introduced to model the fulfillment of the above conditions. The weakness of x in a set of alternatives $\mathscr{O}$, which is calculated as:

$$W(x, \mathscr{O}) = \{y \in \mathscr{O} \ : yPx\}. \quad (5)$$

The strength of x in $\mathscr{O}$ is defined as

$$S(x, \mathscr{O}) = \{y \in \mathscr{O} \ : xR_2 y\}. \quad (6)$$

where R_2 denotes the crisp outranking relation defined in Table 2.

According to conditions (a) and (b), the best compromise solution from a set of actions $\mathscr{O}$ should satisfy:

$$x^* = \underset{x \in \mathscr{O}}{\operatorname{argmin}} \{|W(x, \mathscr{O})|, -|S(x, \mathscr{O})|\}. \quad (7)$$

The RoI contains all the solutions satisfying Eq. (7), with lexicographic priority in favor of $|W(x, \mathscr{O})|$. By extension, the RoI of a MOP may be characterized using $\mathscr{O} = PF$ in Eq. (7), where PF would be the Pareto frontier.

4. Our proposal: The ACO-based Hyper-heuristic

This section describes HyperACO, a hyper-heuristic algorithm intended to treat MaOPs by sequencing the different variants of MOEA/D/O (Table 2). The application of HyperACO demands that the DM's preferences are compatible with outranking models. Fortunately, the preferences of many DMs are non-compensatory. The scientific literature provides recent studies with several satisfactory applications of outranking to real-world problems (e.g., [32–38]).

In the proposed hyper-heuristic, the HLH is an ACO algorithm that searches for the best solution in a sequencing optimization problem with discrete decision variables; here, the objective functions measure the closeness to the RoI to identify the sequence that generates the solutions

[1] A paradox of intransitive preferences arising from the aggregation of individual preferences under a majority rule.

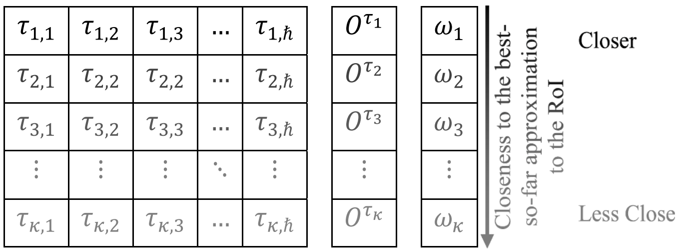

Fig. 1. Pheromone representation.

that best match the DM's preferences.

The LLHs are the different ways to embed the outranking relations into MOEA/D/O. The LLHs search for the best compromise solution in MaOPs with continuous decision variables; here, the two objective functions measure the quality of the solutions in terms of the outranking model (weakness and strength) as stated in Eqs. (5) and (6).

The rest of this section is structured as follows. Subsection 4.1 presents how the HLH represents the sequences and how they are evaluated. Subsection 4.2 describes the pheromone matrix and provides details of the criteria used to archive solutions in this structure. Subsection 4.3 presents how the ants exploit the pheromone matrix to construct new sequences iteratively. Lastly, Subsection 4.4 structures the information provided in the previous subsection in the form of an algorithm, which is explained.

4.1. Solution representation

In HyperACO, a vector $x_i = \langle x_{i,1},\ x_{i,2},\ x_{i,3}, \ldots, x_{i,\hbar}\rangle$ represents the sequence generated by the ith ant, where $\hbar$ is the size of the sequence, $x_{i,l} \in \{0,\ 1,\ 2,\ldots,\ 6\}\ \forall i \in \{1,\ 2,\ 3,\ldots,\kappa\}$, $l \in \{1,\ 2,\ 3,\ldots,\hbar\}$, and κ is the size of the colony. $x_{i,l}$ represents that the variant $V_{x_{i,l}}$ (see Table 2) is set as the lth component of the ith solution ($x_{i,l} = 0$ represents the original MOEA/D, without DM's preferences).

Algorithm 1 presents how HyperACO generates the solution sets by applying the sequence x_i. Here, $\mathscr{r}$ is a parameter defining the number of runs for each sequence, O^i is an array of $\mathscr{r}$ sets, and $\mathscr{g}$ is a parameter defining the number of generations for each component of the sequence. The initial population of the variants $x_{i,1}$ is generated at random (first component of the sequence). The initial population of the variants $x_{i,l}$ for $1 < l \leq \hbar$ is taken from the previous variant $x_{i,l-1}$. Accordingly, a sequence represents the order in which the LLHs should consecutively run—like a relay race—to provide an efficient composite MOEA.

Regarding the computational complexity, the basic operation in Algorithm 1 is the run of MOEA/D/O, which is in O($m^2 n$) (cf. [15]), where m is the number of objectives and n is the number of decision variables. Note that this operation is amortized over $\hbar$ iterations (Lines 3 and 4), so that $\hbar$ does not increase the order of complexity. Therefore, the complexity function is in O($m^2 n \mathscr{r}$).

4.2. Pheromone representation

In HyperACO, a bi-dimensional matrix τ represents the pheromone, which stores the κ best-so-far sequences. Fig. 1 depicts the structure of τ. Here, the required size of τ to store the sequences is $\kappa \times \hbar$.

Furthermore, the sequences in τ are sorted in increasing order based on the distance to the best-so-far approximation to the RoI (the set made of all the solutions generated so far that satisfy Eq. (7)). This distance is calculated by taking the solution sets generated by the sequences, which can be calculated through Algorithm 1. Keep in mind that O^{τ_i} is an array of $\mathscr{r}$ solution sets, one for every single run of the sequence τ_i.

In Fig. 1, the weights ω_i measure the importance of the sequence τ_i in function of its position, expressed as

$$\omega_i = \frac{\mathscr{p}(i)}{\sum_{j=1}^{\kappa} \mathscr{p}(j)},\ \text{where}\ \mathscr{p}(i) = i^{-\varsigma}. \tag{8}$$

Eq. (8) defines $\mathscr{p}(i)$ to be values of a power-law function with exponent ς and argument i, and ω_i to be an adjustment of its probability function to a discrete domain considering κ elements only. Here, ς is a parameter ($\varsigma \geq 1$) that sets the intensification in the algorithm. On the one hand, if $\varsigma \gg 1$, the probability of intensifying the search space around τ_1 increases exponentially. On the other hand, if $\varsigma \approx 1$, the probability of intensifying the search space around any τ_i becomes more linear ($1 \leq i \leq \kappa$).

According to Fig. 1, the criterion to sort τ is a pivotal element. First, the best-so-far approximation to the RoI (represented by the set $\mathscr{O}^{\iota}$) is calculated through Algorithm 2. Then, the distance from any solution set $O_j^{\tau_i}$ to $\mathscr{O}^{\iota}$ may be calculated. Let's consider the following distance metrics between sets:

- $\mathscr{d}_{\min}^{\text{Euclid}}(O_j^{\tau_i}, \mathscr{O}^{\iota})$: It is the Euclidean distance between the closest pair $(x,y) \in O_j^{\tau_i} \times \mathscr{O}^{\iota}$.
- $\mathscr{d}_{\text{avg}}^{\text{Euclid}}(O_j^{\tau_i}, \mathscr{O}^{\iota})$: It is the average Euclidean distance between all the pairs $(x,y) \in O_j^{\tau_i} \times \mathscr{O}^{\iota}$.
- $\mathscr{d}_{\min}^{\text{Cheb}}(O_j^{\tau_i}, \mathscr{O}^{\iota})$: It is the Chebyshev distance between the closest pair $(x,y) \in O^{\tau_i} \times \mathscr{O}^{\iota}$.
- $\mathscr{d}_{\text{avg}}^{\text{Cheb}}(O_j^{\tau_i}, \mathscr{O}^{\iota})$: It is the average Chebyshev distance between all the pairs $(x,y) \in O_j^{\tau_i} \times \mathscr{O}^{\iota}$.

The extreme values of these distances are associated with the performance of a sequence τ_i, being expressed as interval numbers as follows:

$$\mathbf{z}_{\min}^{\text{Euclid}}(\tau_i) = \left[\underline{z_{\min}^{\text{Euclid}}(\tau_i)}, \overline{z_{\min}^{\text{Euclid}}(\tau_i)}\right] \tag{9}$$

Algorithm 2
Approximation to the RoI.

Algorithm name: `approximate_RoI`
Input: The best approximation to the RoI at the previous iteration ($\mathcal{O}^{\iota-1}$), the solution sets of each sequence generated by the ants in the current iteration ($O^{\mathcal{A}_i}\ \forall i \in \{1,2,3,\ldots,\kappa\}$)
Output: The best-so-far approximation to the RoI ($\mathcal{O}^{\iota}$)

$\mathcal{O}_{\text{temp}}^{\iota} \leftarrow \mathcal{O}^{\iota-1}$
For each $i \in \{1,2,3,\ldots,\kappa\}$ ▷ For each ant in the colony
  For each $j \in \{1,2,3,\ldots,\mathscr{r}\}$ ▷ For each run of the sequence
    $\mathcal{O}_{\text{temp}}^{\iota} \leftarrow \mathcal{O}_{\text{temp}}^{\iota} \cup O_j^{\mathcal{A}_i}$
$\mathcal{O}^{\iota} \leftarrow \arg\min_{x \in \mathcal{O}_{\text{temp}}^{\iota}} \{|W(\mathcal{O}_{\text{temp}}^{\iota}, x)|, -|S(\mathcal{O}_{\text{temp}}^{\iota}, x)|\}$ ▷ Equation 7
Return $\mathcal{O}^{\iota}$

Algorithm 3
The HyperACO algorithm.

Algorithm name: `HyperACO`
Input: The parameters of the hyper-heuristic algorithm (ς, κ, ρ and $iter_{\max}$), the parameters of the sequencing optimization problem ($\hbar$, $\mathscr{r}$ and $\mathscr{g}$), the order relation that best fits the DM's preferences on the interval indicators ($\prec_{\mathcal{E}}$, $\prec_{\mathcal{C}}$ or $\prec_{\mathcal{W}}$), the LLHs (variants of MOEA/D/O) and the optimization problem they address.
Output: The sequence that best approximated the RoI (τ_1)

$\mathcal{A} \leftarrow \langle x_1, x_2, x_3, \dots, x_\kappa \rangle$
$\tau \leftarrow \langle \tau_1, \tau_2, \tau_3, \dots, \tau_\kappa \rangle$
$\mathcal{O} \leftarrow \emptyset$
For each $i \in \{1,2,3,\dots,7\}$ ▷ Generate the first seven sequences of the initial population
  $\tau_{i,l} \leftarrow i-1 \quad \forall l \in \{1,2,3,\dots,\hbar\}$
  $O^{\tau_i} \leftarrow$ `run_sequence`$(\tau_i, \mathscr{g}, \mathscr{r})$ ▷ Algorithm 1
For each $i \in \{8,9,10,\dots,\kappa\}$ ▷ Generate the rest of the initial population
  $\tau_{i,l} \leftarrow$ `pick_at_random()` $\quad \forall l \in \{1,2,3,\dots,\hbar\}$
  $O^{\tau_i} \leftarrow$ `run_sequence`$(\tau_i, \mathscr{g}, \mathscr{r})$ ▷ Algorithm 1
$\mathcal{O} \leftarrow$ `approximate_RoI`$(\mathcal{O}, O^{\tau_i}\ \forall i \in \{1,2,3,\dots,\kappa\})$ ▷ Algorithm 2
`sort`$(\tau, \mathcal{O})$ ▷ See Equations 13–16
For $1 \le \iota \le iter_{\max}$ ▷ Begin the main iterated process
  For each $i \in \{1,2,3,\dots,\kappa\}$
    $\jmath \leftarrow$ `roulette_wheel`(ω)
    For each $l \in \{1,2,3,\dots,\hbar\}$
      $x_{i,l} \leftarrow \begin{cases} \tau_{\jmath,l} & \text{if } \wp() < 1-\rho, \\ \texttt{pick\_at\_random()} & \text{otherwise,} \end{cases}$ ▷ Equation 17
    $O^{\mathcal{A}_i} \leftarrow$ `run_sequence`$(x_i, \mathscr{g}, \mathscr{r})$ ▷ Algorithm 1
  $\mathcal{O} \leftarrow$ `approximate_RoI`$(\mathcal{O}, O^{\mathcal{A}_i}\ \forall i \in \{1,2,3,\dots,\kappa\})$ ▷ Algorithm 2
  $\tau \leftarrow$ `join`$(\tau, \mathcal{A})$
  `sort`$(\tau, \mathcal{O})$ ▷ See Equations 13–16
  Remove the $|\tau| - \kappa$ worst solutions from τ
Return τ_1

where

$$\underline{z^{\text{Euclid}}_{\min}}(\tau_i) = \min_{1\le j\le \mathscr{r}}\left\{ d^{\text{Euclid}}_{\min}\left(O^{\tau_i}_j, \mathscr{O}^\iota\right)\right\},$$

and

$$\overline{z^{\text{Euclid}}_{\min}}(\tau_i) = \max_{1\le j\le \mathscr{r}}\left\{ d^{\text{Euclid}}_{\min}\left(O^{\tau_i}_j, \mathscr{O}^\iota\right)\right\}.$$

Similarly, the performance in terms of the other indicators is calculated using Eqs. (10)–(12).

$$\mathbf{z}^{\textbf{Euclid}}_{\textbf{avg}}(\tau_i) = \left[\min_{1\le j\le \mathscr{r}}\left\{ d^{\text{Euclid}}_{\text{avg}}\left(O^{\tau_i}_j, \mathscr{O}^\iota\right)\right\}, \max_{1\le j\le \mathscr{r}}\left\{ d^{\text{Euclid}}_{\text{avg}}\left(O^{\tau_i}_j, \mathscr{O}^\iota\right)\right\}\right] \tag{10}$$

$$\mathbf{z}^{\textbf{Cheb}}_{\textbf{min}}(\tau_i) = \left[\min_{1\le j\le \mathscr{r}}\left\{ d^{\text{Cheb}}_{\min}\left(O^{\tau_i}_j, \mathscr{O}^\iota\right)\right\}, \max_{1\le j\le \mathscr{r}}\left\{ d^{\text{Cheb}}_{\min}\left(O^{\tau_i}_j, \mathscr{O}^\iota\right)\right\}\right] \tag{11}$$

$$\mathbf{z}^{\textbf{Cheb}}_{\textbf{avg}}(\tau_i) = \left[\min_{1\le j\le \mathscr{r}}\left\{ d^{\text{Cheb}}_{\text{avg}}\left(O^{\tau_i}_j, \mathscr{O}^\iota\right)\right\}, \max_{1\le j\le \mathscr{r}}\left\{ d^{\text{Cheb}}_{\text{avg}}\left(O^{\tau_i}_j, \mathscr{O}^\iota\right)\right\}\right] \tag{12}$$

An interval number is a straightforward way to model the variability of the output. This variability stems from the stochastic nature of the composite MOEAs represented by the sequences. According to Eqs. (9)–(12), these intervals are calculated taking the $\mathscr{r}$ runs of the sequences.

The relevance of the four interval indicators depends on the DM's preferences. The Chebyshev indicators would be more appropriate for DMs with worst case-oriented preferences. Minimum and average distances are complementary to the same kind of indicators; minimum indicators measure the quality of the solution set considering the best solution alone, and average indicators measure the overall trend of the complete solution set.

Following the structure depicted in Fig. 1, we propose sorting τ using the possibility function on the interval indicators. In this regard, one of the three following sorting criteria may be applied according to the DM's perspective:

1 Euclidean indicators are considered with lexicographic priority in favor of the minimum distance. Formally, "τ_i precedes τ_j" following Eq. (13). Here, the symbol $\prec_{\mathscr{E}}$ represents a binary order relation taking the Euclidean indicators.

$$\tau_i \prec_{\mathscr{E}} \tau_j = \left\{(\tau_i,\tau_j) \;: Poss\left(\mathbf{z}^{\textbf{Euclid}}_{\textbf{min}}(\tau_j) \ge \mathbf{z}^{\textbf{Euclid}}_{\textbf{min}}(\tau_i)\right) > 0.5 \,\vee\, Poss\left(\mathbf{z}^{\textbf{Euclid}}_{\textbf{min}}(\tau_j) \ge \mathbf{z}^{\textbf{Euclid}}_{\textbf{min}}(\tau_i)\right) = 0.5 \,\wedge\, Poss\left(\mathbf{z}^{\textbf{Euclid}}_{\textbf{avg}}(\tau_j) \ge \mathbf{z}^{\textbf{Euclid}}_{\textbf{avg}}(\tau_i)\right) > 0.5\right\} \tag{13}$$

2 Likewise, Chebyshev indicators may also be taken to construct the binary order relation $\prec_{\mathscr{C}}$ to sort τ, attaching priority to the minimum distance. That is, "τ_i precedes τ_j" according to Eq. (14).

$$\tau_i \prec_{\mathscr{C}} \tau_j = \left\{(\tau_i,\tau_j) \;: Poss\left(\mathbf{z}^{\textbf{Cheb}}_{\textbf{min}}(\tau_j) \ge \mathbf{z}^{\textbf{Cheb}}_{\textbf{min}}(\tau_i)\right) > 0.5 \,\vee\, Poss\left(\mathbf{z}^{\textbf{Cheb}}_{\textbf{min}}(\tau_j) \ge \mathbf{z}^{\textbf{Cheb}}_{\textbf{min}}(\tau_i)\right) = 0.5 \,\wedge\, Poss\left(\mathbf{z}^{\textbf{Cheb}}_{\textbf{avg}}(\tau_j) \ge \mathbf{z}^{\textbf{Cheb}}_{\textbf{avg}}(\tau_i)\right) > 0.5\right\} \tag{14}$$

3 Lastly, an order relation may be defined considering an aggregate score of the four interval indicators. Here, the relation "τ_i precedes τ_j" is expressed as

$$\tau_i \prec_{\mathscr{W}} \tau_j = \left\{(\tau_i,\tau_j) \;: Poss\left(\mathbf{z}_{\mathscr{W}}(\tau_j) \ge \mathbf{z}_{\mathscr{W}}(\tau_i)\right) > 0.5\right\}, \tag{15}$$

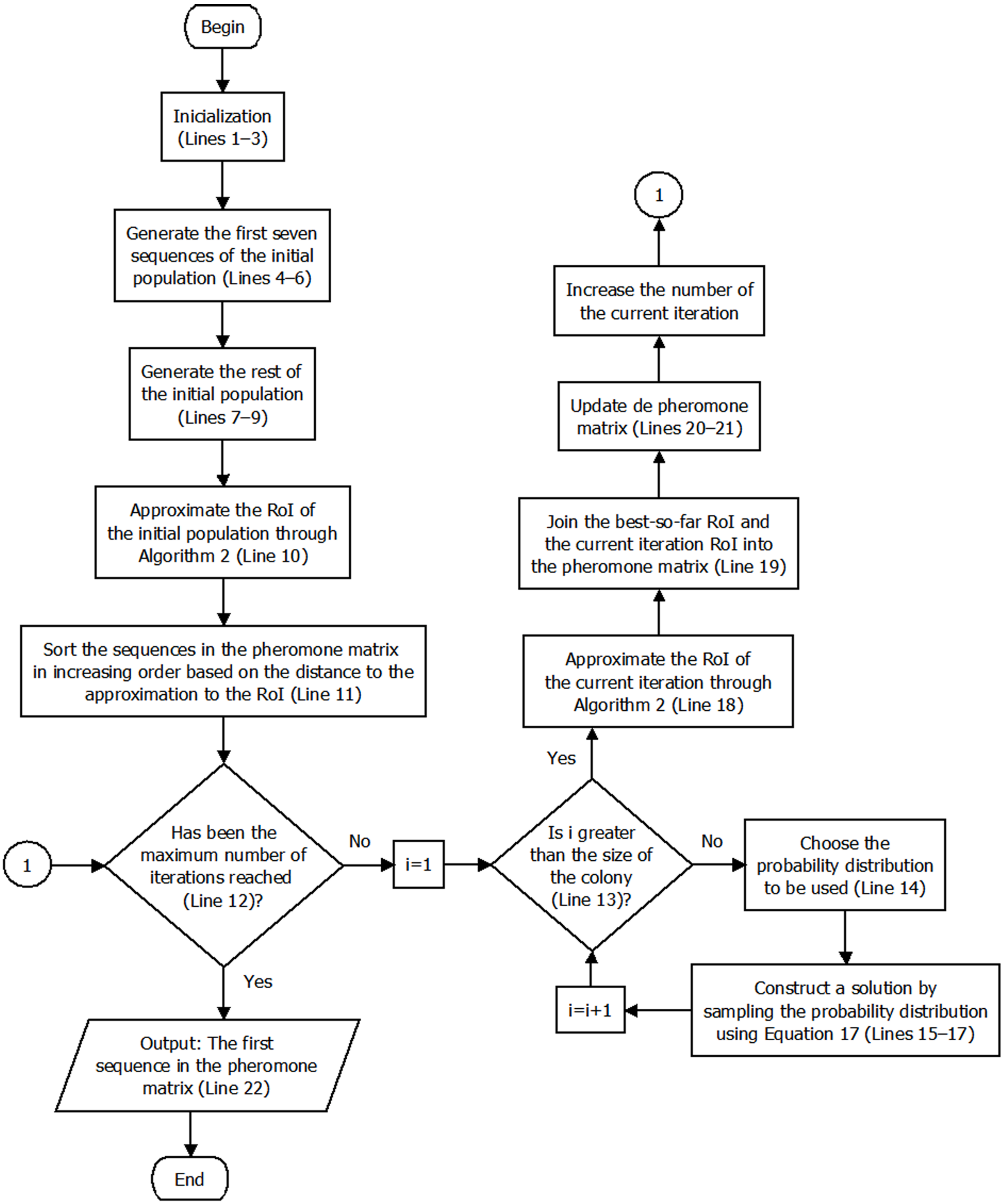

Fig. 2. Flowchart of HyperACO.

where

$$z_{\mathscr{W}}(\tau_i) = w_1 \cdot z^{\mathrm{Euclid}}_{\min}(\tau_i) + w_2 \cdot z^{\mathrm{Euclid}}_{\mathrm{avg}}(\tau_i) + w_3 \cdot z^{\mathrm{Chev}}_{\min}(\tau_i) + w_4 \cdot z^{\mathrm{Chev}}_{\mathrm{avg}}(\tau_i). \quad (16)$$

In Eq. (16), $w = \langle w_1,\ w_2, w_3, w_4 \rangle$ is a weight vector. Note that $z_{\mathscr{W}}(\tau_i)$ is also an interval number, which cumulates the weighted sum of the four interval indicators.

The relation $\prec_{\mathscr{W}}$ models a DM with a preference lying between $\prec_{\mathscr{E}}$ and $\prec_{\mathscr{C}}$. The vector w should be inferred to reflect the DM's preferences about the indicators.

About the computational complexity, the identification of the RoI (Algorithm 2, Line 5) implies the comparison of all the solution pairs considering the m objectives; therefore, it is in $O(\kappa^2 m)$. The calculation of (average and minimum) indicators implies that the distances between all the m-dimensional points in $\tau \times \tau$, considering the r runs, must be measured. In other words, it is in $\kappa^2 m\mathscr{r}$. Additionally, the sorting of the pheromone matrix may be performed in $O(\kappa \log_2 \kappa)$. Ergo, the complexity function of the pheromone update is in $O(\kappa^2 m\mathscr{r})$.

4.3. Solution construction

Ants construct solutions by performing the following steps:

1 A row of the pheromone matrix is selected. Here, the well-known roulette-wheel technique on the associated probability distribution—represented by the weights ω_i—is used. Let j be the row selected by the ith ant.
2 Afterward, the ith ant constructs the ith sequence of the colony, denoted as x_i. The sequence x_i is assigned as follows:

$$x_{i,l} = \begin{cases} \tau_{j,l} & \text{if } \wp() < 1 - \rho, \\ \text{pick at random}() & \text{otherwise,} \end{cases} \quad \forall l \in \{1,\ 2,\ 3,\ \cdots,\ \hbar\}, \tag{17}$$

where $x_{i,l}$ is the lth component of x_i (analogously, $\tau_{j,l}$), $\hbar$ is the size of the sequences, pick_at_random() is a function that randomly chooses one of the seven LLHs, $\wp()$ is a function generating pseudorandom numbers with $\wp() \sim U(0,1)$, and ρ is a parameter to set the balance between exploitation and exploration ($0 < \rho < 1$). This parameter has an effect similar to the evaporation rate in the classic ACO: new regions in the search space are explored with high values; otherwise (with low values), long subsequences in the best solutions are exploited and replicated.

The complexity function of Step 1 depends on the calculation of the weights, which is in $O(\kappa)$. Regarding Step 2, the complexity function is in $O(\kappa\hbar)$ because Eq. (17) must be performed on each decision variable (in the HLH's domain) and each ant of the colony. Ergo, the complexity of the complete construction procedure is in $O(\kappa \cdot (\hbar + 1))$.

4.4. The main algorithm of HyperACO

Algorithm 3 presents an algorithmic outline for HyperACO. Here, Lines 1–3 initialize the main data structures: the array containing the sequences generated in each iteration by the ant colony ($\mathscr{A}$), the pheromone matrix (τ), and the best-so-far approximation to the RoI ($\mathscr{O}$). After that, the initial solutions are generated (Lines 4–9). Note that the first seven initial solutions correspond with the original MOEA/D and the six variants of MOEA/D/O (Lines 4–6), and the rest of the initial solutions are generated at random (Lines 7–9). These initial sequences are run to estimate their performance (Lines 6 and 9). Then, the first approximation of the RoI is calculated, and the pheromone matrix is sorted accordingly (Lines 10 and 11).

Lines 12–21 present the body of the main iterative process. Here, $iter_{\max}$ is a parameter defining the maximum number of iterations for HyperACO. Each ant of the colony constructs a solution following the steps described in Subsection 4.3 (Lines 13–17). These recently constructed sequences are considered to update the best-so-far approximation to the RoI (Line 18). Then, the κ solutions closest to $\mathscr{O}$ are kept in τ (Lines 19–21). Consequently, τ_1 archives the sequence offering the best performance; so, this data structure is the response finally returned by HyperACO (Line 22).

Furthermore, Fig. 2 presents a flowchart depicting the main processes of Algorithm 3. Here, each flowchart symbol indicates the line numbers of Algorithm 3 it represents.

Regarding the computational complexity of Algorithm 3, the following points must be considered:

(a) The complexity of Lines 4–9 is in $O(\kappa \cdot (m^2 n\mathscr{r} + \hbar + 1))$. The term $m^2 n\mathscr{r}$ is because of Algorithm 1, and the term $\hbar + 1$ is because of the construction procedure. Note that this complexity function also applies to Lines 13–17.
(b) The complexity of Lines 10 and 11 is in $O(\kappa^2 m\mathscr{r})$, which represents the pheromone update (as discussed in Section 4.2). Bear in mind that the complexity functions of the sorting ($\kappa \log_2 \kappa$) and approximation of the RoI ($\kappa^2 m$) are dominated by the greater order function associated with the calculation of distances to the RoI. Moreover, note that this complexity function also applies to Lines 18–21.

Therefore, a preliminary version of the complexity function would be in $O(iter_{\max} \cdot \kappa \cdot\ (m^2 n\mathscr{r} + \kappa m\mathscr{r} + \hbar + 1))$. The variables defining the input size are m and n in the LLHs' domain and $\mathscr{r}$ and $\hbar$ in the HLH's domain. Because $iter_{\max}$ and κ are constant parameters; they should be discarded. Finally, the complexity function of Algorithm 3 is in $O(\ m^2 n\mathscr{r} + \hbar)$.

5. Experimental validation

This section presents the numeric results supporting the validity of HyperACO. Subsection 5.1 describes the experimental conditions: the features of hardware and software, the test suites, and the reference solutions to measure performance. Subsection 5.2 presents the results of HyperACO and MOEA/D/O to emphasize the advantages of the proposed hyper-heuristic compared to MOEA/D/O, a state-of-the-art MOEA that incorporates the DM's preferences. In contrast, Subsection 5.3 compares the results of HyperACO with two MOEAs that approximate the complete Pareto frontier: RVEA-iGNG (Reference Vector-guided Evolutionary Algorithm using Improved Growing Neural Gas) by Liu et al. [39] and AR-MOEA (Indicator-based Multi-Objective Evolutionary Algorithm with Reference Point Adaptation) by Tian et al. [40].

5.1. Experimental settings

We implemented Hyper-ACO using C under Linux Ubuntu on a computer with an Intel Core i7 at 2.70 GHz with 16 GB of RAM. The LLHs have the same parameter settings suggested by Fernández et al. [15]. The parameter settings of the HLH were: $\kappa = 50$, $\rho = 0.1$, $\varsigma = 2$, $iter_{\max} = 100$, $\mathscr{r} = 5$, $\mathscr{g} = 50$, and $\hbar = 20$. These settings allow Hyper-ACO to compose MOEAs performing 100,000 evaluations of the objective functions. The reference MOEAs (MOEA/D/O, RVEA-iGNG, and AR-MOEA) are also limited to the same number of evaluations (to ensure a fair comparison).

There are different test suites to assess the performance of for MOEAs; e.g., DTLZ [41], WFG [42], LZ/UF [43] and MaOP [44]. In this paper, we have used the DTLZ and WFG test suites to validate our proposed approach. DTLZ and WFG are widely accepted as standard benchmarks to assess the performance of MOEAs. There are nine continuous problems in both DTLZ (DTLZ1–DTLZ9) and WFG (WFG1–WFG9), which are considered representative because they offer a wide range of geometries in the resulting Pareto frontiers. Moreover, these problems are scalable regarding the number of decision variables and objective functions.

Each problem has been tested with three, five, and ten objective functions. Consequently, we have 54 input instances. Each problem is customized considering n (number of decision variables), m (number of objective functions, $m \in \{3,\ 5,\ 10\}$), and k (number of position-related variables) as follows:

- DTLZ1: $n = m + k - 1$, where $k = 5$.
- DTLZ2–6: $n = m + k - 1$, where $k = 10$.
- DTLZ7: $n = m + k - 1$, where $k = 20$.
- DTLZ8–9: $n = 10(k + 1)$, where $k = m - 1$.
- WFG1–9: $k = 2(m - 1)$ and $n = \begin{cases} 24 & \text{if } m = 3, \\ 47 & \text{if } m = 5, \\ 105 & \text{if } m = 10. \end{cases}$

Table 3
Comparison between the composite MOEAs by HyperACO and MOEA/D/O.

Version of HyperACO	*Problem*	*3 objectives ($m = 3$)* Best variants of MOEAD/D/O	*HyperACO significantly gets closer to the A-RoI*	*5 objectives ($m = 5$)* Best variants of MOEAD/D/O	*HyperACO significantly gets closer to the A-RoI*	*10 objectives ($m = 10$)* Best variants of MOEAD/D/O	*HyperACO significantly gets closer to the A-RoI*
$\prec_{\mathscr{E}}$-HyperACO	DTLZ1	3	✓	1, 2, 4, 5		2, 6	✓
	DTLZ2	5	✓	2, 3	✓	2, 6	✓
	DTLZ3	0, 2	✓	6	✓	0, 2, 6	✓
	DTLZ4	3, 5	✓	3, 5	✓	3, 5	✓
	DTLZ5	0–6		0-6		0–6	
	DTLZ6	0–6		1	✓	0	✓
	DTLZ7	0, 1, 4	✓	1	✓	1, 3, 5	✓
	DTLZ8	0-6		0	✓	4	✓
	DTLZ9	3, 5	✓	2	✓	1, 2, 4, 6	✓
	WFG1	3, 5	✓	2	✓	1, 2, 4, 6	✓
	WFG2	0–6		0	✓	4	✓
	WFG3	3	✓	1, 2, 4, 6		2,6	✓
	WFG4	1–6		1	✓	0	✓
	WFG5	0, 2	✓	6	✓	0, 2, 6	✓
	WFG6	3, 5	✓	3, 5	✓	3,5	✓
	WFG7	0–6		0–6		0–6	✓
	WFG8	5	✓	2, 3	✓	2, 6	✓
	WFG9	0, 1, 4	✓	1	✓	1, 3, 5	✓
$\prec_{\mathscr{C}}$-HyperACO	DTLZ1	0	✓	6	✓	2, 6	✓
	DTLZ2	0	✓	0	✓	0	✓
	DTLZ3	0	✓	6		0, 2, 6	✓
	DTLZ4	0	✓	0	✓	5	✓
	DTLZ5	0	✓	0	✓	0–6	✓
	DTLZ6	0	✓	0	✓	0	✓
	DTLZ7	1, 4	✓	0, 2, 6	✓	1, 3–5	✓
	DTLZ8	0–6		1–6		1–6	
	DTLZ9	0, 2, 6		6	✓	2	✓
	WFG1	0, 2, 6		6	✓	2	✓
	WFG2	0–6		1–6		1–6	
	WFG3	0	✓	6	✓	2, 6	✓
	WFG4	0	✓	0	✓	0	✓
	WFG5	0	✓	6		0, 2, 6	✓
	WFG6	0	✓	0	✓	5	✓
	WFG7	0		0		0–6	
	WFG8	0	✓	0	✓	0	✓
	WFG9	1, 4	✓	0, 2, 6	✓	1, 3–5	✓
$\prec_{\mathscr{W}}$-HyperACO	DTLZ1	0, 3	✓	1–6		2, 6	✓
	DTLZ2	1–6		2	✓	0	✓
	DTLZ3	5	✓	1, 2, 4, 6	✓	0, 2, 6	✓
	DTLZ4	1, 4	✓	1–6		5	✓
	DTLZ5	0		1–6		0–6	✓
	DTLZ6	0	✓	0	✓	0	✓
	DTLZ7	1, 4	✓	2, 4, 6	✓	1, 3–5	✓
	DTLZ8	0–6		2–5	✓	1–6	
	DTLZ9	0, 2, 6		2, 6	✓	2	✓
	WFG1	0, 2, 6		2, 6	✓	2	✓
	WFG2	0–6		2–5	✓	1–6	
	WFG3	0, 3	✓	1–6		2, 6	✓
	WFG4	0	✓	0	✓	0	✓
	WFG5	5	✓	1, 2, 4, 6	✓	0, 2, 6	✓
	WFG6	1, 4	✓	1–6		5	✓
	WFG7	6		1–6		0	✓
	WFG8	0		2	✓	0	✓
	WFG9	1, 4	✓	2, 4, 6	✓	1, 3–5	✓

Regarding the DM's preferences, we took the ten interval outranking models used by Fernández et al. [15]. The parameter values of these models were synthetically generated to represent ten DMs with different systems of preferences. The true RoI of these synthetic DMs for the DTLZ and WFG problems had already been approximated and used in recent studies (e.g., [15,16,45,46]), favoring replicability and comparability of results. The Approximated RoI (A-RoI) considers the solutions satisfying Eq. (7) from a representative sample of 100,000 Pareto-efficient points.

HyperACO and MOEA/D/O were run 30 times for each synthetic DM on each input instance. RVEA-iGNG and AR-MOEA were also run 30 times; however, they do not consider the DM's preferences. All statistical tests for significance were performed through the STAC platform [47].

5.2. A comparison with the variants of MOEA/D/O

In this section, we compare the results of the composite MOEAs obtained by HyperACO with the best variants of MOEA/D/O reported by Fernández et al. [15]. In this experiment, we consider incorporating the three order relations into HyperACO, giving rise to the following three versions: $\prec_{\mathscr{E}}$-HyperACO, $\prec_{\mathscr{C}}$-HyperACO, and $\prec_{\mathscr{W}}$-HyperACO, in accordance with Eqs. (13)–(16). The weights for the relation $\prec_{\mathscr{W}}$ are $w = \langle 0.3,\ 0.1,\ 0.4,\ 0.2 \rangle$.

Table 3 summarizes the results of this experiment. Here, the first column refers to the version of HyperACO, and the second column indicates the problem being treated. Afterward, the results are grouped by the number of objectives: three (the third and fourth columns), five (the fifth and sixth columns), and ten (the seventh and eighth columns).

Table 4
Comparison between the composite MOEAs by HyperACO and two state-of-the-art MOEAs.

Version of HyperACO	*Benchmark*	*Number of objectives*	*Problems in which HyperACO significantly* outperformed*		*Problems in which HyperACO is significantly* outperformed by*	
			(*b*) RVEA-iGNG	(*c*) AR-MOEA	(*b*) RVEA-iGNG	(*c*) AR-MOEA
$\prec_{\mathscr{E}}$-HyperACO	DTLZ	3	5, 7, 9	3–6, 8, 9	1, 2, 3	1
		5	2, 4–6, 8, 9	2, 4–6, 8, 9	3	1, 3
		10	1, 3, 4, 6, 7, 9	1–4, 6, 7, 9		
	WFG	3	2–5	2, 4–7	7, 8, 9	9
		5	1, 2, 4–6, 9	1, 2, 4–6, 8, 9	7	7
		10	1, 5–7, 9	1, 3, 5–9		
$\prec_{\mathscr{C}}$-HyperACO	DTLZ	3	1, 2, 5, 6, 8, 9	2–6, 8, 9	3	7
		5	2, 4–6, 8, 9	2, 4–6, 8, 9	3, 7	1, 7
		10	1, 3, 4, 7, 9	1–4, 6, 7, 9	8	8
	WFG	3	1, 2, 4, 5, 8	1, 2–8	3, 7	3, 9
		5	1, 2, 4, 5, 8, 9	1, 2, 4–6, 8, 9	3, 7	3
		10	1, 5–7, 9	1, 3–9	2	
$\prec_{\mathscr{W}}$-HyperACO	DTLZ	3	2, 4, 5, 7–9	2, 4–6, 8, 9	3	1, 7
		5	1, 3–6, 8, 9	1–4, 6–9		
		10	1, 3–5, 7, 9	1–7, 9	6, 8	8
	WFG	3	1, 3, 5, 6, 8	1, 2, 4–6	7	3, 9
		5	1, 2, 4, 5, 7, 9	1–4, 6–9	8	
		10	3, 5–7, 9	3–9	2	

* According to a Friedman's non-parametric test for statistical significance, and a Nemenyi Post-hoc analysis (both with a 0.95-confidence interval)

Table 5
Borda scores of the reference MOEAs and HyperACO.

Version of HyperACO	*Number of Objectives*	*The Borda score of*		
		(*a*) HyperACO	(*b*) RVEA-iGNG	(*c*) AR-MOEA
$\prec_{\mathscr{E}}$-HyperACO	3	31.5	32.0	44.5
	5	26.0	41.0	41.0
	10	23.5	37.0	47.5
$\prec_{\mathscr{C}}$-HyperACO	3	26.5	39.5	42.0
	5	26.5	40.5	41.0
	10	25.0	34.0	49.0
$\prec_{\mathscr{W}}$-HyperACO	3	28.0	39.5	40.5
	5	22.0	41.5	44.5
	10	25.0	36.0	47.0

Then, given a DTLZ/WFG problem and a number of objectives, the third, fifth, and seventh columns present the best variants of MOEA/D/O (cf. [15]); the results of these variants are the closest to the A-RoI, and there is no statistically significant difference among them. Lastly, the fourth, sixth, and eighth columns indicate (with a checkmark) the problems in which HyperACO composed an *ad hoc* MOEA with a better approximation of the A-RoI than the best variants of MOEA/D/O. These comparisons are supported by a Friedman non-parametric test for statistical significance and a Nemenyi Post-hoc analysis (both with a 0.95-confidence interval). Bear in mind that the initialization of HyperACO injects MOEA/D and the six variants of MOEA/D/O (Lines 4–6 in Algorithm 3) as initial sequences; ergo, the composite MOEA generated by HyperACO is always at least as good as the best of them.

The information provided in Table 3 allows us to highlight the following points:

- The advantages of HyperACO become more tangible as the number of objective functions increases. This conclusion is drafted because the number of problems in which HyperACO outperformed the best variants of MOEA/D/O correlates with the number of objective functions.
- When the Euclidean distance is taken ($\prec_{\mathscr{E}}$-HyperACO), no gain was observed in DTLZ5 regardless of the number of objectives. According to Deb et al. [41], this problem may be particularly easy to address for a well-designed search algorithm because of the geometry of its Pareto frontier. Consequently, we concluded that for this problem, it is not necessary to apply a composite MOEA generated by HyperACO; any of the variants of MOEA/D/O seems to be efficient enough.
- When the Chebyshev distance is considered ($\prec_{\mathscr{C}}$-HyperACO), there is no gain in DTLZ8, WFG2 and WFG7 regardless of the number of objectives. Let's consider the following discussions:
  - On the one hand, according to Deb et al. [41], the Pareto frontier of DTLZ8 is a combination of a straight line (due to side constraints) and a hyper-plane. MOEAs will find severe difficulties maintaining a good distribution while finding solutions in both regions of this problem. Consequently, we concluded that the composite MOEAs by HyperACO do not satisfactorily cope with the geometry of this Pareto frontier. The side constraints could be affecting the performance of our approach. Note that this conclusion is also partially held for $\prec_{\mathscr{W}}$-HyperACO.
  - On the other hand, WFG2 is the single problem in this benchmark with a convex but disconnected region; and WFG7 is concave, separable, and unimodal [48]. Huband et al. [42] identified the WFG problems in which evolutionary algorithms are more likely to converge to the Pareto front. WFG2 and WFG7 are two of these problems. We hypothesized that, in these problems, HyperACO did not contribute because the original versions of MOEA/D/O performed well enough. Note that this behavior also occurred for $\prec_{\mathscr{W}}$-HyperACO with $m = 10$ in WFG2.
- In DTLZ4, DTLZ7, WFG1, WFG5, WFG6, WFG8, and WFG9, the contribution of HyperACO stands out. Focusing on the unconstrained problems, these seven problems are particularly challenging for MOEAs. Zapotecas-Martínez et al. [48] and Huband et al. [42] presented studies on the geometry and properties of these problems. Such challenging conditions allowed the advantages of HyperACO to be remarked.
- Lastly, the results of HyperACO are encouraging because it reached better approximations of the RoI in 122 of the 162 times it was applied (three versions of HyperACO ran on 54 input instances). What is more, the highest contribution was observed in MaOPs ($m > 4$).

Additionally, Appendix B presents the results of Table 3 in more detail. This appendix presents the results of the statistical tests for each version of HyperACO, each value of m, and each problem. This in-detail information is provided for consultation.

Table A1
A numeric example of the calculation of $\sigma(x,y)$ in the framework of the IOA.

Element	*Description*
x, y	A solution x has an image $f(x) = \langle f_1(x),\ f_2(x),\ f_3(x),\ \ldots, f_m(x)\rangle$, where $f_k(x)$ is the value of the kth objective of x. Analogously, $f(y) = \langle f_1(y),\ f_2(y),\ f_3(y),\ \ldots, f_m(y)\rangle$. *Example:* $f(x) = \langle 0.40,\ 0.50,\ 0.60\rangle$ $f(y) = \langle 0.56,\ 0.70,\ 0.54\rangle$ Here, $m = 3$.
Ω	$\Omega = \{\delta_k(x,y) : \delta_k(x,y) > 0\ \forall k \in \{1,\ 2,\ 3,\ \ldots,\ m\}\}$, where m is the number of objectives and $\delta_k(x,y)$ is the credibility degree of the statement "x outranks y with respect to objective k". This index is calculated using $\delta_k(x,y) = Poss(f_k(y) \geq f_k(x))$. *Example:* $\delta_1(x,y) = 1,\ \delta_2(x,y) = 1,\ \delta_3(x,y) = 0$ $\therefore \Omega = \{1,\ 1\}$
$\boldsymbol{w},\ \boldsymbol{v},\ \boldsymbol{\lambda},\ \boldsymbol{\beta}$	Preference model parameters. The DM's value system, denoted as $\boldsymbol{w},\ \boldsymbol{v},\ \boldsymbol{\lambda},\ \boldsymbol{\beta}$, consists of the weight vector $\boldsymbol{w}$, the veto threshold vector $\boldsymbol{v}$, the interval number $\boldsymbol{\lambda}$ that reflects a majority threshold, and the overall credibility threshold $\boldsymbol{b}$. All these parameters are interval numbers. Note that $\sum_{k=1}^{m} \underline{w_k} \leq 1$ and $\sum_{k=1}^{m} \overline{w_k} \geq 1$ in a feasible preference system, as well as $\underline{\beta} > 0.5$ and $\underline{\lambda} > 0.5$. *Example:* $\boldsymbol{w} = \langle \boldsymbol{w}_1,\ \boldsymbol{w}_2,\ \boldsymbol{w}_3\rangle,\ \boldsymbol{w}_1 = [0.35,\ 0.45],\ \boldsymbol{w}_2 = [0.30,\ 0.35],\ \boldsymbol{w}_3 = [0.20,\ 0.25]$ $\boldsymbol{v} = \langle \boldsymbol{v}_1,\ \boldsymbol{v}_2,\ \boldsymbol{v}_3\rangle,\ \boldsymbol{v}_1 = [0.05,\ 0.1],\ \boldsymbol{v}_2 = [0.03,\ 0.08],\ \boldsymbol{v}_3 = [0.07,\ 0.1]$ $\boldsymbol{\lambda} = [0.65,\ 0.75]$ $\boldsymbol{\beta} = [0.51,\ 0.55]$
$C(xS_\gamma y)$	$C(xS_\gamma y) = \{k \in \{1,\ 2,\ 3,\ \ldots, m\}\ : \delta_k(x,y) \geq \gamma\}$ is the set of the objectives in the concordance coalition. *Example:* Let's consider $\gamma \in \Omega$ $C(xS_\gamma y) = \{1,\ 2\}$
$D(xS_\gamma y)$	$D(xS_\gamma y) = \{1,\ 2,\ 3, \ldots,\ m\} \backslash C(xS_\gamma y)$ is the set of the objectives in the discordance coalition. *Example:* $D(xS_\gamma y) = \{3\}$
$\boldsymbol{c}(x,y,\gamma)$	The concordance index according to γ, defined as $\boldsymbol{c}(x,y,\gamma) = \left[\underline{c(x,y)},\ \overline{c(x,y)}\right]$, where: $\underline{c(x,y)} = \begin{cases} \sum_{k\in C(xS_\gamma y)} \underline{w_k} & \text{if } \sum_{k\in C(xS_\gamma y)} \underline{w_k} + \sum_{k\in D(xS_\gamma y)} \overline{w_k} \geq 1, \\ 1 - \sum_{k\in D(xS_\gamma y)} \overline{w_k} & \text{otherwise.} \end{cases}$ $\overline{c(x,y)} = \begin{cases} \sum_{k\in C(xS_\gamma y)} \overline{w_k} & \text{if } \sum_{k\in C(xS_\gamma y)} \overline{w_k} + \sum_{k\in D(xS_\gamma y)} \underline{w_k} \leq 1, \\ 1 - \sum_{k\in D(xS_\gamma y)} \underline{w_k} & \text{otherwise.} \end{cases}$ *Example:* $\underline{c(x,y)} = 0.75 \because \underline{w_1} + \underline{w_2} + \overline{w_3} = 0.90$ $\overline{c(x,y)} = 0.80 \because \overline{w_1} + \overline{w_2} + \underline{w_3} = 1.00$ $\therefore \boldsymbol{c}(x,y,\gamma) = [0.75,\ 0.80]$
$d_k(x,y)$	It is the credibility degree of the assertion "the kth objective alone vetoes the assertion *x outranks y*"; $d_k(x,y) = Poss(f_k(x) \geq f_k(y) + \boldsymbol{v}_k)$, where $\boldsymbol{v}_k$ is the veto threshold associated with the kth objective. *Example:* $d_1(x,y) = Poss(f_1(x) \geq f_1(y) + \boldsymbol{v}_1) = Poss(0.40 \geq 0.56 + [0.05,\ 0.10]) = 0$ $d_2(x,y) = Poss(f_2(x) \geq f_2(y) + \boldsymbol{v}_2) = Poss(0.50 \geq 0.70 + [0.03,\ 0.08]) = 0$ $d_3(x,y) = Poss(f_3(x) \geq f_3(y) + \boldsymbol{v}_3) = Poss(0.60 \geq 0.54 + [0.07,\ 0.10])$ $d_3(x,y) = Poss(0.60 \geq [0.61,\ 0.64]) = 0$
σ_γ	$\sigma_\gamma = \min\left\{\gamma,\ Poss(\boldsymbol{c}(x,y,\gamma) \geq \boldsymbol{\lambda}),\ 1 - \max_{k\in D(xS_\gamma y)}\{d_k(x,y)\}\right\}$ *Example:* $Poss(\boldsymbol{c}(x,y,\gamma) \geq \boldsymbol{\lambda}) = Poss([0.75,\ 0.80] \geq [0.65,\ 0.75]) = 1$ $\sigma_\gamma = \min\{1,\ 1\} = 1$
$\sigma(x,y)$	The credibility index of the assertion "x is at least as good as y", $\sigma(x,y) = \max_{\gamma\in\Omega}\{\sigma_\gamma\}$ *Example:* $\sigma(x,y) = \max\{1,\ 1\} = 1$ Please, note that $\Omega = \{1,\ 1\}$ in our example.

5.3. A comparison with a posteriori MOEAs

This subsection aims to validate the efficiency of HyperACO in comparison with two state-of-the-art MOEAs: RVGEA-iGNG[2] and AR-MOEA[3]. The parameter values of these MOEAs were taken from Liu et al. [39] and Tian et al. [40].

Table 4 presents the results obtained by Hyper-ACO, RVGEA-iGNG, and AR-MOEA in addressing the DTLZ and WFG test suites. Here, the first column indicates the version of Hyper-ACO being considered ($\prec_{\mathscr{E}}$, $\prec_{\mathscr{C}}$ or $\prec_{\mathscr{W}}$); the second column indicates the benchmark, the third column presents the number of objective functions; the fourth and fifth columns indicate the problems in which HyperACO significantly outperformed the reference MOEAs (RVEA-iGNG in the fourth column, and AR-MOEA in the fifth one); and, lastly, the sixth and seventh columns indicate the problems in which the reference MOEAs significantly outperformed HyperACO (RVEA-iGNG in the sixth column, and AR-MOEA in the seventh one).

According to Table 4, the following remarks may be drafted:

- In comparison with RVEA-iGNG and AR-MOEA, HyperACO reached the highest performance when the relation $\prec_{\mathscr{W}}$ was considered.
- Although the choice between HyperACO, RVEA-iGNG, and AR-MOEA depends on the properties of the problem, the number of objectives, and the order relation, HyperACO composed MOEAs that were statistically better on a regular basis.
- The results on some problems are particularly encouraging: DTLZ4 and DTLZ9. Regardless of the number of objectives, HyperACO always composed MOEAs that RVEA-iGNG and AR-MOEA were not able to outperform.
- DTLZ8 is still a challenging problem for HyperACO. In 10-objective problems, none of the versions of HyperACO was able to outperform the reference MOEAs.
- Considering the Euclidean distance and $m = 10$, HyperACO consistently outperformed both algorithms in the 18 problems.
- For the 10-objective problems, the results of HyperACO were consistently and statistically better than AR-MOEA and RVEA-iGNG on DTLZ1, DTLZ3, DTLZ4, DTLZ7, DTLZ9, WFG5–7, and WFG9. Some relevant properties of these problems are:
  - DTLZ1: linear, separable, many-to-one, and multimodal.
  - DTLZ3: concave, separable, many-to-one, and multimodal.
  - DTLZ4: concave, separable, many-to-one, and uni-modal.
  - DTLZ7: mixed (concave/convex), disconnected, separable, and multimodal.
  - DTLZ9: side constrained, and partially degenerate. With $m > 3$, some inconsistencies make it difficult to analyze this problem (cf. [49]).
  - WFG5: concave, separable, many-to-one, and deceptive.
  - WFG6: concave, non-separable, many-to-one, and uni-modal.
  - WFG7: concave, separable, biased, many-to-one, and uni-modal.
  - WFG9: concave, non-separable, biased, many-to-one, and deceptive.
- As can be seen, it is difficult to reach a generalization based on the geometry of the Pareto front. As a note, DTLZ7 is considered one of the most challenging problems in DTLZ; whereas WFG5, WFG6, and WFG9 were presented by Huband et al. [42] in the top five of the most challenging problems in WFG.

Furthermore, Table 5 presents a ranking of the three algorithms based on their Borda scores. Here, the metaheuristics are sorted according to the conducted tests for statistical significance (Friedman) and

[2] Source code taken from https://github.com/BIMK/PlatEMO/tree/master/PlatEMO/Algorithms/Multi-objective%20optimization/RVEA-iGNG

[3] Source code taken from https://github.com/BIMK/PlatEMO/tree/master/PlatEMO/Algorithms/Multi-objective%20optimization/AR-MOEA

Table B1
Ranking of HyperACO, MOEA/D and the six variants of MOEA/D/O according to their closeness to the A-RoI in the DTLZ test suite.

Problem	*Ranking*	$\prec_{\mathscr{E}}$			$\prec_{\mathscr{C}}$			$\prec_{\mathscr{W}}$		
		$m = 3$	$m = 5$	$m = 10$	$m = 3$	$m = 5$	$m = 10$	$m = 3$	$m = 5$	$m = 10$
DTLZ1	1st	H	1, 2, 4, 6, H	H	H	H	H	H	1–6, H	H
	2nd	3	0, 3, 5	2, 6	0	6	2, 6	0, 3	0	2, 6
	3rd	1, 2, 4–6		0, 1	6	1, 2, 4	0, 1	1, 2, 4–6		0, 1, 3–5
	4th	0		4	2	0, 3, 5	4			
	5th			3, 5	1		3, 5			
	6th				3–5					
DTLZ2	1st	H	H	H	H	H	H	1–6, H	H	H
	2nd	5	2, 3	2, 6	0	0	0	0	2	0
	3rd	3	1, 4–6	0	2	1–6	2, 6		0, 3, 5	2, 6
	4th	1, 2, 4, 6	0	1, 3–5	1, 6		1, 3–5		1, 4, 6	1, 3–5
	5th	0			3–5					
DTLZ3	1st	H	H	H	H	6, H	H	H	H	H
	2nd	0, 2	6	0, 2, 6	0	0–2, 4	0, 2, 6	5	1, 2, 4, 6	0, 2, 6
	3rd	6	0–2, 4	1	2	3, 5	1	1–4, 6	0, 3, 5	1, 3–5
	4th	1, 3–5	3, 5	3–5	1, 3–6		3–5	0		
DTLZ4	1st	H	H	H	H	H	H	H	1–6, H	H
	2nd	3, 5	3, 5	3, 5	0	0	5	1, 4	0	5
	3rd	1, 2, 4, 6	4, 1	1, 2, 4	1, 4	6	3	0		1, 3, 4
	4th	0	2, 6	0, 6	2, 3, 5, 6	2	1, 2, 4, 6	2, 3, 5, 6		2, 6
	5th		0			1, 3–5	0			0
	6th									
DTLZ5	1st	0–6, H	0–6, H	0–6, H	0, H	0, H	0–6, H	0, H	1–6, H	H
	2nd				1–6	1–6		6	0	0–6
	3rd							1–5		
DTLZ6	1st	1–6, H	H	H	H	H	H	H	H	H
	2nd	0	1	0	0	0	0	0	0	0
	3rd		6	1, 6	1–6	1, 6	1, 6	1–6	1, 6	1, 6
	4th		2–5	2–5		2–5	2–5		2–5	4
	5th		0							2, 3, 5
DTLZ7	1st	H	H	H	H	H	H	H	H	H
	2nd	0, 1, 4	1	1, 3, 5	1, 4	0, 2, 6	1, 3–5	1, 4	2, 4, 6	1, 3–5
	3rd	6	3–5	2, 4, 6	0	1, 4	2, 6	0	1, 3, 5	2, 6
	4th	2, 3, 5	2, 6	0	6	3, 5	0	6, 2	0	0
	5th		0		2			3, 5		
	6th				3, 5					
DTLZ8	1st	0–6, H	H	H	0–6, H	1–6, H	1–6, H	0–6, H	H	1–6, H
	2nd		0	4		0	0		2–5	0
	3rd		6	1, 3					1, 6	
	4th		1	2, 5, 6					0	
	5th		2	0						
	6th		3–5							
DTLZ9	1st	H	H	H	0, 2, 6, H	H	H	0, 2, 6, H	H	H
	2nd	3, 5	2	1, 2, 4, 6	1, 3–5	6	2	1, 3–5	2, 6	2
	3rd	0–2, 4, 6	6	3		2	1, 6		1, 3–5	1, 6
	4th		1, 3–5	0, 5		1, 3–5	4		0	3, 4
	5th		0			0	0, 3, 5			0, 5

the post-hoc analysis (Nemenyi) for each input instance. Then, the best metaheuristic gets the first position, and the worst one obtains the third position; the position is averaged if a draw occurs. In this context, the Borda score is the cumulative sum of those positions over every single instance. Hence, a general ranking of the metaheuristics can be proposed by following the Borda scores. Such a ranking would describe the average performance of the algorithms.

In Table 5, the first column presents the version of Hyper-ACO; the second column indicates the number of objective functions; the third, fourth, and fifth columns indicate the Borda score obtained by each algorithm taking the 18 problems.

The following points may summarize the information presented in Table 5:

- HyperACO got the best Borda scores when $\prec_{\mathscr{W}}$ is considered. This insight is in line with the remarks raised in Table 4.
- The Borda scores of HyperACO correlate with the number of objectives. HyperACO became better positioned as m increased. This correlation is particularly strong considering $\prec_{\mathscr{E}}$ and $\prec_{\mathscr{C}}$.
- For a given real-world problem—where the properties of the Pareto frontier are unknown—we suggest using HyperACO to compose an *ad hoc* MOEA to address that problem. HyperACO always got the best Borda scores regardless of the number of objective functions and the distance-based relation.

6. Conclusions and directions for future research

This paper has introduced HyperACO, a hyper-heuristic using ACO as the high-level heuristic. The low-level heuristics are taken from the evolutionary algorithms proposed by Zhang and Li [3] and Fernández et al. [15], named MOEA/D and MOEA/D/O, respectively. MOEA/D is an *a posteriori* algorithm based on decomposition, which is quite popular and is still being studied, extended, and applied. MOEA/D/O is indeed a recent extension of MOEA/D, which incorporates the DM's preferences through the interval outranking approach, becoming an *a priori* version of MOEA/D. Fernández et al. [15] explored six preference relations to increase the selective pressure towards the RoI, giving rise to six variants of MOEA/D/O. However, the performance of these variants is sensitive to the properties of the problem, the number of objective functions, and the parameters of the preference model. In these circumstances, the DM should conduct previous exhaustive experimentation to determine the best variant to treat a given problem. HyperACO mitigates this drawback by sequencing the seven low-level heuristics to compose *ad hoc* MOEAs.

Table B2
Ranking of HyperACO, MOEA/D and the six variants of MOEA/D/O according to their closeness to the A-RoI in the WFG test suite.

Problem	*Ranking*	$\prec_{\mathscr{E}}$			$\prec_{\mathscr{C}}$			$\prec_{\mathscr{W}}$		
		$m=3$	$m=5$	$m=10$	$m=3$	$m=5$	$m=10$	$m=3$	$m=5$	$m=10$
WFG1	1st	H	H	H	0, 2, 6, H	H	H	0, 2, 6, H	H	H
	2nd	3, 5	2	1, 2, 4, 6	1, 3–5	6	2	1, 3–5	2, 6	2
	3rd	0–2, 4, 6	6	3		2	1, 6		1, 3–5	1, 6
	4th		1, 3–5	0, 5		1, 3–5	4		0	3, 4
	5th		0			0	0, 3, 5			0, 5
WFG2	1st	0–6, H	H	H	0–6, H	1–6, H	1–6, H	0–6, H	H	1–6, H
	2nd		0	4		0	0		2–5	0
	3rd		6	1, 3					1, 6	
	4th		1	2, 5, 6					0	
	5th		2	0						
	6th		3–5							
WFG3	1st	H	1, 2, 4, 6, H	H	H	H	H	H	1–6, H	H
	2nd	3	0, 3, 5	2, 6	0	6	2, 6	0, 3	0	2, 6
	3rd	1, 2, 4–6		0, 1	6	1, 2, 4	0, 1	1, 2, 4–6		0, 1, 3–5
	4th	0		4	2	0, 3, 5	4			
	5th			3, 5	1		3, 5			
	6th				3–5					
WFG4	1st	1–6, H	H	H	H	H	H	H	H	H
	2nd	0	1	0	0	0	0	0	0	0
	3rd		6	1, 6	1–6	1, 6	1, 6	1–6	1, 6	1, 6
	4th		2–5	2–5		2–5	2–5		2–5	4
	5th		0							2, 3, 5
WFG5	1st	H	H	H	H	6, H	H	H	H	H
	2nd	0, 2	6	0, 2, 6	0	0–2, 4	0, 2, 6	5	1, 2, 4, 6	0, 2, 6
	3rd	6	0–2, 4	1	2	3, 5	1	1–4, 6	0, 3, 5	1, 3–5
	4th	1, 3–5	3, 5	3–5	1, 3–6		3–5	0		
WFG6	1st	H	H	H	H	H	H	H	1–6, H	H
	2nd	3, 5	3, 5	3, 5	0	0	5	1, 4	0	5
	3rd	1, 2, 4, 6	4, 1	1, 2, 4	1, 4	6	3	0		1, 3, 4
	4th	0	2, 6	0, 6	2, 3, 5, 6	2	1, 2, 4, 6	2, 3, 5, 6		2, 6
	5th		0			1, 3–5	0			0
	6th									
WFG7	1st	0–6, H	0–6, H	0–6, H	0, H	0, H	0–6, H	0, H	1–6, H	H
	2nd				1–6	1–6		6	0	0–6
	3rd							1–5		
WFG8	1st	H	H	H	H	H	H	1–6, H	H	H
	2nd	5	2, 3	2, 6	0	0	0	0	2	0
	3rd	3	1, 4–6	0	2	1–6	2, 6		0, 3, 5	2, 6
	4th	1, 2, 4, 6	0	1, 3–5	1, 6		1, 3–5		1, 4, 6	1, 3–5
	5th	0			3–5					
WFG9	1st	H	H	H	H	H	H	H	H	H
	2nd	0, 1, 4	1	1, 3, 5	1, 4	0, 2, 6	1, 3–5	1, 4	2, 4, 6	1, 3–5
	3rd	6	3–5	2, 4, 6	0	1, 4	2, 6	0	1, 3, 5	2, 6
	4th	2, 3, 5	2, 6	0	6	3, 5	0	6, 2	0	0
	5th		0		2			3, 5		
	6th				3, 5					

As far as we know, there is no single way of embedding a preference model in MOEAs that always leads to higher performance, even within the framework of the outranking approach alone. This fact emphasizes the contribution of this paper. Our proposed hyper-heuristic is appealing to address real-world problems because it can offer competitive MOEAs especially designed to treat a particular problem without the need to conduct formal analyses on its properties (e.g., the geometry of the Pareto frontier). Additionally, those analyses are often challenging and time-consuming.

We developed three versions of HyperACO by using the Euclidean distance, the Chebyshev distance, and a weighted distance, each of which reflects a different perspective of the DM about the closeness to the RoI. According to the numeric results, HyperACO regularly composed MOEAs providing solutions with the best approximation to the RoI compared with both *a priori* MOEAs and *a posteriori* MOEAs. This conclusion is supported by statical tests for significance on the results obtained by addressing the DTLZ and WFG test suites with ten synthetic DMs and three different numbers of objectives. According to these results, the advantages of HyperACO became more pronounced as the number of objectives increased.

A promising direction for future research is to develop a strategy for tuning the control parameters in an adaptive way, considering a global setting involving the parameters of both the high-level heuristic and the low-level heuristics combinedly. Additionally, we are going to conduct further experimentation to validate the contribution of this approach in more challenging benchmarks; expressly, the LZ/UF and MaOP test suites [43,44].

It is important to note that HyperACO was designed to address strategic decision problems, where even a slight improvement is significant for organizations, and they would be prepared to spend the computing time needed for such an improvement (for operational decision problems, where the decision maker is often limited in time, we suggest using any of the state-of-the-art algorithms). Fortunately, several programming techniques can make HyperACO much faster, especially parallelization, cloud computing, and distributed computing, which would be another direction for future research.

Lastly, as the low-level heuristics are based on outranking, HyperACO demands that the DM's preferences be non-compensatory about the criteria of the underlying problem (that the low-level heuristics address). Although this is the scope of our current proposal, HyperACO is not necessarily limited to outranking. A proper direction for future research is adding low-level heuristics considering other multi-criteria preference models (e.g., value functions, distances to the ideal solution). It is only natural then that, as future research, we are going to study the identification of the RoI under a broader range of preference models. A hyper-heuristic with such a breakthrough could become the most comprehensive optimization method to treat real-world MaOPs with *a priori* preference incorporation.

CRediT authorship contribution statement

Gilberto Rivera: Methodology, Software, Writing – original draft, Writing – review & editing. **Laura Cruz-Reyes:** Project administration, Writing – original draft, Conceptualization. **Eduardo Fernandez:** Supervision, Conceptualization, Formal analysis. **Claudia Gomez-Santillan:** Writing – review & editing, Investigation, Validation. **Nelson Rangel-Valdez:** Validation, Resources, Project administration. **Carlos A. Coello Coello:** Formal analysis, Investigation, Resources.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

Data will be made available on request.

Acknowledgments

Carlos A. Coello Coello gratefully acknowledges support from CONACyT grant no. 2016-01-1920 (Investigación en Fronteras de la Ciencia 2016) and from a SEP-Cinvestav grant (proposal no. 4). Also, the Basque Government partially supported him through the BERC 2022-2025 program and the Spanish Ministry of Economy and Competitiveness MINECO: BCAM Severo Ochoa excellence accreditation SEV-2017-0718.

Appendix A

Table A.1 develops a numeric example of the calculation of $\sigma(x, y)$ through IOA. Table A.1 extends Table 1 to give the reader a closer look at the function; these extensions are shadowed in gray.

Appendix B

This appendix presents the results of HyperACO compared to MOEA/D and the six variants of MOEA/D/O. For every single run, the performance of the algorithms is evaluated through the four indicators described in Subsection 5.3. The order relations $\prec_{\mathscr{E}}$, $\prec_{\mathscr{C}}$ and $\prec_{\mathscr{W}}$ are used to compare the results of the algorithms. Then, the Friedman non-parametric test is complemented with Nemenyi Post-hoc analysis (both with a 0.95-confidence interval) to determine significant differences and set an order among the results. Tables B.1 and B.2 structure these results. Here, the first column indicates the problem addressed, and the second column refers to the ranking obtained by the statistical tests. Then, the results are grouped by order relation (Columns 3–5 for $\prec_{\mathscr{E}}$, Columns 6–8 for $\prec_{\mathscr{C}}$, and Columns 9–11 for $\prec_{\mathscr{W}}$) and presented incrementally by number of objectives (m). The numbers shown in Tables B.1 ad B.2 indicate the variant of MOEA/D/O (See Table 2); additionally, “0” means the original MOEA/D [3], and “H” stands for HyperACO. As can be observed, Table 3 is indeed a summary of Tables B.1 and B.2.

References

[1] S. Bechikh, M. Elarbi, L. Ben Said, Many-objective Optimization Using Evolutionary Algorithms: A Survey. Recent Advances Evolutionary Multiobjective Optimization, Springer, Cham, 2017, pp. 105–137, https://doi.org/10.1007/978-3-319-42978-6_4.

[2] M.F. Leung, C.A. Coello Coello, C.C. Cheung, S.C. Ng, A.K. Lui, A hybrid leader selection strategy for many-objective particle swarm optimization, IEEE Access 8 (2020) 189527–189545, https://doi.org/10.1109/ACCESS.2020.3031002.

[3] Q. Zhang, H. Li, MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition, IEEE Transactions on Evolutionary Computation 11 (6) (2007) 712–731, https://doi.org/10.1109/TEVC.2007.8927590.

[4] A. Ramirez, J.R. Romero, S. Ventura, A survey of many-objective optimisation in search-based software engineering, Journal of Systems and Software 149 (2019) 382–395, https://doi.org/10.1016/j.jss.2018.12.015.

[5] J.G. Falcón-Cardona, C.A. Coello Coello, A new indicator-based many-objective ant colony optimizer for continuous search spaces, Swarm Intelligence 11 (1) (2017) 71–100, https://doi.org/10.1007/s11721-017-0133-x.

[6] N. Benabbou, C. Leroy, T. Lust, An interactive regret-based genetic algorithm for solving multi-objective combinatorial optimization problems, in: Proceedings of the AAAI Conference on Artificial Intelligence 34, 2020, pp. 2335–2342, https://doi.org/10.1609/aaai.v34i03.5612.

[7] J. Branke, S. Corrente, S. Greco, R. Słowinski, P. Zielniewicz, Using Choquet integral as preference model in interactive evolutionary multi-objective optimization, European Journal of Operational Research 250 (3) (2016) 884–901, https://doi.org/10.1016/j.ejor.2015.10.027.

[8] K. Li, M. Liao, K. Deb, G. Min, X. Yao, Does Preference Always Help? A Holistic Study on Preference-Based Evolutionary Multi-Objective Optimisation Using Reference Points, IEEE Transactions on Evolutionary Computation 124 (6) (2020) 1078–10961, https://doi.org/10.1109/tevc.2020.2987559.

[9] S. Bechikh, M. Kessentini, L.B. Said, K. Ghédira, Preference incorporation in evolutionary multiobjective optimization: a survey of the state-of-the-art, in: Advances in Computers, 98, Elsevier, 2015, pp. 141–207, https://doi.org/10.1016/bs.adcom.2015.03.001.

[10] R. Cheng, T. Rodemann, M. Fischer, M. Olhofer, Y. Jin, Evolutionary Many-Objective Optimization of Hybrid Electric Vehicle Control: From General Optimization to Preference Articulation, IEEE Transactions on Emerging Topics in Computational Intelligence 1 (2) (2017) 97–111, https://doi.org/10.1109/TETCI.2017.2669104.

[11] L. Cruz-Reyes, E. Fernandez, J.P. Sanchez-Solis, C.A. Coello Coello, C. Gomez, Hybrid evolutionary multi-objective optimisation using outranking-based ordinal classification methods, Swarm and Evolutionary Computation 54 (2020), 100652, https://doi.org/10.1016/j.swevo.2020.100652.

[12] B. Roy, The Outranking Approach and the Foundations of Electre Methods, in: Readings Multiple Criteria Decision Aid, Springer Berlin Heidelberg, Berlin, Heidelberg, 1990, pp. 155–183, https://doi.org/10.1007/978-3-642-75935-2_8.

[13] E. Fernández, J. Figueira, J. Navarro, An interval extension of the outranking approach and its application to multiple-criteria ordinal classification, Omega 84 (2019) 189–198, https://doi.org/10.1016/j.omega.2018.05.003.

[14] E. Fernández, J. Navarro, E. Solares, A hierarchical interval outranking approach with interacting criteria, European Journal of Operational Research 298 (1) (2022) 293–307, https://doi.org/10.1016/j.ejor.2021.06.065.

[15] E. Fernández, N. Rangel-Valdez, L. Cruz-Reyes, C.G. Gomez-Santillan, C.A. Coello-Coello, Preference incorporation in MOEA/D using an outranking approach with imprecise model parameters, Swarm and Evolutionary Computation (2022), 101097, https://doi.org/10.1016/j.swevo.2022.101097.

[16] G. Rivera, C.A. Coello Coello, L. Cruz-Reyes, E. Fernandez, C. Gomez-Santillan, N. Rangel-Valdez, Preference Incorporation into Many-Objective Optimization: An Ant Colony Algorithm based on Interval *Outranking*, Swarm and Evolutionary Computation 69 (2022), 101024, https://doi.org/10.1016/j.swevo.2021.101024.

[17] J.H. Drakea, A. Kheiri, R. Özcanc, E.K. Burkea, Recent advances in selection hyper-heuristics, European Journal of Operational Research 285 (2) (2020) 405–428, https://doi.org/10.1016/j.ejor.2019.07.073.

[18] E. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, G Özcan, R. Qu, Hyper-heuristics: A survey of the state of the art, Journal of the Operation Research Society 64 (12) (2013) 1695–1724, https://doi.org/10.1057/jors.2013.71.

[19] G. Rivera, C. Gómez, L. Cruz, R. García, F.A. Balderas, E. Fernández, F. López, Solution to the social portfolio problem by evolutionary algorithms, International Journal of Combinatorial Optimization Problems and Informatics 3 (2) (2012) 21–30.

[20] E. Fernandez, E. Lopez, F. Lopez, C.A. Coello Coello, Increasing selective pressure towards the best compromise in evolutionary multiobjective optimization: The

extended NOSGA method, Information Sciences 181 (1) (2011) 44–56, https://doi.org/10.1016/j.ins.2010.09.007.

[21] R. Raghavjee, N. Pillay, A genetic algorithm selection perturbative hyper-heuristic for solving the school timetabling problem, ORiON 31 (1) (2015) 39–60.

[22] A. Muklason, A.J. Parkes, E. Özcan, B. McCollum, P. McMullan, Fairness in examination timetabling: Student preferences and extended formulations, Applied Soft Computing 55 (2017) 302–318, https://doi.org/10.1016/j.asoc.2017.01.026.

[23] H.L. Jakubovski-Filho, T.N. Ferreira, S.R. Vergilio, Incorporating User Preferences in a Software Product Line Testing Hyper-Heuristic Approach, in: 2018 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2018, pp. 1–8, https://doi.org/10.1109/CEC.2018.8477803.

[24] H.L. Jakubovski-Filho, T.N. Ferreira, S.R. Vergilio, Multiple objective test set selection for software product line testing: evaluating different preference-based algorithms, in: Proceedings of the XXXII Brazilian Symposium on Software Engineering, ACM, 2018, pp. 162–171, https://doi.org/10.1145/3266237.3266275.

[25] T. Macias-Escobar, L. Cruz-Reyes, H. Fraire, B. Dorronsoro, Plane Separation: A method to solve dynamic multi-objective optimization problems with incorporated preferences, Future Generation Computer Systems 110 (2020) 864–875, https://doi.org/10.1016/j.future.2019.10.039.

[26] R.E. Moore, Methods and applications of interval analysis. Studies in Applied and Numerical Mathematics, Society for Industrial and Applied Mathematics, Philadelphia, USA, 1979, pp. 181–190. https://doi.org/10.1137/1.9781611970906.bm.

[27] F. Balderas, E. Fernandez, C. Gomez, N. Rangel, L. Cruz-Reyes, An interval-based approach for evolutionary multi-objective optimisation of project portfolios, International Journal of Information Technology & Decision Making 18 (4) (2019) 1317–1358, https://doi.org/10.1142/S021962201950024X.

[28] J.R. Shi, S.Y. Liu, W.T. Xiong, A new solution for interval number linear programming, Journal of Systems Engineering - Theory and Practice 2 (16) (2005) 101–106, https://doi.org/10.12011/1000-6788(2005)2-101.

[29] S. Yao, Z. Jiang, N. Li, H. Zhang, N. Geng, A multi-objective dynamic scheduling approach using multiple attribute decision making in semiconductor manufacturing, International Journal of Production Economics 130 (1) (2011) 125–133, https://doi.org/10.1016/j.ijpe.2010.12.014.

[30] T. Fliedner, J Liesiö, Adjustable robustness for multi-attribute project portfolio selection, European Journal of Operational Research 252 (3) (2016) 931–946, https://doi.org/10.1016/j.ejor.2016.01.058.

[31] E. Fernández, J.R. Figueira, J. Navarro, Interval-based extensions of two outranking methods for multi-criteria ordinal classification, Omega 95 (2020), 102065, https://doi.org/10.1016/j.omega.2019.05.001.

[32] C. Calafat-Marzal, Á. Gallego-Salguero, M. Segura, S. Calvet-Sanz, GIS-Based and Outranking Approach to Assess Suitable Pig Farming Areas in the Mediterranean Region: Valencian Community, Animals 11 (4) (2021) 1151, https://doi.org/10.3390/ani11041151.

[33] M. Doumpos, J.R. Figueira, A multicriteria outranking approach for modeling corporate credit ratings: An application of the Electre Tri-nC method, Omega 82 (2019) 166–180, https://doi.org/10.1016/j.omega.2018.01.003.

[34] Y. Liang, Y. Ju, L. Martínez, Y. Tu, Sustainable battery supplier evaluation of new energy vehicles using a distributed linguistic outranking method considering bounded rational behavior, Journal of Energy Storage 48 (2022), 103901, https://doi.org/10.1016/j.est.2021.103901.

[35] J. Pereira, E.C. de Oliveira, D.C. Morais, A.P. Costa, P. Arroyo-Lopez, ELECTRE TRI-C with hesitant outranking functions: Application to supplier development, Journal of Intelligent & Fuzzy Systems 37 (6) (2019) 7923–7933, https://doi.org/10.3233/JIFS-190166.

[36] G. Rivera, R. Florencia, M. Guerrero, R. Porras, J.P. Sánchez-Solís, Online multi-criteria portfolio analysis through compromise programming models built on the underlying principles of fuzzy outranking, Information Sciences 580 (2021) 734–755, https://doi.org/10.1016/j.ins.2021.08.087.

[37] G. Rivera, R. Porras, J.P. Sanchez-Solis, R. Florencia, V. García, Outranking-based multi-objective PSO for scheduling unrelated parallel machines with a freight industry-oriented application, Engineering Applications of Artificial Intelligence 108 (2022), 104556, https://doi.org/10.1016/j.engappai.2021.104556.

[38] M. Spiliotis, L. Panagiotou, I. Kagalou, D. Latinopoulos, A fuzzified multicriteria outranking method for water framework directive implementation in a heavily modified urban lake (Pamvotis, Greece), Water Resources Management 34 (14) (2020) 4491–4510, https://doi.org/10.1007/s11269-020-02624-2.

[39] Q. Liu, Y. Jin, M. Heiderich, T. Rodemann, G. Yu, An adaptive reference vector-guided evolutionary algorithm using growing neural gas for many-objective optimization of irregular problems, IEEE Transactions on Cybernetics 52 (5) (2022) 2698–2711, https://doi.org/10.1109/TCYB.2020.3020630.

[40] Y. Tian, R. Cheng, X. Zhang, F. Cheng, Y. Jin, An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility, IEEE Transactions on Evolutionary Computation 22 (4) (2018) 609–622, https://doi.org/10.1109/TEVC.2017.2749619.

[41] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable test problems for evolutionary multiobjective optimization. Evolutionary Multiobjective Optimization, Springer, London, 2002, pp. 105–145, https://doi.org/10.1109/CEC.2002.1007032.

[42] S. Huband, P. Hingston, L. Barone, L. While, A review of multiobjective test problems and a scalable test problem toolkit, IEEE Transactions on Evolutionary Computation 10 (5) (2006) 477–506, https://doi.org/10.1109/TEVC.2005.861417.

[43] H Li, Q. Zhang, Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II, IEEE Transactions on Evolutionary Compututation 12 (2) (2009) 284–302, https://doi.org/10.1109/TEVC.2008.925798.

[44] H. Li, K. Deb, Q. Zhang, P.N. Suganthan, L. Chen, Comparison between MOEA/D and NSGA-III on a set of novel many and multi-objective benchmark problems with challenging difficulties, Swarm and Evolutionary Computation 46 (2019) 104–117, https://doi.org/10.1016/j.swevo.2019.02.003.

[45] A. Castellanos, L. Cruz-Reyes, E. Fernández, G. Rivera, C. Gomez-Santillan, N. Rangel-Valdez, Hybridisation of Swarm Intelligence Algorithms with Multi-Criteria Ordinal Classification: A Strategy to Address Many-Objective Optimisation, Mathematics 10 (3) (2022) 322, https://doi.org/10.3390/math10030322.

[46] A. Castellanos, L. Cruz-Reyes, E. Fernandez, N. Rangel-Valdez, C. Gómez-Santillán, H. Fraire, J.A. Brambila-Hernández, A Method for Integration of Preferences to a Multi-Objective Evolutionary Algorithm Using Ordinal Multi-Criteria Classification, Mathematical and Computational Applications 26 (2) (2021) 27, https://doi.org/10.3390/mca26020027.

[47] I. Rodríguez-Fdez, A. Canosa, M. Mucientes, A. Bugarín, STAC: a web platform for the comparison of algorithms using statistical tests, in: 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2015, pp. 1–8, https://doi.org/10.1109/FUZZ-IEEE.2015.7337889.

[48] S. Zapotecas-Martínez, C.A.C. Coello, H.E. Aguirre, K. Tanaka, A review of features and limitations of existing scalable multiobjective test suites, IEEE Transactions on Evolutionary Computation 23 (1) (2019) 130–142, https://doi.org/10.1109/TEVC.2018.2836912.

[49] I.R. Meneghini, M.A. Alves, A. Gaspar-Cunha, F.G. Guimaraes, Scalable and customizable benchmark problems for many-objective optimization, Applied Soft Computing 90 (2020), 106139, https://doi.org/10.1016/j.asoc.2020.106139.