Results in Control and Optimization 13 (2023) 100315

Contents lists available at ScienceDirect

Results in Control and Optimization

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

A review of classical methods and Nature-Inspired Algorithms (NIAs) for optimization problems

Pawan Kumar Mandal

Indian Institute of Technology Mandi, Mandi 175075, Himachal Pradesh, India

ARTICLE INFO

Keywords:
Classical optimization methods
Nature-Inspired Algorithms (NIAs)
Evolutionary Algorithms (EAs)
Swarm Intelligence (SI) based algorithms
Linear Progrmming (LP)
Mixed Integer Programming (MIP)
Quadratic Programming (QP)

ABSTRACT

Optimization techniques are among the most promising methods to deal with real-world problems, consisting of several objective functions and constraints. Over the decades, many methods have come into existence to solve optimization problems. However, the complexity of these problems is increasing over time. Thereby, it opens up a field of research in developing a robust procedure compatible with such complex optimization problems that provide optimal solutions best suited to the needs of the decision-makers. This review paper presents a survey of the recent use of classical methods and Nature-Inspired Algorithms (NIAs) to solve single and multiple objective problems of optimization in diverse application areas. Moreover, this study briefly describes these widely used solution methods based on the classification of classical approaches and NIAs. Recently published articles based on real-world applications have been included to demonstrate the advantages of each solution technique. In addition, research gaps involving various techniques and future prospects within this field are discussed.

1. Introduction

The process of optimizing an objective function (or collection of objective functions) in a systematic manner is said to be a single-objective optimization (or multi-objective/vector optimization) [1]. There are many real-world problems that can be transformed into single or multi-objective optimization problems, which can be solved by applying specialized and effective optimization techniques to find an optimal solution [2]. Early studies have efficiently dealt with these problems with classical optimization methods. Techniques such as linear programming, quadratic programming, convex optimization, and others were widely applied to solve optimization problems of that era [3–5]. The origin of duality in mathematical programming also played an important role in solving these problems under such circumstances where there has been difficulty directly solving some of the optimization problems [6,7]. However, the presence of nonconvexity in the objective functions or constraints limited the effectiveness of duality for such optimization problems.

Over time, these problems become increasingly complex due to the addition in the number of decision variables, objective functions, and some hard constraints, making them NP-hard problems that cannot be easily tackled by classical methods [8]. By resembling decision-making before a search, traditional approaches for producing the Pareto-optimal set merge the objectives into a single, parameterized objective function [9–11]. On the other hand, the parameters of this function are systematically adjusted by the optimizer rather than specified by the decision maker [12]. In addition, the classical methods are well-suited when the specific optimization problem has relevant mathematical structures such as continuity, differentiability, and convexity [13]. Although, these properties are often not present in the optimization problems of real-world scenarios, which are usually non-smooth and may not be solved by methods designed for smooth optimization problems [14]. Therefore, the applicability of classical approaches is limited

E-mail address: pkmandal321@gmail.com.

https://doi.org/10.1016/j.rico.2023.100315
Received 23 March 2023; Received in revised form 28 September 2023; Accepted 5 October 2023
Available online 7 October 2023

to some specific domains [15]. Nevertheless, traditional approaches remain widely applied to solve many optimization problems because they are easy to use and always provide optimality [16–18].

Classical mathematical programming techniques have some limitations in dealing with such cases of optimization problems [19]. Therefore, researchers have focused on developing new solution methods that are more capable of dealing with NP-hard optimization problems with many variables and objectives and without satisfying some or all of the mathematical properties that are necessarily required for classical techniques [8]. Researchers have discovered that nature is an excellent source of inspiration for developing intelligent systems-based strategies and providing solutions to complex NP-hard optimization problems [20]. Evolutionary, Swarm Intelligence (SI) and other metaheuristics are prominent topics in developing new nature-inspired algorithms. These novel algorithms were developed with the purpose of assessing their ability to handle these complex optimization problems. However, not all these algorithms are efficient. A few have been shown to be highly efficient and have become practical approaches for solving real-world problems [21].

In reality, there is no single algorithm available in the literature that can work efficiently to solve every type of optimization problem and always provides a good optimal solution [22]. Therefore, searching for an efficient procedure to solve complex real-world optimization problems continues in this direction. Hence, new algorithms are continuously being searched according to the specific domain of optimization problems and as per the needs of decision-makers. However, so many classical optimization techniques and NIAs are available in the literature. It is really a challenging task to classify all these algorithms systematically, and all these are reviewed for the current study. The above classifications rely on the benchmarks, and there is no straightforward measures for determining benchmarks in the literature. Therefore, detailed classifications can be challenging for a research article as benchmarks may vary. In this paper, we review only one aspect of these algorithms' characteristics. Namely, we examine the inspiration behind their development and their effectiveness in solving complex optimization problems. We also adopted popularity and how frequently these algorithms are used as criteria for inclusion in this review paper.

This review study briefly introduces broadly used classical solution methods and NIAs for solving various types of optimization problems. In the present literature, five different types of classical optimization approaches are discussed, which further consist of various solution methods used to solve classical optimization problems consisting of well-behaved mathematical properties that are generally required for these methods. Additionally, among the algorithms inspired by the nature categories, three different classifications are discussed: the class of Evolutionary Algorithms (EAs), the class of Swarm Intelligence-based techniques, and other metaheuristics. Among these classifications, EAs are further classified into five different subcategories. These are Genetic Algorithm (GA), Nondominated Shorting Genetic Algorithm II (NSGA-II), Strength Pareto Evolutionary Algorithm 2 (SPEA2), Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D), and Pareto Envelope-based Selection Algorithm II (PESA-II). Swarm Intelligence-based algorithms are classified into five major subcategories: Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Ant Colony Optimization (ACO), Firefly Algorithm (FA), and Fruit Fly Optimization (FFO) algorithm. The field of biology also inspires some other algorithms, but they cannot be placed in the above two classifications. Furthermore, some algorithms are not inspired by biological behavior but are inspired by nature. Physics-based techniques for solving optimization problems are also among them. We are placing these algorithms in other metaheuristics categories, and these algorithms are as follows: Artificial Neural Network (ANN), Flower Pollination Algorithm (FPA), Simulated Annealing (SA) and Gravitational Search Algorithm (GSA). In addition, some research gaps and future prospects are also supplemented in this study. Therefore, this review study may be helpful for readers who are interested in working in this research area.

The remaining of this article is structured as follows. Section 2 discusses some basic concepts and definitions used for optimization problems and solution methods. Popular solution methods (classical and NIAs) are briefly discussed for this review work in Section 3. Section 4 delves into the examination of research gaps and, along with potential future prospects. Subsequently, the conclusions from this review work are discussed in Section 5.

2. Some basic concepts and definitions

In general, single and multi-objective optimization applications vary from different areas of the application domain. One can select the most suitable and efficient optimization technique per the nature of the optimization problem. Readers are referred to follow Refs. [23,24] to better understand the history of single and multi-objective optimization problems and their solution techniques. Only fundamental terminology is defined here for the sake of convenience. Because of the disparities between technical and mathematical jargon, several of these terms have various meanings in the literature. In such circumstances, this study uses the most frequent and relevant terminology.

Decision variables. In an optimization problem, *decision variables* [25] are numerical unknown quantities for which values are assigned. It consists of a domain, which represents the set of all possible values for a variable. These numerical unknown quantities are generally represented as $x_i, i = 1, 2, \ldots, n$, and maybe represented as a *decision vector* $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$.

Decision variable space. The numerical unknown quantities of decision vectors $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T \in \mathbb{R}^n$ span a n-dimensional space, which is said to be *decision variable space* or simply *decision space* [25]. We denote the decision space as $D \subseteq \mathbb{R}^n$.

Constraints. Most real-world optimization problems have certain limitations imposed by particular aspects of the environment. An acceptable solution for decision-making is assumed to satisfy these limitations. These limitations are generally said to be *constraints* of the optimization problem, which express dependencies among parameters and decision variables involved in the problem [25]. These constraints of the optimization problem can be defined mathematically as:

inequality type constraints

$$g_j(\boldsymbol{x}) \geq 0, j = 1, 2, \ldots, p, \tag{1}$$

and/or equality type constraints

$$h_k(\boldsymbol{x}) = 0, k = 1, 2, \ldots, r. \tag{2}$$

Remark 1. It is more important to note that the total number of constraints should always be less than the total number of decision variables; elsewhere, the problem becomes overconstrained as no degrees of freedom are left for optimizing.

Objective functions. In general, the *objective function* [25] is a criterion to evaluate the efficacy of a specific solution. Mathematically these criteria can be stated as computable functions of the decision variables, known as *objective functions*. Let us take a set $S = \{1, 2, \ldots, m\}$, then for each $i \in S, f_i : D \to \mathbb{R}$ represents the ith objective function. In a multi-objective optimization case, some objective functions may conflict with others, in which some of them need to be minimized while others are maximized.

Objective function space. The coordinate space in which vectors from evaluating multi-objective problem solutions are displayed is known as the *objective function space* or *objective space* [25]. Mathematically, all the objective functions in the above definition together form a vector-valued function (or vector function) $\boldsymbol{F} : D \to \mathbb{R}^m$, defined as:

$$\boldsymbol{F}(\boldsymbol{x}) = \left(f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})\right)^T. \tag{3}$$

Clearly, in this definition, $\boldsymbol{F}(\boldsymbol{x})$ is a $m-$dimensional vector of $\mathbb{R}^m$, and therefore the dimension of objective function space is m here.

Preferences. Preferences [1] are decision makers' beliefs on solutions in the criteria space. The decision maker directly imposes preferences on a collection of candidate solutions using procedures that entail a posteriori articulation of preferences. The final solution should then appropriately represent the decision maker's priorities. When expressing preferences a priori, it is necessary to measure ideas before looking at points in the criterion space. The term preference is frequently used in this context to refer to the relative relevance of various objective functions. This statement of preferences, however, is ultimately based on ideas about predicted solutions in the criterion space.

Preference function. A *preference function* [1] is an abstract function of the solution in the criterion space that completely incorporates the decision maker's preferences.

Utility function. The term *utility* indicates the satisfaction level of the decision makers. *Utility functions* are used to model utility, which expresses the level of satisfaction of an individual or group [26]. According to optimization theory, for each objective, an individual utility function is developed that expresses the relative relevance of the objective. It is used to approximate a preference function that cannot normally be described mathematically.

2.1. Classification of optimization problems on the basis of constraints

In general, real-world optimization problems are with or without constraints. An optimization problem with one or more constraints is considered a *constrained optimization problem* [27]. Some recent articles from various domains with constrained optimization problems include [28–30]. In contrast, if no constraints are involved in the optimization problem, it is called an *unconstrained optimization problem* [27]. Recent research articles have also emphasized such optimization problems, which have no constraints in several application areas [31–35]. Simultaneously, constraints of these problems can be the type of *equality* or *inequalities*. In structural optimization, governing equations are generally *equality constraints*. However, *inequalities* arise on constraints usually due to some limitations on the decision variables.

The space of decision variables where all constraints are satisfied is referred to as the *feasible space* [27]. In constrained optimization, one needs to search for the *optimum* of the objective function only in the *feasible space*.

2.2. Classification of optimization problems on the basis of types of decision variables

The decision variables of optimization problems are broadly classified into *discrete, continuous or mixed types of decision variables*. Furthermore, *discrete decision variables* are of the *binary numbers type* (taking values only 0 or 1) or of the *integer valued type* (taking values integers). Some examples of optimization problems with these types of decision variables are; tray and sequencing optimization problems of distillation columns and entire process flow sheets optimization problems [36], transportation problems [37–39], assignment problems in traffic assignment [40–42], assignment problems of diagnosable systems [43,44], human resource allocation problems [45], improving defensive air battle management [46], and discrete optimization in engineering design problems [47,48]. Recent research has also focused on problems involving modeling of dynamic process, for example hybrid discrete-continuous dynamic system optimization [49,50]. Optimization for the integrated process of designing and control [51] and designing of the transient processes [52,53] are two other examples that depend on dynamic models. In all these aforementioned problems, the *discrete decision variables* are usually associated with the structure of process, while the specific *continuous decision variables* are the state of the process.

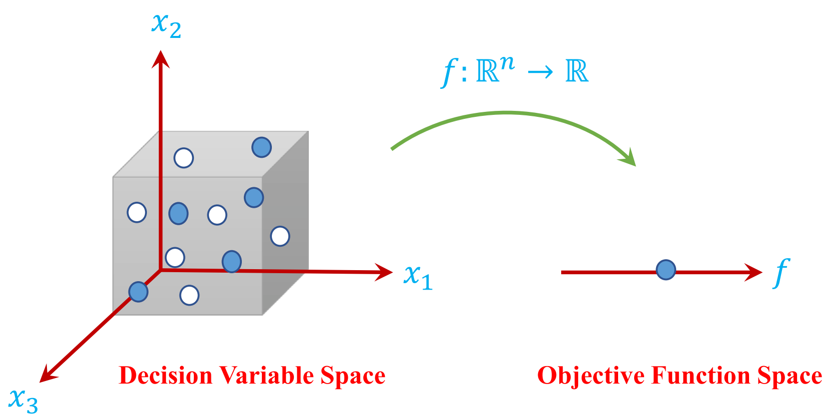

Fig. 1. Evaluation function from the decision variable space (for $n = 3$) to the objective function space for single objective optimization.

2.3. Classification of optimization problems on the basis of objectives

Optimization problems are broadly classified into *single-objective optimization problems* and *multi-objective optimization problems* depending on the number of objectives in the problem. In the subsequent subsections, we shall discuss these two types of classifications of optimization problems.

2.3.1. Single-objective optimization

In a single-objective optimization [25], the problem is concerned with finding a real decision variable that satisfies constraints (1) and (2) and optimizes a scalar single objective function $f(\boldsymbol{x})$. The most general *single-objective optimization problem* may be formulated as follows:

$$\begin{cases} \min \text{ (or max)} f(\boldsymbol{x}), \\ \boldsymbol{x} \in \boldsymbol{\Omega}. \end{cases} \tag{4}$$

Here, f is a real-valued (scalar) function and the set Ω consists of all possible values of $\boldsymbol{x}$, that satisfy the aforementioned constraints (1) and (2). This set $\boldsymbol{\Omega}$ is described as:

$$\boldsymbol{\Omega} = \{\boldsymbol{x} \in \mathbb{R}^n : g_j(\boldsymbol{x}) \geq 0, j = 1, 2, \ldots, p \text{ and } h_k(\boldsymbol{x}) = 0, k = 1, 2, \ldots, r\}. \tag{5}$$

This set $\boldsymbol{\Omega}$ defines the *feasible region* for the solution of the optimization problem and a solution $\boldsymbol{x} \in \Omega$ is said to be a *feasible solution*. Fig. 1 represents an evaluation mapping function from the n-dimensional decision variable space to the real-valued objective function space for a single objective optimization problem.

2.3.2. Multi-objective optimization

The multi-objective optimization problem (sometimes also known as vector optimization or multi-criteria optimization) is concerned with the problem of computing a set of optimal solutions with satisfying constraints and optimizing all the objective functions simultaneously of the optimization problem [25]. These functions combine to produce a mathematical representation of performance criteria that are typically conflicting in nature. Therefore, the word "optimize" in multi-objectives reflects finding a solution that satisfies all the objective functions at an acceptable level for the decision makers.

Without loss of generality, let $f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})$ are the objective functions of the given $m-$dimensional multi-objective optimization problem. Our estimated goal is to find a set of possible optimal solutions by solving all these objective functions simultaneously. Combining all these functions, we formulate a vector-valued function $\boldsymbol{F}(\boldsymbol{x})$ (refereed to Eq. (3)). Fig. 2 represents the illustration of the vector valued function $\boldsymbol{F}(\boldsymbol{x})$ from n-dimensional decision space (or solution space) to the m-dimensional objective space for a multi-objective problem of optimization. A vector from the feasible region of the decision space denotes a solution. These vectors produce a specific vector in the objective space when it is operated by evaluation mapping $\boldsymbol{F}(\boldsymbol{x})$, which specifies the solution quality based on the objective function values.

For a basic understanding of finding an appropriate solution method to a multi-objective problem of optimization, the mathematical definition of these problems becomes more necessary. This section presents a more general mathematical definition of the multi-objective optimization problem for the readers. Since it possesses more than one objective function, one needs to optimize these functions simultaneously. However, this leads to the main concern for the person who wishes to solve such problems having conflicting nature of some or all the objective functions (i.e., minimizing some objective functions and maximizing some objective functions). After optimizing all objectives, a set of possible optimal solutions are determined rather than a single optimal solution

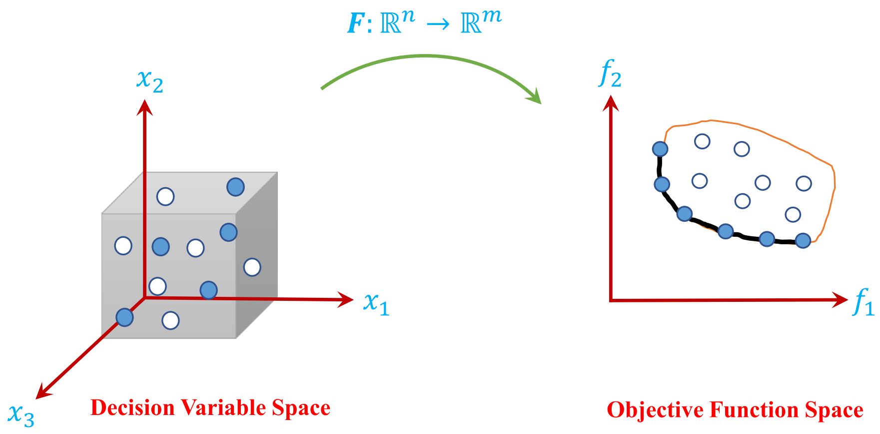

Fig. 2. Evaluation function from the decision variable space (for $n = 3$) to the objective function space (for $m = 2$) for multi objective optimization.

as a possible case in a single-objective problem. This combined solution set can be obtained using Pareto optimality theory [54]. A general multi-objective optimization problem is formulated as follows:

$$\begin{cases} \min\ (\text{or}\max)\,\boldsymbol{F}(\boldsymbol{x}), \\ \boldsymbol{x} \in \boldsymbol{\Omega}. \end{cases} \tag{6}$$

Here, $\boldsymbol{F}$ is an m-dimensional vector-valued function, as shown in Eq. (3), and the set $\boldsymbol{\Omega}$ is a feasible region of the solutions $\boldsymbol{x}$, denoted by Eq. (5).

2.4. Concept of Pareto optimality

A solution to a multi-objective type of optimization problem, in contrast to a single-objective type of optimization problem, is more of a concept than a definition. There is no single global solution in the case of a multi-objective type of optimization problem, and it is, therefore, essential to identify a collection of solutions that all satisfy a preset definition for an optimal solution. The idea of Pareto optimality [55] is the central concept in defining optimal solutions.

Definition 1 (*Pareto Dominance [25]:*)**.** Let $S = \{1, 2, \ldots, m\}$, and let us take two vectors $\boldsymbol{x}, \boldsymbol{y} \in D$ then, the vector $\boldsymbol{x}$ is said to *dominate* the other vector $\boldsymbol{y}$, (denoted as $\boldsymbol{y} \prec \boldsymbol{x}$) iff for all $i \in S, f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{y})$, (i.e., $\boldsymbol{F}(\boldsymbol{x}) \leq \boldsymbol{F}(\boldsymbol{y})$) and there exists at-least one $i \in S, f_i(\boldsymbol{x}) < f_i(\boldsymbol{y})$.

Definition 2 (*Pareto Optimal Solution (or Efficient Solution) [25]:*)**.** A feasible solution, $\boldsymbol{x}^* \in D$, is known as a *Pareto optimal solution* iff no other solution, $\boldsymbol{x} \in D$ exists, that dominates $\boldsymbol{x}^*$.

Definition 3 (*Pareto Optimal Set [25]:*)**.** The set of all Pareto optimal solutions is said to be the *Pareto optimal set*, which is denoted and defined as:

$$\mathcal{P}_S = \left\{ \boldsymbol{x}^* \in D : \nexists \boldsymbol{x} \in D \text{ such that } \boldsymbol{x}^* \prec \boldsymbol{x} \right\}. \tag{7}$$

Definition 4 (*Pareto Optimal Front [25]:*)**.** A set consisting of all such vector valued objective functions $\boldsymbol{F}(\boldsymbol{x}^*)$, for which $\boldsymbol{x}^* \in \mathcal{P}_S$, is said to be the *Pareto optimal front*. Mathematically this set is denoted and defined as:

$$\mathcal{P}_F = \left\{ \boldsymbol{F}(\boldsymbol{x}^*) : \boldsymbol{x}^* \in \mathcal{P}_S \right\}. \tag{8}$$

Since the Pareto optimal set consists of many solutions, all of these are equivalently acceptable solutions to the multi-objective problem. However, from a decision-making perspective, one needs to select only one solution from the solutions available in the Pareto optimal set. This selection is made by a person who wishes to make the decision. It is expected that the person (decision maker) should be well-informed about the requirements and goals of the given optimization problem. The decision maker can also use other intelligence to select the final solution [56]. The preference for selecting the final solution can be determined by estimating reference points, such as the ideal point, utopian point, and nadir point. With the help of these reference points, a decision maker can obtain information about the ranges of the objective functions, which are essential for making preferences for decision makers [57]. Additionally, reference points are used to normalize the objective space, which reduces the computational complexity that is necessary for solving the optimization problem [58]. These reference points for a multi-objective minimization problem are defined as follows:

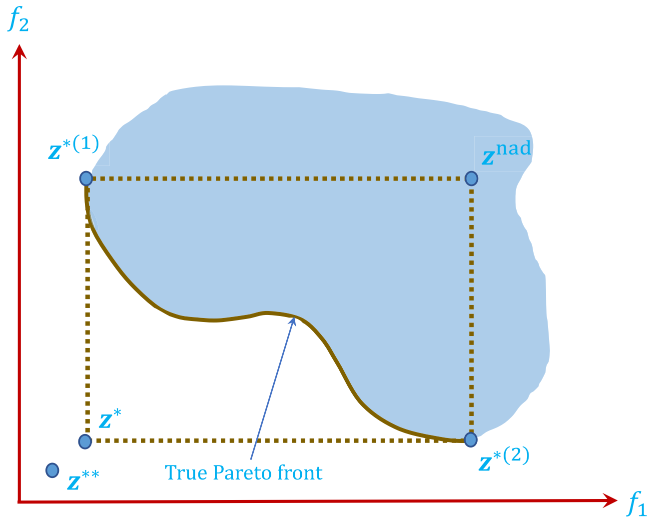

Fig. 3. A schematic illustration of reference points in two dimensional objective space.

Definition 5 (*Ideal Objective Vector [59]:*). The *ideal objective vector* $z^* = \left(z_1^*, z_2^*, \ldots, z_m^*\right)^T \in \mathbb{R}^m$ can be obtained by minimizing all objective functions independently for a feasible reason. Mathematically, it is defined as:

$$z_i^* = \min_{x \in \Omega} f_i(x), \forall i \in S. \tag{9}$$

Remark 2. The ideal objective vector is an array containing the minimum limit of all objective function values. As a result, there is always a feasible solution for each objective function with the same value as the corresponding component in the ideal solution.

Definition 6 (*Utopian Objective Vector [59]:*). The utopian objective vector denoted as $z^{**} = \left(z_1^{**}, z_2^{**}, \ldots, z_m^{**}\right)^T \in \mathbb{R}^m$ has every elements slightly smaller than the ideal objective vector. Mathematically, the utopian objective vector is defined as:

$$z_i^{**} = z_i^* - \epsilon_i, \forall i \in S, \tag{10}$$

where $\epsilon_i > 0$ for all $i \in S$.

Remark 3. It is worth noting that the ideal objective vector is strictly superior to the utopian objective vector.

Definition 7 (*Nadir Objective Vector [59]:*). A nadir objective vector which is denoted as $z^{\mathrm{nad}} = \left(z_1^{\mathrm{nad}}, z_2^{\mathrm{nad}}, \ldots, z_m^{\mathrm{nad}}\right)^T \in \mathbb{R}^m$, is the objective vector consisting of the worst objective function values for the Pareto optimal set $\mathcal{P}_S$. Mathematically, it is defined as:

$$z_i^{\mathrm{nad}} = \max_{x \in \mathcal{P}_S} f_i(x), \forall i \in S. \tag{11}$$

Remark 4. In contrast to the ideal objective vector, which indicates the minimum limit of each objective function value in the feasible region, the nadir objective vector depicts the maximum limit of all objective function values in the whole Pareto optimal set.

The diagram in Fig. 3 illustrates various reference points in a two-dimensional objective space.

As shown in Fig. 3, for a two dimensional multi-objective optimization problem, $z^{*(1)} = \left(z_1^{*(1)}, z_2^{*(1)}\right)$ and $z^{*(2)} = \left(z_1^{*(2)}, z_1^{*(2)}\right)$ represents the coordinates of the minimum solutions of objectives f_1 and f_2 respectively in the objective function space. Then, the nadir objective vector and the ideal objective vector can be computed from $z^{*(1)}$ and $z^{*(2)}$ as $z^{\mathrm{nad}} = \left(z_1^{*(2)}, z_2^{*(1)}\right)$, and $z^* = \left(z_1^{*(1)}, z_2^{*(2)}\right)$, respectively.

3. Solution techniques adopted for optimization problems

In general, optimization problems of the real-world are constrained and highly complex in nature and involve multiple objectives and constraints. Hence to solve such challenging problems, there is a need for an efficient procedure that is well-suitable according to the nature of the problem. The solution methodologies widely adopted to solve such optimization problems are classified into two major categories: classical methods and Nature-Inspired Algorithms (NIAs). Subsequently, NIAs are further classified into different subcategories of solution methods, which include Evolutionary Algorithms (EAs), Swarm Intelligence (SI) based techniques, and other metaheuristics. These NIAs have been further subclassified into various algorithms, which we discuss in the subsequent subsections.

3.1. Classical optimization approaches

This subsection briefly discusses some important classical approaches. These solution strategies have been adopted to address various optimization problems in some of the application areas. These classical optimization methods are advantageous if the problems consist of certain mathematical properties according to the requirements of the solution methods. The following paragraphs discuss five classifications of classical optimization approaches based on various attributes, efficacy, and popularity.

Linear Programming (LP). Since the inception of optimization problems, linear programming-based methods have been among the most pro-dominant techniques for solving these problems [60]. These are the most widely applied adaptation methods around the world. Generally, LP-based methods are used to solve single linear objective optimization problems consisting of linear constraints only. If the goal of an LP problem is to find an integer value solution, it is called integer programming [61]. In integer programming, a linear programming variant, the decision variables are of the integer type. Readers are referred to the following Refs. [4,61–63] for a better understanding of the concept of linear programming. Some of the very popular linear programming-based techniques for solving linear optimization problems are the simplex method [64], graphical method [4], primal–dual method [65], relaxation-based methods [66], cutting plane method [67], branch-and-bound method [63], branch-and-cut method [68], and ellipsoid method [69]. Some recent applications of real-world linear optimization problems solved using these methods include the vehicle routing problem [70,71], traveling salesman problems [70,72], and theory of the firm [73].

Mixed Integer Programming (MIP). MIP problems consist of at least one constraint of integer type (i.e., some decision variables are constrained only for integer values). This type of constraint usually captures the discrete nature of real-world decision-making problems through MIP models. One such example is the cardinality constraint on the portfolio optimization problem, in which a few of the decision variables are bounded to taking only binary integer values and are used to decide whether an asset is part of the portfolio [74]. It, therefore, broadens the usefulness of optimization techniques in handling more complex real-world problems. However, the inclusion of such combinatorial type constraints makes the MIP model an NP-hard problem and, therefore, becomes computationally more challenging [75]. The cutting plane method [67], and branch-and-bound method [63] are the two important techniques used to solve MILP problems.

MIPs are further classified into two categories: Mixed Integer Linear Programming (MILP) and Mixed Integer Non-Linear Programming (MINLP) problems. If all the objectives and constraints (equality, non-equality, or both) of MIP problems are of the linear type, and there is one integer-valued constraint, then such a problem is referred to as the MILP problem. Some of the recent studies make use of MILP for optimization [76–78]. On the other hand, if an MIP problem consists of either non linear objectives, non linear constraints, or both with continuous and discrete variables, then it is referred to as MINLP [79]. Compared to MILP, MINLP problems are relatively more challenging to solve, since these problems comprise all the challenges of its subfamily, such as the combinatorial nature of the MIP problem and the difficulty in solving nonconvex (or even for the case of convex) problems. Nevertheless, MINLPs have gained popularity in various applications, including but not limited to the finance, management science, engineering, and operations research domains [16,18,80]. Claudia and Andrea [81] reviewed the various tools available in the literature used to solve MINLP problems. Some popular methods to solve MINLP problems include the nonlinear branch and bound method [63,82], outer approximation method [83], and extended cutting plane method [84].

Quadratic Programming (QP). Quadratic programming is the procedure of solving single or multi-objective optimization problems that involve quadratic objective functions. QP methods are one of the most famous and broadly used techniques to solve convex and non-convex optimization problems [85]. The QP problems consist of quadratic objective functions and linear equality, non-equality, or both types of constraints. One of the main advantages of quadratic programs is that they require less computation cost. Nonetheless, the effort needed to find a solution is strongly influenced by the number of inequality constraints and the characteristics of the objective function [86]. Some widely used methods to solve QP problems are interior point method [87], active set method [88], the Lagrange multipliers method [89], augmented Lagrangian method [90–92], penalty method [89,93], Wolfe's method [94], proximal point algorithm [95], prox-like method such as proximal gradient descent method [96], proximal decomposition method [97], and accelerated proximal gradient method [98], alternating direction multipliers method (ADMM) [99], and the numerical method for nonconvex quadratic programming [100]. Recently published articles with the use of applications of quadratic programming in various real-world optimization problems are optimal power flow [101,102], multi-robot and task-space force control [103], positioning of marine vessels [104], and energy management system [105,106].

DC (Difference of Convex) Programming. DC programming [107] is an optimization method that deals with a superclass of convex functions known as Difference of Convex (DC) functions. It is a powerful optimization method used to solve nonconvex optimization problems where the objective function is not convex but can be decomposed into the difference between two convex functions [108]. The DC programming problem aims to minimize a DC function subject to constraints. By decomposing the objective function in this way, DC programming allows for efficient and effective optimization techniques to be applied, leading to high-quality solutions even in the presence of nonconvexity. DC programming has applications in various fields, including machine learning [109–111], statistics [112], and finance [113,114].

Table 1
Popular classical methods used to solve single or multi-objective optimization problems.

Classical methods	Linear Programming	Simplex Method Graphical Method Primal–Dual Method Relaxation-based Methods Cutting Plane Method Branch and Bound Method Branch and Cut Method Ellipsoid Method	
	Mixed Integer Programming	Mixed Integer Linear Programming	Cutting Plane Method Branch and Bound Method
		Mixed Integer Non Linear Programming	Non-Linear Branch and Bound Method Outer Approximation Method Extended Cutting Plane Method
	Quadratic Programming	Interior Point Method Active Set Method Lagrange Multipliers Method Augmented Lagrangian Method Penalty Method Wolfe's Method Proximal Point Algorithm	
		Prox-Like Method	Proximal Gradient Descent Method Proximal Decomposition Method Accelerated Proximal Gradient Method
		Alternating Direction Multipliers Method Numerical Method for Nonconvex Quadratic Programming	
	DC Programming Polynomial Goal Programming		

Polynomial Goal Programming (PGP). PGP is a versatile approach for incorporating decision makers' preferences of different objectives into optimization problems [115]. The strategy is simple and well suited for adding to the preference weights of decision makers in the objective functions [10]. PGP is a two-stage process; in the first step, a multi-objective optimization problem is converted into single-objective sub-problems corresponding to each objective, and then each sub-problem is solved independently to obtain the aspired levels. In the next step, goal variables corresponding to each objective are used to minimize deviations from aspired levels [9]. The PGP approach has also been adopted in some recent research to address many real-world problems of optimization. Some of them are portfolio selection problems [116–120], scheduling and planning in service systems [121], chickpea cultivar selection under stress conditions [122], and sustainable development goals of India agenda 2030 [123].

A summary of the popular classical methods adopted to solve various single or multi-objective optimization problems is provided in the Table 1.

3.2. Nature-Inspired Algorithms (NIAs)

The family of NIAs consists of a class of novel problem-solving techniques that are somehow inspired by natural phenomena. In general, optimization problems of real-world are highly complex in nature. They are of multi-objective and multi-dimensional types [19]. Solving such problems with deterministic techniques is sometimes very difficult. These techniques are typically applied only if the problem is bound to obey some important mathematical properties.

Furthermore, finding the optimal solution to such optimization problems from conventional approaches requires more effort and involves higher time complexity [8]. To overcome all these limitations, many algorithms inspired by nature came into existence. These NIAs consider a stochastic technique to find the optimal solution in a larger pool of search space. In this subsection, we discuss a few such classes of nature-inspired based algorithms.

3.2.1. Evolutionary Algorithms (EAs)

The group of EAs is one of the most substantial classes of NIAs that consists of all such algorithms based on the natural evolution process. EAs have been used effectively since their conception to solve problems in a different range of application disciplines. Some of them include; finance, engineering, operation research, and social sciences [124–128]. The major advantage of using EAs is that they require a little domain knowledge to operate. Therefore it can be used even with less subjective knowledge of mathematical aspects. Hence, it can be implemented even in a discontinuous, non-differentiable, and/or non-convex objective space [128]. In the past literature, most of the financial problems solved through EAs are of the single-objectives type [2].

Table 2
A short list of important operators used in EAs with their definitions and types.

Operator	Definition	Types
Selection	The selection operator is used to give preference to better solutions (chromosomes)	Parent selection, Tournament selection, Roulette wheel selection, Rank selection
Crossover	The crossover operator is used to combine two individuals' parents to produce offspring	One-point crossover, Two-point crossover, Uniform crossover, Exponential crossover, Simulated binary crossover
Mutation	The mutation operator is used to provide a perturbation into the population	Bit-flip mutation, Swap mutation, Inversion mutation, Polynomial mutation, Power mutation

Nevertheless, over the last three decades, many researchers have applied EAs for multi-objective type of optimization problems of economics, finance, and operations research, in which the objectives are conflicting in nature. In this direction, Bäck and Schwefel [129] present a comprehensive overview of EAs for parameter optimization and examine three kinds of EAs: evolutionary programming, evolution strategies, and GAs. The authors discussed how these algorithms differ and resemble each other in four aspects: representation, mutation, recombination, and selection. Finally, they evaluate the advantages and disadvantages of each algorithm and offer some possible future research directions. Zhou et al. [130] presented a state-of-the-art review of MOEAs for a survey period of eight years based on several evolutionary parameters. Ponsich et al. [2] surveyed a review on MOEAs to address portfolio selection problems and other economics and finance applications. Metaxiotis and Liagkouras [126] thoroughly reviewed the implementations of MOEAs in portfolio management problems.

We will first start with the definitions of some primary attributes and their types used in various EAs. Table 2 below shows some of these primary attributes with their types. Later, we briefly discuss some of the important EAs in details.

The foundation of this idea lies in Charles Darwin's theory of natural selection, which elucidates the mechanism behind biological evolution.

Genetic Algorithm (GA). GA is among the popular algorithms used to solve optimization problems in the class of EAs and is one of the oldest nature-inspired population-based algorithms proposed by Holland [131]. The foundation of this idea lies in Charles Darwin's theory of natural selection [132], which elucidates the phenomenon of biological evolution. At present, there are many modern algorithms from the EA class that are somehow directly or indirectly related to GA [8]. The essence of GA includes the following steps: (i) Encoding of objective functions (this is often determined by the nature of the problem, and some popular encoding schemes include binary encoding, real number encoding, and natural number encoding); (ii) Selection of the population based on their fitness; (iii) Generate new population by performing evolution cycle that includes crossover and mutation operators; (iv) Replace the old population by new population and perform the iteration again; (v) Decode the final result to get the desired solution of the problem. Genetic algorithms are being widely used in many real-world applications. A few of recent studies in which GA is used to solve various real world optimization problems are image processing [133], game theory [134], diagnosis [135], treatment planning [136], and healthcare management [137]. A review of recent advancements in genetic algorithms was recently published by Katoch et al. [138].

Nondominated Shorting Genetic Algorithm II (NSGA-II). Eversince its inception, NSGA-II [139] has proven to be the most popular and broadly used technique for solving different types of optimization problems with many objectives due to its applicability in a wide range of applications. The NSGA-II is based on the concept of the Pareto-dominance principle. For each solution, first, we need to identify all the solutions it dominates and the set of solutions from which it is dominated. Thereafter, the nondominant sorting mechanism is applied to divide the combined parent and offsprings populations into multiple nondominant fronts. The first front consists of a set of solutions that are not dominated by the rest of the solutions in the new population. This front consists of the best solutions for the combined population. Additionally, it is the better approximation of the true Pareto optimal front. This means that solutions on the present front are given higher priority than solutions on subsequent fronts. The crowding comparison operator is used to find which solution should be in the nondominated fronts. This is done by picking a fixed solution from the population and computing the mean distance of any two points on either side of this reference solution with each objective of the problem. A solution in a less crowded region is more likely to be selected for the subsequent generation. This operator ensures the diversity among the nondominated solutions in the front.

NSGA-II, in its original form, takes simulated binary crossover as the recombination operator and polynomial mutation as the mutation operator. The overall computational complexity of the NSGA-II algorithm is $O(MN^2)$, making it one of the fastest algorithms in the class of EAs. Recently, Verma et al. [140] carried out a survey on NSGA-II for selected combinatorial multi-objective optimization problems, that include assignment problems, scheduling problems, traveling salesman problems, allocation problems, vehicle routing problems, and knapsack problems. Some of the recently published articles with the use of NSGA-II in various applications are; portfolio management [141–143], trading systems [144], healthcare [145–147], workflow management [148–150], water resource management [151,152], wireless sensor networks [153,154], power systems [155], and many more.

Strength Pareto Evolutionary Algorithm 2 (SPEA2). SPEA2 [156] is among the most improved versions of its predecessor algorithm, SPEA [157]. The three main features in SPEA2, in contrast to the previous version, that make it more advantageous are: (i) an advanced fitness assignment process is provided that takes into account, for the each individual, how many solutions it dominates, and how many solutions it is dominated by; (ii) implementation of a nearest-neighbor density estimation technique, which provides for more precise search process guidance; and (iii) a novel archive truncation method ensures the preservation of boundary solutions. SPEA2 maintains an outer archive of nondominated solutions found at the time of search and updates these solutions at each generation. This updation is performed through a density estimation technique [158] inspired by the $K-$nearest neighbor method to separate solutions with the same fitness. Some recent publications with SPEA2 in various application areas are; water resource management, power systems, and investment decision making [159–161].

Multi Objective Evolutionary Algorithm based on Decomposition (MOEA/D). The MOEA/D introduced by Zhang and Li [162] is based on the concept of the aggregation method in which a multi-objective optimization problem is first decomposed into a number of (e.g., N) scalar sub-problems, with each sub-problem interrelated through $T-$neighboring sub-problems in a Euclidean sense. Here, N is a user-defined parameter used to set a collection of uniformly distributed weight vectors, each of which has dimensions equal to those of the objective space of the stated problem. Then, using recombination operators, MOEA/D iteratively optimizes all of the sub-problems. Furthermore, older solutions are replaced by the offspring solutions of the sub-problem if their decomposition cost is better than that of the existing solution. One of the important features of MOEA/D is its decomposition technique, weighted sum approach, and weighted Tchebysheff approach. Trivedi et al. [163] presented a review survey of multi-objective evolutionary algorithms based on decomposition. Some recent applications of multi-objective optimization problems solved using MOEA/D are shop scheduling problem [164,165], wireless sensor network coverage optimization [166], and resource allocation strategy [167].

Pareto Envelope-based Selection Algorithm (PESA-II). PESA-II, an EA class algorithm proposed by Corne et al. [168], works on region-based selection rather than of an individual selection strategy, as reported in SPEA2 and NSGA-II. In this method, a hyperbox is used as a unit of region-based selection. Moreover, an external archive is utilized to store approximated Pareto solutions. In contrast, new solutions from generation are stored in a separate population. However, the size of the external archive varies at the time of the evolutionary process. This algorithm preserves solution diversity by partitioning the objective space into smaller pieces of hyperboxes. Relative fitness is allocated to each hyperbox according to the number of solutions in the hyperbox. Both parent and offspring are selected from the external archive based on the density of hyperboxes. That is, rather than individuals, a hyperbox is selected using a predefined selection criterion. Hyperboxes with a lower density of the solution are more likely to be selected for the recombination process. Power systems [169,170], driving assistance system [171], and radial distribution systems [172] are some real-world applications of multi-objective optimization problems that are solved with the use of PESA-II.

3.2.2. Swarm Intelligence (SI) based algorithms

Another popular class of the NIA family is the group of SI-based metaheuristic algorithms. SI algorithms are based on natural swarm intelligence behavior, such as fish and bird schooling, ants searching for food, the intelligent foraging behavior of honey bees, and so on. The word swarm generally refers to any restrained group of interacting individuals or agents. Over the years, SIs have gained significant attention for their capability to solve real-world optimization problems in a broad area of applications. The flexibility and simplicity of SI-based algorithms have made them one of the most widely used NIAs in the field of computational intelligence and optimization. Chakraborty and Kar [173] delve into the core concept, explore the potential domains of application, and provide an extensive overview of various Swarm Intelligence (SI) algorithms. These algorithms include insect-based and animal-based approaches, which are examined in meticulous detail. In particular, the authors concentrate on algorithms inspired by ants, bees, fireflies, glow-worms, bats, monkeys, lions, and wolves. Brezočnik et al. [174] reviewed 64 various SI algorithms for feature selection across multiple application domains. Ertenlice and Kalayci [175] conducted a review survey on the recently published literature on SI-based algorithms used to solve various types of portfolio selection problems and explored the future scope of SI opportunities in various fields. Recently, Chen et al. [176] also presented a comprehensive review on the overview and recent advances of SI algorithms for solving classic portfolio selection models and argued some future directions for the research. Sun et al. [177] present a systematic literature survey of the use of SI algorithms in Internet of Things (IoT) domain. The authors' primary focused on analyzing SI-enabled applications to Wireless Sensor Network (WSN) and discussing associated WSN research concerns. The survey also covers other IoT domains where SI techniques have been applied. Some of the important SI-based algorithms used in various real-world applications are briefly discussed in the paragraphs below.

Particle Swarm Optimization (PSO). PSO is one of the most prominent and the first SI class metaheuristic algorithms introduced by Kennedy and Eberhart [178]. It is based on nature's swarm intelligence behavior, which simulates the integrated function of fish and birds to exchange knowledge in a group in order to increase decision-making ability in a synchronized manner. PSO has emerged as a very intriguing and ever-growing topic of research in the SI class since its inception. It has been applied to solve many challenging single and multi-objective multi-criteria decision-making problems from various real-world areas of applications [175]. In PSO, a member, called a particle, continues to adjust its position in search of food and motion by assessing its own experience and the experiences from other members in the swarm. Therefore, a particle has to keep moving in multi-dimensional space to move from one place to another, striving to converge on the food (i.e., an optimal solution). Here a position of the solution space means a solution to the optimization problem. Considering the PSO algorithms, Pradhan et al. [179] presented a research survey for cloud computing environment for metaheuristic scheduling mechanisms. Many recent studies have used PSO in various decision-making problems, such as resource allocations [180,181], battery management systems [182,183], power systems [184,185], landslide susceptibility mapping [186], and path planning [187].

Ant Colony Optimization (ACO). Dorigo et al. [188] introduced an SI-based algorithm, called the ACO algorithm, that imitates the swarm behavior of natural ants. The concept of this algorithm is motivated by the communication skills of ants that are continuously moving to find the shortest route toward searching for a food source [189]. For this, a chemical called a pheromone that they release along the route indicates a preferred path so that other colony ants can follow the same. The route with the higher pheromone residue is more likely to be the shortest route and therefore the activity of ants is more frequent here. The pheromone model in the ACO algorithm uses the same concept to solve single and multi-objective optimization problems where the pheromone residue is updated with each iteration. García et al. [190] examined the ACO algorithms from various literature used to solve challenging optimization problems of combinatorial nature and analyzed the relationship of ACOs with other metaheuristic algorithms.

Artificial Bee Colony (ABC) algorithm. The ABC algorithm is another SI class based algorithm proposed by Karaboga et al. [191]. This is inspired on the swarm intelligence behavior of honey bees. Bees are social insects and prefer to live in natural colonies. The collective intelligence of a swarm of bees for forage selection requires food sources, employed foragers, and unemployed foragers. Based on this information, an ABC algorithm is devised, with three different types of bees in the colony: planned, spectator, and scout bees. The first half portion of the colony posses of employed bees, while the other half portion houses onlookers. With the use of waggle dance, employed bees transmit and exploit nectar information to the colony.

On the other hand, an onlooker bee in the hive observes this waggle dance and modifies its trajectory depending on the quality of nectar to target the right nutrient source. Scout bees look for new nutrient sources after all identified sources have been exhausted up. Scout bees are responsible for discovering new food sources at random (here, the location of a food source refers to a problem solution). At the same time, the amount of nectar describes the quality of that solution. Several authors [192–194] have performed a research survey based on ABC variants and applications, their developments, and future research directions.

Firefly Algorithm (FA). Yang proposed the Firefly Algorithm (FA) [195], a metaheuristic algorithm that utilizes fireflies' flashing patterns and behavior to solve optimization problems. In essence, the flashing characteristics of the FA are explained by three rules: (i) all fireflies are of unisexual in nature; (ii) firefly attractiveness is directly proportional to the brightness of the fireflies; and (iii) the brightness of a firefly is obtained by the landscape of objective functions. In case of a maximization problem, fireflies' brightness is directly proportional to that of the objective function. Other types of brightness can be describe in the same manner as the fitness function is defined in GAs. Some of the recent applications areas wherein FA is used to solve various optimization problems are image processing [196], biomedical and health care [197], wireless sensor networks [198], network intrusion detection [199] and many more. Kumar and Kumar [200] have recently presented a systematic review work on FA based on various parameters and suggest possible future research directions.

Fruit Fly Optimization (FFO) algorithm. Pan [201] introduced a metaheuristic FFO algorithm for finding global solutions to optimization problems. The FFO algorithm is inspired by the biologically intelligent foraging behavior of fruit flies to search for their food. Incorporating this algorithm to solve optimization problems has several advantages, such as, ease of implementation, a straightforward structure, and finding solutions in a reasonable amount of time [202]. The FFO algorithm undergoes through four stages: (i) *initialization* – set termination criteria, population size, and initialize the swarm location of fruit flies; (ii) *osphresis foraging* – randomly search for food sources around fruit fly swarms; (iii) *population evaluation* – evaluate the fitness value (smell concentration value) of every food sources; and (iv) *vision foraging* – find the best food source having the maximum fitness value. Once this is achieved, a swarm of fruit flies will fly to the best source. FFO algorithm is applied in various recent applications areas to solve optimization problems like science, engineering, medical, agriculture, and many more [203–207].

3.2.3. Other metaheuristics

Biology has played a crucial role in the development of NIAs. *Bio-inspired optimization algorithms* are an emerging approach based on the principles and motivations of nature's biological evolution to develop new and robust competing technologies. Many algorithms in the EA and SI class are inspired in some way by biological phenomena. Nevertheless, we are placing those algorithms in a separate basket here because these algorithms are more closely related to the class mentioned earlier. On the other hand, some of the NIAs have drawn motivation from the laws of physics as well. Such algorithms have been formulated by mimicking specific physical laws from the field of physics such as mechanics, gravity, electric charge, and many others. In this subsection, some important other metaheuristics that have not been part of the EA or SI classes are discussed here.

Artificial Neural Network (ANN). The ANN approach is among the most popular techniques in the vast field of human applications [208]. It operates by leveraging the functionality of the biological nervous system in the human brain [209]. Many organizations continuously strive to invent new ANN ideas to solve complex real-world optimization problems [210]. Implementing an ANN can make the model more accessible and precise than complex systems with large-scale inputs [211]. An ANN has an input layer of neurons (sometimes called nodes or units), hidden layers (perhaps one or more), and an output layer of neurons [212]. Inputs from the individual are assessed in many ways through the input layer. After that, the hidden layers compute all the hidden patterns and features through an activation function where artificial neurons take in a set of weighted inputs. Finally, the output layer is the last layer of the ANN, where the desired predictions of the individual are obtained. Some recent studies in ANNs for various applications include portfolio management [213,214], applications to pattern recognition [215], solar energy systems [216], diabetes prediction [217,218], and energy efficiency prediction [219].

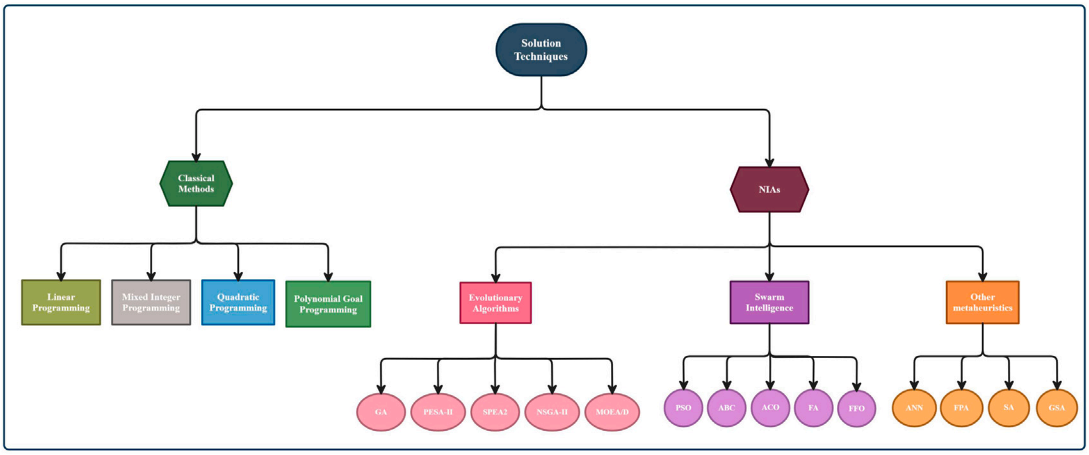

Fig. 4. A tree diagram showing the various solution techniques used to solve single or multi-objective optimization problems.

Flower Pollination Algorithm (FPA). The FPA is a bio-inspired algorithm first introduced by Yang [220], inspired from the natural pollination procedure of flowering plants. Flower pollination is often related to pollen transmission. It is followed by pollinators, for example, birds, bats, insects, and other animals. These characteristics can be formulated into algorithms by using four rules: (i) biotic and cross-pollination act as global pollination, where pollen-carrying pollinators follow Levy flights; (ii) abiotic and self-pollination processes perform the local search; (iii) because of the similarity between the two flowers, a flower constancy is involved; and (iv) a switch probability between 0 and 1 manages local and global pollination. Due to its physical proximity and other factors, it is slightly biased toward local pollination. Recently, Mohamed and Laila [221] presented a review work on FPA and adopted several important parameters, such as biological inspiration, existing studies and the comparisons, variants of FPA, implementation, and FPA application areas. A comparison between the FPA and six other NIAs is also performed for the constrained optimization problem.

Simulated Annealing (SA) algorithm. SA is among the most popular, widely used, and oldest NIAs, first introduced by Kirkpatrick et al. [222]. It is based on the physical annealing phenomenon inspired by the law of physics and applied to solve almost every type of optimization problem. In essence, it replicates the annealing process utilized for heating physical materials, wherein metals possessing the least energy and large crystal size are cooled and solidified, leading to their transformation into a crystalline state that effectively reduces defects within the metal structures. The process of annealing consists careful temperature control and the cooling rate. The main advantage of using SA is that, unlike deterministic search methods or gradient-based methods, it has the potential not to be trapped in local optima [8]. Therefore, convergence toward global optima in SA is highly likely if sufficient randomness is employed in conjunction with very slow cooling. Recent developments in some areas of applications with SA include medical [223,224], supply chain [225,226], feature selection [227,228], portfolio management [229,230], and power systems [231].

Gravitational Search Algorithm (GSA). GSA, as its name suggests, is a metaheuristic algorithm inspired by the theory of Newton's law of gravity of physics. The concept of mass and gravity was the main idea of the motivation behind the GSA [232]. In GSA, search agents are assumed to be objects having a specific mass that measures their performance. In contrast, through the force of gravity, each object in the universe of discourse interacts with other objects. In GSA, search agents are assumed to have a specific mass that measures their performance. Due to this, the object with maximum mass (heavier object) attracts all the other objects of lighter mass by its gravitational force. Hence the heavier mass object moves more slowly than the object with light mass, which exhibits a good solution, and this ensures the exploitation step of the GSA. GSA has been applied in various applications areas and has been used to solve various real world optimization problems such as power system [233], sensor networks [234], software design [235] and many more. Recently, Rashedi et al. [236] presented a review survey on the development of GSAs to solve various single or multi-objective optimization problems and provided some suggestions and future directions to explore new areas.

Below, Table 3 provides a brief description of all the NIAs mentioned above on the following basis: year, author name, concept used, key operators, and parameters.

A summary of all the above-discussed solution techniques adopted to solve various optimization problems is exhibited in the form of a tree diagram in Fig. 4.

Besides the NIAs discussed above, there are many other NIAs in the literature that are used to solve single or multi-objective optimization problems in various real-world domains. However, these NIAs are not considered in this review work and are not included in this study. Nevertheless, it should be acknowledged that these NIAs have their own merits and advantages in solving optimization problems of different kinds. Some of the NIAs that are not covered in this review are:

- **Harmony Search (HS) algorithm** [237]**:** It is based on the improvisation process of musicians. The main limitation of this algorithm is that it is trapped between local optimal solutions and is computationally more expensive than the other popular metaheuristics.

Table 3
A short list of well known NIAs with their features in chronological order.

	Algorithms	Author(s)	Year	Source of Inspirations	Key Operators	Parameters
Evolutionary Algorithms (EAs)	GA	Holland [131]	1992	Natural selection of biological evolution	Selection, Crossover, Mutation	Population size, Crossover and Mutation probability
	SPEA2	Zitzler et al. [156]	2001	Pareto dominance, archive truncation method, nearest-neighbor density estimation technique	Selection, Crossover, Mutation	population size, Archive size, Number of generations
	PESA-II	Corne et al. [168]	2001	Region-based selection	Selection, Crossover, Mutation	Population size, Crossover and Mutation probability, Chromosome lengths, Hyper-grid size
	NSGA-II	Deb et al. [139]	2002	Nondominant sorting & Crowding distance	Selection, Crossover, Mutation	Population size, Number of generations, Crossover and Mutation probability
	MOEA/D	Zhang and Li [162]	2007	Aggregation method (Decomposition), weighted sum approach, and weighted Tchebysheff approach	Differential Evolution, Mutation	Population size, Scaling factor, Croosover and Mutation probability
Swarm Intelligence (SI) based algorithms	PSO	Kennedy and Eberhart [178]	1995	Nature's swarm intelligence behavior (fish and birds)	Fitness evaluation, Updation	Swarm size, Inertia weight, Cognitive parameter, Social parameter
	ABC	Karaboga et al. [191]	2005	Intelligence behavior of honey bees	Employed bee phase, Onlooker bee phase, Scout bee phase	Number of employed bees, Number of onlooker bees
	ACO	Dorigo et al. [188]	2006	Imitates the swarm behavior of natural ants	Pheromone updating operator, Ant decision rule operator	Number of ants, Pheromone evaporation rate
	FA	Yang [195]	2009	Concept of fireflies' flashing patterns and behavior	Attraction operator, Absorption operator	Attractiveness coefficient, Absorption coefficient, Step size
	FFO	Pan [201]	2012	Intelligent foraging behavior of fruit flies	Initialization, Osphresis foraging, Population evaluation, Vision foraging	Number of fruit flies in the swarm, Maximum number of iterations, Step size, Weight of the global best position
Other metaheuristics	ANN	McCulloch and Pitts [209]	1943	Functionality of the biological nervous system of the human brain	Activation function, Weight initialization, Backpropagation	Learning rate, Number of hidden layers, Number of neurons per hidden layer, Batch size and Number of epochs
	SA	Kirkpatrick et al. [222]	1983	Physical annealing phenomenon	Swap, Scramble, Insertion, Reversion, Crossover operators	Initial temperature, Cooling rate, Stopping criterion
	GSA	Rashedi et al. [232]	2009	Theory of Newton's law of gravity of physics	Mass assignment, Force acting on the objects, Movement using the Newton's second low of motion	Inertial mass, Active and Passive gravitational mass, Position of the mass
	FPA	Yang [220]	2012	Natural pollination procedure of flowering plants	Global pollination, Local pollination, Levy flight	Population size, Number of iterations, Mutation probability, Step size

Acronyms: ABC — Artificial Bee Colony, ACO — Ant Colony Optimization, ANN — Artificial Neural Network, BA — Bio-inspired Algorithm, EA — Evolutionary Algorithm, FA — Firefly Algorithm, FFO — Fruit Fly Optimization, FPA — Flower Pollination Algorithm, GA — Genetic Algorithm, GSA — Gravitational Search Algorithm, MOEA/D — Multi-Objective Evolutionary Algorithm based on Decomposition, NSGA-II — Non-dominated Shorting Genetic Algorithm-II, PB — Physics Based, PESA-II — Pareto Envelope-based Selection Algorithm II, PSO — Particle Swarm Optimization, SA — Simulated Annealing, SI — Swarm Intelligence, SPEA2 — Strength Pareto Evolutionary Algorithm 2.

- **Honey Bee Algorithm (HBA)** [191]**:** HBA is based on the behavior of honey bees in search of food sources. It is a parameter optimization strategy that has been extensively worked out in several research publications [191,238]. The Honey Bee algorithm has been observed to have a slow convergence rate and to be prone to premature convergence [239].
- **Bat Algorithm (BA)** [240]**:** It is among those optimization algorithms inspired by the echolocation behavior of bats. According to Ramli et al. [241], while the basic BA achieves better convergence for optimum solutions than other classic optimization approaches, inability to jump out of local optima, slowing down the convergence rate and decreasing accuracy.
- **Cuckoo Search (CS) algorithm** [242]**:** It is inspired by some cuckoo species' nest parasitism, as well as Levy flights' random walks. The cuckoo search parameters are typically continuously monitored during the optimization process. This, however, might result in disadvantages, such as slow and premature convergence [243].
- **Grey Wolf Optimizer (GWO) algorithm** [244]**:** GWO algorithm is inspired by the models of the social hierarchy and hunting behavior of grey wolves. Some of its advantages are fewer parameters, basic ideas, and straightforward implementation. Its drawbacks include a slow rate of convergence, poor solution accuracy, and a propensity for the local optimum. Because of this, many scholars have made significant advancements in this algorithm [245–248].
- **Bacterial Foraging Optimization (BFO) algorithm** [249]**:** BFO algorithm is a swarm intelligence algorithm that mimics the bacterial foraging behavior in order to maximize energy while exploring. The conventional BFO approach, on the other hand, has several limitations, which include poor convergence speed, susceptible to being stranded in local minima, and fixed step length [250]. Many other types of enhanced BFO algorithms have been developed to address these shortcomings [251].
- **Cat Swarm Optimization (CSO) algorithm** [252]**:** It is a swarm-based optimization approach that resembles the hunting behavior of cats. Because of its superior search capacity for optimal solutions and higher robustness, the CSO algorithm has been used in various applications [253]. Nevertheless, low convergence and higher memory usage are its main flaws, limiting optimization efficiency to a greater extent.
- **Black Hole Optimization (BHO) algorithm** [254]**:** It is an NIA that mimics the black hole phenomenon created by a huge star with an extremely strong gravitational force. Nevertheless, the approach has some drawbacks, such as the number of clusters must be known ahead of time; the performance is heavily dependent on the initial centroids and can be trapped in locally optimal solutions [255].
- **Spider Optimization (SO) algorithm** [256]**:** SO algorithm is a population-based metaheuristic that resembles the cooperative behavior of a social spider. This algorithm has a significant flaw in that it frequently produces sub-optimal results due to an improper balance of exploration and exploitation in its search strategies [257]. Further disadvantages of the SO algorithm include slower and premature convergence, time consumption, and high spatial complexity [258].
- **Teaching–Learning-Based Optimization (TLBO) algorithm** [259]**:** It is inspired by classroom teaching and learning processes. The basic TLBO algorithm has a slower convergence rate for optimization problems than the existing well-known metaheuristics.
- **Water Wave Optimization (WWO) algorithm** [260]**:** WWO algorithm is an NIA that mimics the movement of water waves concept. It has fewer downsides, such as poor computation accuracy, falling quickly into locally optimal solutions, and slower convergence speed [261,262].

In summary, there are numerous NIAs available for solving optimization problems, and while some are more commonly used than others, it is important to explore and consider all options to find the most effective solution for a particular problem. These NIAs may not be as well-known as some of their counterparts, but they offer unique approaches to optimization problems and should be noticed in research or practical applications. As the field of optimization continues to grow and evolve, it is crucial to explore and evaluate different NIAs to find the most effective solutions for real-world problems. Some of them are still in the experimental stage and require further research and development before they can be applied in practical scenarios. However, with the increasing interest in optimization techniques and the advancement in computing technology, it is expected that more efficient and effective NIAs will be developed in the future to tackle complex optimization problems.

4. Research gaps and future prospects

This study reviews real-world single or multi-objective optimization problem-based research articles that use classical and NIA approaches to solve the problem. The classical methods are more effective when the optimization problem has mathematical properties that are well-suited [13]. However, optimization problems involving several objective functions do not typically have such great mathematical structures, especially when combinatorial type constraints are included. One of the common drawbacks of utilizing traditional approaches to address multi-objective optimization problems is that the original problem needs first to be transformed into an similar set of parameterized single-objective optimization problems. Additionally, solving the respective parameterized single-objective problems multiple times for a different parameters combinations is expected to provide a good approximation of the Pareto optimal front of the original multi-objective problem. However, this strategy generally does not provide a decent estimate of the Pareto frontier of the problem.

Over the last two decades, NIAs have been widely applied to address optimization problems with realistic constraints. These procedures, unlike conventional methods, do not depend on the mathematical attributes of the problems. Because of their population-based design, these approaches require a significant time to obtain an approximate solution for the large-scale portfolio selection problem [263]. On the contrary, a single run of the multi-objective class of NIAs can yield a fairly accurate estimate to the Pareto optimal solution. Because of its population-based structure and candidate generation procedure, NIAs are also ideally suited for parallel implementations.

Future research might concentrate on developing robust and efficient population-based metaheuristics for addressing multi-objective optimization problems with a pretty balance of exploration and exploitation while requiring less computation time. Recently developed techniques to address many objectives may also be used to attempt to solve the optimization problems that have a higher number of objectives together with many decision variables [264,265]. The increase in the number of decision variables for a multi-objective optimization problem sometimes makes optimization algorithms more computationally expensive and more challenging to handle. To overcome this situation, a robust procedure may be designed so that the optimization algorithm can work faster and more efficiently even for many decision variables cases.

Introducing practical constraints to a real-world optimization problem poses yet another significant hurdle. The inclusion of these constraints significantly alters the mathematical nature of the problem. As an illustration, the incorporation of combinatorial type constraints in different optimization problems can pose a significant challenge. Further addition of some hard constraints (which typically do not adhere to some of the mathematical properties that are essential for classical techniques) converts the optimization problem into a complex one. In such instances, the general constraint-handling strategies fail to deliver the best feasible solution. To address these cases, repair-based procedures have been found to be a good replacement in the literature [175,266–270]. Recent research has also revealed several effective repair strategies to address the addition of further round-lot constraints [271]. A meticulous examination of the literature suggests that the potential of these repair-based mechanisms has not been fully harnessed in the context of many-objective optimization problems. Subsequent research endeavours may explore the feasibility of incorporating these established repair strategies to augment the efficacy of the solution methodologies.

Finally, it is also important to carefully choose the performance metrics used to evaluate the quality of solutions in the multi-objective optimization case. Metrics such as hypervolume and epsilon indicators are most commonly used in multi-objective optimization, but they may not always capture all aspects of solution quality [272]. Therefore, it is important to consider multiple metrics and perform a thorough analysis of solution quality before making any decisions based on optimization results. Thus the above discussion provides directions and suggestions for finding answers to some of the open questions linked with single and multi-objective optimization algorithms for the researchers willing to work in the field of classical optimization methods and NIAs for solving optimization-related problems.

5. Conclusion

This review paper briefly highlights different solution methods that are used to solve various single and multi-objective optimization problems in diverse real-world application areas. First, some applicable definitions and basic concepts related to general optimization problems are discussed. After that, classifications of optimization problems based on various parameters are emphasized. Furthermore, this study analyzes the primary solution techniques, classified into classical methods and NIAs. In this classification, five different types of classical optimization approaches are examined, which are further classified into various solution methods used to solve classical optimization problems consisting of well-behaved mathematical properties generally required for these methods.

Moreover, three different classifications among the algorithms inspired by the nature family are discussed. These classes consist of the class of evolutionary algorithms, the class of swarm intelligence-based techniques, and the other metaheuristics. From these classifications, evolutionary algorithms are further grouped into five different subcategories. These are Genetic Algorithm (GA), Nondominated Shorting Genetic Algorithm II (NSGA-II), Multi Objective Evolutionary Algorithm based on Decomposition (MOEA/D), Strength Pareto Evolutionary Algorithm 2 (SPEA2), and Pareto Envelope-based Selection Algorithm II (PESA-II). Swarm intelligence-based algorithms are classified into five broader subcategories: Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Ant Colony Optimization (ACO), Firefly Algorithm (FA), and Fruit Fly Optimization (FFO) algorithm. Some of the algorithms possess similar characteristics to those discussed above and get inspiration from the biological phenomenon, but they cannot be put into the above classifications. Whereas some algorithms are inspired from the law of physics. Among these algorithms are the Artificial Neural Network (ANN), Flower Pollination Algorithm (FPA), Simulated Annealing (SA), and Gravitational Search Algorithm (GSA). In addition, recently published articles on real-world applications have been included to demonstrate each solution technique's advantages. The research survey and analysis carried out in this work provide a comprehensive perspective on adopting the most appropriate and frequently used solution method for different optimization problems depending on the complexity of the problem. In addition, research issues and gaps related to the solution methods are identified, and the scope of future developments is also discussed.

Some of these key areas are identified as a future scope for researchers to further development of NIAs. This future scope consists of designing new robust and efficient population-based optimization algorithms for solving multi-objective optimization problems that have a pretty balance of exploration and exploitation and require less computation time. Another key area is to design a robust procedure that can handle optimization problems involving many decision variables, hard practical constraints, and multi objective scenarios. Also, more focus can be drawn for the optimization algorithm that is easier to implementation, faster, and more efficient.

Declaration of competing interest

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

Data availability

No data was used for the research described in the article.

References

[1] Marler RT, Arora JS. Survey of multi-objective optimization methods for engineering. Struct Multidiscip Optim 2004;26(6):369–95.
[2] Ponsich A, Jaimes AL, Coello CAC. A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications. IEEE Trans Evol Comput 2012;17(3):321–44.
[3] Hiriart-Urruty J-B, Lemaréchal C. Convex analysis and minimization algorithms I: Fundamentals, vol. 305. Springer science & business media; 1996.
[4] Bertsimas D, Tsitsiklis JN. Introduction to linear optimization, vol. 6. Athena Scientific Belmont, MA; 1997.
[5] Boyd SP, Vandenberghe L. Convex optimization. Cambridge University Press; 2004.
[6] Rockafellar RT. Conjugate duality and optimization. SIAM; 1974.
[7] Auslender A, Teboulle M. Duality in optimization problems. Springer; 2003.
[8] Yang X-S. Nature-inspired optimization algorithms. Academic Press; 2020.
[9] Aksaraylı M, Pala O. A polynomial goal programming model for portfolio optimization based on entropy and higher moments. Expert Syst Appl 2018;94:185–92.
[10] Proelss J, Schweizer D. Polynomial goal programming and the implicit higher moment preferences of US institutional investors in hedge funds. Financial Mark Portfolio Manag 2014;28(1):1–28.
[11] Tamiz M, Jones D, Romero C. Goal programming for decision making: An overview of the current state-of-the-art. European J Oper Res 1998;111(3):569–81.
[12] Zitzler E. Evolutionary algorithms for multiobjective optimization: Methods and applications, vol. 63. Shaker Ithaca; 1999.
[13] Hazan E. Introduction to online convex optimization. 2019, arXiv preprint arXiv:1909.05207.
[14] Elhedhli S, Goffin J-L, Vial J-P. Nondifferentiable optimization: Introduction, applications and algorithms. Groupe d'études et de recherche en analyse des décisions, École des hautes …; 2000.
[15] Deb K, et al. Evolutionary algorithms for multi-criterion optimization in engineering design. Evol Algorithms Eng Comput Sci 1999;2:135–61.
[16] Boukouvala F, Misener R, Floudas CA. Global optimization advances in mixed-integer nonlinear programming, MINLP, and constrained derivative-free optimization, CDFO. European J Oper Res 2016;252(3):701–27.
[17] Shukla PK, Deb K. On finding multiple Pareto-optimal solutions using classical and evolutionary generating methods. European J Oper Res 2007;181(3):1630–52.
[18] Bussieck MR, Drud AS, Meeraus A. MINLPLib—a collection of test models for mixed-integer nonlinear programming. INFORMS J Comput 2003;15(1):114–9.
[19] Tanabe R, Ishibuchi H. An easy-to-use real-world multi-objective optimization problem suite. Appl Soft Comput 2020;89:106078.
[20] Zang H, Zhang S, Hapeshi K. A review of nature-inspired algorithms. J Bionic Eng 2010;7(4):S232–7.
[21] Fister Jr I, Yang X-S, Fister I, Brest J, Fister D. A brief review of nature-inspired algorithms for optimization. 2013, arXiv preprint arXiv:1307.4186.
[22] Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Trans Evol Comput 1997;1(1):67–82.
[23] Stadler W. Multicriteria optimization in engineering and in the sciences, vol. 37. Springer Science & Business Media; 1988.
[24] De Weck OL. Multiobjective optimization: History and promise. In: Invited keynote paper, GL2-2, the third China-Japan-Korea joint symposium on optimization of structural and mechanical systems, vol. 2. 2004, p. 34.
[25] Coello CAC, Lamont GB, Van Veldhuizen DA, et al. Evolutionary algorithms for solving multi-objective problems, vol. 5. Springer; 2007.
[26] Mansfield E, Yohe G. Microeconomics: Theory/applications. 7th ed. USA: WW Norton & Company, Inc.; 2004.
[27] Rao SS. Engineering optimization: Theory and practice. John Wiley & Sons; 2019.
[28] Fioretto F, Pontelli E, Yeoh W. Distributed constraint optimization problems and applications: A survey. J Artificial Intelligence Res 2018;61:623–98.
[29] Chen B, Zhong J, Chen Y. A hybrid approach for portfolio selection with higher-order moments: Empirical evidence from Shanghai Stock Exchange. Expert Syst Appl 2020;145:113104.
[30] Antil H, Khatri R, Löhner R, Verma D. Fractional deep neural network via constrained optimization. Mach Learn: Sci Technol 2020;2(1):015003.
[31] Fan S-KS, Jen C-H. An enhanced partial search to particle swarm optimization for unconstrained optimization. Mathematics 2019;7(4):357.
[32] Mohamed AK, Mohamed AW. Real-parameter unconstrained optimization based on enhanced agde algorithm. In: Machine learning paradigms: Theory and application. Springer; 2019, p. 431–50.
[33] Rezaiee-Pajand M, Entezami A, Sarmadi H. A sensitivity-based finite element model updating based on unconstrained optimization problem and regularized solution methods. Struct Control Health Monit 2020;27(5):e2481.
[34] Abubakar AB, Kumam P, Malik M, Chaipunya P, Ibrahim AH. A hybrid FR-DY conjugate gradient algorithm for unconstrained optimization with application in portfolio selection. AIMS Math 2021;6(6):6506–27.
[35] Abubakar AB, Kumam P, Malik M, Ibrahim AH. A hybrid conjugate gradient based approach for solving unconstrained optimization and motion control problems. Math Comput Simulation 2022;201:640–57.
[36] Floudas CA. Nonlinear and mixed-integer optimization: Fundamentals and applications. Oxford University Press; 1995.
[37] Aneja YP, Nair KP. Bicriteria transportation problem. Manage Sci 1979;25(1):73–8.
[38] Meyer JR, Kain JF, Wohl M. The urban transportation problem. Harvard University Press; 2013.
[39] Chizat L, Peyré G, Schmitzer B, Vialard F-X. Scaling algorithms for unbalanced optimal transport problems. Math Comp 2018;87(314):2563–609.
[40] Ribeiro CC, Minoux M, Penna MC. An optimal column-generation-with-ranking algorithm for very large scale set partitioning problems in traffic assignment. European J Oper Res 1989;41(2):232–9.
[41] Xu S, Jiang W, Deng X, Shou Y. A modified Physarum-inspired model for the user equilibrium traffic assignment problem. Appl Math Model 2018;55:340–53.
[42] Long J, Chen J, Szeto W, Shi Q. Link-based system optimum dynamic traffic assignment problems with environmental objectives. Transp Res D 2018;60:56–75.
[43] Preparata FP, Metze G, Chien RT. On the connection assignment problem of diagnosable systems. IEEE Trans Electron Comput 1967;(6):848–54.
[44] Antonante P, Spivak DI, Carlone L. Monitoring and diagnosability of perception systems. In: 2021 IEEE/RSJ international conference on intelligent robots and systems. IEEE; 2021, p. 168–75.
[45] Grillo H, Alemany M, Caldwell E. Human resource allocation problem in the industry 4.0: A reference framework. Comput Ind Eng 2022;169:108110.
[46] Liles IV JM, Robbins MJ, Lunday BJ. Improving defensive air battle management by solving a stochastic dynamic assignment problem via approximate dynamic programming. European J Oper Res 2023;305(3):1435–49.
[47] Bergman D, Cire AA, Van Hoeve W-J, Hooker JN. Discrete optimization with decision diagrams. INFORMS J Comput 2016;28(1):47–66.
[48] Hussien AG, Hassanien AE, Houssein EH, Amin M, Azar AT. New binary whale optimization algorithm for discrete optimization problems. Eng Optim 2020;52(6):945–59.

[49] Bähner FD, Prado-Rubio OA, Huusom JK. Discrete-continuous dynamic simulation of plantwide batch process systems in MATLAB. Chem Eng Res Des 2020;159:66–77.
[50] Gerlach AR, Leonard A, Rogers J, Rackauckas C. The koopman expectation: an operator theoretic method for efficient analysis and optimization of uncertain hybrid dynamical systems. 2020, arXiv preprint arXiv:2008.08737.
[51] Bansal V, Perkins JD, Pistikopoulos EN. A case study in simultaneous design and control using rigorous, mixed-integer dynamic optimization models. Ind Eng Chem Res 2002;41(4):760–78.
[52] Oldenburg J, Marquardt W, Heinz D, Leineweber DB. Mixed-logic dynamic optimization applied to batch distillation process design. AIChE J 2003;49(11):2900–17.
[53] Cheng L, Liang X, Bai J, Chen Q, Lemon J, To A. On utilizing topology optimization to design support structure to prevent residual stress induced build failure in laser powder bed metal additive manufacturing. Addit Manuf 2019;27:290–304.
[54] Censor Y. Pareto optimality in multiobjective problems. Appl Math Optim 1977;4(1):41–59.
[55] Pareto V. Cours dEconomie Politique: Professe a Universjte de Lausanne, vol. 1. F. Rouge; 1896.
[56] Miettinen K. Nonlinear multiobjective optimization, vol. 12. Springer Science & Business Media; 2012.
[57] Bechikh S, Said LB, Ghedira K. Estimating nadir point in multi-objective optimization using mobile reference points. In: IEEE congress on evolutionary computation. IEEE; 2010, p. 1–9.
[58] Miettinen K, Mäkelä MM, Kaario K. Experiments with classification-based scalarizing functions in interactive multiobjective optimization. European J Oper Res 2006;175(2):931–47.
[59] Deb K. Multi-objective optimization using evolutionary algorithms: an introduction. In: Multi-objective evolutionary optimization for product design and manufacturing. Springer; 2011, p. 3–34.
[60] Raidl GR, Puchinger J. Combining (integer) linear programming techniques and metaheuristics for combinatorial optimization. Hybrid Metaheuristics 2008;31–62.
[61] Wolsey LA, Nemhauser GL. Integer and combinatorial optimization, vol. 55. John Wiley & Sons; 1999.
[62] Dantzig G. Linear programming and extensions. Princeton University Press; 2016.
[63] Lawler EL, Wood DE. Branch-and-bound methods: A survey. Oper Res 1966;14(4):699–719.
[64] Nelder JA, Mead R. A simplex method for function minimization. Comput J 1965;7(4):308–13.
[65] Luenberger DG, Ye Y, et al. Linear and nonlinear programming, vol. 2. Springer; 1984.
[66] Fisher ML. The Lagrangian relaxation method for solving integer programming problems. Manage Sci 1981;27(1):1–18.
[67] Kelley Jr JE. The cutting-plane method for solving convex programs. J Soc Ind Appl Math 1960;8(4):703–12.
[68] Padberg M, Rinaldi G. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. SIAM Rev 1991;33(1):60–100.
[69] Bland RG, Goldfarb D, Todd MJ. The ellipsoid method: A survey. Oper Res 1981;29(6):1039–91.
[70] Khoufi I, Laouiti A, Adjih C. A survey of recent extended variants of the traveling salesman and vehicle routing problems for unmanned aerial vehicles. Drones 2019;3(3):66.
[71] Zhen L, Li M, Laporte G, Wang W. A vehicle routing problem arising in unmanned aerial monitoring. Comput Oper Res 2019;105:1–11.
[72] Roberti R, Ruthmair M. Exact methods for the traveling salesman problem with drone. Transp Sci 2021;55(2):315–35.
[73] Dorfman R. Application of linear programming to the theory of the firm. University of California Press; 2020.
[74] Konno H, Suzuki K-i. A mean-variance-skewness portfolio optimization model. J Oper Res Soc Japan 1995;38(2):173–87.
[75] Mansini R, Ogryczak W, Speranza MG. Linear and mixed integer programming for portfolio optimization. In: EURO: The association of European operational research societies, vol. 21. Springer; 2015.
[76] Fernández-Navarro F, Martínez-Nieto L, Carbonero-Ruz M, Montero-Romero T. Mean squared variance portfolio: A mixed-integer linear programming formulation. Mathematics 2021;9(3):223.
[77] Mashayekh S, Stadler M, Cardoso G, Heleno M. A mixed integer linear programming approach for optimal DER portfolio, sizing, and placement in multi-energy microgrids. Appl Energy 2017;187:154–68.
[78] Benati S, Rizzi R. A mixed integer linear programming formulation of the optimal mean/value-at-risk portfolio problem. European J Oper Res 2007;176(1):423–34.
[79] Bussieck MR, Pruessner A, et al. Mixed-integer nonlinear programming. SIAG/OPT Newslett Views News 2003;14(1):19–22.
[80] Díaz J, Cortés M, Hernández J, Clavijo Ó, Ardila C, Cabrales S. Index fund optimization using a hybrid model: genetic algorithm and mixed-integer nonlinear programming. Eng Econ 2019;64(3):298–309.
[81] D'Ambrosio C, Lodi A. Mixed integer nonlinear programming tools: an updated practical overview. Ann Oper Res 2013;204(1):301–20.
[82] Gupta OK, Ravindran A. Branch and bound experiments in convex nonlinear integer programming. Manage Sci 1985;31(12):1533–46.
[83] Viswanathan J, Grossmann IE. A combined penalty function and outer-approximation method for MINLP optimization. Comput Chem Eng 1990;14(7):769–82.
[84] Westerlund T, Pettersson F. An extended cutting plane method for solving convex MINLP problems. Comput Chem Eng 1995;19:131–6.
[85] Mangasarian OL. Nonlinear programming. SIAM; 1994.
[86] Nocedal J, Wright SJ. Quadratic programming. Numerical optimization 2006;448–92.
[87] Wright SJ. Primal-dual interior-point methods. SIAM; 1997.
[88] Wong E. Active-set methods for quadratic programming. San Diego: University of California; 2011.
[89] Bertsekas DP. Constrained optimization and lagrange multiplier methods. Academic Press; 2014.
[90] Hestenes MR. Multiplier and gradient methods. J Optim Theory Appl 1969;4(5):303–20.
[91] Powell MJ. A method for nonlinear constraints in minimization problems. Optimization 1969;283–98.
[92] Hamdi A, Mishra SK. Decomposition methods based on augmented Lagrangians: a survey. In: Topics in nonconvex optimization: Theory and applications. Springer; 2011, p. 175–203.
[93] Rockafellar RT. A dual approach to solving nonlinear programming problems by unconstrained optimization. Math Program 1973;5(1):354–73.
[94] Wolfe P. The simplex method for quadratic programming. Econometrica 1959;382–98.
[95] Rockafellar RT. Monotone operators and the proximal point algorithm. SIAM J Control Optim 1976;14(5):877–98.
[96] Bubeck S, et al. Convex optimization: Algorithms and complexity. Found Trends Mach Learn 2015;8(3–4):231–357.
[97] Chen G, Teboulle M. A proximal-based decomposition method for convex minimization problems. Math Program 1994;64(1–3):81–101.
[98] Parikh N, Boyd S, et al. Proximal algorithms. Found Trends Optim 2014;1(3):127–239.
[99] Gabay D, Mercier B. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. Comput Math Appl 1976;2(1):17–40.
[100] Gould NI, Toint PL. Numerical methods for large-scale non-convex quadratic programming. Trends Ind Appl Math 2002;149–79.
[101] Montoya OD, Gil-González W, Garces A. Sequential quadratic programming models for solving the OPF problem in DC grids. Electr Power Syst Res 2019;169:18–23.
[102] Wu C, Gu W, Zhou S, Chen X. Coordinated optimal power flow for integrated active distribution network and virtual power plants using decentralized algorithm. IEEE Trans Power Syst 2021;36(4):3541–51.

[103] Bouyarmane K, Chappellet K, Vaillant J, Kheddar A. Quadratic programming for multirobot and task-space force control. IEEE Trans Robot 2018;35(1):64–77.
[104] Witkowska A, Śmierzchalski R. Adaptive dynamic control allocation for dynamic positioning of marine vessel based on backstepping method and sequential quadratic programming. Ocean Eng 2018;163:570–82.
[105] Killian M, Zauner M, Kozek M. Comprehensive smart home energy management system using mixed-integer quadratic-programming. Appl Energy 2018;222:662–72.
[106] Yang C, Wang M, Wang W, Pu Z, Ma M. An efficient vehicle-following predictive energy management strategy for PHEV based on improved sequential quadratic programming algorithm. Energy 2021;219:119595.
[107] Horst R, Thoai NV. DC programming: overview. J Optim Theory Appl 1999;103:1–43.
[108] Nhat PD, Le HM, Le Thi HA. Accelerated difference of convex functions algorithm and its application to sparse binary logistic regression. In: IJCAI. 2018, p. 1369–75.
[109] Le Thi HA, Le HM, Nguyen VV, Pham Dinh T. A DC programming approach for feature selection in support vector machines learning. Adv Data Anal Classif 2008;2:259–78.
[110] López J, Maldonado S, Carrasco M. Double regularization methods for robust feature selection and SVM classification via DC programming. Inform Sci 2018;429:377–89.
[111] El Halabi M, Orfanides G, Hoheisel T. Difference of submodular minimization via DC programming. In: International Conference on Machine Learning. PMLR; 2023, p. 9172–201.
[112] Thi HAL, Dinh TP, Luu HPH, Le HM. Deterministic and stochastic DCA for DC programming. In: Springer Handbook of Engineering Statistics. Springer; 2023, p. 675–702.
[113] Pham Dinh T, Le Thi HA, Pham VN, Niu Y-S. DC programming approaches for discrete portfolio optimization under concave transaction costs. Optim Lett 2016;10:261–82.
[114] Hooshmand F, MirHassani S. Efficient DC algorithm for the index-tracking problem. In: Intelligent computing & optimization: Proceedings of the 4th international conference on intelligent computing and optimization 2021. Springer; 2022, p. 566–76.
[115] Tayi GK. A polynomial goal programming approach to a class of quality control problems. J Oper Manage 1985;5(2):237–46.
[116] Mandal PK, Thakur M. Higher-order moments in portfolio selection problems: A comprehensive literature review. Expert Syst Appl 2023;121625. http://dx.doi.org/10.1016/j.eswa.2023.121625.
[117] Jalota H, Mandal PK, Thakur M, Mittal G. A novel approach to incorporate investor's preference in fuzzy multi-objective portfolio selection problem using credibility measure. Expert Syst Appl 2023;118583.
[118] Mittal SK, Srivastava N. Mean-variance-skewness portfolio optimization under uncertain environment using improved genetic algorithm. Artif Intell Rev 2021;54(8):6011–32.
[119] Pahade JK, Jha M. Credibilistic variance and skewness of trapezoidal fuzzy variable and mean–variance–skewness model for portfolio selection. Results Appl Math 2021;11:100159.
[120] Kaucic M, Barbini F, Camerota Verdù FJ. Polynomial goal programming and particle swarm optimization for enhanced indexation. Soft Comput 2020;24(12):8535–51.
[121] Gür Ş, Eren T. Scheduling and planning in service systems with goal programming: Literature review. Mathematics 2018;6(11):265.
[122] Karacan I, Senvar O, Arslan O, Ekmekçi Y, Bulkan S. A novel approach integrating intuitionistic fuzzy analytical hierarchy process and goal programming for chickpea cultivar selection under stress conditions. Processes 2020;8(10):1288.
[123] Haq A, Modibbo UM, Ahmed A, Ali I. Mathematical modeling of sustainable development goals of India agenda 2030: a Neutrosophic programming approach. Environ Dev Sustain 2022;24(10):11991–2018.
[124] Slowik A, Kwasnicka H. Evolutionary algorithms and their applications to engineering problems. Neural Comput Appl 2020;32(16):12363–79.
[125] Dasgupta D, Michalewicz Z. Evolutionary algorithms in engineering applications. Springer Science & Business Media; 2013.
[126] Metaxiotis K, Liagkouras K. Multiobjective evolutionary algorithms for portfolio management: A comprehensive literature review. Expert Syst Appl 2012;39(14):11685–98.
[127] Tapia MGC, Coello CAC. Applications of multi-objective evolutionary algorithms in economics and finance: A survey. In: 2007 IEEE congress on evolutionary computation. IEEE; 2007, p. 532–9.
[128] Coello CAC, Lamont GB. Applications of multi-objective evolutionary algorithms, vol. 1. World Scientific; 2004.
[129] Bäck T, Schwefel H-P. An overview of evolutionary algorithms for parameter optimization. Evol Comput 1993;1(1):1–23.
[130] Zhou A, Qu B-Y, Li H, Zhao S-Z, Suganthan PN, Zhang Q. Multiobjective evolutionary algorithms: A survey of the state of the art. Swarm Evol Comput 2011;1(1):32–49.
[131] Holland JH. Genetic algorithms. Sci Am 1992;267(1):66–73.
[132] Darwin C. On the origin of species, 1859. Routledge; 2004.
[133] Sun Y, Xue B, Zhang M, Yen GG, Lv J. Automatically designing CNN architectures using the genetic algorithm for image classification. IEEE Trans Cybern 2020;50(9):3840–54.
[134] Jiacheng L, Lei L. A hybrid genetic algorithm based on information entropy and game theory. IEEE Access 2020;8:36602–11.
[135] Reddy GT, Reddy MPK, Lakshmanna K, Rajput DS, Kaluri R, Srivastava G. Hybrid genetic algorithm and a fuzzy logic classifier for heart disease diagnosis. Evol Intell 2020;13:185–96.
[136] Hemanth DJ, Anitha J. Modified genetic algorithm approaches for classification of abnormal magnetic resonance brain tumour images. Appl Soft Comput 2019;75:21–8.
[137] Luan J, Yao Z, Zhao F, Song X. A novel method to solve supplier selection problem: Hybrid algorithm of genetic algorithm and ant colony optimization. Math Comput Simulation 2019;156:294–309.
[138] Katoch S, Chauhan SS, Kumar V. A review on genetic algorithm: past, present, and future. Multimedia Tools Appl 2021;80(5):8091–126.
[139] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 2002;6(2):182–97.
[140] Verma S, Pant M, Snasel V. A comprehensive review on NSGA-II for multi-objective combinatorial optimization problems. IEEE Access 2021;9:57757–91.
[141] Lu S, Zhang N, Jia L. A multiobjective multiperiod mean-semientropy-skewness model for uncertain portfolio selection. Appl Intell 2021;51(8):5233–58.
[142] Li C, Wu Y, Lu Z, Wang J, Hu Y. A multiperiod multiobjective portfolio selection model with fuzzy random returns for large scale securities data. IEEE Trans Fuzzy Syst 2020;29(1):59–74.
[143] Pal R, Chaudhuri TD, Mukhopadhyay S. Portfolio formation and optimization with continuous realignment: a suggested method for choosing the best portfolio of stocks using variable length NSGA-II. Expert Syst Appl 2021;186:115732.
[144] Sadeghi A, Daneshvar A, Zaj MM. Combined ensemble multi-class SVM and fuzzy NSGA-II for trend forecasting and trading in forex markets. Expert Syst Appl 2021;185:115566.
[145] Rabbani M, Oladzad-Abbasabady N, Akbarian-Saravi N. Ambulance routing in disaster response considering variable patient condition: NSGA-II and MOPSO algorithms. J Ind Manag Optim 2022;18(2):1035.
[146] Soui M, Mansouri N, Alhamad R, Kessentini M, Ghedira K. NSGA-II as feature selection technique and AdaBoost classifier for COVID-19 prediction using patient's symptoms. Nonlinear Dynam 2021;106(2):1453–75.

[147] Ala A, Alsaadi FE, Ahmadi M, Mirjalili S. Optimization of an appointment scheduling problem for healthcare systems based on the quality of fairness service using whale optimization algorithm and NSGA-II. Sci Rep 2021;11(1):1–19.
[148] Li H, Wang B, Yuan Y, Zhou M, Fan Y, Xia Y. Scoring and dynamic hierarchy-based NSGA-II for multiobjective workflow scheduling in the cloud. IEEE Trans Autom Sci Eng 2021;19(2):982–93.
[149] Xu X, Fu S, Li W, Dai F, Gao H, Chang V. Multi-objective data placement for workflow management in cloud infrastructure using NSGA-II. IEEE Trans Emerging Top Comput Intell 2020;4(5):605–15.
[150] Xu X, Fu S, Yuan Y, Luo Y, Qi L, Lin W, et al. Multiobjective computation offloading for workflow management in cloudlet-based mobile cloud using NSGA-II. Comput Intell 2019;35(3):476–95.
[151] Lalehzari R, Boroomand Nasab S, Moazed H, Haghighi A, Yaghoobzadeh M. Simulation–optimization modelling for water resources management using nsgaii-oip and modflow. Irrigation Drainage 2020;69(3):317–32.
[152] Liu D, Huang Q, Yang Y, Liu D, Wei X. Bi-objective algorithm based on NSGA-II framework to optimize reservoirs operation. J Hydrol 2020;585:124830.
[153] Kanwar V, Kumar A. Multiobjective optimization-based DV-hop localization using NSGA-II algorithm for wireless sensor networks. Int J Commun Syst 2020;33(11):e4431.
[154] Harizan S, Kuila P. A novel NSGA-II for coverage and connectivity aware sensor node scheduling in industrial wireless sensor networks. Digit Signal Process 2020;105:102753.
[155] Wang S, Zhao D, Yuan J, Li H, Gao Y. Application of NSGA-II algorithm for fault diagnosis in power system. Electr Power Syst Res 2019;175:105893.
[156] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm. TIK-Rep 2001;103.
[157] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Trans Evol Comput 1999;3(4):257–71.
[158] Silverman BW. Density estimation for statistics and data analysis. Routledge; 2018.
[159] Dariane AB, Sabokdast MM, Karami F, Asadi R, Ponnambalam K, Mousavi SJ. Integrated operation of multi-reservoir and many-objective system using fuzzified hedging rule and strength Pareto evolutionary optimization algorithm (SPEA2). Water 2021;13(15):1995.
[160] Biswal SR, Shankar G. Simultaneous optimal allocation and sizing of DGs and capacitors in radial distribution systems using SPEA2 considering load uncertainty. IET Gener Transm Distrib 2020;14(3):494–505.
[161] Liu X, Zhang D. An improved SPEA2 algorithm with local search for multi-objective investment decision-making. Appl Sci 2019;9(8):1675.
[162] Zhang Q, Li H. MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 2007;11(6):712–31.
[163] Trivedi A, Srinivasan D, Sanyal K, Ghosh A. A survey of multiobjective evolutionary algorithms based on decomposition. IEEE Trans Evol Comput 2016;21(3):440–62.
[164] Jiang E-d, Wang L. An improved multi-objective evolutionary algorithm based on decomposition for energy-efficient permutation flow shop scheduling problem with sequence-dependent setup time. Int J Prod Res 2019;57(6):1756–71.
[165] Zhang B, Pan Q-k, Gao L, Meng L-L, Li X-Y, Peng K-K. A three-stage multiobjective approach based on decomposition for an energy-efficient hybrid flow shop scheduling problem. IEEE Trans Syst Man Cybern 2019;50(12):4984–99.
[166] Xu Y, Ding O, Qu R, Li K. Hybrid multi-objective evolutionary algorithms based on decomposition for wireless sensor network coverage optimization. Appl Soft Comput 2018;68:268–82.
[167] Kang Q, Song X, Zhou M, Li L. A collaborative resource allocation strategy for decomposition-based multiobjective evolutionary algorithms. IEEE Trans Syst Man Cybern 2018;49(12):2416–23.
[168] Corne DW, Jerram NR, Knowles JD, Oates MJ. PESA-II: Region-based selection in evolutionary multiobjective optimization. In: Proceedings of the 3rd annual conference on genetic and evolutionary computation. 2001, p. 283–90.
[169] Omidi Brojeni P, Abazari S, Madani M. PESA II algorithm-based optimal coordination of directional overcurrent relays in microgrid. Comput Intell Electr Eng 2022;13(2):51–64.
[170] Chakkarapani K, Thangavelu T, Dharmalingam K. Thermal analysis of brushless DC motor using multiobjective optimization. Int Trans Electr Energy Syst 2020;30(10):e12546.
[171] Khanra M, Nandi AK. Optimal driving based trip planning of electric vehicles using evolutionary algorithms: A driving assistance system. Appl Soft Comput 2020;93:106361.
[172] Ahmed W, Selim A, Kamel S, Yu J, Jurado F. Probabilistic load flow solution considering optimal allocation of SVC in radial distribution system. Int J Interact Multimed Artif Intell IJIMAI 5.
[173] Chakraborty A, Kar AK. Swarm intelligence: A review of algorithms. In: Nature-inspired computing and optimization: Theory and applications. Springer; 2017, p. 475–94.
[174] Brezočnik L, Fister Jr I, Podgorelec V. Swarm intelligence algorithms for feature selection: a review. Appl Sci 2018;8(9):1521.
[175] Ertenlice O, Kalayci CB. A survey of swarm intelligence for portfolio optimization: Algorithms and applications. Swarm Evol Comput 2018;39:36–52.
[176] Chen Y, Zhao X, Yuan J. Swarm intelligence algorithms for portfolio optimization problems: Overview and recent advances. Mob Inf Syst 2022;2022.
[177] Sun W, Tang M, Zhang L, Huo Z, Shu L. A survey of using swarm intelligence algorithms in IoT. Sensors 2020;20(5):1420.
[178] Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks, vol. 4. IEEE; 1995, p. 1942–8.
[179] Pradhan A, Bisoy SK, Das A. A survey on PSO based meta-heuristic scheduling mechanism in cloud computing environment. J King Saud Univ Comput Inf Sci 2021.
[180] Deng W, Ni H, Liu Y, Chen H, Zhao H. An adaptive differential evolution algorithm based on belief space and generalized opposition-based learning for resource allocation. Appl Soft Comput 2022;127:109419.
[181] Deng W, Xu J, Zhao H, Song Y. A novel gate resource allocation method using improved PSO-based QEA. IEEE Trans Intell Transp Syst 2020.
[182] Ren X, Liu S, Yu X, Dong X. A method for state-of-charge estimation of lithium-ion batteries based on PSO-LSTM. Energy 2021;234:121236.
[183] Ghorbani N, Kasaeian A, Toopshekan A, Bahrami L, Maghami A. Optimizing a hybrid wind-PV-battery system using GA-PSO and MOPSO for reducing cost and increasing reliability. Energy 2018;154:581–91.
[184] Shaheen MA, Hasanien HM, Alkuhayli A. A novel hybrid GWO-PSO optimization technique for optimal reactive power dispatch problem solution. Ain Shams Eng J 2021;12(1):621–30.
[185] Priyadarshi N, Padmanaban S, Holm-Nielsen JB, Blaabjerg F, Bhaskar MS. An experimental estimation of hybrid ANFIS–PSO-based MPPT for PV grid integration under fluctuating sun irradiance. IEEE Syst J 2019;14(1):1218–29.
[186] Moayedi H, Mehrabi M, Mosallanezhad M, Rashid ASA, Pradhan B. Modification of landslide susceptibility mapping using optimized PSO-ann technique. Eng Comput 2019;35(3):967–84.
[187] Song B, Wang Z, Zou L. An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve. Appl Soft Comput 2021;100:106960.
[188] Dorigo M, Birattari M, Stutzle T. Ant colony optimization. IEEE Comput Intell Mag 2006;1(4):28–39.
[189] Deneubourg J-L, Aron S, Goss S, Pasteels JM. The self-organizing exploratory pattern of the argentine ant. J Insect Behav 1990;3(2):159–68.
[190] Cordón García O, Herrera Triguero F, Stützle T. A review on the ant colony optimization metaheuristic: Basis, models and new trends. Mathware Soft Comput 9(2).

[191] Karaboga D, et al. An idea based on honey bee swarm for numerical optimization. Technical Report-tr06, Erciyes University, Engineering Faculty, Computer; 2005.
[192] Karaboga D, Akay B. A comparative study of artificial bee colony algorithm. Appl Math Comput 2009;214(1):108–32.
[193] Bansal JC, Sharma H, Jadon SS. Artificial bee colony algorithm: a survey. Int J Adv Intell Paradigms 2013;5(1/2):123–59.
[194] Bolaji AL, Khader AT, Al-Betar MA, Awadallah MA. Artificial bee colony algorithm, its variants and applications: A survey. J Theor Appl Inf Technol 2013;47(2).
[195] Yang X-S. Firefly algorithms for multimodal optimization. In: International symposium on stochastic algorithms. Springer; 2009, p. 169–78.
[196] Dey N, Chaki J, Moraru L, Fong S, Yang X-S. Firefly algorithm and its variants in digital image processing: A comprehensive review. In: Applications of firefly algorithm and its variants: Case studies and new developments. Springer; 2020, p. 1–28.
[197] Nayak J, Naik B, Dinesh P, Vakula K, Dash PB. Firefly algorithm in biomedical and health care: advances, issues and challenges. SN Comput Sci 2020;1(6):311.
[198] Zivkovic M, Bacanin N, Tuba E, Strumberger I, Bezdan T, Tuba M. Wireless sensor networks life time optimization based on the improved firefly algorithm. In: 2020 international wireless communications and mobile computing. IEEE; 2020, p. 1176–81.
[199] Selvakumar B, Muneeswaran K. Firefly algorithm based feature selection for network intrusion detection. Comput Secur 2019;81:148–55.
[200] Kumar V, Kumar D. A systematic review on firefly algorithm: past, present, and future. Arch Comput Methods Eng 2021;28(4):3269–91.
[201] Pan W-T. A new fruit fly optimization algorithm: taking the financial distress model as an example. Knowl-Based Syst 2012;26:69–74.
[202] Pan Q-K, Sang H-Y, Duan J-H, Gao L. An improved fruit fly optimization algorithm for continuous function optimization problems. Knowl-Based Syst 2014;62:69–83.
[203] Aggarwal A, Dimri P, Agarwal A, Verma M, Alhumyani HA, Masud M. IFFO: an improved fruit fly optimization algorithm for multiple workflow scheduling minimizing cost and makespan in cloud computing environments. Math Probl Eng 2021;2021:1–9.
[204] Bezdan T, Stoean C, Naamany AA, Bacanin N, Rashid TA, Zivkovic M, et al. Hybrid fruit-fly optimization algorithm with k-means for text document clustering. Mathematics 2021;9(16):1929.
[205] Liu A, Deng X, Ren L, Liu Y, Liu B. An inverse power generation mechanism based fruit fly algorithm for function optimization. J Syst Sci Complex 2019;32(2):634–56.
[206] Qin S, Pi D, Shao Z, Xu Y. Hybrid collaborative multi-objective fruit fly optimization algorithm for scheduling workflow in cloud environment. Swarm Evol Comput 2022;68:101008.
[207] Fan Y, Wang P, Mafarja M, Wang M, Zhao X, Chen H. A bioinformatic variant fruit fly optimizer for tackling optimization problems. Knowl-Based Syst 2021;213:106704.
[208] Haykin S, Network N. A comprehensive foundation. Neural Netw 2004;2(2004):41.
[209] McCulloch WS, Pitts W. A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 1943;5(4):115–33.
[210] Abiodun OI, Jantan A, Omolara AE, Dada KV, Mohamed NA, Arshad H. State-of-the-art in artificial neural network applications: A survey. Heliyon 2018;4(11):e00938.
[211] Filip FG. Decision support and control for large-scale complex systems. Annu Rev Control 2008;32(1):61–70.
[212] Yegnanarayana B. Artificial neural networks. PHI Learning Pvt. Ltd.; 2009.
[213] Mohammadi S, Nazemi A. On portfolio management with value at risk and uncertain returns via an artificial neural network scheme. Cogn Syst Res 2020;59:247–63.
[214] Vo NN, He X, Liu S, Xu G. Deep learning for decision making and the optimization of socially responsible investments and portfolio. Decis Support Syst 2019;124:113097.
[215] Abiodun OI, Jantan A, Omolara AE, Dada KV, Umar AM, Linus OU, et al. Comprehensive review of artificial neural network applications to pattern recognition. IEEE Access 2019;7:158820–46.
[216] Elsheikh AH, Sharshir SW, Abd Elaziz M, Kabeel A, Guilan W, Haiou Z. Modeling of solar energy systems using artificial neural network: A comprehensive review. Sol Energy 2019;180:622–39.
[217] Harz HH, Rafi AO, Hijazi MO, Abu-Naser SS. Artificial neural network for predicting diabetes using JNN. Int J Acad Eng Res (IJAER) 2020;4(10).
[218] El_Jerjawi NS, Abu-Naser SS. Diabetes prediction using artificial neural network. Int J Adv Sci Technol 2018;121.
[219] Khalil AJ, Barhoom AM, Abu-Nasser BS, Musleh MM, Abu-Naser SS. Energy efficiency prediction using artificial neural network. Int J Acad Pedagog Res (IJAPR) 2019;3(9).
[220] Yang X-S. Flower pollination algorithm for global optimization. In: International conference on unconventional computing and natural computation. Springer; 2012, p. 240–9.
[221] Abdel-Basset M, Shawky LA. Flower pollination algorithm: a comprehensive review. Artif Intell Rev 2019;52(4):2533–57.
[222] Kirkpatrick S, Gelatt Jr CD, Vecchi MP. Optimization by simulated annealing. Science 1983;220(4598):671–80.
[223] Elgamal ZM, Yasin NBM, Tubishat M, Alswaitti M, Mirjalili S. An improved harris hawks optimization algorithm with simulated annealing for feature selection in the medical field. IEEE Access 2020;8:186638–52.
[224] Al-Turjman F, Zahmatkesh H, Mostarda L. Quantifying uncertainty in internet of medical things and big-data services using intelligence and deep learning. IEEE Access 2019;7:115749–59.
[225] Goodarzian F, Wamba SF, Mathiyazhagan K, Taghipour A. A new bi-objective green medicine supply chain network design under fuzzy environment: Hybrid metaheuristic algorithms. Comput Ind Eng 2021;160:107535.
[226] Fathollahi-Fard AM, Govindan K, Hajiaghaei-Keshteli M, Ahmadi A. A green home health care supply chain: New modified simulated annealing algorithms. J Clean Prod 2019;240:118200.
[227] Hosseini FS, Choubin B, Mosavi A, Nabipour N, Shamshirband S, Darabi H, et al. Flash-flood hazard assessment using ensembles and Bayesian-based machine learning models: application of the simulated annealing feature selection method. Sci Total Environ 2020;711:135161.
[228] Abdel-Basset M, Ding W, El-Shahat D. A hybrid Harris Hawks optimization algorithm with simulated annealing for feature selection. Artif Intell Rev 2021;54(1):593–637.
[229] Yang X-Y, Chen S-D, Liu W-L, Zhang Y. A multi-period fuzzy portfolio optimization model with short selling constraints. Int J Fuzzy Syst 2022;1–15.
[230] Som A, Kayal P. A multicountry comparison of cryptocurrency vs gold: Portfolio optimization through generalized simulated annealing. Blockchain Res Appl 2022;3(3):100075.
[231] Micev M, Ćalasan M, Ali ZM, Hasanien HM, Aleem SHA. Optimal design of automatic voltage regulation controller using hybrid simulated annealing–Manta ray foraging optimization algorithm. Ain Shams Eng J 2021;12(1):641–57.
[232] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: a gravitational search algorithm. Inform Sci 2009;179(13):2232–48.
[233] Shilaja C, Arunprasath T. Optimal power flow using moth swarm algorithm with gravitational search algorithm considering wind power. Future Gener Comput Syst 2019;98:708–15.
[234] Ebrahimi Mood S, Javidi MM. Energy-efficient clustering method for wireless sensor networks using modified gravitational search algorithm. Evol Syst 2020;11(4):575–87.
[235] Palanikkumar D, Anbuselvan P, Rithu B. A gravitational search algorithm for effective Web service selection for composition with enhanced QoS in SOA. Int J Comput Appl 2012;42(8):12–5.
[236] Rashedi E, Rashedi E, Nezamabadi-Pour H. A comprehensive survey on gravitational search algorithm. Swarm Evol Comput 2018;41:141–58.

[237] Geem ZW, Kim JH, Loganathan GV. A new heuristic optimization algorithm: harmony search. Simulation 2001;76(2):60–8.
[238] Fathian M, Amiri B, Maroosi A. Application of honey-bee mating optimization algorithm on clustering. Appl Math Comput 2007;190(2):1502–13.
[239] Pham DT, Castellani M. A comparative study of the Bees Algorithm as a tool for function optimisation. Cogent Eng 2015;2(1):1091540.
[240] Yang X-S. A new metaheuristic bat-inspired algorithm. In: Nature inspired cooperative strategies for optimization. Springer; 2010, p. 65–74.
[241] Ramli M, Abas ZA, Desa M, Abidin ZZ, Alazzam MB. Enhanced convergence of Bat Algorithm based on dimensional and inertia weight factor. J King Saud Univ Comput Inf Sci 2019;31(4):452–8.
[242] Yang X-S, Deb S. Engineering optimisation by cuckoo search. Int J Math Model Numer Optim 2010;1(4):330–43.
[243] Joshi AS, Kulkarni O, Kakandikar GM, Nandedkar VM. Cuckoo search optimization-a review. Mater Today Proc 2017;4(8):7262–9.
[244] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. Adv Eng Softw 2014;69:46–61.
[245] Hou Y, Gao H, Wang Z, Du C. Improved grey wolf optimization algorithm and application. Sensors 2022;22(10):3810.
[246] Teng Z-j, Lv J-l, Guo L-w. An improved hybrid grey wolf optimization algorithm. Soft Comput 2019;23:6617–31.
[247] Mittal N, Singh U, Sohi BS. Modified grey wolf optimizer for global engineering optimization. Appl Comput Intell Soft Comput 2016;2016.
[248] Gupta S, Deep K. A novel random walk grey wolf optimizer. Swarm Evol Comput 2019;44:101–12.
[249] Passino KM. Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Syst Mag 2002;22(3):52–67.
[250] Chen H, Zhang Q, Luo J, Xu Y, Zhang X. An enhanced bacterial foraging optimization and its application for training kernel extreme learning machine. Appl Soft Comput 2020;86:105884.
[251] Guo C, Tang H, Niu B, Lee CBP. A survey of bacterial foraging optimization. Neurocomputing 2021;452:728–46.
[252] Chu S-C, Tsai P-W, Pan J-S. Cat swarm optimization. In: PRICAI 2006: Trends in artificial intelligence: 9th Pacific Rim international conference on artificial intelligence Guilin, China, August 7-11, 2006 proceedings 9. Springer; 2006, p. 854–8.
[253] Li J, Gao M, Pan J-S, Chu S-C. A parallel compact cat swarm optimization and its application in DV-Hop node localization for wireless sensor network. Wirel Netw 2021;27:2081–101.
[254] Hatamlou A. Black hole: A new heuristic optimization approach for data clustering. Inform Sci 2013;222:175–84.
[255] Deeb H, Sarangi A, Mishra D, Sarangi SK. Improved Black Hole optimization algorithm for data clustering. J King Saud Univ Comput Inf Sci 2022;34(8):5020–9.
[256] Cuevas E, Cienfuegos M, Zaldívar D, Pérez-Cisneros M. A swarm optimization algorithm inspired in the behavior of the social-spider. Expert Syst Appl 2013;40(16):6374–84.
[257] Luque-Chang A, Cuevas E, Fausto F, Zaldivar D, Pérez M. Social spider optimization algorithm: modifications, applications, and perspectives. Math Probl Eng 2018;2018:1–29.
[258] Abd El Aziz M, Hassanien AE. An improved social spider optimization algorithm based on rough sets for solving minimum number attribute reduction problem. Neural Comput Appl 2018;30:2441–52.
[259] Rao RV, Savsani VJ, Vakharia D. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des 2011;43(3):303–15.
[260] Zheng Y-J. Water wave optimization: a new nature-inspired metaheuristic. Comput Oper Res 2015;55:1–11.
[261] Zhang J, Zhou Y, Luo Q. An improved sine cosine water wave optimization algorithm for global optimization. J Intell Fuzzy Systems 2018;34(4):2129–41.
[262] Zhao F, Shao D, Wang L, Xu T, Zhu N, et al. An effective water wave optimization algorithm with problem-specific knowledge for the distributed assembly blocking flow-shop scheduling problem. Knowl-Based Syst 2022;243:108471.
[263] Li C, Wu Y, Lu Z, Wang J, Hu Y. A multi-period multi-objective portfolio selection model with fuzzy random returns for large scale securities data. IEEE Trans Fuzzy Syst 2020.
[264] Deb K, Jain H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. IEEE Trans Evol Comput 2013;18(4):577–601.
[265] Jain H, Deb K. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach. IEEE Trans Evol Comput 2013;18(4):602–22.
[266] Meghwani SS, Thakur M. Multi-objective heuristic algorithms for practical portfolio optimization and rebalancing with transaction cost. Appl Soft Comput 2018;67:865–94.
[267] Chang T-J, Meade N, Beasley JE, Sharaiha YM. Heuristics for cardinality constrained portfolio optimisation. Comput Oper Res 2000;27(13):1271–302.
[268] Krink T, Paterlini S. Multiobjective optimization using differential evolution for real-world portfolio optimization. Comput Manag Sci 2011;8(1–2):157–79.
[269] Lwin K, Qu R, Kendall G. A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization. Appl Soft Comput 2014;24:757–72.
[270] Lwin KT, Qu R, MacCarthy BL. Mean-VaR portfolio optimization: A nonparametric approach. European J Oper Res 2017;260(2):751–66.
[271] Meghwani SS, Thakur M. Multi-criteria algorithms for portfolio optimization under practical constraints. Swarm Evol Comput 2017;37:104–25.
[272] Riquelme N, Von Lücken C, Baran B. Performance metrics in multi-objective optimization. In: 2015 Latin American computing conference. IEEE; 2015, p. 1–11.