

A Comprehensive Review on the Classification of Multi-Objective Optimization Techniques

Mohamd Hassan Gadallah, Abdelrahman Ali M. Ahmed*

Department of Mechanical Design and Production Engineering, Faculty of Engineering, Cairo University, Giza, Egypt
Email: *abdel.ahmed.a@eng-st.cu.edu.eg, mohamed@aucegypt.edu

How to cite this paper: Gadallah, M.H. and Ahmed, A.A.M. (2025) A Comprehensive Review on the Classification of Multi-Objective Optimization Techniques. *Open Journal of Optimization*, 14, 181-228.
<https://doi.org/10.4236/ojop.2025.144010>

Received: September 25, 2025

Accepted: November 28, 2025

Published: December 1, 2025

Copyright © 2025 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Multi-objective optimization remains a significant and realistic problem in engineering. A trade-off among conflicting objectives subject to equality and inequality constraints is known as the Pareto Front. The size problem is common to multi-objective optimization. Recently, several multi-objective optimization techniques have been developed; each technique has its own and different characteristics rather than others. This involves search techniques, selection methods, preferences made by the decision makers, an optimization framework, fitness evaluation, mutation or crossover methods, the nature of the problem and constraints, and the degree of complexity. Moreover, classification of problems into static and dynamic further attempts to balance some criteria such as diversity, coverage, and convergence. This paper provides an elaboration on all multi-objective optimization techniques, and shows the drawbacks addressed in the literature, which will help researchers' understanding of the various formulations in the field.

Keywords

Hybrid, Heuristic-Based Methods, Swarm-Based Methods, Prior Preference-Based Methods, Posterior Preference-Based Methods, Interactive Methods

1. Introduction

Multi-objective optimization results in a tradeoff between local and global optimal solutions, influenced by numerous attractors within the solution space. This complexity complicates the task for optimization algorithms [1], particularly in scenarios where the importance of objectives varies among decision makers. In

multi-objective optimization, multiple conflicting goals make it impossible to identify a single best solution without knowing the decision makers' preferences, as all trade-off solutions are equally optimal and important [2] [3].

In optimization problems, decision-makers usually assign preferences to reflect the importance of different objectives using various techniques (a priori, a posteriori, interactive, or no-preference). However, in multi-criteria optimization involving qualitative aspects like aesthetics—where preferences are unclear—resulting trade-off solutions may seem unsatisfactory and hinder creativity or design distinctiveness [4]. In scenarios where some objectives are more important than others, prioritizing these objectives is essential. Lexicographic Goal Programming is a common method used to solve such problems by addressing goals in order of their importance.

It ranks the importance of objectives using strategic planning by breaking down objectives into several levels of proactive priorities [5]. Hence, optimization formulations can be categorized into three different categories: no preferences, preference-based (weighing methods), and Prioritization, which utilizes a prioritizing strategy to assign full relative priority to the higher-level ranked objective; afterwards the next rank's objective, The goal of the multi-decision-maker needs prioritizing problem is to determine the optimal ranking that considers the ratings of each decision-maker separately. Decision-makers must first determine alternative global rankings based on their opinions and then choose the final ranking using a contextual and strategic data-driven process [6].

It is thought that an optimization method may be appropriate for one problem but not for another; in other words, the ability of an optimization method to solve an optimization problem and generate high-quality tradeoff solutions depends on a variety of factors, including search strategy, decision-maker preferences, optimizer framework, selection process, fitness estimation method, degree of complexity, and the point or phase at which a human interacts to enter his preferences [3], in the following sections classification of Multi objective optimization methods will be classified with respect to the aforementioned aspects.

2. Multi-Objective Search Techniques for Optimization

Figure 1 demonstrates the three main categories into which optimization search methods can be generally separated: classical, meta-heuristic, and enumerative. The two primary types of classical algorithms are gradient-based and direct search, which are both known for being accurate methods that guarantee the best possible result. These serve as the cornerstone for the creation of meta-heuristic methods. However, these techniques have a limited scope in practical applications [7]-[9]. Enumerative searches are deterministic, but they are distinguished here because they don't use heuristics. It is the simplest one; it systematically explores the entire solution space. They take into account every potential answer within a predetermined search space. In which the solution is evaluated, however, it is easily seen that this technique is inefficient or even infeasible as search spaces become large

and it takes a longer time for large problems [7] [8].

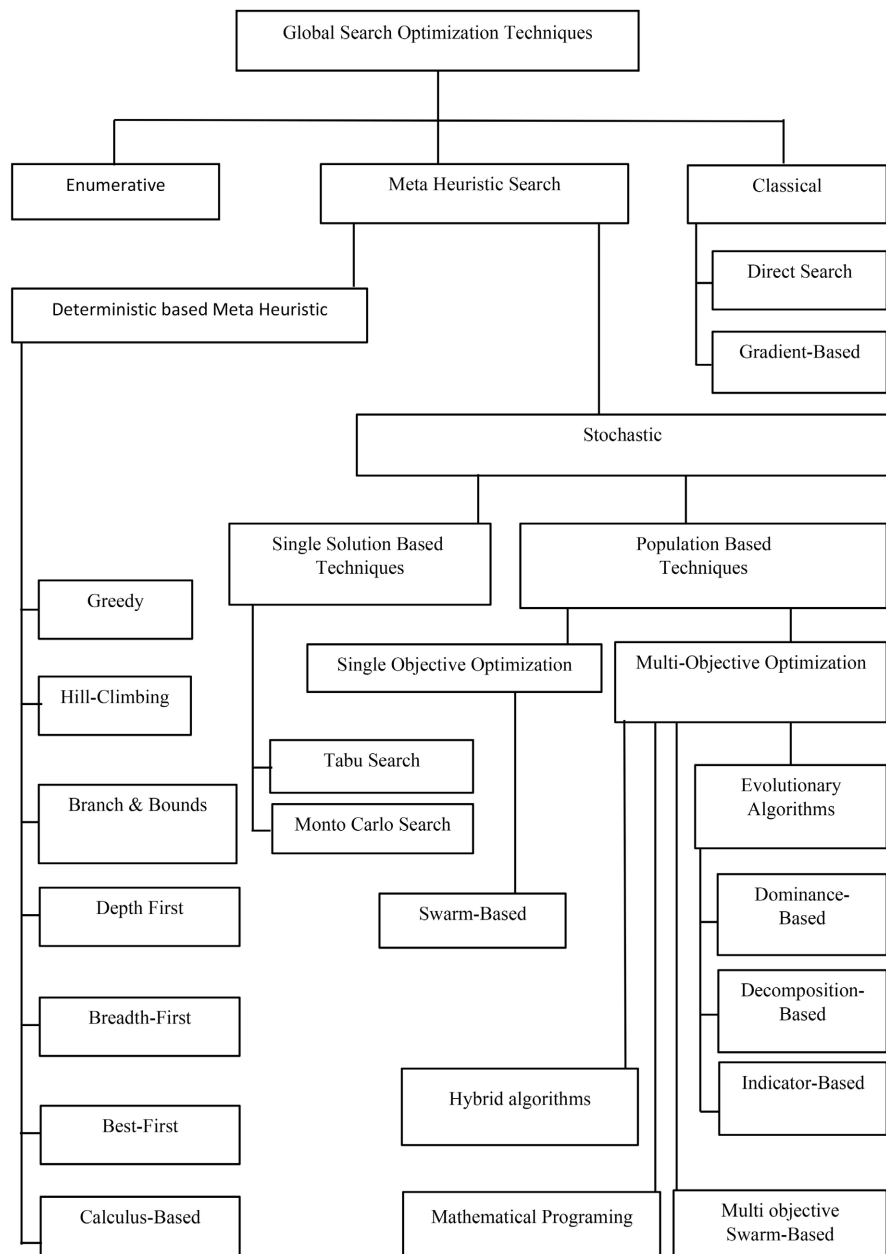


Figure 1. Multi-objective optimization search techniques.

Meta-heuristic methods are best for dealing with real-world problems in several fields; it can serve in engineering, management, politics, and others, and can be applied for discontinuous, large-dimensional, multi-modal problems and can even be used to solve black box optimization problems without understanding the problem physics. An obvious drawback back that it is unable to provide an exact result. It can be categorized into probabilistic or stochastic and deterministic-based approaches. Metaheuristic stochastic algorithms use the Monte Carlo method or random number generation for initialization, creating uniform pseudo-ran-

dom sequences based on the uniform probability distribution [10]. Deterministic meta heuristics address optimization problems by applying preset judgments, guaranteeing that the result stays constant [9]. Deterministic meta-heuristic-based search is simple and comprehensible, while the stochastic approach is complex and incomprehensible [11].

Greedy, Hill climbing, Bound and Branch search technique, Branch & Bounds [7]; depth-first search [12] [13], breadth-first search [13], Best-first search [12], and calculus-based search technique [13] are well-known techniques under the Deterministic-based Meta Heuristic family. These methods vary in their approach and suitability depending on the problem at hand, whether it involves searching through discrete states (DFS, BFS, Best-First) or optimizing continuous functions (Calculus-Based). Every technique has advantages, and the selection of a particular strategy is dependent on the specific needs and features of the optimization problem [14].

The hill-climbing approach, widely used among human problem solvers, focuses on local optimization. It derives its name from climbers aiming to reach the mountain peak swiftly by choosing the steepest ascent path from their current location. Similarly, when adjusting the knobs of a television to achieve a better image, each knob is turned sequentially, maximizing progress toward the desired quality before moving to the next. This method hinges on continuously evaluating image quality relative to a benchmark and making incremental improvements based on local data, ensuring gradual enhancements in the most promising direction [14].

Hill-climbing is popular due to its low computational cost; it doesn't require remembering past steps. It works by always choosing the most promising local option. However, this simplicity has downsides: it can get stuck at local maxima, follow misleading paths, or revisit fewer promising options. Since it doesn't backtrack, it's called an irrevocable method. Hill-climbing performs better when the evaluation function is accurate and when operators are commutative, meaning their order doesn't affect outcomes, ensuring the solution can still be reached, even if delayed. This makes it useful for problems like the least spanning tree, though designing the right operators can be challenging [14].

Branch and Bound is a systematic search technique for solving optimization problems by partitioning the solution space and pruning regions that cannot yield better solutions than the best one found so far. It is used when either exhaustive search is too expensive or greedy/local methods are not able to discover the global optimum [14].

Branching the problem and dividing it into smaller subproblems (branches), and bounding for each subproblem, a bound (the best possible solution in that branch) is computed. If this bound is worse than the best-known solution, we can prune (discard) that branch. Pruning discards subproblems that can't result in a better solution. Search Tree: Problems are represented as nodes in a tree. The root node is the original problem; children are subproblems [14].

Depth-First Search (DFS) investigates each branch as far as feasible before backtracking. Commonly used in graph traversal problems to explore as deeply as much as feasible along each path before backtracking. It is advanced in memory efficiency as it explores deeply, suitable for problems with a large state space [14].

Breadth-First Search (BFS) explores all neighbor nodes at the present depth level before moving on to nodes at the next depth level. Effective for finding the graph's unweighted shortest path, or when all paths need to be explored systematically. It is advanced in Guarantees the quickest route in terms of number of edges [14].

Best-First Search chooses the node with the most promise according to some heuristic, typically the estimated distance from the goal. Widely used in informed search algorithms like A*, where the heuristic guides the search towards the goal efficiently. Advances in efficiency in finding a solution quickly based on the Calculus-Based Optimization, which uses mathematical calculus (derivatives, gradients) to optimize continuous functions. Applied in a number of fields, such as engineering, economics, and physics, to find optimal solutions to problems described by mathematical functions. Provides exact solutions when the objective and constraint functions are differentiable, allowing for precise optimization in continuous domains [14].

Stochastic search is divided into two categories: population-based methods and single-solution methods. In the single solution the work only to modify a single solution, until reaching a defined criterion [15]. In contrast, population-based methods alter and modify a set of solutions during optimization. However, throughout the process of optimization, a collection of solutions is altered and transformed using population-based strategies [16]. Local search, Tabu search, etc., are some examples of stochastic single-solution-based methods [17].

One of the significant issues that faces the meta heuristic optimizers in stochastic search is highly sensitive to the initial population's characteristics. Population-based metaheuristic algorithms have different capacities to arrive at a global optimum when the initialization method is altered [15]. In this search technique, it is possible for the same starting solutions to lead to various final answers when using probabilistic approaches since they employ random rules throughout search. In a short length of time, these strategies produce solutions of excellent quality [16].

Single-objective-based techniques can only produce the best results for one competing objective at a time and have a single objective function and a unique optimal solution. Multi-objective methods deal simultaneously with several objectives and offer Pareto-optimal solutions [15] [18]. Evolutionary and swarm-based approaches, mathematical programming, and hybrid algorithms are the four primary groups into which multi-objective techniques fall. Hybrid algorithms combine the concepts of swarm and evolutionary algorithms or others [15]. The integration of different methods allows the algorithm to improve each component and mitigate its weaknesses. Here are a few ways in which multi-objective algorithms can be hybridized [19] [20]:

- Integrating Local Search with Evolutionary Algorithms: Multi-objective algorithms like the Strength Pareto Evolutionary Algorithm 2 (SPEA2) and the Non-dominated Sorting Genetic Algorithm II (NSGA-II) can be enhanced by incorporating local search techniques. Local search refines solutions in the vicinity of the current population, improving both the convergence to the Pareto front and the diversity of the solutions.
- Hybrid algorithms can incorporate machine learning methods to enhance decision-making and guide the search process. For instance, reinforcement learning or surrogate modeling can predict the performance of candidate solutions, enabling us to concentrate on the most promising regions inside the search space.
- Integrating Swarm Intelligence with Evolutionary Algorithms: Combining multi-objective algorithms with swarm intelligence methods, such as Particle Swarm Optimization (PSO), results in hybrid approaches like Multi-Objective Particle Swarm Optimization (MOPSO). This integration enhances the efficient exploration of the area under search.
- Hybrid Algorithms: These approaches combine evolutionary algorithms with mathematical programming methods, such as linear or integer programming, to enhance their ability to address specific constraints or objectives effectively.

Evolutionary algorithms are techniques inspired by the principles of natural evolution. They allow for the generation of a group of trade-off solutions in a single run while requiring significantly less computational power to identify these solutions [15]. Convergence, variety or diversity, and coverage are the fundamental objectives [19]. Evolutionary algorithms are distinguished in three main categories: decomposition-based, indicator-based, and dominance-based [21]. The Pareto dominance principle is used to determine the fitness of solutions in dominance-based algorithms.

The Normal Boundary Intersection (NBI) is a well-known example of a posteriori technique based on mathematical programming [22], Modified Normal Boundary Intersection (NBIm) [23], Normal Constraint (NC) [24], and Successive Pareto Optimization (SPO) [25]. And Directed Search Domain (DSD) [26], which builds many scalarizations to tackle the multi-objective optimization issue. A Pareto optimum solution, either locally or globally, is produced by the answer to each scalarization. In **Figure 4**, the NBI, NBIm, NC, and DSD scalarizations are designed to produce uniformly distributed Pareto points that provide a decent approximation of the actual Pareto point set.

Swarm-based techniques leverage the intelligent behavior of self-organizing and decentralized systems inspired by biological phenomena. These algorithms are well-suited for solving both single-objective and multi-objective optimization problems. One of the most renowned swarm-based approaches for multi-objective optimization is Multi-Objective Particle Swarm Optimization (MOPSO) [27], Multi-Objective Ant Colony Optimization (MOACO) [28], and Multi-Objective Marine Predator Algorithm (MMPOA) [29]. Multi-objective Whale Optimization Algorithm (WOA) [30], and Multi-objective Spotted Hyena Optimization (MOSHO)

[31]. These techniques successfully minimize the objective function without becoming trapped in local minima (Figure 2).

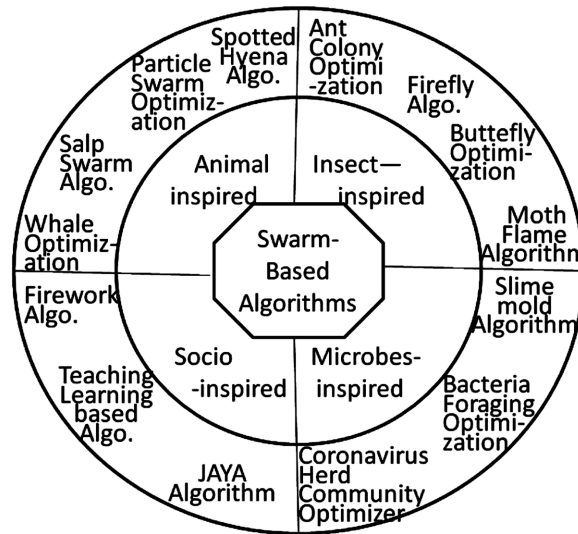


Figure 2. Swarm-based optimization algorithms.

Swarm-based approaches rely on influencing individual behavior by local or global individuals on others, similar to the crossover operator in evolutionary algorithms. They incorporate potential solutions traveling via hyperspace, allowing individuals to benefit from past experiences. One of the most significant accomplishments of PSO is its effectiveness in solving both discrete binary and continuous nonlinear optimization problems [32].

Through interactions at the local level, the self-organizing characteristic of a swarm system facilitates the evolution of solutions at the global level.

3. Classification of Multi-Objective Optimization Based on Preferences

There are three main approaches to optimization with multiple objectives, depending on when the decision maker interacts at which provides additional preference information [4] [10] [11] [33] [34].

- A priori method requires the decision-maker to provide additional preferences, such as objective weights, before the optimization process begins. In other words, preference information must be incorporated into the mathematical programming problem prior to initiating the solution search [10].
- The posteriori method involves the user expressing their preferences after gaining an understanding of the trade-offs among non-dominated alternatives. These approaches can be generally divided into two categories: those that employ mathematical programming techniques and those based on Evolutionary Multi-objective Optimization (EMO) algorithms. In these methods, the decision-maker evaluates all or a subset of the Pareto optimal options and selects what they consider to be the best or most suitable solution [4] [10] [34].

- In interactive approaches, the decision-maker provides preferences during each iteration, gradually guiding the process toward the final solution. The execution of this approach involves refining objective functions and constraints by incorporating user feedback on preferences at different stages of the process [4] [10] [34].

In addition to the three previously mentioned approaches, multi-objective optimization (MOO) can also produce solutions that do not rely on preference-based trade-offs. However, such solutions are less suitable for multi-criteria optimization, which often considers qualitative factors like aesthetics. This limitation can hinder one of the most vital aspects of design: originality [34]. Examples of non-preference-based methods include Gradient Descent, a commonly used optimization technique, as well as Simulated Annealing, Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Random Search. Random Search, which involves sampling points randomly from the solution space and evaluating their performance, can be surprisingly effective for certain optimization issues, particularly when the solution space is poorly understood.

3.1. Priori Preference-Based Multi-Objective Optimization Procedure

Classical single or multi-objective optimization techniques are frequently used to support the priori approach, while the posteriori method necessitates intriguing alterations to theorems and optimization algorithms. In real-world applications, interactive methods are quite fascinating, although they usually rely on computational techniques employed in a priori and a posteriori approach and combine them with intermediate phases of preference elicitation [34]. **Figure 3** shows the numerous kinds of multi-objective optimization methods based on the role of humans in the optimization process [34].

As illustrated in **Figure 4**, the NBI, NBI_m, NC, and DSD scalarization methods are formulated to generate uniformly distributed Pareto points, offering a reasonable approximation of the true Pareto front.

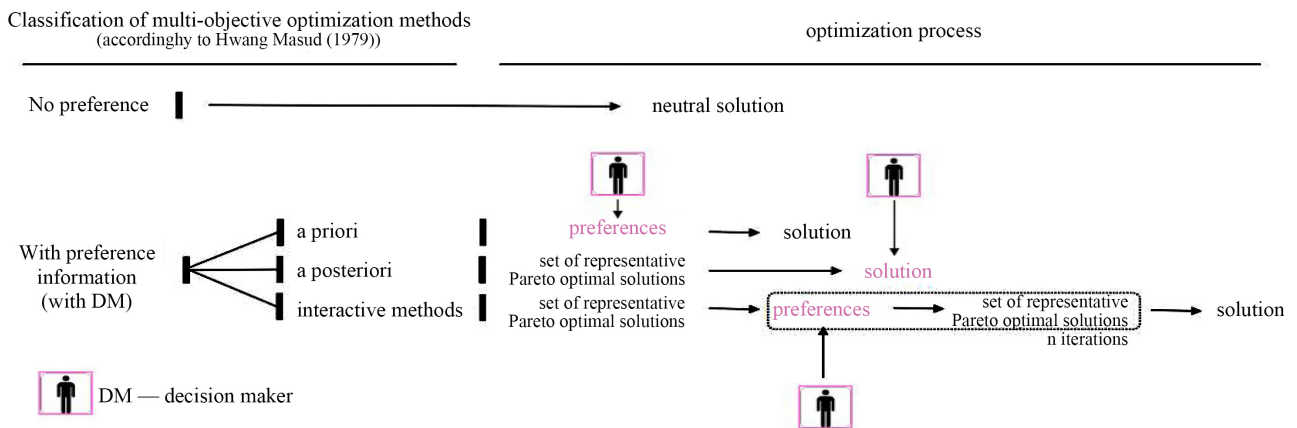


Figure 3. Grouping multi-objective optimization techniques according on how humans are involved in the optimization process.

Figure 4 gives examples of optimization methods under each search classification; the most well-known a priori approaches are the utility function method, the lexicographic method, and goal programming, and others [33].

Figure 5 illustrates schematically the a priori approach, which begins with the selection of preference vector w using higher-level information. The composite function is then created using the preference vector, and a single-objective optimization method is applied to optimize it in order to get a single solution. The method is applicable to find several trade-off solutions by using an alternative preference vector and repeating the steps [33].

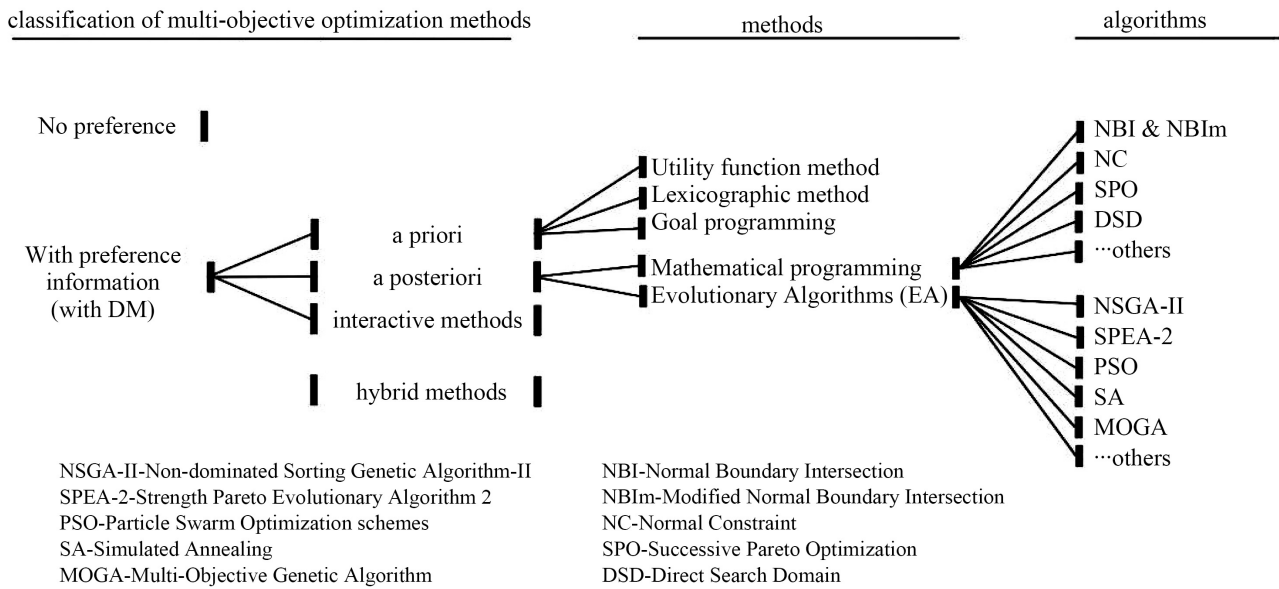


Figure 4. Goal-oriented multi-objective optimization methods and algorithms classification.

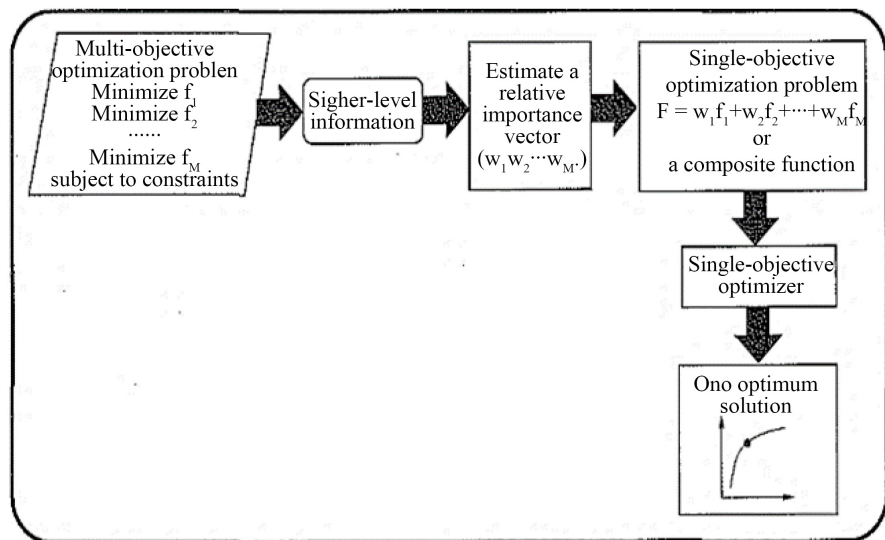


Figure 5. Schematic of a priori preference-based multi-objective optimization procedure (Classical MOO Procedure).

Tradeoff results/solutions obtained by the preference-based procedure is very sensitive to the relative preference vector used in forming the composite function. Aside from the problem, it is obvious that determining a relative preference vector is highly subjective and challenging [33].

Classical optimization mostly works according to the priori preference-based approach. Starting with the weighted sum approach, the ϵ -constraint method, Tchebycheff methods [9], value function methods and goal programming methods and other methods will be presented below [33].

As described above, since the classical procedure requires higher preference information before starting the optimizer, it could be categorized as a priori multi-objective optimization [34]. The **Weighted Sum Method** is one of the priori approaches examples; it combines multiple objectives into a single objective function by assigning a user-defined weight to each one and multiplying accordingly. This approach is the simplest and most widely used traditional technique. It's the most intuitive method for addressing multiple objectives, as it involves minimizing a weighted total. For example, if the goals are to lower production costs and reduce material waste during manufacturing, this method naturally balances the trade-offs between the two [33].

The vector of objectives function must be normalized in order to solve the issue of varying order of magnitude for distinct objective functions before being transformed into a single weighted objective function.

This strategy can be outlined as follows:

$$\text{Minimize } f(x) = \sum_{m=1}^M w_m f_m(x)$$

$$\text{Subject to } g_j(x) \geq 0, j = 1, 2, \dots, J$$

$$h_k(x) = 0, k = 1, 2, \dots, K$$

$$x_i^{(L)} \leq x_i \leq x_i^{(U)}, i = 1, 2, \dots, n$$

Weighted Metric Methods integrates numerous objectives into a single objective can be used instead of a weighted sum of the objectives. Weighted metrics, such as l_p and l_∞ distance measurements, are frequently employed for this purpose. The weighted l_∞ distance measure of any solution x from the ideal solution z^* can be minimized for non-negative weights as follows:

$$l_p(x) = \sum_{m=1}^M w_m |f_m(x) - z_m^*|^{1/p},$$

$$\text{Subject to } g_i(x) \geq 0, i = 1, 2, \dots, J$$

$$h_k(x) = 0, k = 1, 2, \dots, K$$

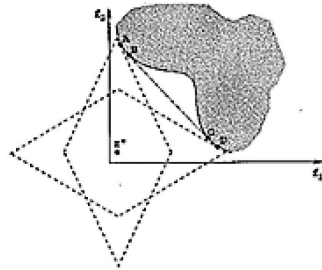
$$x_i^{(L)} \leq x_i \leq x_i^{(U)}, i = 1, 2, \dots, n$$

Figure 6 shows that, when $p = 1$, the resultant problem corresponds to the weighted sum technique. When $p = 2$, the weighted Euclidean distance from the ideal point of any point in the objective space is reduced [33].

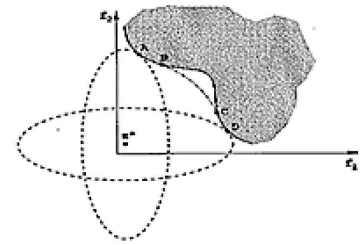
Generally, when using higher value of p , the goal is to reduce the largest deviation $|f_m(x) - z_m^*|$, this is called weighted Tchebycheff metric [3] [9].

$$\text{Minimize } l_\infty(x) = \sum_{m=1}^M \max_{m=1}^M w_m |f_m(x) - z_m^*|,$$

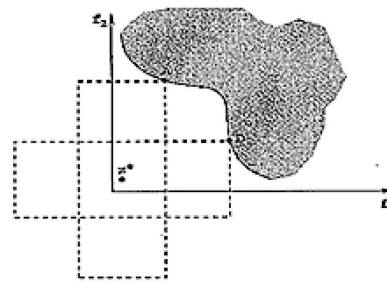
Subject to $g_i(x) \geq 0, i = 1, 2, \dots, J$
 $h_k(x) = 0, k = 1, 2, \dots, K$
 $x_i^{(L)} \leq x_i \leq x_i^{(U)}, i = 1, 2, \dots, n$
 The figures below for $p = 1, p = 2, p = \infty$



The weighted metric method with $p = 1$.



The weighted metric method with $p = 2$.



The weighted metric method with $p = \infty$.

Figure 6. Effect of p parameter on Pareto frontier.

It is not possible to find every optimal solution when $p = 1, p = 2$, where all optimum solutions can be found if we apply the weighted Tchebycheff metric, which should satisfy the following theorem [33].

Theorem 3 Let x^* be a Pareto-optimal solution. Then the weighted Tchebycheff problem can be solved by x^* if there is a positive weighting vector shown in the above equation, where the reference point is the utopian objective vector z^{**} .

It's also critical to keep in mind that, as p increases, the problem can no longer be differentiated, which means that many gradient-based approaches cannot be used to identify the minimum solution to the single-objective optimization problem [3].

Normalizing the objective functions is essential when using this method since different objectives may require values with different orders of magnitude, and understanding the lowest and maximum function values for every target is the main challenge faced using this method. Additionally, the real solution z^* is needed for this strategy to work. Consequently, prior to optimizing the l_p metric, each of the M objectives must be improved independently [3].

ϵ Constraint Method is shown in **Figure 7**. It may address the non-convex objective space; this is a limitation of the weighted approach. To address this, Haimes *et al.* (1971) suggested reformulating the MOOP by retaining a single objective

while constraining the others to user-defined values. The revised problem is expressed as follows [3]

$$\begin{aligned}
 &\text{Minimize } f_\mu(x), \\
 &\text{Subject to } f_m(x) \leq \epsilon_m, m = 1, 2, \dots, M \text{ and } m \neq \mu \\
 &g_i(x) \geq 0, i = 1, 2, \dots, J \\
 &h_k(x) = 0, k = 1, 2, \dots, K \\
 &x_i^{(l)} \leq x_i \leq x_i^{(u)}, i = 1, 2, \dots, n
 \end{aligned}$$

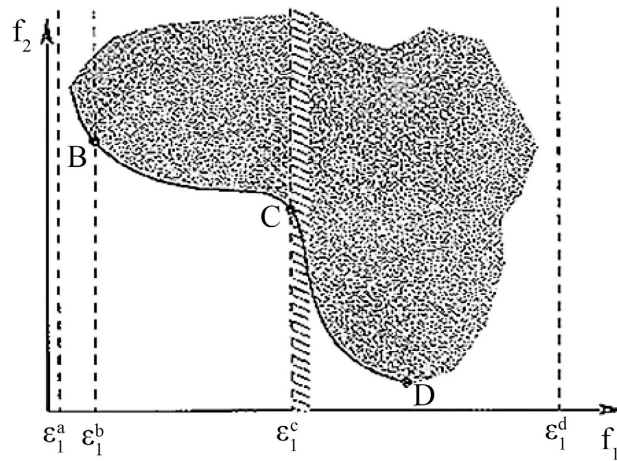


Figure 7. The ϵ -constraint method.

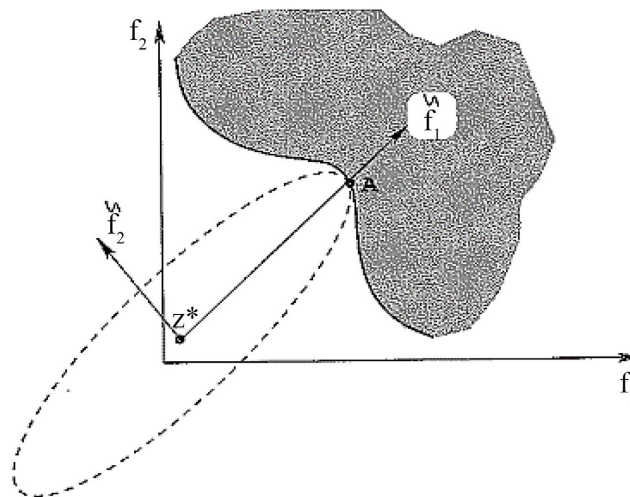


Figure 8. The proposed correlated weighted metric method with $p = 2$.

The parameter ϵ_m represents an upper bound of the value of $f_m(x)$ and need not necessarily mean a small value close to zero. In this method, one objective is treated as objective function, and the others as constraints $f_m(x) \leq \epsilon_m$, if $\epsilon_m = \epsilon^c$ divides the objective space into two different parts, $f_1 > \epsilon^c$, and $f_1 \leq \epsilon^c$.

The resulting problem expressed in the preceding equation has a feasible solution that is located in the left section of the objective space. Finding a solution that reduces this viable zone is now the challenge of the resulting problem [3].

The obvious minimum solution is “C”. Using the ε -constraint method, intermediate Pareto-optimal solutions to nonconvex objective space problems can be obtained.

The following theorem proves the ε -Constraint Method’s utility in dealing with either convex or non-convex problems.

Theorem 3 The unique solution of the ε -constraint problem stated in equation is Pareto-optimal for any given upper bound vector $\varepsilon = (\varepsilon_1, \varepsilon_{\mu-1}, \varepsilon_{\mu+1}, \dots, \varepsilon_M)^T$.

Kuhn-Tucker optimality conditions (Deb, 1995) and by using a Lagrange multiplier for μ_1 the constraint.

Using different ε_M values, different Pareto-optimal solutions can be found. The same strategy can be applied to issues with convex or nonconvex objective spaces.

One obstacle lies in using ε -Constraint Method when choosing ε_M not between the minimum and maximum function value, thus no feasible solutions will be obtained [3].

Rotated Weighted Metric Method. Alternatively, as shown in **Figure 8**, the l_p metric can be applied with an arbitrary rotation from the ideal point, as opposed to applying it directly as indicated in the weighted metric method. Let us consider the following relationship between the rotated objective reference axes \tilde{f} and the original objective axes f [3].

$$\tilde{f} = Rf$$

where R is the rotation matrix of size $M \times M$, thus, the modified l_p metric becomes

$$\tilde{l}_p = \left(\sum_{m=1}^M w_m \left| \tilde{f}_m(x) - z_m^* \right|^p \right)^{1/p}$$

By using $p = 2$, then

$$\left[(f(x) - z^*)^T C (f(x) - z^*) \right]^{1/2},$$

where,

$$C = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}^T \begin{bmatrix} w1 & 0 \\ 0 & w2 \end{bmatrix} \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}$$

Different w and α will give different solutions, where w takes the value from 0 to 1, and the angle α from 0 to 90 degrees.

Adapting the Optimal Solution Dynamically

The problem faced l_p metric is when p is smaller, it will be challenging to locate some optimum solutions, this difficulty can be overcome also, by update of point z^* for every time that a Pareto-optimal solution is found, by this way l_p distance comes closer to Pareto optimal front, and the solutions that aren’t covered before will be easier to be obtained [3].

As shown in **Figure 9**, with each Pareto-optimal solution discovered thus far, all potential combinations of solutions can be created to produce new candidate

ideal solutions. Following that, a candidate solution that does not dominate any of the other alternatives can be chosen as the new z^* , **Figure 9** showing that by moving z^* closer to Pareto front, more Pareto optimal solutions can be found

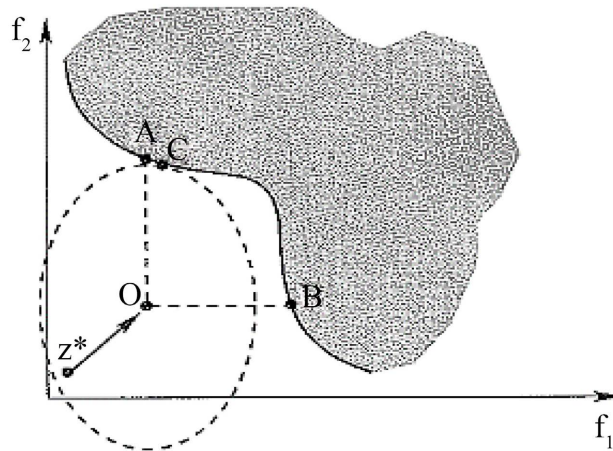


Figure 9. Effect of z^* location on number of optimum Pareto solutions.

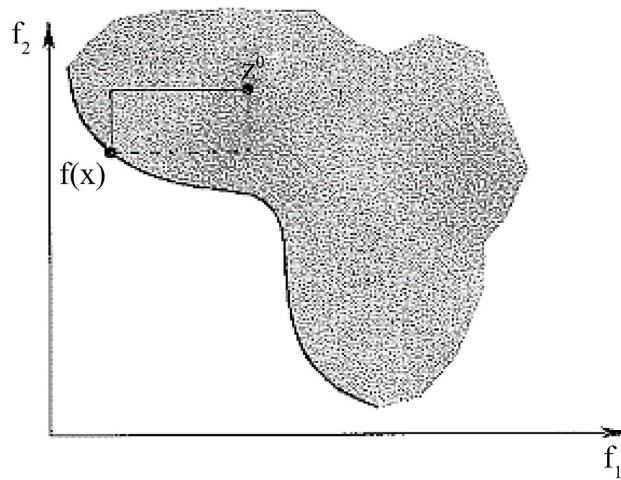


Figure 10. Benson's method.

Benson's Method is most similar to **weighted metric approach**, except that the reference solution is assumed to be a viable non-Pareto-optimal solution. The working principle is shown in **Figure 10**; it depends on a solution z^0 which is generated randomly from the feasible region. Afterwards, the non-negative difference $(z^n - f_m(x))$ of each objective is then evaluated, and their sum is maximized [3];

$$\text{maximize } \sum_{m=1}^M \max(0, (z_m^0 - f_m(x))),$$

$$\text{Subject to } f_m(x) \leq z_m^0, m = 1, 2, \dots, M$$

$$g_i(x) \geq 0, i = 1, 2, \dots, J$$

$$h_k(x) = 0, k = 1, 2, \dots, K$$

$$x_i^{(L)} \leq x_i \leq x_i^{(u)}, i = 1, 2, \dots, n$$

Maximizing the above objective involves identifying a hypercube with the largest perimeter. The optimal solution to this problem lies on the Pareto-optimal front, as the Pareto-optimal region is located at the limit of what a practical search space is [3].

The method, while superior to the weighted metric, has limitations such as additional constraints to limit the search to the region dominated by the chosen solution z^0 and a non-differentiable objective function, making gradient-based methods challenging to address. Although Ehrgott (2000) modified the formulation for differentiable objective functions also presents equality requirements.

Value Function Method (Utility Function) depends on utilizing the interaction of function value with the others, it allows for interactions between different objectives, where $f(x) = (f(1), f(2), \dots, f_m(x))$. When comparing solutions i and j , if the utility function $U(f_i) > U(f_j)$ solution i is selected over solution j . According to Rosenthal (1985), the value function must be significantly decreasing before being employed in multi-objective optimization. This indicates that if one of the objective function values is decreased while the other objective function values remain constant, the preference for a solution must grow. As a result, Miettinen (1999) established the following theorem:

Theorem 4

Let the value function $U: R^M \rightarrow R$ be strongly decreasing. Let U attain its maximum at f^* . Then, f^* is Pareto-optimal.

For the contours shown in **Figure 11**, solution A is the best fit, as its value function contour is tangential to the Pareto-optimal front. It's necessary to remember that this approach can only identify one answer at once. By adjusting the value function's parameters, several Pareto-optimal solutions can be attained [3].

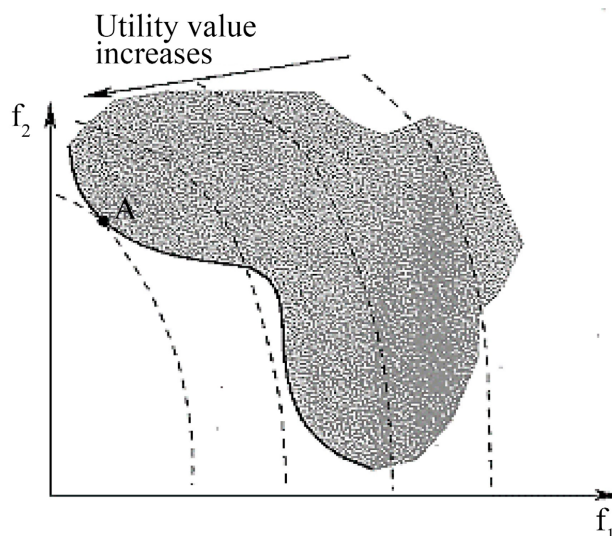


Figure 11. Contours of the value function.

In the utility function, the goal is to maximize the value function as follows:
Where $U: R^M \rightarrow R$ for all M objectives

$$\begin{aligned} & \text{maximize } U(f(x)), \\ & \text{Subject to } g_i(x) \geq 0, i = 1, 2, \dots, J \\ & h_k(x) = 0, k = 1, 2, \dots, K \\ & x_i^{(L)} \leq x_i \leq x_i^{(u)}, i = 1, 2, \dots, n \end{aligned}$$

The primary limitation of this approach is that the resultant answer is totally reliant on the value function that was employed. Additionally, the user must create a value function that is globally applicable throughout the whole search space; consequently, there is a risk of utilizing an overly simplistic value function [3].

Min-Max Goal Programming, in this technique, instead of reducing the total weighted deviations from the objectives, the largest deviation within each goal is minimized [3]. This method involves achieving target values for every objective, so that it can be considered as an extension of traditional linear programming [35]. The Mathematical expression of Min-max Goal Programming is as follows:

Min d

$$\begin{aligned} & \text{Subject to } f_i(x) + n_i - p_i = b_i \\ & \alpha_i n_i + \beta_i p_i \leq d \\ & x \in F \\ & x \geq 0, n_i, p_i \geq 0, i = 1, 2, 3 \dots k \end{aligned}$$

where p_i , and n_i are the positive and negative deviations from the desired value of the i^{th} objective, d is the largest deviation, and b_i is the precise targeted level for the i^{th} goal [35].

Because this technique necessitates selecting weight factors α_i and β_i , the user is ultimately in control of the technique. This method resembles the weighted Tchebycheff method in several ways, but it substitutes the target solution for the ideal solution [3].

This approach was updated by Zeleny, who suggested a new Goal programming procedure instead of determining the optimum in a system with fixed resources, to construct an optimal system by expanding resources [36]. This method is then called De Novo programming. In the above procedure, the objective is to determine each objective function's best and worst performances; based on these performances, a suitable solution is then obtained.

3.2. Posteriori Preference-Based Multi-Objective Optimization Procedure

This procedure is called a posteriori multi-objective optimization, where the algorithm searches for a minimal set of solutions based on a partial order (usually the Pareto order) within the target space. After generating the set of nondominated solutions, the user then analyzes the trade-offs and selects a preferred solution afterward [34].

There is a key difference between the two methods: priori methods require a preference vector before knowing the outcomes, while posteriori methods use problem data to choose from a set of trade-off solutions. A posteriori approaches are generally more systematic and practical, but if a reliable preference vector is

available, then priori methods are sufficient and more direct.

In **Figure 4**, the **posteriori approaches** are divided into two main categories: mathematical programming techniques and algorithms for Evolutionary Multi-objective Optimization (EMO), NBI, NC, SPO, DSD are examples for the mathematical programming techniques, while NSGA II, NSGA III, SPEA-2, PSO, SA, and MOGA are good examples for the evolutionary algorithms [14].

As shown in **Figure 12**, Step 1 (downward and vertically) finds multiple and a well-distributed trade-off solution without using any relative preference information. Once a well-distributed set of trade-off options has been identified, afterwards, higher-level information is used to select one of the trade-off options in Step 2 (horizontally, towards the right) [3].

For a problem with a single objective, Step 1 will result in an exact one global optimal solution, preventing it from moving on to Step 2. Both steps are required in the event of a single-objective optimization with several global optima to locate all or most of the global optima initially. And then to select one from them using the problem's higher-level knowledge [3].

Mathematical programming and Evolutionary Algorithms (EA) are two main categories of a postriori optimization methods. It is also called the ideal multi-objective optimization techniques, which are depicted schematically in **Figure 12**.

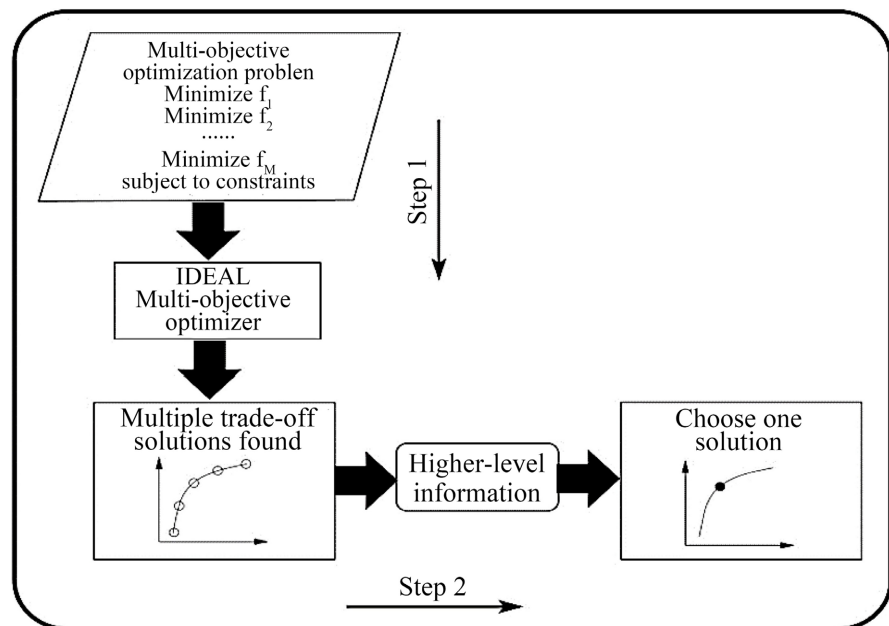


Figure 12. Schematic of a two-step multi-objective optimization procedure.

These techniques are effective for black-box optimization as they don't require knowledge of the underlying physics. They can also handle discontinuous, high-dimensional, and multi-modal problems, but they do not guarantee precise or satisfactory solutions [9].

Among these, mathematical computations methods, and the evolutionary approach (EA) uses the principles of nature's evolution to guide its search for the

best answer, four different evolutionary algorithms, i.e., genetic algorithms (GAs), evolution strategy (ES), evolutionary programming (EP) and genetic programming (GP) which are considered a population based, or an ideal multi-objective optimization procedure, in the following section the most famous methods will be outlined [9].

Multi-objective Evolutionary Algorithms (MOEAs)

Figure 13 demonstrates that multi-objective evolutionary algorithms can be classified into three main categories, Dominance-based, indicator-based, and decomposition-based algorithms are the three types of evolution-based algorithms.

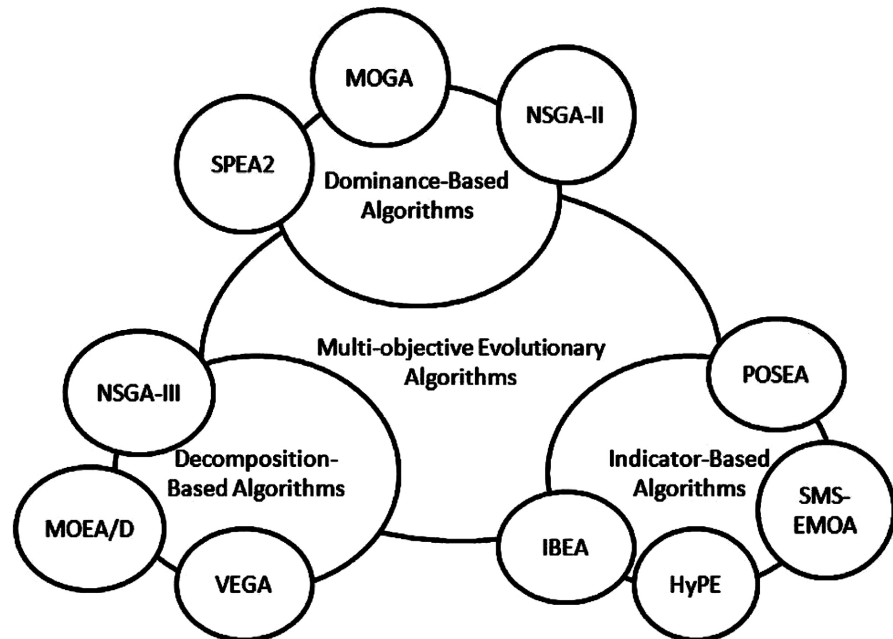


Figure 13. Classification of multi-objective evolutionary algorithms.

Dominance-based algorithms: Fitness is ascribed to solutions based on the Pareto dominance principle. Examples of dominance-based algorithms are Multi-Objective Optimization Algorithm (MOGA), Strength Pareto Evolutionary Algorithm (SPEA), Pareto archived Evolutionary Algorithm (PAES), and Non-Dominated Sorting Genetic Algorithm II (NSGA II) [9]. Dominance-based is also called Pareto-based based where the MOEAs employ a two-tiered ranking system. The first ranking is governed by the relationship of Pareto dominance, while the ranking at the second level is predicated on how each point contributes to diversity [34].

In MOEAs, all trade-off solutions are treated equally. If objective importance is known, the problem can be simplified using weighted sum functions. In priori methods, preferences are provided before solving, allowing conversion to a single-objective problem. In a posteriori method, preferences are used after generating a well-distributed Pareto front, from which the decision-maker selects the preferred solution—this is known as an ideal preference approach [10].

Numerous preference-based techniques have been developed. The initial at-

tempt was by Fonseca and Fleming [33], who proposed ranking the population members based on both Pareto dominance and the decision maker's preferences. A fuzzy approach was also introduced, where the person who makes decisions specifies preferences using preference points. Branke and Deb incorporated preference information into NSGA-II by modifying the dominance definition and applying a biased crowding distance based on weights [37] proposed a multi-objective evolutionary algorithm (MOEA) that partitions the objective space into levels, integrates prior preferences using meaningful parameters, and automatically constructs scalar objectives without requiring weight selection.

Indicator-based algorithms (IBEA): in which the ranking or selection of individuals is dependent on the value of the indicator measure, some of these (IBEA) algorithms, S-Metric Selection Evolutionary Multi-Objective Algorithm (SMS-EMOA), Portfolio Selection Multi-Objective Evolutionary Algorithm (POSEA) and Hypervolume Estimation Algorithm (HypE) [9].

These types of MOEAs are equipped with quality indicators to direct the search methodology. Hypervolume and generational distances are good examples of these metrics using an arbitrary indicator to compare two pairs of possible solutions. This approach is called the indicator-based evolutionary algorithm (IBEA), which was developed by Zitzler and Künzli. Another approach, introduced by Brockhoff and Zitzler, focuses on objective reduction techniques and hypervolume-based methods [38]. This approach was further advanced by Bader and Zitzler, who proposed a fast hypervolume-based MOEA for addressing many-objective optimization problems. It has since become a widely adopted method for resolving problems with several objectives. Subsequently, [35] explored the robustness of hypervolume-based multi-objective search methods. In their work, they introduced three novel strategies for addressing robustness within the field of evolutionary computing: modifying the objective functions, incorporating additional objectives, and imposing robustness constraints. These strategies were then integrated into a multi-objective hypervolume-based search framework [35].

Decomposition-based algorithms: utilize scalarization techniques to decompose the problem into many smaller subproblems, which are subsequently solved cooperatively. Among these methods are the following: MOEA/D based on decomposition (MOEA/D) [20], VEGA (Vector Evaluating Genetic Algorithm) [39], Non-Dominated Sorting Genetic Algorithm III (NSGA-III) [40], MOEA/D based on dynamic resource allocation (MOEA/D-DRA) [41], and MOEA/DD based on dominance and decomposition (MOEA/DD) [28].

MOEA/D is a multi-objective evolutionary algorithm framework that uses conventional aggregation approaches to decompose MOP into scalar sub-objective optimization problems. These subproblems can be linearly or nonlinearly weighted, and their neighboring relations are established based on distances between their aggregation weighting vectors. Each subproblem keeps one solution in its memory, and the algorithm generates a new solution if it's better than the old one. Each individual neighboring subproblem will update its current solutions if it receives

a solution better than the old one. A key advantage of MOEA/D is that local search can be naturally applied to each subproblem, as each one involves optimizing a scalar objective function [20].

3.3. Interactive Optimization Methods (IMO)

The decision-maker is given access to intermediate search results in an interactive method, which helps them comprehend the issue and gives more preference data to direct the search. User input on preferences is requested at several points throughout the algorithm's execution in order to enhance the objective functions, constraints, and their priority.

The goal of interactive multi-objective optimization (IMO) is to determine a decision maker's most favored solution while considering a series of progressive preferences. The decision maker can modify their choices and narrow their search space exploration to only areas they are interested in. Over the last few decades, two distinct approaches—the evolutionary multi-objective optimization (EMO) and multiple criteria decision making (MCDM)—have progressively come to share an interest in IMO. While IMO methods rooted in EMO often use evolutionary algorithms to generate a representative set of solutions in the decision maker's preferred region, IMO methods developed by the MCDM community typically use mathematical programming methodology. One preferred, Pareto's solution [42] [43].

Two essential components of interactive multi-objective optimization are the DM and machine (algorithm). To demonstrate how the DM and the machine communicate when employing IMO approaches, a DM-Machine interaction system is displayed in **Figure 14**.

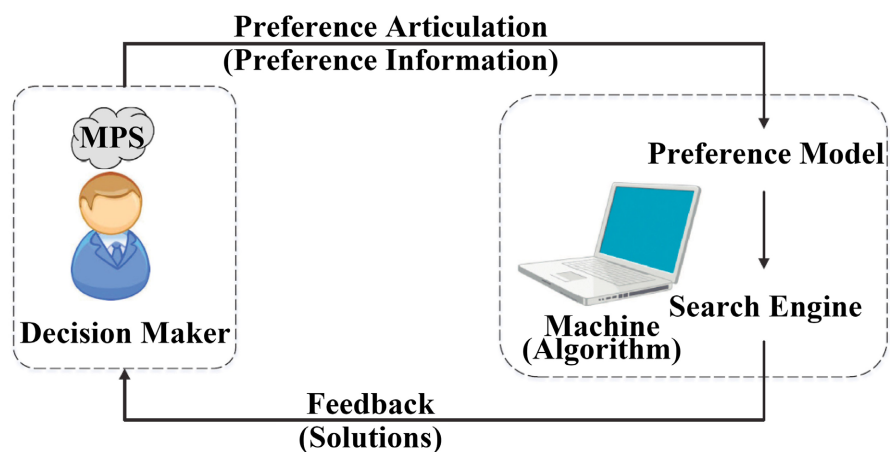


Figure 14. DM-machine interaction system.

The machine is essentially an algorithm that combines a search engine (optimization algorithm) with a preference model. The DM is a human DM who wants to determine his or her most preferred solution [42].

The figure above depicts the entire interaction process. The decision-maker ex-

presses their preferences based on their comprehension of the issue and the algorithm-generated answers. Then, acting as a conduit between the algorithm and the decision-maker, it builds a preference model utilizing the input supplied [42].

The findings are shared with the decision-maker, enabling them to refine their preferences. Through interaction with the algorithm, the decision-maker gains insights and adjusts their preferences to determine the most favored solution [42]. The development of Interactive Multi-Objective (IMO) techniques consists of three main components: search engine, preference model, and preference information. The decision maker (DM) provides various forms of preference information, each affecting their cognitive load differently. The preference model defines how the machine utilizes the DM's preferences, without the DM needing to be aware of it. The algorithm that the search engine uses determines the quality of the results [42].

Any interactive optimization methods will be built on four fundamental design elements: Search engine, preference model, preference information, and interaction pattern come in order of importance; one of two interaction patterns is utilized either interaction after run (IAR) or interaction during run (IDR).

The IMO methodologies have mostly embraced both IAR and IDR patterns. As seen in **Figure 15(a)** and **Figure 15(b)**, an IMO technique that follows the IAR pattern can produce one or a set of (approximate) Pareto optimum solutions at each iteration since it runs the search engine through from start to finish between two neighboring interactions. The DM can contribute preferences such as reference points, weights, tradeoffs, and the categorization of objectives. If just one solution is developed and presented to them, DM is presented with more than one solution.

The solutions obtained in the first three iterations of IMO methods are demonstrated. The curve in each sub-figure represents the Pareto front. (a) and (b) show a single solution (z) and a set of solutions (F) obtained by methods adopting the IAR pattern, respectively. (c) illustrates how a population (P) changes over time in accordance with the decision maker's (DM's) preferences using methods that follow the IDR pattern [14].

In addition to providing tradeoffs or a categorization of objectives on the most desirable option, decision makers may also give reference points or weights and compare these solutions. Classical interactive MCDM techniques use the IAR pattern to find one or more Pareto optimum solutions for each iteration. Additionally, it is utilized in certain interactive MOEAs, such as those that rely on reference point data. Section III has specifics on various IMO techniques [14].

As seen in **Figure 15(c)**, IMO approaches that follow the IDR pattern allow the DM to provide preferences at regular intervals throughout the search engine run in order to direct the search towards the DM's area of interest (ROI). In most cases, the population does not approach the Pareto front until much later in the optimization process. According to [6] [11], the DM is more active in the whole optimization-cum-decision making process in this scenario since they have more

opportunities to offer fresh information. Thus, the IDR pattern’s IMO procedure is more DM-focused. It should be mentioned that approaches using the IAR pattern are better suited for preferences like tradeoffs and objective categorization, since it is more relevant to take these tradeoffs into account.

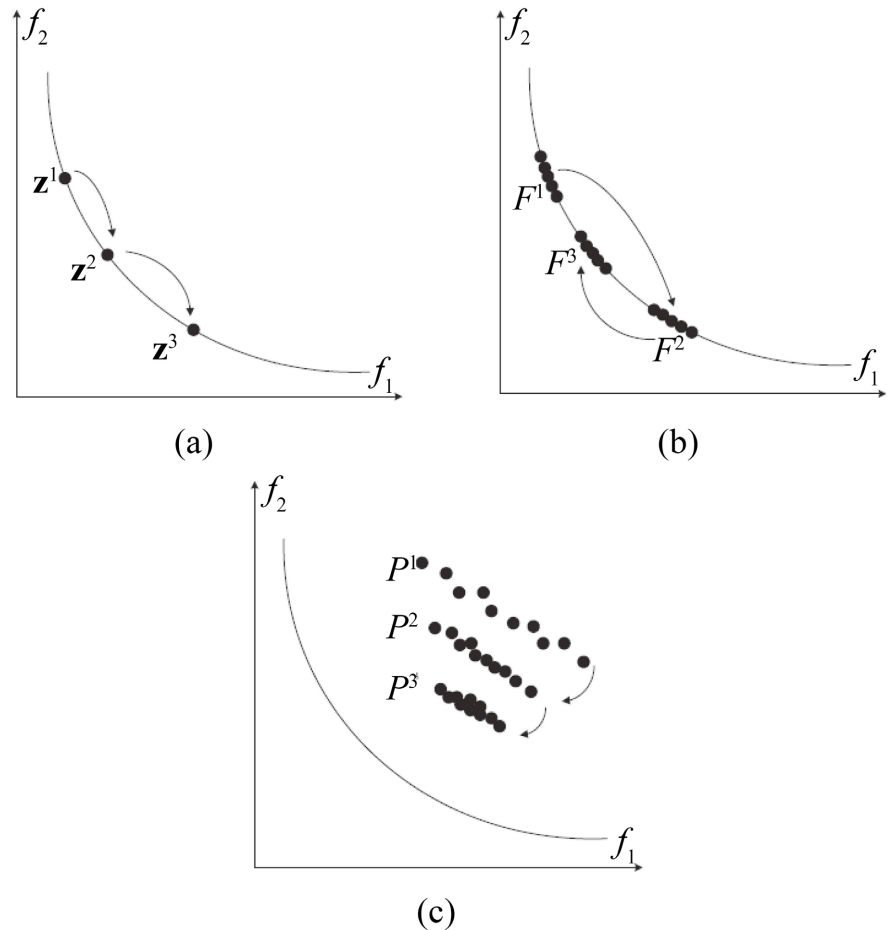


Figure 15. Dm’s preferences for methods adopting the IDR or IAR pattern [19].

Preference Information can be categorized into three different ways: expectation that DM wants to achieve, the comparison of objective functions and the comparison of solutions, which requires the DM to make comparisons when expressing preferences. It is possible to compare objective functions using weights, trade-offs, objective categorization, etc. Weights are frequently employed to represent the relative significance of goals, and finally, a tradeoff is when one solution is given up in order to improve another at a feasible solution [44] [45].

Reference Model widely employs one of three types of preference models: choice rules, dominance relations, and value functions (also known as utility functions). A value function (VF) is a scalar function that assesses solutions quantitatively for all objectives. The DM either explicitly specifies its parameters or indirectly determines them depending on the DM’s preferences. The DM’s preferences are expressed as a relationship between two solutions, which is known as a dominant relation.

In the selection operator of interactive MOEAs, it frequently takes the role of the relationship of Pareto dominance [42]. The following table shows a summary of the preference types.

Table 1 illustrates the advantages and disadvantages of search engines, preference model, and preference information.

Table 1. Common kinds of information about preferences and their pros and cons [14].

Category	Preference Information	Features	Advantages and Disadvantages
Expectation	Reference point	A reference point consists of k continuous valued aspiration levels representing desirable objective values (quantitative).	The reference point could either be accessible or not, and the DM has the discretion to define it. Calculating the objective ranges is frequently necessary, and additional computational effort or prior knowledge may be required
	Weights	The purpose of weights is to show how important one objective is in relation to others. They can be derived from the decision-maker's (DM's) pairwise comparisons of objectives or explicitly provided by the DM as k scalar values.	There is a widespread misunderstanding that weights will indicate how important each goal is in relation to the others. However, it is unclear what these notions actually represent. Additionally, employing weights to control the solution process isn't always easy for the DM [46].
Comparison of objective functions	Tradeoffs	In general, taking one objective as the reference objective, the DM needs to give the amount of increment of each of the remaining $k-1$ objectives to compensate one unit decrement of the reference objective (quantitative).	Although tradeoffs can facilitate the quest for a more ideal solution, they also put a heavy cognitive burden on the DM because they must decide how much compromise is acceptable between objectives.
	Classifying of objectives	The DM is tasked with categorizing objectives based on desirable changes and specifying aspiration levels or upper bounds for some objectives.	Establishing reference points is strongly related to classifying objectives. The decision-maker has more control over the process of finding a solution when objectives are categorized and the degree of relaxation permitted for those that can be compromised is specified. But it's crucial to remember that the decision-maker's burden will rise as a result of these new responsibilities.

Continued

Comparison of solutions	Pairwise comparison of solutions	determining whether solution is better than the other, whether they are equal, indifferent, or neither (qualitative).	Relatively less mental work is required of the decision-maker while comparing solutions. However, as the number of solutions increases, their duties will probably also expand.
	Classification of solutions	Dividing a set of solutions into different categories where solutions in each category incomparable or indifferent (qualitative).	
	Selecting the most referred solution	Selecting the most preferred solution among group of solutions (qualitative).	

- Models of preference have been widely utilized in the literature, including value functions (or utility functions), dominance relations, and decision rules [47]. Decision rules typically consist of two parts: the premise, which outlines the criteria that objectives and/or solutions must satisfy, and the decision section, which establishes the relationships between solutions or assigns them a score [42].
- Many methods model DM’s underlying value function (VF) dynamically based on DM’s preferences. Three types of popular VFs are explained; a) A weighted metric measures the distance between the objective vector and a certain point, b) Wierzbicki introduced a successful scalarizing function (ASF) as part of the reference point technique, According to Wierzbicki, it is a modified value function (VF) that communicates both the usefulness of meeting aspirations and the disutility of failing to do so [48]. ASFs have been created in various forms [49] [50], c) The DM is symbolized by only one compatible additive value function [51]. The term “compatible” refers to the IMO approach’s ability to create a preference model that ranks solutions similarly to the DM method [52] [53] examines all compatible additive VFs.
- But for the dominance relations, dominance relation indicates the DM’s choice between two solutions: one is chosen over the other, or both are indifferent or incomparable. The binary relation between two solutions, x and y, can be expressed as xDy (x dominates y), yDx (y dominates x), or $xDNy$ (indifferent). Dominance relations sometimes combine Pareto and DM preferences, allowing for comparison of non-dominated solutions [13] [42] [54].
- Decision-makers usually apply “IF-THEN” principles according to their personal preferences. Decision rules can explain the relationships between solutions or assign solution scores to help choose the best options if certain criteria are satisfied by the objectives and/or solutions. Because of their inherent error, preference models in decision rules are more general and simpler for DMs to

understand. The DM's qualitative preferences may be transformed into quantitative data on goals or solutions using fuzzy rules, a sort of decision rule [55].

- In hybrid preference model some IMO approaches use a mixed preference model. In [56], the r -dominance relation is defined using the weighted Euclidean distance. A solution x is said to r -dominate a solution y if they meet any of the following two conditions: 1) x Pareto dominates y ; 2) x and y are non-dominated; x is closer to the reference point than y in terms of weighted Euclidean distance; and the absolute value of their distance difference exceeds a threshold. [57] [58], after getting the strengths of solutions using the fuzzy inference system, a strength superior relation is established to define a fitness function.

IMO approaches utilize search engines classified into mathematical programming (MP) techniques and non-MP strategies. techniques for optimization, including linear, nonlinear, and multi-objective programming, are specifically designed for software systems. In the MCDM community, IMO approaches frequently achieve Pareto-optimal solutions [42].

3.4. Non-Preference-Based Methods

In contrast to preference-based procedures, population-based approaches do not require predefined information before searching or solving problems; instead, for every simulation run, a collection of solutions is produced, and the best solutions are selected to form the Pareto frontier set of solutions through additional sorting. In recent years, a variety of non-classical, unconventional, and stochastic search and optimization algorithms have been introduced, transforming the field of search and optimization [3].

3.5. Hybridization Techniques

In **Figure 4**, Hybrid techniques, which, although not fully studied yet, may comprise various Evolutionary Multi-objectives Optimization algorithms or a combination of MCDM (multi-criteria decision making) and EMO; its future uses might supplant existing technologies that depend on goal-oriented approaches. Speed-constrained Multi-objective Particle Swarm Optimization (SMPSO) is one of the potential high-performance algorithms, since the single objective version of the method (PSO) outperformed the EA-based alternatives [54] [59]. Other approaches like the Parallel Re-Combinative Simulated Annealing technique [60], address the issue of genetic drift by integrating the principle of the cooling sequence from simulated annealing [61], Boltzmann tournament selection [62], and standard genetic operators. One of the well-known hybridization methods is the Local Search Memetic Genetic Algorithms (GAs) employ a global search strategy; however, they often require a relatively long time to converge to the global optimum. To address this, a population-based GA can be combined with an individual learning procedure to improve the fine-tuning of global search.

When a genetic algorithm is integrated with a local search procedure, it forms

a hybrid genetic algorithm (HGA). The basic steps of an HGA are as follows [63]:

- 1) Specify the objective or fitness function, and configure the genetic algorithm parameters, including population size, parent-to-offspring ratio, selection strategy, crossover count and mutation rate.
- 2) Create the first population at random as the current parent population.
- 3) Evaluate the objective function for each individual (chromosome or solution) in the initial population.
- 4) Produce the offspring population by applying GA operations like crossover, mutation, and selection.
- 5) Determine the objective function values for all individuals in the offspring population.
- 6) Perform a local search on each offspring, evaluating fitness of each new location, and replace the offspring if a better local option is available.
- 7) Select the individuals that will form the next generation; this process is known as replacement in that individuals from the current parent population are “replaced” by a new population formed from individuals selected from the offspring and/or parent generations.
- 8) If the criterion for stopping is met, the process terminates; otherwise, return to Step 4.

Among these learning searching techniques (LS) are the Nelder-Mead Simplex method and the three-dimensional local search. The Best-Offspring Hybrid Genetic Algorithm is one of the developed algorithms that is built on earlier principles (BOHGAS) [64]. The objective of this algorithm is to lower the overall LS expenses. It has been observed that the LS may be run repeatedly on the same “valley” (to find the lowest) or “mountain” (to find the maximum) [65]. Therefore, after local search, it is likely that several chromosomes within a generation become clustered closely together, positioned on the same or closer to the peak or at the same or closer to the valley. This may make it harder for the GA to maintain diversity in its population, an important consideration in avoiding converging to a local optimum [66].

3.6. Advantages and Disadvantages of a Priori, a Posteriori, and Interactive Preference Approaches

When the DM is expected to offer preference, information is a crucial component of MCDM. To achieve this, there are three options:

- Prior to the search (a priori approaches).
- Posteriori.
- Interactive preferences.

Its widespread use is due to the fact that any optimization procedure that makes use of this a priori knowledge becomes easy. The primary challenge (and disadvantage of the method) lies in locating this first global preference data [42].

Interactive approaches have been normally favored by researchers [67] for several reasons:

- The entire set of components of a situation, besides the context in which it is embedded, can affect perception.

- Individual preference functions or value structures cannot be expressed analytically, although it is assumed that the DM subscribes to a set of beliefs.
- Value structures change over time, and preferences of the DM can change over time as well.
- Aspirations or desires change as a result of learning and experience.
- The DM normally looks at trade-offs satisfying a certain set of criteria, rather than at optimizing all the objectives one after the other.

Interactive techniques also present several challenges, particularly concerning the preference data the decision-maker (DM) must provide while conducting the search. For instance, the DM may be asked to evaluate a set of options for each goal, assign weights, or adjust ambition levels. These tasks are often complex, and DMs frequently find it difficult to offer input that effectively guides searching for the best compromise solution.

In the field of OR, posteriori methods are also often used [68] [69]. These methods' primary benefit is that, because they are predicated on the idea that "more is better," no utility function is needed for the analysis [70]. The primary drawbacks of posteriori methods are:

- The algorithms used with these approaches are normally very complex and tend to be difficult for the DM to understand.
- Many real-world problems are too large and complex to be solved using this sort of approach.
- The number of solutions in the Pareto optimal set is typically too large for the decision maker to analyze effectively.

Combining two or more of these techniques is also a possibility. An approach that may be developed is to ask the DM for some pre-search information and then ask them to modify their choices while the search is underway. Comparing this to using the two methods separately, it might be more effective.

3.7. Classification Strategies: Static and Dynamic Strategies

Fei Wu *et al.* (2023) developed a dynamic optimization algorithm to face the following problems [70]:

- 1) How to deal with diversity, keeping the population convergence and diversity balanced?
- 2) What is the best way to design a prediction strategy that is both robust and efficient?
- 3) From the strategy, the classification of decision variables is not clear.

Mohammad Reza Sharif *et al.* (2021) introduced a multi-objective moth swarm algorithm featuring a novel definition of pathfinder moths and moonlight, aimed at improving synchronization and preserving a well-distributed set of non-dominated solutions. Furthermore, the crowding-distance method was used to choose the most efficient solutions within the population. The efficiency of the suggested MOMSA was evaluated using a series of multi-objective benchmark problems ranging from 7 to 30 dimensions [70].

They were compared with the results. Against three well-known meta-heuristics, including the decomposition-based multi-objective evolutionary algorithm (MOEA/D) and the Pareto envelope-based selection algorithm II (PESA-II), and multi-objective ant lion optimizer (MOALO). To facilitate comparison, four measures were used: generational distance (GD), spread (Δ), spacing (S), and maximum spread (MS). Sandra C. Cerda-Flores *et al.* (2022) introduced a similar article on potential multi-objective methods [70].

3.8. Classification of Meta-Heuristic Algorithms According to Their Nature: Swarm-Based, Evolutionary-Based, Physics-Based, and Human-Based

Meta-heuristic algorithms are analyzed and compared across five key aspects, followed by a classification of the algorithms based on these criteria and These classifications are according to:

- The type of algorithms (swarm-based, evolutionary-based, physics-based and human-based).
- Nature-inspired vs. non-nature-inspired.
- The inspiration's origin.
- Population-based vs. single solution-based.
- Based on the country of origin.

Today, algorithms are generally classified into four main categories: physics/chemistry-based, bio-inspired (excluding swarm intelligence), swarm intelligence (SI)-based, and a miscellaneous group. The latter includes algorithms that are difficult to categorize under the first three groups, as they incorporate diverse characteristics from various domains such as social behavior and emotional processes [3].

3.9. Working Principles for Well-Known Meta Heuristic Optimization Algorithms

Table 2 below explains briefly the most well-known multi-objective algorithms, meta-heuristic-based, work principles and their category.

Table 2. A wide range of 110 meta-heuristic algorithms.

Method	Brief explanation
Evolutionary Programming (EP)	This algorithm defines a number of operators, such as crossover and mutations, and is a basic method used in the majority of modern and evolutionary approaches [71]
Genetic Algorithm (GA)	This algorithm leverages biological concepts such as heredity and mutation to find approximate results. It is one of the most widely used population-based heuristic algorithms. The primary operator of the algorithm is combination, but it also employs the mutation operator to prevent poor convergence Stay away from local optima traps. The algorithm is relied on the survival of the fittest theory and Darwin's theory, with the core idea being that traits are inherited through genes [72]
Scatter Search Algorithm (SSA)	t differs from other evolutionary algorithms in that it is based on the concept of using systematic techniques to generate new solutions, offering certain advantages over solutions chosen purely at random. This approach is used to enhance and diversify the search process [73]

Continued

Simulated Annealing (SA)	The SA algorithm is a probability-based, single-solution method for finding the global optimal solution in large solution spaces, based on the melting and freezing process of metals [74]
Tabu Search (TS)	The algorithm employs the concept of a Tabu List to avoid local optima traps. It begins with an initial solution and searches for the best neighboring solution. It moves to the best solution unless it is on the tabu list, with the length of the list representing the maximum number of iterations [17]
Cultural Algorithms (CA)	This algorithm is a type of evolutionary algorithm that, unlike others, incorporates both a population component and a knowledge component. The belief space in the algorithm is classified into various types, including temporal knowledge, domain-specific knowledge, situational knowledge, and spatial knowledge [75]
Particle Swarm Optimization (PSO)	Inspired by bird flight, the algorithm calculates objective function values for each particle and determines direction based on its current location, best location, and the group's best particles. This process is repeated using speed and position update operators until a stopping criterion is met [76]
Ant Colony Optimization (ACO)	ACO is a population-based metaheuristic algorithm inspired by nature, specifically the actions of ants. As social creatures focused on survival, ants locate food and communicate through pheromones, leaving traces on the ground. When selecting between two paths, ants typically prefer those with higher pheromone concentrations [77]
Differential Evolution (DE)	Based on the theory of natural evolution, this algorithm utilizes mutation and combination operators. However, it differs from the Genetic Algorithm (GA) in terms of comparison and solution selection [78]
Variable Neighborhood Search (VNS)	The algorithm applies basic local search rules by dividing the space of solution into neighborhoods, randomly selecting one, and performing a local search. If the new solution is better than the best recorded one, a new neighborhood is generated around that solution [79]
Sheep Flocks Heredity Model (SFHM)	This algorithm is an evolutionary computation algorithm based on sheep flocks' heredity. It simulates heredity of sheep flocks in a prairie [80]
Harmony Search (HS)	Harmony in music and optimization problem solving is closely related, as it relates to the optimal relationship between sound waves and frequencies for optimal audience experience [81]
Bacterial Foraging Optimization (BFO)	The algorithm, inspired by bacterial movement, focuses on the search behavior of individual bacteria, their reproduction probability, and bacterial decomposition, with well-nourished bacteria being the key elements [82]
Social Cognitive Optimization (SCO)	This algorithm emphasizes the development of human knowledge and intellect, considering behavioral, environmental, and personal factors. It also incorporates learning from others and the impact of their actions [83]
Shuffled Frog Leaping Algorithm (SFLA)	This algorithm combines probabilistic and deterministic methods to simulate frogs' search for food in wetlands. It seeks to balance exploring potential solutions with in-depth research within the solution space. The population is a group of frogs with chromosome structures similar to those in genetic algorithms [84]
Electromagnetism-like algorithm (EMA)	The proposed algorithm utilizes electrostatic system rules, adjusting each particle's virtual electric charge based on its optimal position [85]
Space Gravitational Algorithm (SGA)	The SGA algorithm, inspired by space simulations, applies Einstein's theory and Newton's law of gravitation to identify the global optimal solution. This approach reduces computational complexity and helps avoid local optima traps [86]
Particle Collision Algorithm (PCA)	The PCA algorithm, inspired by nuclear collision reactions, features a structure similar to simulated annealing (SA) but does not require user-defined parameters or a cooling schedule [87]
Big Bang-Big Crunch (BB-BC)	The BB-BC algorithm employs the Big Bang theory and the concept of universe freezing to create a weighted center of gravity, generating new solutions through the use of a Gaussian distribution [88]
Group Search Optimizer (GSO)	The algorithm is motivated by the actions of animals searching for food resources, categorizing them into producers, scroungers, and rangers, each exhibiting distinct behaviors [89]
Invasive Weed Optimization (IWO)	The algorithm is modeled after robust, random, and adaptable weed colonies, which pose a threat to beneficial plants while demonstrating their ability to adapt to environmental changes [90]

Continued

Small-world Optimization Algorithm (SWOA)	The proposed algorithm, inspired by scientific studies on human communication and networking, utilizes local short-range and random long-range search agents to perform both local and global searches [91]
Cat Swarm Optimization (CSO)	The algorithm employs tracking and searching sub-models, treating cats as solutions with parameters such as relative velocity, proportional value, and status [92]
Saplings Growing UP Algorithm (SGA)	The algorithm is based on the planting and growth of saplings, aiming for uniform distribution of agents in the solution space through mating, branching, and vaccination [93]
Imperialist Competitive Algorithm (ICA)	The algorithm divides countries into colonial and colonizer groups, building empires. Over iterations, weak empires collapse, leading to convergence when only one empire remains [94]
Artificial Bee Colony Algorithm (ABC)	This algorithm uses bee group behavior to identify food resources, modeling worker bees, watchdogs, and scouts' performance. Operators consider deterministic, probabilistic, and searching for new areas if improvement isn't achieved [95]
Central Force Optimization (CFO)	CFO is a deterministic algorithm that uses gravity to slow the search process, focusing on the searches with the highest proportion or mass [96]
Integrated Radiation Algorithm (IRA)	The algorithm uses Einstein's theory of general relativity to find the optimal solution in a simplified astrophysics search space, assuming the optimal solution is an incompletely symmetrically growing supernova [97]
Multi Point Simulated Annealing Algorithm (MPSA)	This method uses multiple stages simulated annealing strategy to compare multiple candidates designs simultaneously [98]
River Formation Dynamics Algorithm (RFDA)	The algorithm relying on the formation of rivers and their beds through erosion and sedimentation processes, considering factors like water movement and sedimentation [99]
Big Crunch Algorithm (BCA)	According to the algorithm, which depends on closed the universe theory, the Big Bang, the initial cosmic bang, produces an endless amount of heat and energy until there is just one mass left [100]
Biogeography Based Optimization (BBO)	This algorithm draws inspiration from the spatial dispersion of biological entities and employs fundamental operators such as migration and mutation [101]
Firefly Algorithm (FFA)	The algorithm, inspired by firefly insects' ability to generate light, attracts weaker ones inversely related to their distance [102]
Paddy Field Algorithm (PFA)	The method disperses seeds at random, with the goal of avoiding local optimal traps. Higher-growth plants have a greater likelihood of reused [103]
Gravitational Search Algorithm (GSA)	The algorithm optimizes a system by applying the laws of motion and gravity. Because mass and distance are proportionate, each particle inside the cosmos is drawn to every other particle. The object with the highest mass is anticipated to be the ideal global solution [104]
Cuckoo Search (CS)	The proposed algorithm optimizes chicken production by avoiding detection by host birds, based on cuckoo behavior in bird nests [105]
Hunting Search (HuS)	The algorithm, motivated by collective animal hunting like Dolphins, wolves, and lions, suggests that despite their distinct hunting methods, they share a common approach: hunters encircle prey, tighten sieges, and re-establish the group if prey escapes [106]
Intelligent Water Drops (IWD)	The method models variations in water speed, soil content, and soil bed as it flows from one point to another using intelligent water droplets acting as search agents. [107]
Artificial Physics Optimization Algorithm (APOA)	To look through the solution space, APOA employs physical forces to transfer agents to regions with higher fitness based on user-defined agent mass [108]
Bacterial Evolutionary Algorithm (BEA)	This evolutionary clustering technique maximizes the number of groups in datasets by using gene transfer and bacterial mutation [109]

Continued

Human-inspired Algorithm (HIA)	The algorithm resembles the approach of modern climbers, splitting the search space into equal sub-spaces and distributing an equivalent quantity of agents across each one [110]
League Championship Algorithm (LCA)	The algorithm's inspiration comes from a sports league competition, where search agents compete over several weeks. The fittest team emerges as the winner, while all teams prepare for adjustments in the following week [111]
Locust Swarms (LS)	Inspired by locusts, the algorithm uses greedy local search, the PSO approach, and clever starting positions to explore the search space, starting a little bit away from prior solutions [112]
Consultant-Guided Search (CGS)	CGS is a collective intelligence method that emphasizes direct information sharing among members of a population, inspired by real-world decision-making strategies [113]
Bat Algorithm (BA)	The algorithm, inspired by bat echolocation, adjusts each bat's speed and position to detect prey, barriers, and nests in darkness, based on optimal positions [114]
Charged System Search (CSS)	The algorithm uses the laws of motion to direct charged particles (CPs) in solving problems, with the position, acceleration, and velocity of each agent being affected by the other CPs. [115]
Chemical Reaction Optimization (CRO)	The algorithm models chemical reactions and energy transfers, employing combination and decomposition processes to reduce potential energy [116]
Eagle Strategy Algorithm (ESA)	This two-step hybrid search algorithm blends the firefly algorithm with random search [117]
Group Counseling Optimization (GCO)	With iterations that resemble counseling sessions, where members gradually better their positions with help from the counseling team or themselves, the algorithm mimics human problem-solving behavior through consultation [118]
Social Emotional Optimization (SEO)	The algorithm model human efforts to improve social status, treating individuals as members of society. It updates their emotional scores based on feedback and selects the members with the highest status [119]
Galaxy Based Search Algorithm (GbSA)	GbSA algorithm uses spiral arms of galaxies to look for the best solutions, improving spiral movements with chaos to escape local optimal traps [120] [121]
Spiral Dynamics Inspired Optimization (SDIO)	The method, which draws inspiration from natural spiral occurrences, searches the solution space using a multidimensional spiral and balances variation and intensification using control parameters [122]
Teaching-learning based Optimization (TLBO)	The approach uses a mutual teaching-learning relationship, treating population members as class students. It involves two phases: teacher influence and mutual interactions [123]
Anarchic Society Optimization (ASO)	It illustrates the difficulty of overcoming local optimal constraints when a group of members are aberrant, unstable, and disruptive [124]
Current Search (CS)	This technique is based on how electricity flows in circuits, where current usually selects the channel with the least resistance [125]
Water Cycle Algorithm (WCA)	The algorithm, inspired by water cycle processes, moves creeks towards rivers and rivers towards the sea, with creeks qualifying to connect the river or sea [126]
Wolf Search Algorithm (WSA)	The algorithm, inspired by wolves' survival strategies; while keeping the prior location in the memory, it allows each agent to independently search and if the new a position is superior than the previous ones, including [127]
Mine Blast Algorithm (MBA)	The method, relies on a real-world mine-blast event, selects the explosive component causing the most damage to enlarge the most recent mine [128]
Atmosphere Clouds Model (ACM)	In order to intensify, diversify, and examine the search area., the algorithm mimics the formation, movement, and dissemination of a natural cloud [129]
Black Holes Algorithm (BHA)	Inspired by the phenomena of black hole, the algorithm views the ideal solution in every cycle as a black hole that attracts more stars and randomly produces novel solutions [130]

Continued

Egyptian Vulture Optimization (EVO)	The algorithm, inspired by Egyptian vultures' natural behaviors, was initially developed for hybrid optimization functions [131]
Penguins Search Optimization Algorithm (PSOA)	The group has the more food, such fish, is chosen as the optimal option by this method, which is based on penguins' collective hunting behavior [132]
Swallow Swarm optimization (SSO)	The algorithm used divides particles into three groups—explorer, aimless, and leader—based on the movements and behavior of swallows [133]
Grey Wolf Optimizer (GWO)	The method relies on the predation strategy and hierarchy of gray wolves, including Omega, Delta, Beta, and Alpha kinds and the three primary hunting stages [134]
Golden Ball (GB)	This approach uses the principles of a football game and is predicated on the multiple population approach [135] [136]
Animal Migration Optimization Algorithm (AMOA)	The way that animals migrate in groups and leave a single group to live in another served as the model for this algorithm [137]
Soccer League Competition Algorithm (SLC)	Soccer leagues served as inspiration., focuses on team and player competition for league ranking and personal development, resulting in faster and more accurate convergence to global optimality [138] [139]
Chicken Swarm (CS)	This algorithm replicates the hierarchical conduct of a flock of chickens, with hens, roosters, and chickens organized in a specific order [140]
Forest Optimization Algorithm (FOA)	The algorithm, based on planting seeds in a forest, assumes that seeds under trees cannot grow, but those scattered elsewhere may [141]
Heart Algorithm (HA)	This algorithm is modeled to mimics the circulatory system and heart. The best-fitting population member is considered the heart, while the others are blood molecules. Blood molecules gravitate toward or away from the heart to improve their fitness [142]
Kaizen Programming (KP)	Each expert contributes an idea to the final solution which combines all of the ideas. The fitness of each concept is determined by its contribution in the Japanese problem-solving technique known as kaizen [143]
Exchange Market Algorithm (EMA)	The process of exchanging shares on the stock market served as the model for this evolutionary algorithm [144]
African Buffalo Optimization (ABO)	The algorithm draws inspiration from the actions of the behavior of African buffalos, wild domestic cattle, who roam the continent during rainy seasons for green pastures [145]
Elephant Herding Optimization (EHO)	The behavior of African buffalos, wild domestic cattle that travel the continent in search of new pastures during wet seasons, served as the model for the algorithm [146]
Ions Motion Algorithm (IMA)	This algorithm uses tensile force, pressure, and ionic motions of anions and cations. Positive ions (cations) and negative ions (anions) are the two groups into which candidate solutions are separated. Ions introduce potential solutions, and their movement within the search area is driven by tensile and pressure forces [147]
General Relativity Search Algorithm (GRSA)	The algorithm draws inspiration from the actions of the general theory of relativity. In this approach, population members are represented as space-based particles, influenced only by gravity. These particles move along the shortest paths to reach their most stable positions, with step lengths and directions determined by the shortest paths and speed [148]
Jaguar Algorithm with Learning Behavior (JALB)	This algorithm is modeled after the hunting behavior of jaguars, who charge toward their prey, sometimes hunting in groups. It balances exploration and exploitation by mimicking this hunting strategy [149]
Optics Inspired Optimization (OIO)	This method models optical phenomena, using convex and concave mirrors to diverge and converge light beams. Within the search area, peaks and valleys function as convex and concave mirrors, respectively, reflecting the search space [150]

Continued

Runner-Root Algorithm (RRA)	This algorithm models the behavior of plant runners and roots, with roots focusing on small regions and runners covering larger areas with big steps. It includes two functions representing the runners as well as roots for exploitation and exploration, respectively [151]
Vortex Search Algorithm (VSA)	Based on the vertical movement of fluids, this meta-heuristic algorithm improves and expands searches using an adjustable step length mechanism [152]
Stochastic Fractal Search (SFS)	A mathematical idea is used by the program known as fractals, which are taken from the natural phenomena of growth [153]
Prey-Predator Algorithm (PPA)	The predator-prey interaction found in animals served as the model for the algorithm [154]
Water Wave Optimization (WVO)	This method, which was inspired by water waves, makes use of breaking, reflection, and propagation to produce an efficient search mechanism in a high-dimensional solution space [155]
Bull Optimization Algorithm (BOA)	This algorithm changes the genetic algorithm's selection procedure so that only superior individuals are eligible to take part in crossover [156]
Elephant Search Algorithm (ESA)	A population's members are viewed by this algorithm as a herd of elephants. Members who identify as male and female are regarded as local and exploratory search agents, respectively [157]
Ant Lion Optimizer (ALO)	Five steps make up this technique, which was inspired by ant-lion hunting: random search, sieging, trapping, capturing prey, and rebuilding the trap [158]
Lion Optimization Algorithm (LOA)	This application simulates lions' social behavior. Male adult lions can be either resident or nomadic, and they can live in packs or roam freely. While resident lions represent local searchers, nomadic lions act as global search agents for identification and exploration [159]
Whale Optimization Algorithm (WOA)	The three main parts of this algorithm—sifting prey, assaulting prey (exploitation), and looking for prey (exploration)—were motivated by natural selection and whale social behavior [160]
Dynamic Virtual Bats Algorithm (DVBA)	The ability of bats to produce different frequencies and wavelengths during the hunting phase served as the inspiration for this approach [161]
Tug of War Optimization (TWO)	This population-based algorithm was inspired by the tug-of-war game. Every potential solution is considered by this algorithm collectively to engage in the game [162]
Virus Optimization Algorithm (VOA)	This algorithm imitates how viruses behave when they target cells. Using this technique, the immune system regulates the quantity of viruses in every cycle to stop the unintended increase in viruses [163]
Virus colony search (VCS)	This program imitates how viruses replicate and spread when they infect host cells [164]
Crow Search Algorithm (CSA)	This method is a population-based approach motivated by the brilliant actions of crows. The core principle of the Algorithm of Crow Search (CSA) is that crows store and hide surplus food, retrieving it when necessary [165]
Dragonfly Algorithm (DA)	This algorithm is modeled on the social behavior of dragonflies during activities such as foraging, group navigation, and evading predators [166]
Camel Algorithm (CA)	The strategy incorporates elements like temperature, water supply, stability, visibility, and ground conditions, drawing inspiration from camel behavior during desert marches [167]
Water Evaporation Optimization (WEO)	The algorithm models water molecule behavior during evaporation from solid surfaces, treating the solid surface as the solution space and water molecules as the population. Search parameters include surface wettability and molecular features [168]
Thermal Exchange Optimization (TEO)	The plain and easy-to-understand Newton's law of cooling serves as the foundation for this method [169]
Electro-Search Algorithm (ESA)	The flow of electrons around an atom's nucleus served as the model for this algorithm [170]
Grasshopper Optimization Algorithm (GOA)	This approach models optimization problems using a group of grasshoppers, treating the newborn stage the same as the adult stage. While adult grasshoppers make larger, abrupt movements, juveniles take smaller, slower steps [171]

Continued

Sperm Motility Algorithm (SMA)	This algorithm is inspired by the human reproductive system. Sperm, serving as search agents, randomly disperse throughout the solution space, mimicking their motility in seeking an ovum. The ovum's chemical secretions attract the sperm toward the optimal solution [172]
Beetle Swarm Optimization Algorithm (BSOA)	This approach is suggested by using beetle foraging concepts to improve swarm optimization performance [173]
Chaotic Bird Swarm Optimization Algorithm	To increase the quality of exploitation, this algorithm blends foraging and privilege behaviors with chaotic-based approaches [174]
Butterfly Optimization Algorithm	In order to achieve global optimization, this method imitates the behavior of butterflies during mating and food hunting [175]
Chaotic grasshopper Optimization Algorithm (CGOA)	This approach integrates chaos theory into the GOA optimization process, using chaotic maps to effectively balancing extraction and exploration [176]
Quantum Dolphin Swarm Algorithm	To get around the local optimum, this technique incorporates the quantum search algorithm with the Dolphin Swarm Algorithm [177]
Emperor Penguins Colony	The method is motivated by Emperor Penguin behavior, specifically their body heat radiation and spiraling movements within their colony [178]
Shell Game Optimization	This technique creates an algorithm for resolving optimization issues by simulating the rules of a game called shell games [179]
Darts Game Optimizer (DGO)	This technique creates an algorithm to solve optimization problems by simulating the laws of the game of darts [180]
Capuchin Search Algorithm (CapSA)	The capuchin monkeys' dynamic behavior served as the model for this algorithm [176]
Red deer Algorithm	The behavior of Scottish red deer, especially their distinctive mating rituals seen throughout the breeding season, serves as the model for this algorithm [177]
Dynamic Multi Objective Optimization Algorithm	A dynamic multi-objective optimization algorithm utilizing classification prediction. The algorithm comprises three key components: a classification strategy, a dynamic classification adjustment mechanism, and tailored prediction strategies that adapt to the outcomes of the dynamic classification [178] [181]
Three main types of MOEA algorithms are evaluation index-based, decomposition-based, and dominant relation-based.	Combining these typical search techniques with the chaotic evolution algorithm to enhance the search functionality of multi-objective optimization algorithms [20]

4. Conclusions

- Multi-objective optimization formulation can be mainly categorized into three families, the first is prioritization, in which optimization starts with the highest priority objective, and the consequence objective. The process starts with the most important one, lexicographic is a good procedure for this formulation, the second family is the preference-based or the weighing formulations, which in turn are divided into, priori, postriori, or interactive approaches, depend on where the decision maker puts his input, and finally the non-preference methods in which tradeoff solutions are generated without any decision-making inputs. Still, multi-objective methods lack a methodology for dealing with multi-

objective Pareto fronts.

- Each family involves several different methods; each one has its own search technique, preference conditions, and attributes. The multi-objective optimization procedure can also be a hybrid of two or more different methods/formulations in order to fix a drawback of individual methods. By merging different techniques, one can create a hybrid that has multiple advantages that cannot be reached when using a single optimization method.
- The ability of an optimization method depends on many factors, like searching techniques, decision makers' interaction, the point at which the decision maker puts his references, type of references, a priori, postriori, interactive, or no-preference based, type and complexity, and number of objectives and constraints in the optimization problems.
- Clustering of solutions is still a drawback of all methods utilizing metaheuristics; this should be considered in future research.

Availability of Data and Materials

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Authors' Contributions

The authors confirm their contribution to the paper as follows: study conception, data collection, and draft manuscript preparation, data analysis, validation, conceptualization, writing—review and editing: M.H., and A.A., all authors reviewed the results and approved the final version of the manuscript.

Acknowledgements

The authors extended their thanks to Cairo University for providing the research facilities to complete the article.

Conflicts of Interest

The authors declared no potential conflict of interest with respect to the research, authorship, and/or publication of this article.

References

- [1] Zitzler, E. and Thiele, L. (1998) An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach. TIK Report, 43.
- [2] Zitzler, E. and Thiele, L. (1999) Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, **3**, 257-271. <https://doi.org/10.1109/4235.797969>
- [3] Deb, K. (2001) Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley & Sons, Inc.
- [4] Cichocka, J. and Browne, W.N. (2016) Multicriteria Optimization in Architectural Design Goal-Oriented Methods and Computational Morphogenesis. In: Symonowicz, J., Pakowska, M. and Nisztuk, M., Eds., *Shapes of Logic. Everything What Surround Us*

- Can Be Described*, Oficyna Wydawnicza Politechniki Wrocławskiej, 107-116.
- [5] Felfernig, A., Stettinger, M., Atas, M., Samer, R., Nerlich, J., Scholz, S., *et al.* (2018) Towards Utility-Based Prioritization of Requirements in Open Source Environments. 2018 *IEEE 26th International Requirements Engineering Conference (RE)*, Banff, 20-24 August 2018, 406-411. <https://doi.org/10.1109/re.2018.00-17>
 - [6] Kifetew, F.M., Susi, A., Muñante, D., Perini, A., Siena, A. and Busetta, P. (2017) Towards Multi-Decision-Maker Requirements Prioritisation via Multi-Objective Optimisation. In: *CAiSE-Forum-DC*, 137-144.
 - [7] Coello, C.A.C. (2007) *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer.
 - [8] Michalewicz, Z. (2013) *How to Solve It: Modern Heuristics*. Springer Science & Business Media.
 - [9] Sharma, S. and Kumar, V. (2022) A Comprehensive Review on Multi-Objective Optimization Techniques: Past, Present and Future. *Archives of Computational Methods in Engineering*, **29**, 5605-5633. <https://doi.org/10.1007/s11831-022-09778-9>
 - [10] Miettinen, K. (1999) *Nonlinear Multiobjective Optimization*. Springer Science & Business Media, Vol. 12, 201-203.
 - [11] Deb, K. (2011) Multi-Objective Optimisation Using Evolutionary Algorithms: An Introduction. In: Wang, L.H., Ng, A.H.C. and Deb, K., Eds., *Multi-Objective Evolutionary Optimisation for Product Design and Manufacturing*, Springer, 3-34. https://doi.org/10.1007/978-0-85729-652-8_1
 - [12] Rajabi Moshtaghi, H., Toloie Eshlaghy, A. and Motadel, M.R. (2021) A Comprehensive Review on Meta-Heuristic Algorithms and Their Classification with Novel Approach. *Journal of Applied Research on Industrial Engineering*, **8**, 63-89.
 - [13] Pearl, J. (1984) *Heuristics*. Addison-Wesley.
 - [14] Ahmed, A.A.M. and Gadallah, M.H. (2025) Improved Strength Pareto Global Adaptive Multi-Objective Optimization Search and Selection Algorithm (ISPGAEA). In: *Advanced Research Trends in Sustainable Solutions, Data Analytics, and Security*, IGI Global, 297-350. <https://doi.org/10.4018/979-8-3693-7117-6.ch009>
 - [15] Agushaka, J.O. and Ezugwu, A.E. (2022) Initialisation Approaches for Population-Based Metaheuristic Algorithms: A Comprehensive Review. *Applied Sciences*, **12**, Article No. 896. <https://doi.org/10.3390/app12020896>
 - [16] Pant, M., Thangaraj, R., Grosan, C. and Abraham, A. (2008) Improved Particle Swarm Optimization with Low-Discrepancy Sequences. 2008 *IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, Hong Kong SAR, 1-6 June 2008, 3011-3018. <https://doi.org/10.1109/cec.2008.4631204>
 - [17] Venkateswarlu, C. (2022) A Metaheuristic Tabu Search Optimization Algorithm: Applications to Chemical and Environmental Processes. In: Tsuzuki, M.S.G., *et al.*, Eds., *Engineering Problems—Uncertainties, Constraints and Optimization Techniques*, IntechOpen. <https://doi.org/10.5772/intechopen.98240>
 - [18] Bandyopadhyay, S. and Saha, S. (2013) Some Single- and Multiobjective Optimization Techniques. In: Bandyopadhyay, S. and Saha, S., Eds., *Unsupervised Classification: Similarity Measures, Classical and Metaheuristic Approaches, and Applications*, Springer, 17-58. https://doi.org/10.1007/978-3-642-32451-2_2
 - [19] Xu, Q., Xu, Z. and Ma, T. (2020) A Survey of Multiobjective Evolutionary Algorithms Based on Decomposition: Variants, Challenges and Future Directions. *IEEE Access*, **8**, 41588-41614. <https://doi.org/10.1109/access.2020.2973670>

- [20] Zhang, Q.F. and Li, H. (2007) MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, **11**, 712-731. <https://doi.org/10.1109/tevc.2007.892759>
- [21] Schütze, O. and Hernández, C. (2021) Archiving in Evolutionary Multi-Objective Optimization: A Short Overview. In: Schütze, O. and Hernández, C., Eds., *Archiving Strategies for Evolutionary Multi-Objective Optimization Algorithms*, Springer International Publishing, 17-20. https://doi.org/10.1007/978-3-030-63773-6_3
- [22] Das, I. and Dennis, J.E. (1998) Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM Journal on Optimization*, **8**, 631-657. <https://doi.org/10.1137/s1052623496307510>
- [23] de S. Motta, R., Afonso, S.M.B. and Lyra, P.R.M. (2012) A Modified NBI and NC Method for the Solution of N-Multiobjective Optimization Problems. *Structural and Multidisciplinary Optimization*, **46**, 239-259. <https://doi.org/10.1007/s00158-011-0729-5>
- [24] Messac, A., Ismail-Yahaya, A. and Mattson, C.A. (2003) The Normalized Normal Constraint Method for Generating the Pareto Frontier. *Structural and Multidisciplinary Optimization*, **25**, 86-98. <https://doi.org/10.1007/s00158-002-0276-1>
- [25] Mueller-Gritschneider, D., Graeb, H. and Schlichtmann, U. (2009) A Successive Approach to Compute the Bounded Pareto Front of Practical Multiobjective Optimization Problems. *SIAM Journal on Optimization*, **20**, 915-934. <https://doi.org/10.1137/080729013>
- [26] Erfani, T. and Utyuzhnikov, S.V. (2011) Directed Search Domain: A Method for Even Generation of the Pareto Frontier in Multiobjective Optimization. *Engineering Optimization*, **43**, 467-484. <https://doi.org/10.1080/0305215x.2010.497185>
- [27] Lotfi, S. and Karimi, F. (2017) A Hybrid MOEA/D-TS for Solving Multi-Objective Problems. *Journal of AI and Data Mining*, **5**, 183-195.
- [28] Alaya, I., Solnon, C. and Ghedira, K. (2007) Ant Colony Optimization for Multi-Objective Optimization Problems. *19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)*, Vol. 1, 450-457. <https://doi.org/10.1109/ictai.2007.108>
- [29] Abdel-Basset, M., Mohamed, R., Mirjalili, S., Chakraborty, R.K. and Ryan, M. (2021) An Efficient Marine Predators Algorithm for Solving Multi-Objective Optimization Problems: Analysis and Validations. *IEEE Access*, **9**, 42817-42844. <https://doi.org/10.1109/access.2021.3066323>
- [30] Kumawat, I.R., Nanda, S.J. and Maddila, R.K. (2017) Multi-Objective Whale Optimization. *TENCON 2017-2017 IEEE Region 10 Conference*, Penang, 5-8 November 2017, 2747-2752. <https://doi.org/10.1109/tencon.2017.8228329>
- [31] Dhiman, G. and Kumar, V. (2018) Multi-Objective Spotted Hyena Optimizer: A Multi-Objective Optimization Algorithm for Engineering Problems. *Knowledge-Based Systems*, **150**, 175-197. <https://doi.org/10.1016/j.knsys.2018.03.011>
- [32] Coello, C.A.C. (2011) An Introduction to Multi-Objective Particle Swarm Optimizers. In: Gaspar-Cunha, A., et al., Eds., *Soft Computing in Industrial Applications*, Springer, 3-12. https://doi.org/10.1007/978-3-642-20505-7_1
- [33] Deb, K. (1997) Limitations of Evolutionary Computation Methods. In: Bäck, T., Fogel, D.B. and Michalewicz, Z., Eds., *Handbook of Evolutionary Computation*, IOP Publishing and Oxford University Press, B2.9.
- [34] Emmerich, M.T.M. and Deutz, A.H. (2018) A Tutorial on Multiobjective Optimization: Fundamentals and Evolutionary Methods. *Natural Computing*, **17**, 585-609.

- <https://doi.org/10.1007/s11047-018-9685-y>
- [35] Umarusman, N. (2013) Min-Max Goal Programming Approach for Solving Multi-Objective de Novo Programming Problems. *International Journal of Operations Research*, **10**, 92-99.
- [36] Bhattacharya, D. and Chakraborty, S. (2018) Solution of the General Multi-Objective De-Novo Programming Problem Using Compensatory Operator under Fuzzy Environment. *Journal of Physics: Conference Series*, **1039**, Article ID: 012012. <https://doi.org/10.1088/1742-6596/1039/1/012012>
- [37] Fonseca, C.M. and Fleming, P.J. (1993) Genetic Algorithms for Multiobjective Optimization: Formulation Discussion and Generalization. *ICGA*, Vol. 93, 416-423.
- [38] Hosseini, H.S. (2009) The Intelligent Water Drops Algorithm: A Nature-Inspired Swarm-Based Optimization Algorithm. *International Journal of Bio-Inspired Computation*, **1**, 71-79. <https://doi.org/10.1504/ijbic.2009.022775>
- [39] Myers, R.H., Montgomery, D.C., Vining, G.G., Borrer, C.M. and Kowalski, S.M. (2004) Response Surface Methodology: A Retrospective and Literature Survey. *Journal of Quality Technology*, **36**, 53-77. <https://doi.org/10.1080/00224065.2004.11980252>
- [40] Mishra, V. and Singh, V. (2016) Vector Evaluated Genetic Algorithm-Based Distributed Query Plan Generation in Distributed Database. In: Afzalpulkar, N., *et al.*, Eds., *Proceedings of the International Conference on Recent Cognizance in Wireless Communication & Image Processing*, Springer India, 325-337. https://doi.org/10.1007/978-81-322-2638-3_37
- [41] Deb, K. and Jain, H. (2014) An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems with Box Constraints. *IEEE Transactions on Evolutionary Computation*, **18**, 577-601. <https://doi.org/10.1109/tevc.2013.2281535>
- [42] Khan Mashwani, W. and Salhi, A. (2012) A Decomposition-Based Hybrid Multi-Objective Evolutionary Algorithm with Dynamic Resource Allocation. *Applied Soft Computing*, **12**, 2765-2780. <https://doi.org/10.1016/j.asoc.2012.03.067>
- [43] Xin, B., Chen, L., Chen, J., Ishibuchi, H., Hirota, K. and Liu, B. (2018) Interactive Multiobjective Optimization: A Review of the State-of-the-Art. *IEEE Access*, **6**, 41256-41279. <https://doi.org/10.1109/access.2018.2856832>
- [44] Sindhya, K., Ruiz, A.B. and Miettinen, K. (2011) A Preference Based Interactive Evolutionary Algorithm for Multi-Objective Optimization: PIE. In: Takahashi, R.H.C., *et al.*, Eds., *International Conference on Evolutionary Multi-Criterion Optimization*, Springer, 212-225. https://doi.org/10.1007/978-3-642-19893-9_15
- [45] Blickle, T. and Thiele, L. (1996) A Comparison of Selection Schemes Used in Evolutionary Algorithms. *Evolutionary Computation*, **4**, 361-394. <https://doi.org/10.1162/evco.1996.4.4.361>
- [46] Miettinen, K., Ruiz, F. and Wierzbicki, A.P. (2008) Introduction to Multiobjective Optimization: Interactive Approaches. In: Branke, J., *et al.*, Eds., *Multiobjective Optimization: Interactive and Evolutionary Approaches*, Springer, 27-57. https://doi.org/10.1007/978-3-540-88908-3_2
- [47] Roy, B. and Mousseau, V. (1996) A Theoretical Framework for Analysing the Notion of Relative Importance of Criteria. *Journal of Multi-Criteria Decision Analysis*, **5**, 145-159. [https://doi.org/10.1002/\(sici\)1099-1360\(199606\)5:2<145::aid-mcda99>3.0.co;2-5](https://doi.org/10.1002/(sici)1099-1360(199606)5:2<145::aid-mcda99>3.0.co;2-5)
- [48] Słowiński, R., Greco, S. and Matarazzo, B. (2002) Axiomatization of Utility, Outranking and Decision Rule Preference Models for Multiple-Criteria Classification Prob-

- lems under Partial Inconsistency with the Dominance Principle. *Control and Cybernetics*, **31**, 1005-1035.
- [49] Wierzbicki, A.P. (1982) A Mathematical Basis for Satisficing Decision Making. *Mathematical Modelling*, **3**, 391-405. [https://doi.org/10.1016/0270-0255\(82\)90038-0](https://doi.org/10.1016/0270-0255(82)90038-0)
- [50] Miettinen, K. and Mäkelä, M.M. (2002) On Scalarizing Functions in Multiobjective Optimization. *OR Spectrum*, **24**, 193-213. <https://doi.org/10.1007/s00291-001-0092-9>
- [51] Nikulin, Y., Miettinen, K. and Mäkelä, M.M. (2010) A New Achievement Scalarizing Function Based on Parameterization in Multiobjective Optimization. *OR Spectrum*, **34**, 69-87. <https://doi.org/10.1007/s00291-010-0224-1>
- [52] Branke, J., Greco, S., Słowiński, R. and Zielniewicz, P. (2015) Learning Value Functions in Interactive Evolutionary Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, **19**, 88-102. <https://doi.org/10.1109/tevc.2014.2303783>
- [53] Figueira, J.R., Greco, S., Mousseau, V. and Słowiński, R. (2008) Interactive Multi-Objective Optimization Using a Set of Additive Value Functions. In: Branke, J., et al., Eds., *Multiobjective Optimization: Interactive and Evolutionary Approaches*, Springer, 97-119. https://doi.org/10.1007/978-3-540-88908-3_4
- [54] Branke, J., Greco, S., Słowiński, R. and Zielniewicz, P. (2010) Interactive Evolutionary Multiobjective Optimization Driven by Robust Ordinal Regression. *Bulletin of the Polish Academy of Sciences: Technical Sciences*, **58**, 347-358. <https://doi.org/10.2478/v10175-010-0033-3>
- [55] Le, N., Xuan, H.N., Brabazon, A. and Thi, T.P. (2016) Complexity Measures in Genetic Programming Learning: A Brief Review. 2016 *IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, 24-29 July 2016, 2409-2416. <https://doi.org/10.1109/cec.2016.7744087>
- [56] Greco, S., Matarazzo, B. and Slowinski, R. (2001) Rough Sets Theory for Multicriteria Decision Analysis. *European Journal of Operational Research*, **129**, 1-47. [https://doi.org/10.1016/s0377-2217\(00\)00167-3](https://doi.org/10.1016/s0377-2217(00)00167-3)
- [57] Ching, L. and Masud, A.S.M. (1979) Multiple Objective Decision Making-Methods and Applications.
- [58] Ben Said, L., Bechikh, S. and Ghedira, K. (2010) The R-Dominance: A New Dominance Relation for Interactive Evolutionary Multicriteria Decision Making. *IEEE Transactions on Evolutionary Computation*, **14**, 801-818. <https://doi.org/10.1109/tevc.2010.2041060>
- [59] Shen, X., Guo, Y., Chen, Q. and Hu, W. (2008) A Multi-Objective Optimization Evolutionary Algorithm Incorporating Preference Information Based on Fuzzy Logic. *Computational Optimization and Applications*, **46**, 159-188. <https://doi.org/10.1007/s10589-008-9189-2>
- [60] Cichocka, J., Browne, W. and Rodriguez, E. (2015) Evolutionary Optimization Processes as Design Tools: Implementation of a Revolutionary Swarm Approach. *Proceedings of 31th International PLEA Conference*, Bologna, 9-11 September 2015, 75-78.
- [61] Mahfoud, S.W. and Goldberg, D.E. (1995) Parallel Recombinative Simulated Annealing: A Genetic Algorithm. *Parallel Computing*, **21**, 1-28. [https://doi.org/10.1016/0167-8191\(94\)00071-h](https://doi.org/10.1016/0167-8191(94)00071-h)
- [62] Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. (1983) Optimization by Simulated Annealing. *Science*, **220**, 671-680. <https://doi.org/10.1126/science.220.4598.671>
- [63] Bäck, T., Fogel, D.B. and Michalewicz, Z. (1997) Handbook of Evolutionary Computation. *Release*, **97**, B1.

- [64] Asoh, H. and Mühlenbein, H. (1994) On the Mean Convergence Time of Evolutionary Algorithms without Selection and Mutation. In: Davidor, Y., *et al.*, Eds., *International Conference on Parallel Problem Solving from Nature*, Springer, 88-97. https://doi.org/10.1007/3-540-58484-6_253
- [65] Wan, W. and Birch, J.B. (2013) An Improved Hybrid Genetic Algorithm with a New Local Search Procedure. *Journal of Applied Mathematics*, **2013**, Article ID: 103591. <https://doi.org/10.1155/2013/103591>
- [66] Seront, G. and Bersini, H. (2000) A New GA-Local Search Hybrid for Continuous Optimization Based on Multi-Level Single Linkage Clustering. *Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation*, Las Vegas, 8-12 July 2000, 90-95.
- [67] Haupt, R.L. and Haupt, S.E. (2003) *Practical Genetic Algorithms*. Wiley. <https://doi.org/10.1002/0471671746>
- [68] Fathollahi-Fard, A.M., Hajiaghayi-Keshteli, M. and Tavakkoli-Moghaddam, R. (2020) Red Deer Algorithm (RDA): A New Nature-Inspired Meta-Heuristic. *Soft Computing*, **24**, 14637-14665. <https://doi.org/10.1007/s00500-020-04812-z>
- [69] Wu, F., Wang, W., Chen, J. and Wang, Z. (2023) A Dynamic Multi-Objective Optimization Method Based on Classification Strategies. *Scientific Reports*, **13**, Article No. 15221. <https://doi.org/10.1038/s41598-023-41855-2>
- [70] Sharifi, M.R., Akbarifard, S., Qaderi, K. and Madadi, M.R. (2021) A New Optimization Algorithm to Solve Multi-Objective Problems. *Scientific Reports*, **11**, Article No. 20326. <https://doi.org/10.1038/s41598-021-99617-x>
- [71] Cerda-Flores, S.C., Rojas-Punzo, A.A. and Nápoles-Rivera, F. (2022) Applications of Multi-Objective Optimization to Industrial Processes: A Literature Review. *Processes*, **10**, Article No. 133. <https://doi.org/10.3390/pr10010133>
- [72] Fogel, D.B. and Fogel, L.J. (1996) An Introduction to Evolutionary Programming. In: Alliot, J.-M., *et al.*, Eds., *European Conference on Artificial Evolution*, Springer, 21-33. https://doi.org/10.1007/3-540-61108-8_28
- [73] Holland, J.H. (1992) *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. The MIT Press. <https://doi.org/10.7551/mitpress/1090.001.0001>
- [74] Glover, F. (1977) Heuristics for Integer Programming Using Surrogate Constraints. *Decision Sciences*, **8**, 156-166. <https://doi.org/10.1111/j.1540-5915.1977.tb01074.x>
- [75] Glover, F. (1986) Future Paths for Integer Programming and Links to Artificial Intelligence. *Computers & Operations Research*, **13**, 533-549. [https://doi.org/10.1016/0305-0548\(86\)90048-1](https://doi.org/10.1016/0305-0548(86)90048-1)
- [76] Reynolds, R.G. (1994) An Introduction to Cultural Algorithms. *Proceedings of the 3rd Annual Conference on Evolutionary Programming*, Vol. 24, 131-139.
- [77] Kennedy, J. and Eberhart, R. (1995) Particle Swarm Optimization. *Proceedings of ICNN'95-International Conference on Neural Networks*, Vol. 4, 1942-1948. <https://doi.org/10.1109/icnn.1995.488968>
- [78] Dorigo, M., Maniezzo, V. and Coloni, A. (1996) Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, **26**, 29-41. <https://doi.org/10.1109/3477.484436>
- [79] Storn, R. and Price, K. (1997) Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, **11**, 341-359. <https://doi.org/10.1023/a:1008202821328>
- [80] Mladenović, N. and Hansen, P. (1997) Variable Neighborhood Search. *Computers &*

- Operations Research*, **24**, 1097-1100. [https://doi.org/10.1016/s0305-0548\(97\)00031-2](https://doi.org/10.1016/s0305-0548(97)00031-2)
- [81] Kim, H. and Ahn, B. (2001) A New Evolutionary Algorithm Based on Sheep Flocks Heredity Model. 2001 *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Vol. 2, 514-517.
- [82] Geem, Z.W., Kim, J.H. and Loganathan, G.V. (2001) A New Heuristic Optimization Algorithm: Harmony Search. *Simulation*, **76**, 60-68.
<https://doi.org/10.1177/003754970107600201>
- [83] Passino, K.M. (2002) Biomimicry of Bacterial Foraging for Distributed Optimization and Control. *IEEE Control Systems Magazine*, **22**, 52-67.
- [84] Xie, X.F., Zhang, W.J. and Yang, Z.L. (2002) Social Cognitive Optimization for Non-linear Programming Problems. *Proceedings. International Conference on Machine Learning and Cybernetics*, Vol. 2, 779-783.
<https://doi.org/10.1109/icmlc.2002.1174487>
- [85] Eusuff, M.M. and Lansey, K.E. (2003) Optimization of Water Distribution Network Design Using the Shuffled Frog Leaping Algorithm. *Journal of Water Resources Planning and Management*, **129**, 210-225.
[https://doi.org/10.1061/\(asce\)0733-9496\(2003\)129:3\(210\)](https://doi.org/10.1061/(asce)0733-9496(2003)129:3(210))
- [86] Birbil, Ş.İ. and Fang, S.C. (2003) An Electromagnetism-Like Mechanism for Global Optimization. *Journal of Global Optimization*, **25**, 263-282.
<https://doi.org/10.1023/a:1022452626305>
- [87] Hsiao, Y.T., Chuang, C.L., Jiang, J.A. and Chien, C.C. (2005) A Novel Optimization Algorithm: Space Gravitational Optimization. 2005 *IEEE International Conference on Systems, Man and Cybernetics*, Vol. 3, 2323-2328.
<https://doi.org/10.1109/icsmc.2005.1571495>
- [88] Sacco, W.F. and de Oliveira, C.R. (2005) A New Stochastic Optimization Algorithm Based on Particle Collisions. *Transactions of the American Nuclear Society*, **92**, 657-659.
- [89] Erol, O.K. and Eksin, I. (2006) A New Optimization Method: Big Bang-Big Crunch. *Advances in Engineering Software*, **37**, 106-111.
<https://doi.org/10.1016/j.advengsoft.2005.04.005>
- [90] He, S., Wu, Q.H. and Saunders, J.R. (2006) A Novel Group Search Optimizer Inspired by Animal Behavioural Ecology. 2006 *IEEE International Conference on Evolutionary Computation*, Vancouver, 16-21 July 2006, 1272-1278.
<https://doi.org/10.1109/cec.2006.1688455>
- [91] Mehrabian, A.R. and Lucas, C. (2006) A Novel Numerical Optimization Algorithm Inspired from Weed Colonization. *Ecological Informatics*, **1**, 355-366.
<https://doi.org/10.1016/j.ecoinf.2006.07.003>
- [92] Du, H., Wu, X. and Zhuang, J. (2006) Small-World Optimization Algorithm for Function Optimization. *Advances in Natural Computation: Second International Conference, ICNC 2006*, Xi'an, 24-28 September 2006, 264-273.
https://doi.org/10.1007/11881223_33
- [93] Chu, S.C., Tsai, P.W. and Pan, J.S. (2006) Cat Swarm Optimization. *PRICAI 2006: Trends in Artificial Intelligence: 9th Pacific Rim International Conference on Artificial Intelligence*, Guilin, 7-11 August 2006, 854-858.
https://doi.org/10.1007/978-3-540-36668-3_94
- [94] Karci, A. and Alatas, B. (2006) Thinking Capability of Saplings Growing up Algorithm. In: Corchado, E., et al., Eds., *Intelligent Data Engineering and Automated Learning*, Springer, 386-393. https://doi.org/10.1007/11875581_47

- [95] Atashpaz-Gargari, E. and Lucas, C. (2007) Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition. 2007 *IEEE Congress on Evolutionary Computation*, Singapore, 25-28 September 2007, 4661-4667. <https://doi.org/10.1109/cec.2007.4425083>
- [96] Karaboga, D. and Basturk, B. (2007) A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. *Journal of Global Optimization*, **39**, 459-471. <https://doi.org/10.1007/s10898-007-9149-x>
- [97] Formato, R.A. (2007) Central Force Optimization: A New Metaheuristic with Applications in Applied Electromagnetics. *Progress in Electromagnetics Research*, **77**, 425-491. <https://doi.org/10.2528/pier07082403>
- [98] Chuang, C.L. and Jiang, J.A. (2007) Integrated Radiation Optimization: Inspired by the Gravitational Radiation in the Curvature of Space-Time. 2007 *IEEE Congress on Evolutionary Computation*, Singapore, 25-28 September 2007, 3157-3164. <https://doi.org/10.1109/cec.2007.4424875>
- [99] Lamberti, L. and Pappalettere, C. (2007) Weight Optimization of Skeletal Structures with Multi-Point Simulated Annealing. *Computer Modeling in Engineering and Sciences*, **18**, 183.
- [100] Rabanal, P., Rodríguez, I. and Rubio, F. (n.d.) Using River Formation Dynamics to Design Heuristic Algorithms. In: Akl, S.G., *et al.*, Eds., *International Conference on Unconventional Computation*, Springer, 163-177. https://doi.org/10.1007/978-3-540-73554-0_16
- [101] Kripka, M. and Kripka, R.M.L. (2008) Big Crunch Optimization Method. *International Conference on Engineering Optimization*, Rio de Janeiro, 1-5 June 2008, 1-7.
- [102] Simon, D. (2008) Biogeography-Based Optimization. *IEEE Transactions on Evolutionary Computation*, **12**, 702-713. <https://doi.org/10.1109/tevc.2008.919004>
- [103] Yang, X.S. (2009) Firefly Algorithms for Multimodal Optimization. In: Watanabe, O. and Zeugmann, T., Eds., *International Symposium on Stochastic Algorithms*, Springer, 169-178. https://doi.org/10.1007/978-3-642-04944-6_14
- [104] Premaratne, U., Samarabandu, J. and Sidhu, T. (2009) A New Biologically Inspired Optimization Algorithm. 2009 *International Conference on Industrial and Information Systems (ICIIS)*, Peradeniya, 28-31 December 2009, 279-284. <https://doi.org/10.1109/iciinfs.2009.5429852>
- [105] Rashedi, E., Nezamabadi-Pour, H. and Saryazdi, S. (2009) GSA: A Gravitational Search Algorithm. *Information Sciences*, **179**, 2232-2248. <https://doi.org/10.1016/j.ins.2009.03.004>
- [106] Yang, X.S. and Deb, S. (2009) Cuckoo Search via Lévy Flights. 2009 *World Congress on Nature & Biologically Inspired Computing (NaBIC)*, Coimbatore, 9-11 December 2009, 210-214. <https://doi.org/10.1109/nabic.2009.5393690>
- [107] Oftadeh, R. and Mahjoob, M.J. (2009) A New Meta-Heuristic Optimization Algorithm: Hunting Search. 2009 *5th International Conference on Soft Computing, Computing with Words and Perceptions in System Analysis, Decision and Control*, Famagusta, 2-4 September 2009, 1-5. <https://doi.org/10.1109/icscw.2009.5379451>
- [108] Xie, L., Zeng, J. and Cui, Z. (2009) General Framework of Artificial Physics Optimization Algorithm. 2009 *World Congress on Nature & Biologically Inspired Computing (NaBIC)*, Coimbatore, 9-11 December 2009, 1321-1326. <https://doi.org/10.1109/nabic.2009.5393736>
- [109] Das, S., Chowdhury, A. and Abraham, A. (2009) A Bacterial Evolutionary Algorithm for Automatic Data Clustering. 2009 *IEEE Congress on Evolutionary Computation*,

- Trondheim, 18-21 May 2009, 2403-2410. <https://doi.org/10.1109/cec.2009.4983241>
- [110] Zhang, L.M., Dahlmann, C. and Zhang, Y. (2009) Human-Inspired Algorithms for Continuous Function Optimization. 2009 *IEEE International Conference on Intelligent Computing and Intelligent Systems*, Vol. 1, 318-321. <https://doi.org/10.1109/icipsys.2009.5357838>
- [111] Kashan, A.H. (2009) League Championship Algorithm: A New Algorithm for Numerical Function Optimization. 2009 *International Conference of Soft Computing and Pattern Recognition*, Malacca, 4-7 December 2009, 43-48. <https://doi.org/10.1109/socpar.2009.21>
- [112] Chen, S. (2009) Locust Swarms—A New Multi-Optima Search Technique. 2009 *IEEE Congress on Evolutionary Computation*, Trondheim, 18-21 May 2009, 1745-1752. <https://doi.org/10.1109/cec.2009.4983152>
- [113] Iordache, S. (2010) Consultant-Guided Search: A New Metaheuristic for Combinatorial Optimization Problems. *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, Portland, 7-11 July 2010, 225-232. <https://doi.org/10.1145/1830483.1830526>
- [114] Yang, X.S. (2010) A New Metaheuristic Bat-Inspired Algorithm. In: González, J.R., et al., Eds., *Nature Inspired Cooperative Strategies for Optimization*, Springer, 65-74. https://doi.org/10.1007/978-3-642-12538-6_6
- [115] Kaveh, A. and Talatahari, S. (2010) A Novel Heuristic Optimization Method: Charged System Search. *Acta Mechanica*, **213**, 267-289. <https://doi.org/10.1007/s00707-009-0270-4>
- [116] Lam, A.Y.S. and Li, V.O.K. (2010) Chemical-Reaction-Inspired Metaheuristic for Optimization. *IEEE Transactions on Evolutionary Computation*, **14**, 381-399. <https://doi.org/10.1109/tevc.2009.2033580>
- [117] Yang, X. and Deb, S. (2010) Eagle Strategy Using Lévy Walk and Firefly Algorithms for Stochastic Optimization. In: González, J.R., et al., Eds., *Nature Inspired Cooperative Strategies for Optimization*, Springer, 101-111. https://doi.org/10.1007/978-3-642-12538-6_9
- [118] Xu, Y., Cui, Z. and Zeng, J. (2010) Social Emotional Optimization Algorithm for Non-linear Constrained Optimization Problems. *1st International Conference on Swarm, Evolutionary, and Memetic Computing, SEMCCO 2010*, Chennai, 16-18 December 2010, 583-590. https://doi.org/10.1007/978-3-642-17563-3_68
- [119] Shah-Hosseini, H. (2011) Otsu's Criterion-Based Multilevel Thresholding by a Nature-Inspired Metaheuristic Called Galaxy-Based Search Algorithm. 2011 *3rd World Congress on Nature and Biologically Inspired Computing*, Salamanca, 19-21 October 2011, 383-388. <https://doi.org/10.1109/nabic.2011.6089621>
- [120] Shah-Hosseini, H. (2011) Principal Components Analysis by the Galaxy-Based Search Algorithm: A Novel Metaheuristic for Continuous Optimisation. *International Journal of Computational Science and Engineering*, **6**, 132-140. <https://doi.org/10.1504/ijcse.2011.041221>
- [121] Tamura, K. and Yasuda, K. (2011) Spiral Dynamics Inspired Optimization. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, **15**, 1116-1122. <https://doi.org/10.20965/jaciii.2011.p1116>
- [122] Rao, R.V., Savsani, V.J. and Vakharia, D.P. (2011) Teaching-Learning-Based Optimization: A Novel Method for Constrained Mechanical Design Optimization Problems. *Computer-Aided Design*, **43**, 303-315. <https://doi.org/10.1016/j.cad.2010.12.015>
- [123] Shayeghi, H. and Dadashpour, J. (2012) Anarchic Society Optimization Based PID Control of an Automatic Voltage Regulator (AVR) System. *Electrical and Electronic*

- Engineering*, **2**, 199-207. <https://doi.org/10.5923/j.eee.20120204.05>
- [124] Sakulin, A. and Puangdownreong, D. (2012) A Novel Meta-Heuristic Optimization Algorithm: Current Search. *Proceedings of the 11th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases*, Cambridge, 22-24 February 2012, 125-130. <https://dl.acm.org/doi/proceedings/10.5555/2183067>
- [125] Eskandar, H., Sadollah, A., Bahreininejad, A. and Hamdi, M. (2012) Water Cycle Algorithm—A Novel Metaheuristic Optimization Method for Solving Constrained Engineering Optimization Problems. *Computers & Structures*, **110**, 151-166. <https://doi.org/10.1016/j.compstruc.2012.07.010>
- [126] Tang, R., Fong, S., Yang, X. and Deb, S. (2012) Wolf Search Algorithm with Ephemeral Memory. *7th International Conference on Digital Information Management (ICDIM 2012)*, Macau SAR, 22-24 August 2012, 165-172. <https://doi.org/10.1109/icdim.2012.6360147>
- [127] Sadollah, A., Bahreininejad, A., Eskandar, H. and Hamdi, M. (2012) Mine Blast Algorithm for Optimization of Truss Structures with Discrete Variables. *Computers & Structures*, **102**, 49-63. <https://doi.org/10.1016/j.compstruc.2012.03.013>
- [128] Yan, G.W. and Hao, Z.J. (2013) A Novel Optimization Algorithm Based on Atmosphere Clouds Model. *International Journal of Computational Intelligence and Applications*, **12**, Article ID: 1350002. <https://doi.org/10.1142/s1469026813500028>
- [129] Hatamlou, A. (2013) Black Hole: A New Heuristic Optimization Approach for Data Clustering. *Information Sciences*, **222**, 175-184. <https://doi.org/10.1016/j.ins.2012.08.023>
- [130] Sur, C., Sharma, S. and Shukla, A. (2013) Egyptian Vulture Optimization Algorithm—A New Nature Inspired Meta-Heuristics for Knapsack Problem. *The 9th International Conference on Computing and Information Technology (IC2IT2013)*, North Bangkok, 9-10 May 2013, 227-237. https://doi.org/10.1007/978-3-642-37371-8_26
- [131] Gheraibia, Y. and Moussaoui, A. (2013) Penguins Search Optimization Algorithm (PeSOA). *Recent Trends in Applied Artificial Intelligence: 26th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2013*, Amsterdam, 17-21 June 2013, 222-231. https://doi.org/10.1007/978-3-642-38577-3_23
- [132] Neshat, M., Sepidnam, G. and Sargolzaei, M. (2012) Swallow Swarm Optimization Algorithm: A New Method to Optimization. *Neural Computing and Applications*, **23**, 429-454. <https://doi.org/10.1007/s00521-012-0939-9>
- [133] Mirjalili, S., Mirjalili, S.M. and Lewis, A. (2014) Grey Wolf Optimizer. *Advances in Engineering Software*, **69**, 46-61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
- [134] Osaba, E., Diaz, F. and Onieva, E. (2014) Golden Ball: A Novel Meta-Heuristic to Solve Combinatorial Optimization Problems Based on Soccer Concepts. *Applied Intelligence*, **41**, 145-166. <https://doi.org/10.1007/s10489-013-0512-y>
- [135] Thierens, D., Goldberg, D.E. and Pereira, A.G. (1998) Domino Convergence, Drift, and the Temporal-Saliency Structure of Problems. 1998 *IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence*, Anchorage, 4-9 May 1998, 535-540. <https://doi.org/10.1109/iccc.1998.700085>
- [136] Li, X., Zhang, J. and Yin, M. (2013) Animal Migration Optimization: An Optimization Algorithm Inspired by Animal Migration Behavior. *Neural Computing and Applications*, **24**, 1867-1877. <https://doi.org/10.1007/s00521-013-1433-8>
- [137] Moosavian, N. and Roodsari, B.K. (2014) Soccer League Competition Algorithm: A

- Novel Meta-Heuristic Algorithm for Optimal Design of Water Distribution Networks. *Swarm and Evolutionary Computation*, **17**, 14-24.
<https://doi.org/10.1016/j.swevo.2014.02.002>
- [138] Moosavian, N. and Roodsari, B.K. (2014) Soccer League Competition Algorithm, a New Method for Solving Systems of Nonlinear Equations. *International Journal of Intelligence Science*, **4**, 7-16. <https://doi.org/10.4236/ijis.2014.41002>
- [139] Meng, X., Liu, Y., Gao, X. and Zhang, H. (2014) A New Bio-Inspired Algorithm: Chicken Swarm Optimization. *Advances in Swarm Intelligence. 5th International Conference, ICSI 2014*, Hefei, 17-20 October 2014, 86-94.
https://doi.org/10.1007/978-3-319-11857-4_10
- [140] Ghaemi, M. and Feizi-Derakhshi, M. (2014) Forest Optimization Algorithm. *Expert Systems with Applications*, **41**, 6676-6687.
<https://doi.org/10.1016/j.eswa.2014.05.009>
- [141] Hatamlou, A. (2014) Heart: A Novel Optimization Algorithm for Cluster Analysis. *Progress in Artificial Intelligence*, **2**, 167-173.
<https://doi.org/10.1007/s13748-014-0046-5>
- [142] De Melo, V.V. (2014) Kaizen Programming. *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, Vancouver, 12-16 July 2014, 895-902. <https://doi.org/10.1145/2576768.2598264>
- [143] Ghorbani, N. and Babaei, E. (2014) Exchange Market Algorithm. *Applied Soft Computing*, **19**, 177-187. <https://doi.org/10.1016/j.asoc.2014.02.006>
- [144] Odili, J.B., Kahar, M.N.M. and Anwar, S. (2015) African Buffalo Optimization: A Swarm-Intelligence Technique. *Procedia Computer Science*, **76**, 443-448.
<https://doi.org/10.1016/j.procs.2015.12.291>
- [145] Wang, G., Deb, S. and Coelho, L.d.S. (2015) Elephant Herding Optimization. 2015 *3rd International Symposium on Computational and Business Intelligence (ISCBI)*, Bali, 7-9 December 2015, 1-5. <https://doi.org/10.1109/iscbi.2015.8>
- [146] Javidy, B., Hatamlou, A. and Mirjalili, S. (2015) Ions Motion Algorithm for Solving Optimization Problems. *Applied Soft Computing*, **32**, 72-79.
<https://doi.org/10.1016/j.asoc.2015.03.035>
- [147] Beiranvand, H. and Rokrok, E. (2015) General Relativity Search Algorithm: A Global Optimization Approach. *International Journal of Computational Intelligence and Applications*, **14**, Article ID: 1550017. <https://doi.org/10.1142/s1469026815500170>
- [148] Chen, C.C., Tsai, Y.C., Liu, I.I., Lai, C.C., Yeh, Y.T., Kuo, S.Y., *et al.* (2015) A Novel Metaheuristic: Jaguar Algorithm with Learning Behavior. 2015 *IEEE International Conference on Systems, Man, and Cybernetics*, Hong Kong SAR, 9-12 October 2015, 1595-1600. <https://doi.org/10.1109/smc.2015.282>
- [149] Kashan, A.H. (2015) A New Metaheuristic for Optimization: Optics Inspired Optimization (OIO). *Computers & Operations Research*, **55**, 99-125.
<https://doi.org/10.1016/j.cor.2014.10.011>
- [150] Merrikh-Bayat, F. (2015) The Runner-Root Algorithm: A Metaheuristic for Solving Unimodal and Multimodal Optimization Problems Inspired by Runners and Roots of Plants in Nature. *Applied Soft Computing*, **33**, 292-303.
<https://doi.org/10.1016/j.asoc.2015.04.048>
- [151] Doğan, B. and Ölmez, T. (2015) A New Metaheuristic for Numerical Function Optimization: Vortex Search Algorithm. *Information Sciences*, **293**, 125-145.
<https://doi.org/10.1016/j.ins.2014.08.053>
- [152] Salimi, H. (2015) Stochastic Fractal Search: A Powerful Metaheuristic Algorithm.

- Knowledge-Based Systems*, **75**, 1-18. <https://doi.org/10.1016/j.knsys.2014.07.025>
- [153] Tilahun, S.L. and Ong, H.C. (2015) Prey-Predator Algorithm: A New Metaheuristic Algorithm for Optimization Problems. *International Journal of Information Technology & Decision Making*, **14**, 1331-1352. <https://doi.org/10.1142/s021962201450031x>
- [154] Zheng, Y. (2015) Water Wave Optimization: A New Nature-Inspired Metaheuristic. *Computers & Operations Research*, **55**, 1-11. <https://doi.org/10.1016/j.cor.2014.10.008>
- [155] Findik, O. (2015) Bull Optimization Algorithm Based on Genetic Operators for Continuous Optimization Problems. *Turkish Journal of Electrical Engineering and Computer Sciences*, **23**, 2225-2239. <https://doi.org/10.3906/elk-1307-123>
- [156] Deb, S., Fong, S. and Tian, Z. (2015) Elephant Search Algorithm for Optimization Problems. 2015 10th International Conference on Digital Information Management (ICDIM), Jeju, 21-23 October 2015, 249-255. <https://doi.org/10.1109/icdim.2015.7381893>
- [157] Mirjalili, S. (2015) The Ant Lion Optimizer. *Advances in Engineering Software*, **83**, 80-98. <https://doi.org/10.1016/j.advengsoft.2015.01.010>
- [158] Yazdani, M. and Jolai, F. (2015) Lion Optimization Algorithm (LOA): A Nature-Inspired Metaheuristic Algorithm. *Journal of Computational Design and Engineering*, **3**, 24-36. <https://doi.org/10.1016/j.jcde.2015.06.003>
- [159] Mirjalili, S. and Lewis, A. (2016) The Whale Optimization Algorithm. *Advances in Engineering Software*, **95**, 51-67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
- [160] Topal, A.O. and Altun, O. (2016) A Novel Meta-Heuristic Algorithm: Dynamic Virtual Bats Algorithm. *Information Sciences*, **354**, 222-235. <https://doi.org/10.1016/j.ins.2016.03.025>
- [161] Kaveh, A. and Zolghadr, A. (2016) A Novel Meta-Heuristic Algorithm: Tug of War Optimization. *International Journal of Optimization in Civil Engineering*, **6**, 469-492.
- [162] Cuevas Juarez, J.R. and Liang, Y.C. (2015) A Novel Metaheuristic for Continuous Optimization Problems: Virus Optimization Algorithm. *Engineering Optimization*, **48**, 73-93. <https://doi.org/10.1080/0305215x.2014.994868>
- [163] Li, M.D., Zhao, H., Weng, X.W. and Han, T. (2016) A Novel Nature-Inspired Algorithm for Optimization: Virus Colony Search. *Advances in Engineering Software*, **92**, 65-88. <https://doi.org/10.1016/j.advengsoft.2015.11.004>
- [164] Askarzadeh, A. (2016) A Novel Metaheuristic Method for Solving Constrained Engineering Optimization Problems: Crow Search Algorithm. *Computers & Structures*, **169**, 1-12. <https://doi.org/10.1016/j.compstruc.2016.03.001>
- [165] Mirjalili, S. (2015) Dragonfly Algorithm: A New Meta-Heuristic Optimization Technique for Solving Single-Objective, Discrete, and Multi-Objective Problems. *Neural Computing and Applications*, **27**, 1053-1073. <https://doi.org/10.1007/s00521-015-1920-1>
- [166] Ibrahim, M.K. and Ali, R.S. (2016) Novel Optimization Algorithm Inspired by Camel Traveling Behavior. *Iraqi Journal for Electrical and Electronic Engineering*, **12**, 167-177. <https://doi.org/10.37917/ijeee.12.2.7>
- [167] Kaveh, A. and Bakhshpoori, T. (2016) Water Evaporation Optimization: A Novel Physically Inspired Optimization Algorithm. *Computers & Structures*, **167**, 69-85. <https://doi.org/10.1016/j.compstruc.2016.01.008>
- [168] Kaveh, A. and Dadras, A. (2017) A Novel Meta-Heuristic Optimization Algorithm: Thermal Exchange Optimization. *Advances in Engineering Software*, **110**, 69-84.

- <https://doi.org/10.1016/j.advengsoft.2017.03.014>
- [169] Tabari, A. and Ahmad, A. (2017) A New Optimization Method: Electro-Search Algorithm. *Computers & Chemical Engineering*, **103**, 1-11. <https://doi.org/10.1016/j.compchemeng.2017.01.046>
- [170] Mirjalili, S., Sandemi, S. and Lewis, A. (2017) Grasshopper Optimisation Algorithm: Theory and Application. *Advances in Engineering Software*, **105**, 30-47. <https://doi.org/10.1016/j.advengsoft.2017.01.004>
- [171] Hezam, I.M. and Raouf, O.A. (2017) Sperm Motility Algorithm: A Novel Metaheuristic Approach for Global Optimisation. *International Journal of Operational Research*, **28**, 143-163. <https://doi.org/10.1504/ijor.2017.081473>
- [172] Yang, L. and Wang, T. (2018) Beetle Swarm Optimization Algorithm: Theory and Application.
- [173] Houssein, E.H., Hassanien, A.E. and Ismail, F.H. (2018) Chaotic Bird Swarm Optimization Algorithm. In: Hassanien, A.E., et al., Eds., *International Conference on Advanced Intelligent Systems and Informatics*, Springer, 294-303. https://doi.org/10.1007/978-3-319-99010-1_27
- [174] Arora, S. and Singh, S. (2018) Butterfly Optimization Algorithm: A Novel Approach for Global Optimization. *Soft Computing*, **23**, 715-734. <https://doi.org/10.1007/s00500-018-3102-4>
- [175] Arora, S. and Anand, P. (2018) Chaotic Grasshopper Optimization Algorithm for Global Optimization. *Neural Computing and Applications*, **31**, 4385-4405. <https://doi.org/10.1007/s00521-018-3343-2>
- [176] Qiao, W. and Yang, Z. (2019) Solving Large-Scale Function Optimization Problem by Using a New Metaheuristic Algorithm Based on Quantum Dolphin Swarm Algorithm. *IEEE Access*, **7**, 138972-138989. <https://doi.org/10.1109/access.2019.2942169>
- [177] Harifi, S., Khalilian, M., Mohammadzadeh, J. and Ebrahimnejad, S. (2019) Emperor Penguins Colony: A New Metaheuristic Algorithm for Optimization. *Evolutionary Intelligence*, **12**, 211-226. <https://doi.org/10.1007/s12065-019-00212-x>
- [178] Dehghani, M., Montazeri, Z., Malik, O., Givi, H. and Guerrero, J. (2020) Shell Game Optimization: A Novel Game-Based Algorithm. *International Journal of Intelligent Engineering and Systems*, **13**, 246-255. <https://doi.org/10.22266/ijies2020.0630.23>
- [179] Dehghani, M., Montazeri, Z., Givi, H., Guerrero, J. and Dhiman, G. (2020) Darts Game Optimizer: A New Optimization Technique Based on Darts Game. *International Journal of Intelligent Engineering and Systems*, **13**, 286-294. <https://doi.org/10.22266/ijies2020.1031.26>
- [180] Braik, M., Sheta, A. and Al-Hiary, H. (2020) A Novel Meta-Heuristic Search Algorithm for Solving Optimization Problems: Capuchin Search Algorithm. *Neural Computing and Applications*, **33**, 2515-2547. <https://doi.org/10.1007/s00521-020-05145-6>
- [181] Wang, Z., Pei, Y. and Li, J. (2023) A Survey on Search Strategy of Evolutionary Multi-Objective Optimization Algorithms. *Applied Sciences*, **13**, Article No. 4643. <https://doi.org/10.3390/app13074643>

List of Abbreviations

MOEA	Multi Objective Optimization Algorithms
SPEA2	Strength Pareto Evolutionary Algorithm 2
NSGA-II	Non-dominated Sorting Genetic Algorithm II
PSO	particle swarm optimization
MOPSO	Multi-Objective Particle Swarm Optimization
NBI	Normal Boundary Intersection
NBI _m	Modified Normal Boundary Intersection
NC	Normal Constraint
SPO	Successive Pareto Optimization
DSD	Directed Search Domain
MOACO	Multi-objective Ant Colony Optimization
MMPOA	Multi-Objective Marine Predator Algorithm
WOA	Multi-objective Whale optimization Algorithm
MOSHO	Multi-objective Spotted Hyena Optimization
EMO	Evolutionary Multi-objective Optimization
MOO	Multi-objective optimization
ACO	Ant Colony Optimization
EA	Evolutionary Algorithms
PAES	Pareto archived Evolutionary Algorithm
IBEA	Indicator-based algorithms
SMS-EMOA	S-Metric Selection Evolutionary Multi-Objective Algorithm
POSEA	Portfolio Selection Multi-Objective Evolutionary Algorithm
HypE	Hypervolume Estimation Algorithm
MOEA/D	Multi Objective Optimization based on decomposition Algorithm
MOEA/D-DRA	Multi Objective Optimization based on dynamic resource allocation Algorithm
MOEA/DD	Multi Objective Optimization based on dominance and decomposition Algorithm
IMO	Interactive Optimization Methods
MCDM	Multiple criteria decisions making