Archives of Computational Methods in Engineering (2024) 31:4485–4519
https://doi.org/10.1007/s11831-024-10168-6

REVIEW ARTICLE

Metaheuristics for Solving Global and Engineering Optimization Problems: Review, Applications, Open Issues and Challenges

Essam H. Houssein[1] · Mahmoud Khalaf Saeed[1] · Gang Hu[2,3] · Mustafa M. Al-Sayed[1]

Received: 19 December 2023 / Accepted: 25 March 2024 / Published online: 21 August 2024

Abstract

The greatest and fastest advances in the computing world today require researchers to develop new problem-solving techniques capable of providing an optimal global solution considering a set of aspects and restrictions. Due to the superiority of the metaheuristic Algorithms (MAs) in solving different classes of problems and providing promising results, MAs need to be studied. Numerous studies of MAs algorithms in different fields exist, but in this study, a comprehensive review of MAs, its nature, types, applications, and open issues are introduced in detail. Specifically, we introduce the metaheuristics' advantages over other techniques. To obtain an entire view about MAs, different classifications based on different aspects (i.e., inspiration source, number of search agents, the updating mechanisms followed by search agents in updating their positions, and the number of primary parameters of the algorithms) are presented in detail, along with the optimization problems including both structure and different types. The application area occupies a lot of research, so in this study, the most widely used applications of MAs are presented. Finally, a great effort of this research is directed to discuss the different open issues and challenges of MAs, which help upcoming researchers to know the future directions of this active field. Overall, this study helps existing researchers understand the basic information of the metaheuristic field in addition to directing newcomers to the active areas and problems that need to be addressed in the future.

1 Introduction

As the world moves towards competition in all fields, people need to best use the limited resources to score a better result and thus achieve a better place in the competition. In this context, optimization is strongly needed. Optimization is a process of picking up the optimal values of the optimization problem's parameters from a given set of values to achieve the desired output, which specifically means output minimization or maximization. In other words, we need to obtain the best optimal solution under a set of limitations and constraints by tuning the parameters of the problem to be addressed. As mentioned in [1], the optimization process includes a set of steps which starts with formulating the problem to be in the form of an optimization problem, constructing the objectives (cost or fitness) function, determining the decision variables and the restrictions on these variables, then simplifying the reality of the problem by generating the mathematical model that represents the problem. Finally, the problem solver seeks to generate the most acceptable solution by maximizing or minimizing the value of the objective function.

Stochastic optimization algorithms are the most promising type under the umbrella of optimization, which can be classified as heuristic Algorithms (HAs) and metaheuristic Algorithms (MAs). In simple words, stochastic optimization is the general class of algorithms that depends on the random nature in the process of getting the optimal or near-optimal solution. HAs are iterative algorithms, iterating many times seeking a better solution than the solution obtained previously. HAs are used to find a feasible and reasonable

✉ Essam H. Houssein
essam.halim@mu.edu.eg

Mahmoud Khalaf Saeed
mahmoud.elsaeed@mu.edu.eg

Gang Hu
hugang@xaut.edu.cn

Mustafa M. Al-Sayed
mustafa.al-sayed@mu.edu.eg

1 Faculty of Computers and Information, Minia University, Minia, Egypt

2 Department of Applied Mathematics, Xi'an University of Technology, Xi'an 710054, China

3 School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China

solution but may not be the optimal one. In addition, HAs do not provide any evidence of the optimality of the solution obtained. A set of issues can be found in HAs, such as being problem-dependent algorithms specifically designed for a particular problem [2]. Another challenge in HAs is immeasurable success as there is no information about how close the obtained solution is to the optimal. Finally, there is a dilemma in measuring the computational time. Disclosing these gaffes and achieving a trade-off between solution quality and computational time is a main purpose of the appearance of metaheuristics [3]. As it is used to solve different types of problems, metaheuristics are the most preferable type of these algorithms. Metaheuristics were introduced for the first time by Glover in [4].

1.1 Understanding Optimization: What and Why?

In this section, we will help researchers understand the fascinating world of optimization. First, we will present examples of what optimization is, then we will try to answer the question, "Why do we study optimization?" Finally, this section will clarify how to optimize anything you have.

What is Optimization? In simple words, optimization is the art of perfectionism—how to perfectly make something in the best way. Optimization answers the question: how to obtain the best solution for a problem while applying a set of limitations? Maximizing profit, minimizing mass, pollution minimization, noise reduction, and drag reduction are all practical examples that can be achieved by using optimization. In most cases, optimization helps in the design process as a replacement for the traditional approach, which depends mainly on trials or humans. To clarify the simplicity and practical power of optimization in the design process, a diagrammatic view of how optimization methods help in the design process is presented in Fig. 1.

Why Optimization? People may ask why we study optimization. In most cases, we do not have the opportunity to physically perform trials; instead, we use optimization to simulate a solution for a specific problem to see what the result of this trial would be. Hence, we can decide whether this trial is applicable or not. People may benefit from applying optimization in the industry to achieve a better position in competition under limited resources.

1.2 Paper Structure

The rest of this paper is structured as follows: various metaheuristic taxonomies and the development process are illustrated in Sects. 2 and 3. Taxonomies of optimization problems based on many criteria and their performance assessment are introduced in Sects. 4 and 5. The applications of metaheuristic algorithms (MAs) in different fields are presented in Sect. 6. The open issues and challenges of MAs and the observations from the experiment are introduced in Sect. 7. Finally, the research review is concluded in Sect. 8. The outline of the article is illustrated in Fig. 2.

2 Metaheuristics Optimization Algorithms Taxonomies

Due to the rapid growth of the optimization field, many metaheuristic (MA) algorithms have been proposed recently. These algorithms need to be classified according to four main taxonomies: inspiration source, number of search agents, the mechanisms followed in the optimization process, and solution updating, in addition to the number of parameters included in the algorithm. In this section, these new algorithms will be classified.

2.1 Taxonomy According to the Inspiration Source

This is the most familiar and oldest classification of metaheuristic algorithms (MAs) and is suitable for studying the subcategory of MAs, which are nature-inspired metaheuristic algorithms. In general, by including the source of inspiration in the calculation, different studies use different classifications according to the inspiration, as illustrated

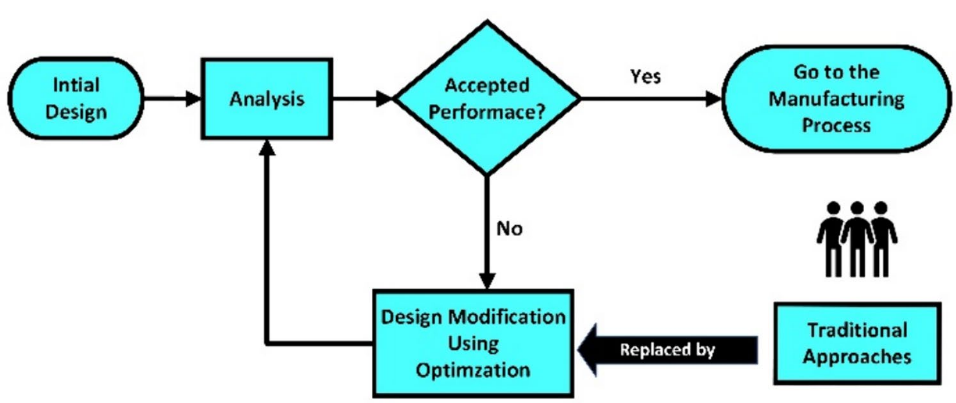


Fig. 1 The optimization methods as a replacement of traditional method in the design process

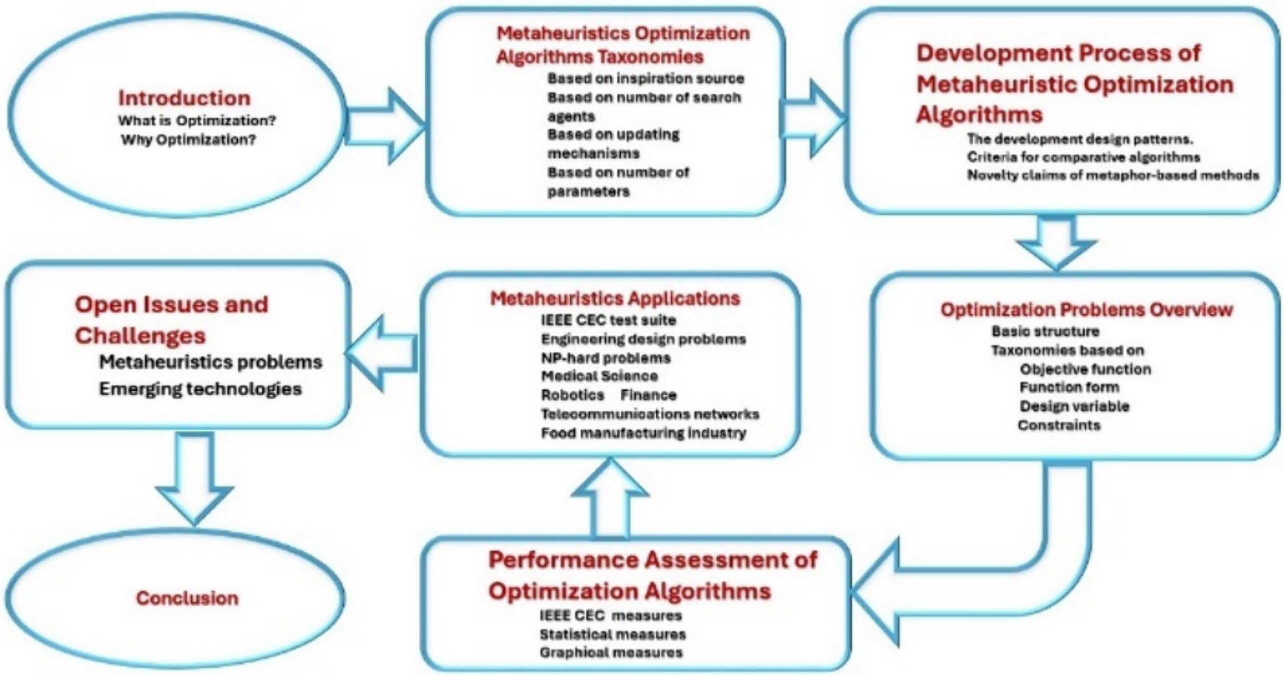


Fig. 2 The outline of the article

in Table 1. In this study, Fig. 3 shows a more comprehensive taxonomy for MAs.

As follows, the subcategories of the source of inspiration for MAs shown in Fig. 3 are illustrated in detail.

Swarm Intelligence (SI) is a self-organized system of collaborative behavior. SI has a set of characteristics, such as good communication skills between individuals, the ability to share information among its individuals, and the ability to learn from doing (adaptable beings). On the other hand, organisms do not have the ability to defend themselves against predators; they need to be in a swarm to perform the search or attack process for food. Mimicking the behavior of beings that live in flocks or herds seeking to hunt for prey or find food is the main inspiration for SI algorithms [8]. One of the most famous algorithms in this category is Particle Swarm Optimization (PSO) [9], which is inspired by mimicking the intelligent behavior of a flock of birds. Monkey Search Optimization (MSO) [10] is another example of SI algorithms that simulate the tree climbing process during the food discovery process. Hunting strategy and hierarchy-based leadership are the inspiration for Grey Wolf Optimizer (GWO) [11], Ant Colony Optimization (ACO) [12], Cuckoo Search (CS) [13], Ant Lion Optimizer (ALO) [14], and Honey Badger

Table 1 Different trials of classifying the MAs according to inspiration source

	Optimization algorithms categories	Authors
Metaheuristics optimization algorithms (MAs)	Swarm intelligence (SI) based algorithms Bio-inspired (not SI) based algorithms Physics-chemistry based algorithms Another algorithm	Fister Jr et al. [5]
	Physics-based algorithms Chemistry-based algorithms Biology-based algorithms	Siddique and Adeli [6]
	Breeding-based evolutionary algorithms Swarm intelligence-based algorithms Physics-chemistry-based algorithms Human social behavior-based algorithms Plant-based algorithms Miscellaneous	Molina et al. [7]

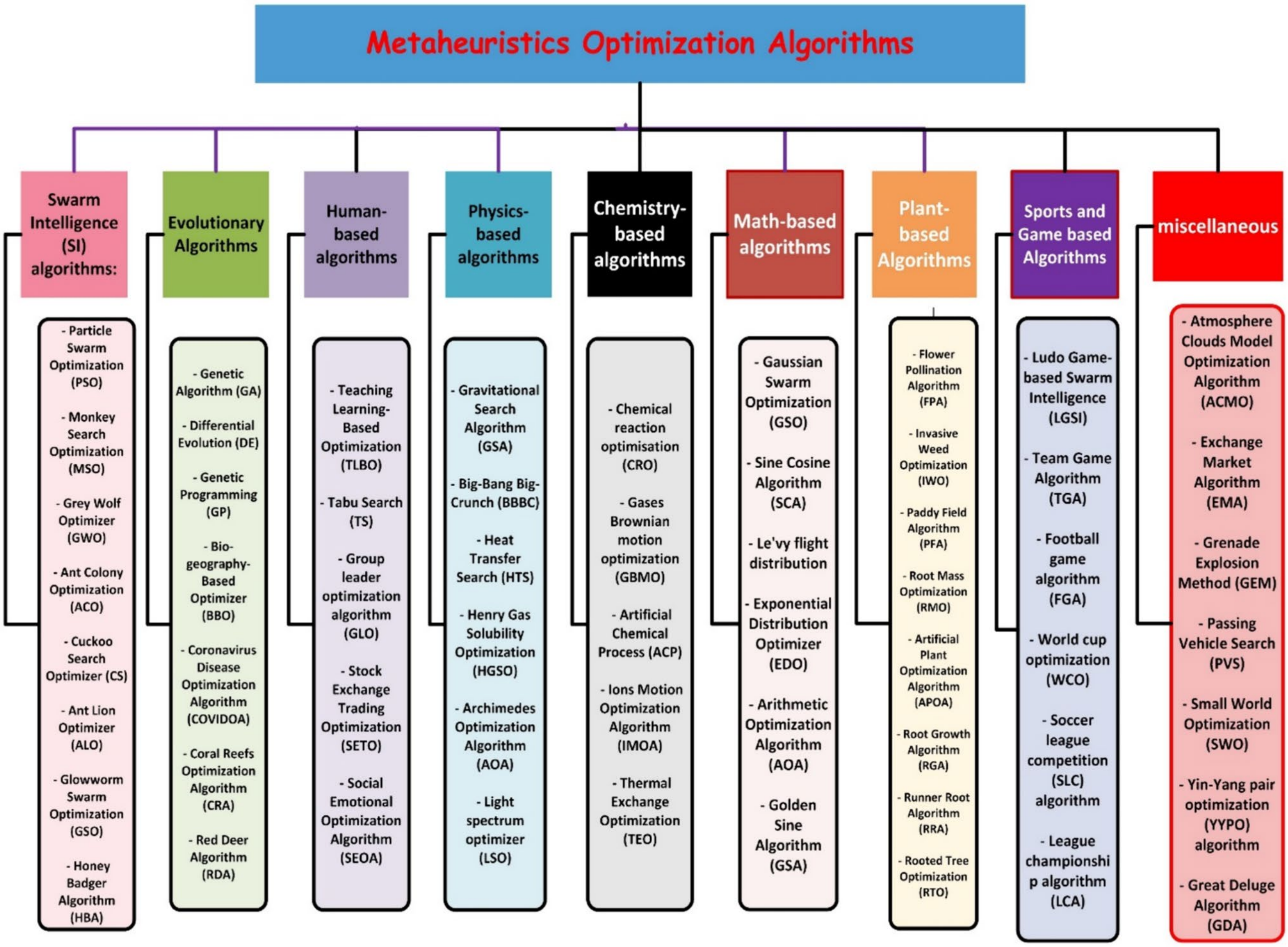


Fig. 3 The proposed classification of MAs based on the source of inspiration

Algorithm (HBA) [15], which are well-known instances of SI algorithms.

Evolutionary Algorithms (EA) simulate the behavior of evolution, including recombination, mutation, crossover, and selection. EA begins by generating a random population; this population is then evaluated to choose the most fit individuals to contribute to the next generation. After several iterations, the population evolves to find the optimal solution. Genetic Algorithm (GA) [16] is the oldest algorithm in this class, mimicking Charles Darwin's theory of natural evolution. Other well-known EA methods include Differential Evolution (DE) [17], Genetic Programming (GP) [18], Coronavirus Disease Optimization Algorithm (COVIDOA) [19], Liver Cancer Algorithm [20], and Red Deer Algorithm (RDA) [21].

Human-based Algorithms (HA) is the main inspiration for this category. Mimicking the learning process between teachers and students led to the introduction of Teaching Learning-Based Optimization (TLBO) [22]. Tabu Search (TS) [23] enhances the search process through long and short memory. Other well-known HA algorithms include Group Leader Optimization Algorithm (GLO) [24], Stock Exchange Trading Optimization (SETO) [25], and Social Emotional Optimization Algorithm (SEOA) [26].

Physics-based Algorithms (PhA) are inspired by the physical laws or simulating a physical phenomenon such as gravitation, Big Bang, black hole, galaxy, and field. In other words, the physical rules are used in the process of generating new solutions. The most popular instances of this class are the Gravitational Search Algorithm (GSA) [27], Big Bang Big Chain (BBBC) [28], Heat Transfer Search [29], Henry Gas Solubility Optimization (HGSO) [30], Archimedes Optimization Algorithm [31], and Light Spectrum Optimizer (LSO) [32], which are some of the most common algorithms in the PhA category.

Chemistry based algorithms (ChAs) are algorithms that concentrate on the principle of chemical reactions such as molecular reaction, Brownian motion, molecular radiation. A list of algorithms that fall into this category are Chemical Reaction Optimization (CRO) [33], Gases

Brownian Motion Optimization (GBMO) [34], Artificial Chemical Process (ACP) [35], Ions Motion Optimization Algorithm (IMOA) [36], and Thermal Exchange Optimization (TEO) [37], are common instances of the ChA category.

Math-based algorithms (MathA) Math-based optimization algorithms are algorithms that can be inspired from the mathematical theorems, concept and rules. Some algorithms fall into this group including; Gaussian Swarm Optimization (GSO) [38], Sine Cosine Algorithm (SCA) [39], Lévy flight distribution [40], Exponential Distribution Optimizer (EDO) [41], and Golden Sine Algorithm (GSA) [42], are common instances of the MathA category.

Plant-based Algorithms (PlA) The PLAs is relays on the simulation of the intelligent behavior of the plants. Specifically, a set of concepts in plant nature is used to inspire new metaheuristic optimization algorithms such as the flower flow pollination process, the phenomenon of colonization of invasive weeds in nature, the ecology and weed biology. Some algorithms fall into this group including; Flower Pollination Algorithm (FPA) [43], Invasive Weed Optimization (IWO) [44], Paddy Field Algorithm (PFA) [45], Artificial Plant Optimization Algorithm (APOA) [46], Plant Growth Optimization (PGO) [47], Root Growth Algorithm (RGA) [48], Rooted Tree Optimization (RTO) [49] are common instances of the PlA category.

Sports and Game based Algorithms (SpGA) Depending in the information and rules applied in the sports and gaming, a set of optimization algorithms can be inspired from team game strategies used in football, Basketball, and volleyball, Ludo Game. Ludo Game-Based Swarm Intelligence (LGSI) [50], Team Game Algorithm (TGA) [51], Football game algorithm (FGA) [52], World Cup Optimization (WCO) [53], Soccer League Competition (SLC) algorithm [54], and League championship algorithm (LCA) [55] are common instances of SpGA algorithms.

Miscellaneous The rest of metaheuristics optimization algorithms can be collected to be belongs to the miscellaneous class, the purpose of using the term miscellaneous is the miscellaneous ideas such as politics, Artificial thoughts, atmosphere, trade and other topics. Work occurring in clouds such as cloud movement, spread, and creation is the basic idea behind the inspiration of the Atmosphere Cloud Model Optimization Algorithm (ACMO) [56], the exchange of information in the stock market occurs, and is the basic motivation behind the Exchange Market Algorithm (EMA) [57]. The Grenade Explosion Method (GEM) [58], Passing Vehicle Search (PVS) [59], Small World Optimization (SWO) [60], Yin-Yang Pair Optimization (YYPO) algorithm [61], Political Optimizer (PO) [62], and the Great Deluge Algorithm (GDA) [63] are other examples of this category.

2.2 Taxonomy According to the Number of Search Agents

The classification according to the source of inspiration is the most familiar and is usually introduced in studies to summarize the concept of classification. However, this classification is not enough to tackle the classification process, as it does not provide any information about the internal mathematical structure or programming ideas of the algorithms. Hence a new angle of classification is used. Metaheuristics can be categorized based on the number of search agents seeking to find the optimal into two groups of single-solution-based MAs (SMAs), and population-based MAs (PMAs). The following two paragraphs provide more information about each group. Figure 4 is a clarification view of this taxonomy.

Single-solution based MAs (SMAs) SMAs is also called Trajectory-based algorithms (TAs) as the algorithms in this class depends on single trajectory nature in its work. In other words, in each iteration, the solution is directed to a single trajectory. The optimization procedure (searching about the optimal solution) of SMAs is started with single solution (from one search agent), later, and in the subsequent iterations, the solution is refined with the aim of achieving the optimal solution. We can say that the algorithm generates a single path to the optimal solution over the course of the iteration. For SMAs, the Simulated Annealing (SA) [64] is one of the familiar algorithms. where a single search agent moves through the design or search space of the problem being tackled. Over the course of iteration, a better solution or moves is accepted to participate in determining the optimal solution while the weak movements and solution are more likely to participate in the optimization process. Applying these actions guarantee generating an optimal path through the search space with a great probability of achieving a global optimal solution. Hill climbing (HC) reviewed in [65], Tabu Search (TS) [23], Great Deluge Algorithm (GDA) [63], Iterated Local Search (ILS) [66], and Greedy Randomized Adaptive Search Procedures (GRASP) [67] are some instances of this class.

Population-based MAs (PMAs) In contrast, and taking advantage of sharing information among agents, Collaborative work and data remembering, the PMAs is introduced. First, we can say that more than one agent is superior to a single agent in achieving the optimal solution. Specifically, a great number of search agents work together to extensively explore the search space, so we can call PMAs explorative-based algorithms. The optimization procedure starts with employing a population of search agents positioned at many distinct positions in the search space, and over the course of iterations, the population uses the advantage of sharing information to better achieve the best global solution. In simple words, a set of lines is drawn in the search space to

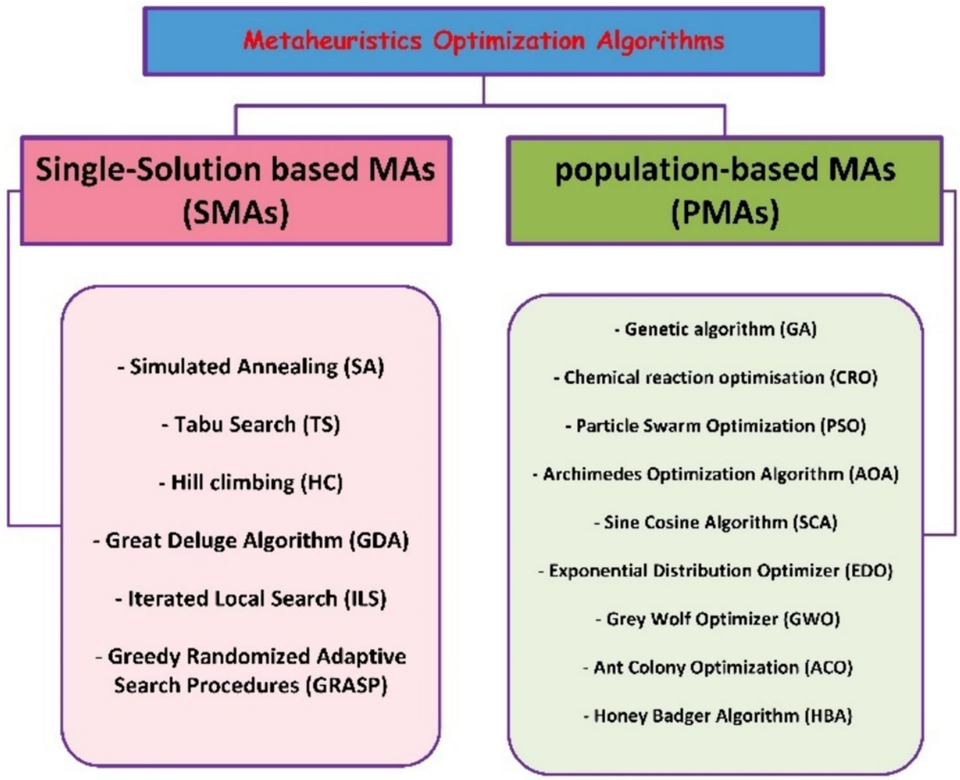


Fig. 4 The classification of MAs based on the number of search agents

extensively search the search space in order to obtain the best optimal solution achieved by all search agents. One of the oldest and widely used algorithms in PMAs is the Genetic Algorithm (GA), Chemical reaction optimization (CRO), Particle Swarm Optimization (PSO), Archimedes Optimization Algorithm (AOA), Sine Cosine Algorithm (SCA), Exponential Distribution Optimizer (EDO), Grey Wolf Optimizer (GWO), Ant Colony Optimization (ACO) and Honey Badger Algorithm (HBA) are some instances from this category.

In general, no class is totally better than the other where PMAs escape from the local optima dilemma in contrast to SMAs, also SMAs consume less computational time than PMAs, for a *Itr* number of iterations, the SMAs perform a lower number of objective function evaluation which equals $1 \times Itr$ while the PMAs perform $N \times Itr$ evaluation of the objective function. N here stands for the number of search agents employed by the algorithm to obtain the optimal solution. But overall, the scientists prefer to use the PMAs as it has a greater probability of achieving global optimal solution in a considerable amount of time.

2.3 Taxonomy According to Updating Mechanisms

However, the classification according to the number of search agents provides information about the internal structure of the algorithm, but it cannot be treated as a uniform classification due to the few algorithms belonging to one group while the remainder (majority) falls under the other group. In this context, we need to provide a different classification angle to achieve an acceptable degree of uniform classification. According to the most important step of any algorithm, which is the solution update process. From this prospective MAs can be classified as solution creation-based algorithms (SCBAs) and differential vector movement-based algorithms (DVMs) [68]. In the following paragraph, we introduce a simple classification based on the behavior of the algorithms. Figure 5 is a clarification view of this taxonomy.

Solution Creation Based Algorithms (SCs) In SCs, A set of parent solutions are merged to generate the new solution, in other words no single solution is used to create the fresh solution. Furthermore, the SCs can be categorized into two subcategories which are combination-based algorithms and stigmergy-based algorithms. In combination-based algorithms several solutions are combined or crossover-ed. Genetic Algorithm (GA), Gene Expression (GE), Harmony Search (HS), Bee Colony Optimization (BCO), Cuckoo Search (CS), Dolphin Search (DS) are some examples of this subcategory. On the other hand, in strategy-based solutions different solutions are indirectly coordinated by intermediate structure to generate new solutions. Ant Colony Optimization (ACO), Termite Hill Algorithm (THA), River Formation Dynamics (RFD), Intelligence Water Drops Algorithm (IWDA),

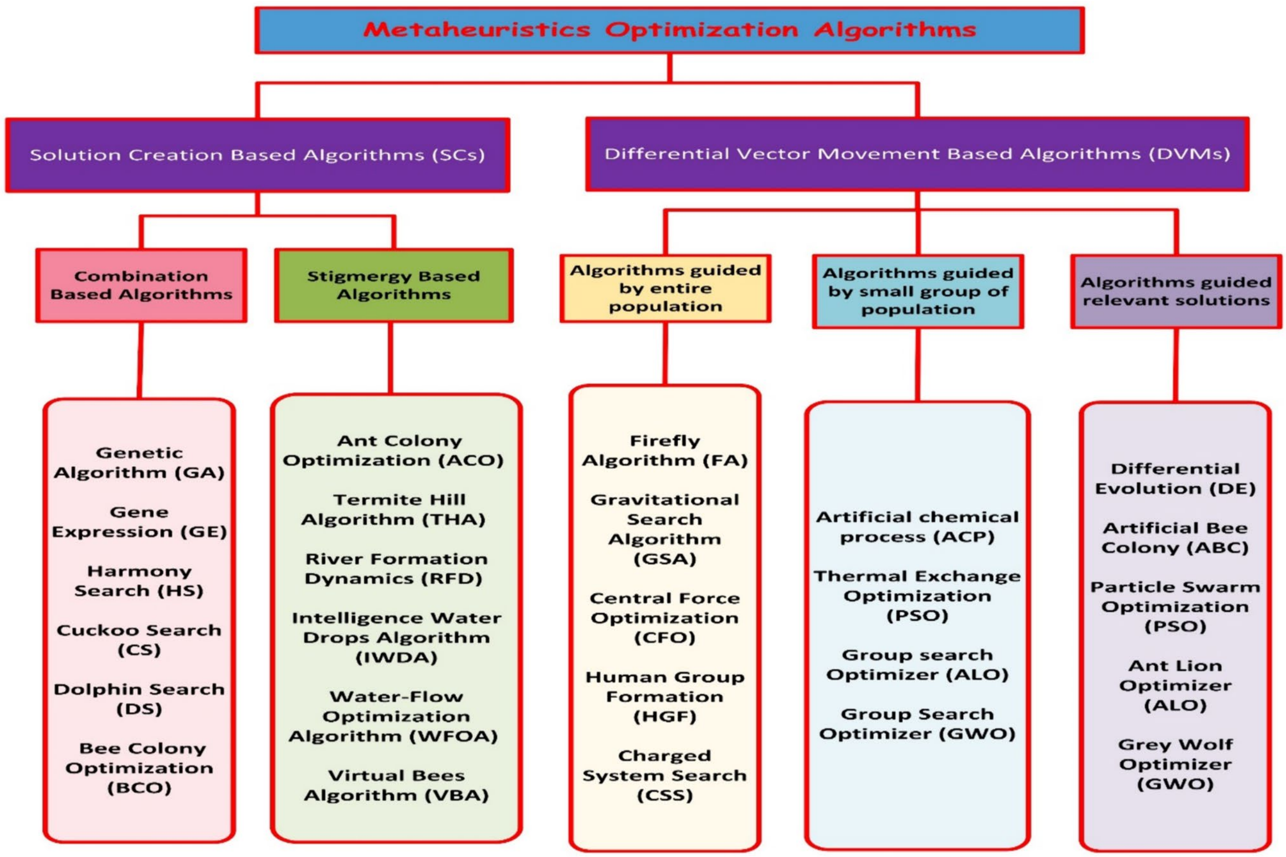


Fig. 5 The classification of MAs based on population update mechanisms

Water-Flow Optimization Algorithm (WFOA), and Virtual Bees Algorithm (VBA) are some examples of the second subcategory.

Differential Vector Movement Based Algorithms (DVMs) Applying the mutation or shifting operation on the algorithm in order to generate a new solution is called Differential Vector Movement method. The fresh generated solution needs to be fitted to the previous one to participate in the next iteration of the optimization procedure. In this context, DVMs is categorized into three subcategories. In the first subcategory, the whole population's solution is used to generate the new solution, such operation occurs in Firefly Algorithm (FA) Gravitational Search Algorithm (GSA), Central Force Optimization (CFO), Human Group Formation (HGF) and Charged System Search (CSS). In the second sub-category, a small number of solutions (neighbourhoods) in population is employed to generate a new solution such as Artificial chemical process (ACP), Thermal Exchange Optimization (PSO), Group Search Optimizer (ALO), and Group Search Optimizer (GWO). In the last sub-category, only the relevant (best/worst) solutions are employed to generate the new solution such as Differential Evolution (DE), Artificial Bee Colony (ABC), Particle Swarm Optimization (PSO), Ant Lion Optimizer (ALO), and Grey Wolf Optimizer (GWO).

2.4 Taxonomy According to Number of Parameters

To deeply consider the internal configuration of the algorithm for this type of classification. Tuning the parameter of the algorithm plays a vital role in the performance of the algorithm when solving a specific problem. As mentioned in [1], it is a complicated task to choose the best values of the parameter that scores a better solution. Furthermore, the parameters can enhance the robustness and flexibility of the MAs if they are adjusted correctly. The optimization problem plays a vital role in defining the values of parameters. From a complexity perspective, the complexity of an algorithm is affected by the number of parameters. In this context and taking into account the importance of the parameters, this classification is introduced. Kanchan Rajwar et al. in [68] first classify the MAs according to the number of primary parameters employed in the MAs as illustrated in Fig. 6.

The number of parameters changes from one algorithm to another, which can be ***0, 1, 3, 4,*** etc. For simplicity we

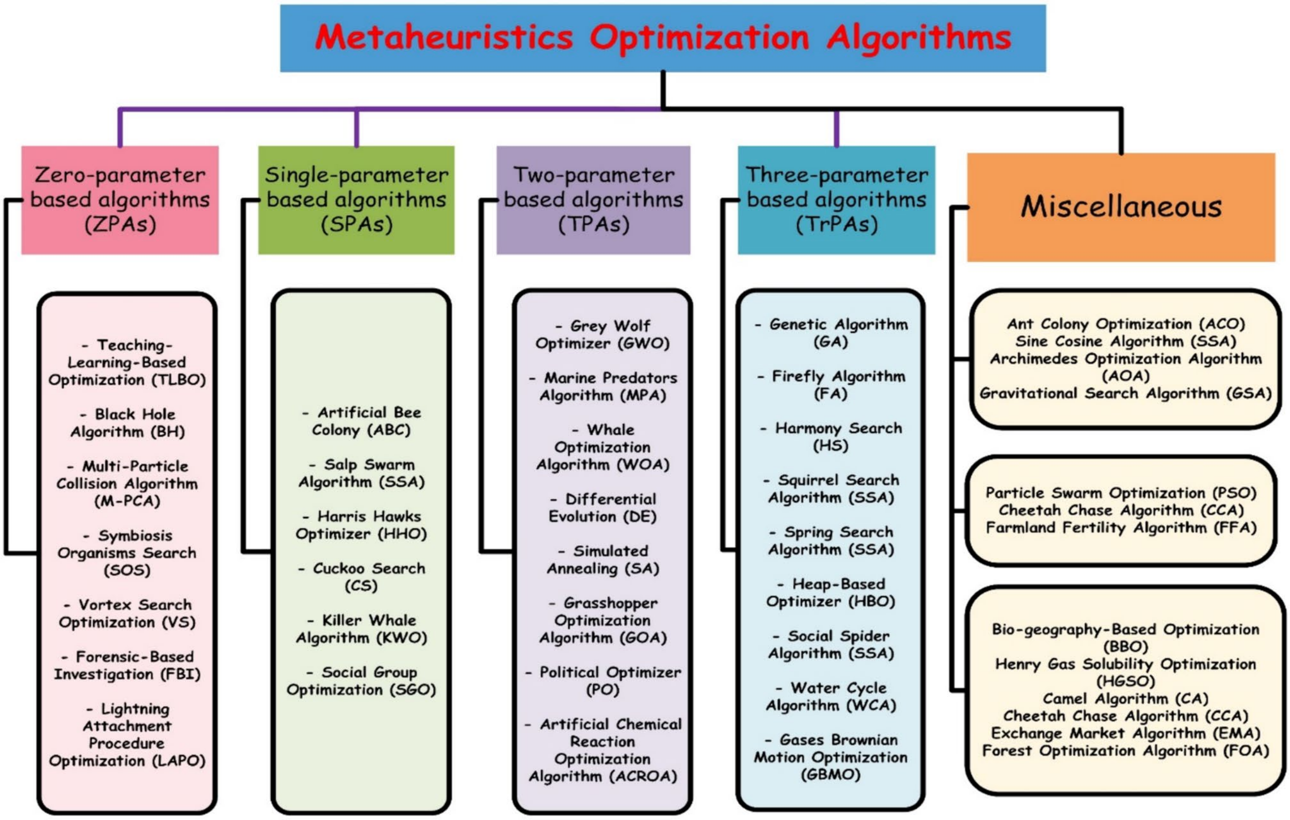


Fig. 6 The classification of MAs according to the number of primary parameters

will consider four main groups holding algorithm parameter numbers up to ***3*** and the rest fall into the miscellaneous group. The following paragraphs provide a detailed explanation of the five groups in this classification.

Zero-parameter-based algorithms (ZPAs): The ZPAs do not have any parameter in their internal configuration so it also called Free-parameter-based algorithms. The absence of parameters in ZPAs gives the user the opportunity to easily adapt the algorithm to be utilized in different optimization problems. Hence, the algorithms belong to this group considered as flexible, adaptive, and easy-to-use algorithms. Teaching–Learning-Based Optimization (TLBO) [22], Black Hole Algorithm (BH) [69], Multi-Particle Collision Algorithm (M-PCA) [70], Symbiosis Organisms Search (SOS) [71], Vortex Search Optimization (VS) [72], Forensic-Based Investigation (FBI) [73], and Lightning Attachment Procedure Optimization (LAPO) [74] are some examples of ZPAs.

Single-parameter-based algorithms (SPAs): SPAs is the type of algorithms that own a single primary parameters in their internal configuration. So, it also is called monoparameter-based algorithms. Mostly, this single parameter has the ability to change the amount of exploration and exploitation that occurred in the algorithm. For example, in the Artificial Bee Colony (ABC) algorithm the single parameter ***Limit*** is used to determine the amount of food source left [75], in the Salp Swarm Algorithm (SSA) ***c1*** is the parameter used to achieve a better balance between explorative and exploitative capabilities [76], and in Harris Hawks Optimizer (HHO) [77] the switch between soft and hard besiege is achieved by the magnitude value parameter ***E***. Cuckoo Search (CS), Killer Whale Algorithm (KWO), and Social Group Optimization (SGO) are another example of this group.

Two-parameter-based algorithms (TPAs): In TPAs only two primary parameters exist in the internal structure of the algorithm. For example, in the Grey Wolf Optimizer (GWO), the two primary parameters ***a*** and ***c*** must be adjusted. The a is adjusted to be equal to **2** to **0**, allowing the algorithm to perform a smooth transition from exploration and exploitation while the ***c*** parameter is used to allow the algorithm to reach distinct locations around the optimal agent relative to the current location, In the Marine Predators Algorithm (MPA), ***P*** and ***FADs*** are the two primary control parameters. To overstate the predator or prey move, ***P*** is adjusted, while ***FADs*** is used to manage exploration behavior. Finally, in the Whale Optimization Algorithm (WOA) the two primary parameters ***A*** and ***C*** need to be modified to perform the exploration-to-exploitation transition and to allow the algorithm to explore several positions around the optimal agent relative to the present location. Differential Evolution (DE), Simulated Annealing (SA), Grasshopper Optimization Algorithm (GOA), Political Optimizer (PO), and Artificial Chemical Reaction Optimization Algorithm (ACROA) are just a few instances of TPAs.

Three-parameter-based algorithms (TrPAs): In TPAs only three primary parameters exist in the internal structure of the algorithm. For example, the mutation rate ***mr***, the crossover rate ***cr***, and the new population selection criterion are the three parameters used in the Genetic Algorithm (GA) to allow the algorithm to escape from the local optima, improve the accuracy of the solution, and generate a most fit new generation, respectively. The randomization, attractiveness, and absorption are the three parameters included in the Firefly Algorithm (FA) to manage the execution of the algorithm and the random walks of fireflies. Finally, the distance bandwidth (BW), the harmony memory considering rate (HMCR), and the pitch adjusting rate (PAR) are the three primary parameters used in Harmony Search (HS) to increase the opportunity of achieving a global search and improve the local search problem. Squirrel Search Algorithm (SSA), Krill Herd (KH), Spring Search Algorithm (SSA), Artificial Algae Algorithm (AAA), Gases Brownian Motion Optimization (GBMO), Hurricane-Based Optimization Algorithm (HOA), Orca Optimization Algorithm (OOA), Social Spider Algorithm (SSA), Water Cycle Algorithm (WCA), Equilibrium Optimizer (EO), Parasitism Predation Algorithm (PPA), and Heap-Based Optimizer (HBO) are few instances of this group.

Miscellaneous: The rest of algorithms that own over three parameters in their internal configuration fall under the category of the miscellaneous group. It is not easy to cover all three-parameter algorithms. so, only three subgroups are introduced. the first subgroup is the four parameter-based algorithms such as Ant Colony Optimization (ACO), Sine Cosine Algorithm (SSA), Archimedes Optimization Algorithm (AOA), and Gravitational Search Algorithm (GSA). The second subgroup holds algorithms that employed five primary parameters in their internal structure such as Particle Swarm Optimization (PSO), Cheetah Chase Algorithm (CCA) and Farmland Fertility Algorithm (FFA). The last subgroup is algorithms with more than five primary parameters in their internal configuration. Biogeography-Based Optimization (BBO) with ***six*** parameters, Henry Gas Solubility Optimization (HGSO) with ***twelve*** primary parameters and the Camel Algorithm (CA) with ***seven***} primary parameters are the most familiar algorithms in this subgroup. Cheetah Chase Algorithm (CCA), Exchange Market Algorithm (EMA), and Forest Optimization Algorithm (FOA) are also instances of this subgroup.

In general algorithms with few parameter-based MAs are easy to be adapted and hence the applicability of these algorithms to handle any optimization problem will increase and, on the other hand, large parameter-based MAs cause a disability of these algorithms to handle the optimization problems, as we encounter a problem in adapting all of their parameters to be suited for problem being tackled. hence the applicability will be decreed.

2.5 Metaheuristic Algorithms Merits

The MAs have a priority to be studied by the researcher than HAs, as they have four characteristics [78], which can be summarized as follows.

Metaheuristics simplicity It is painless to inspire a MAs as we can use a natural concept, physical phenomena or an animal behavior in the inspiration process. Utilizing the merit of simplicity, the researchers Seize the opportunity to make an extension in the metaheuristics works as they develop a new method by mimicking a natural idea, use the ability of search enhancement techniques to boost the performance of an existing algorithm, or even take the advantages in two metaheuristics algorithms and generate a new metaheuristics algorithm by applying a hybridization process. Furthermore, simplicity encourages computer scientists and other researchers to easily study the existing MAs and then apply them to solve a wide range of problems.

Metaheuristics flexibility In the other techniques there is a need to modify the structure of the algorithm to be matched with the problem being solved, unlike these techniques metaheuristics flexibility virtue allows the researchers to easily apply the MAs on any problem as the MAs have the capability of treating the problem as black box, in other words it need the input(s), output(s) of a problem on hand. No effort is used in modifying the structure; all effort is directed towards formulating the problem being solved in the form of an optimization problem.

Metaheuristics stochastic nature Computing the derivation of the search space of the problem is a necessity for the gradient-based optimization techniques to achieve an optimal solution. Dissimilar to these techniques, the preponderance of MAs is considered as a derivative-free mechanism when applying the process of optimization, specifically the MAs follow a stochastic nature during the search process as they start the optimization process by employing a set of search agents to generate random solutions without computing the derivative of the search space. The collaborative work of these search agents allows the algorithm to get the optimal solution. This merit allows researchers to easily use the MAs algorithms to perfectly tackle compound, expensive, and difficult problems that suffer from the trouble of obtaining the derivative information.

According to the previous features, the research community has increased, and researchers from different fields and application areas have been using the metaheuristic optimization algorithm in their work. About 4,476 documents founded in the Scopus have used the word metaheuristics in the last decade. Figure 7a is introduced to visualize the distribution of research studies according to the subject area, while Fig. 7b is used to depict the number of studies generated in each year of the previous decade.

3 Development Process of Metaheuristic Optimization Algorithms

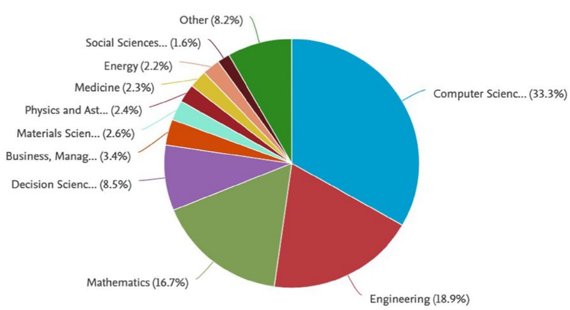
The simplicity merit of MAs allows researchers to easily develop a large number of algorithms in different application areas. To develop a new metaheuristic algorithm, a researcher can follow one of the following development processes according to the type of algorithm that is being developed, and some processes can also be used together.

Develop a new optimization algorithm The most of work for developing an optimization algorithm done by inspire the main idea of the algorithm from a different metaphors or concepts. These metaphors or concepts are mainly a simulation of rules or processes in different disciplines such as Chemistry, Physics, Biology, Psychology, Computation, Maths, and Human. Figure 3 is used to visualize a different source of inspiration with examples in each category. In general, most metaheuristics have been designed to mimic the system of living and survival of beings such as animals, birds, and insects, in addition to mimicking natural evolution. Insects (specifically, bees and ants) are the most popular metaphor for the development of a new optimization method by researchers.

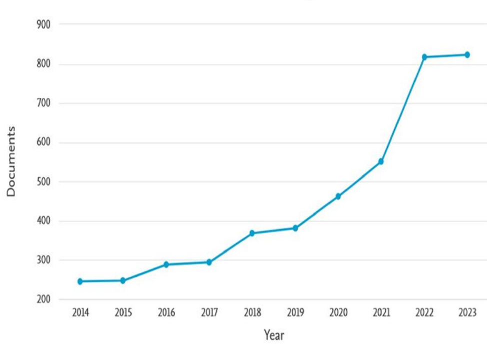
Develop a new optimization algorithm from existing one One of the most popular ways to develop a new optimization method is to benefit from the operators of a specific algorithm in enhancing the structure of another algorithm. In simple words, the operators of other algorithms can emerge into the basic structures of the algorithm to boost the performance of the previously developed algorithm, and hence use it in solving different types of problems and issues. There are many enhancement operators used in the field; one of the most used ones is opposition-based learning (OBL). OBL is a machine learning mechanism that is used to increase the performance of the optimization algorithm by considering the opposite position of the solution in the search space. Specifically, two values are computed, the main and opposite positions, according to the objective function value, one of the two values maintained in the optimization process, and the other discarded. Taking into account only the best values, the optimization process became more accurate and a high level of performance is achieved. The orthogonal learning (OL) strategy is another example of an operator used as an enhancement strategy for MAs. The OL strategy mainly improves the exploitation capabilities. For example, the OL strategy was used to improve the Archimedes optimization algorithm, the cuckoo search algorithm and the artificial bee colony optimization algorithm, respectively. Enhanced solution quality (ESQ) is another mechanism used in the MA enhancement process. The ESQ was used to improve the performance of the reptile search algorithm and the Harris Hawks optimization (HHO) algorithm, respectively. Finally, the Local Escaping Operator (LEO) is used to develop an optimized version of the MPA called the enhanced marine predator algorithm (EMPA).

Hybridizing two or more optimization algorithm As a trial for enhancing the performance and applicability of the optimization method, researchers can benefit from hybridizing two or more optimization algorithms together in order to take the main strengths of each algorithm. The idea behind hybridization is to choose one algorithm better in exploration capabilities and another better in exploitation capabilities. Many challenges are encountered when we develop a new algorithm using the hybridization process, such as how to select the algorithm and how to merge them together, and is the new algorithm better than each one separately?

As shown in the previous paragraphs, there is a different development process for developing a new optimization method, although there is a set of limitations that must be considered during the development process such as the



(a) Metaheuristics documents in the last 10 years according to the subject area.



(b) Metaheuristics documents published in the last 10 years.

Fig. 7 Scopus statistics from 2014 to 2023 difficulty of transforming all the concepts with details into a mathematical form, how the algorithm totally manages the change in information about the source of inspiration, in addition to how people with low familiarity with the inspiration sources develop new methods.

3.1 Criteria for Comparative Algorithms

To gauge the effectiveness of newly developed algorithms, it is crucial for research to present the process of comparing them with existing algorithms. This should include a discussion of the selection criteria for comparative algorithms and the methodology used for comparison. The selection criteria for comparative algorithms depend mainly on the nature of the algorithm and the development process followed in developing the algorithm. In all cases, comparative algorithms should contain common criteria, which are state-of-the-art algorithms, newly developed algorithms, CEC winner algorithms, and high-performance algorithms. Specifically in case of developing the algorithm using the inspiration of a phenomenon process, the comparative algorithms list must contain algorithms with the same inspiration source or concept if there exist in addition to the common criteria algorithms. In case of developing an algorithm using the restructure method (i.e., merging a new operator or strategy), the comparative algorithms must contain the basic algorithm, algorithms developed using the same strategy if exists, algorithms that contain the strategy itself, in addition to the common criteria algorithms. In the case of developing algorithms using the hybridization process, the comparative algorithm list must contain the two basic algorithms that participate in the hybridization process, in addition to the common criteria algorithms.

3.2 Novelty Claims of Metaphor-Based Methods

The different ways of developing an optimization algorithm and the simplicity merit of the metaheuristic allow researchers to easily develop a large number of MAs. But a question must be asked here: Does this inspiration convey a novelty? In this section, we will present a set of claims and myths in the inspiration process of the metaheuristic optimization algorithms. As introduced in [79] a six widely used algorithms have been analyzed to prove that all components of the six (grey wolf, moth-flame, whale, firefly, bat and ant lion) are equivalent to a component of well-known techniques such as evolutionary algorithms and particle swarm optimization. Hence the authors called these algorithms misleading or tricky optimization algorithms, as they were inspired by bestial or duplicated metaphors and did not bring any novelty or useful principles in the metaheuristics field. We will present what considerations must be taken when developing a new novel algorithm and how to judge about the novelty of the new proposed algorithm in the field of metaheuristics.

Recently, a large number of publications have developed self-proclaimed or novel metaphor-based methods, but it is not obvious why they used them and what the novelty ideas are behind these methods. The set of all negative points, criticizes about novelty claims of various metaphor-based methods, can be introduced in the following points:

- The metaphor-based methods redefine a well-known concept in the field of optimization and deliver it as a new concept or under new terminologies.
- weak translation of the metaphors into a mathematical model or equations, and the model cannot be used totally to reflect the metaphors correctly. Finally, the proposed algorithm does not translate the mathematical model obtained from the metaphor correctly.
- There is a myth in introducing the motivations behind the use of metaphor where instead of delivering the motivations as a sound or scientific basis they use accurate motivations such as a new metaphor "has never been used before" or a new mathematical model "has never been appeared in the past". Additionally, there is no concentration on the optimization process itself and how this process is employed to introduce effective design choices.
- Instead of applying the evaluations of the proposed algorithms mainly on the state-of-the-art problems, the authors of these methods depend on the comparison with other algorithms or experimental analysis of low complexity problems in evaluating the performance or applicability of the proposed algorithm.

To prevent these negative points, the authors must apply two metrics analyses of the proposed algorithm before naming it as a "novel", which are:

- Usefulness: in which the author must clearly introduce what are the useful ideas that come from the metaphor and how this metaphor helps in solving the optimization problems.
- Novelty: When proposing a new method in the field of metaheuristics, was this new metaphor novel used to convey ideas?

4 Optimization Problems Overview

Achieving an acceptable solution is the main goal of any algorithm. Due to the rapid expansion of the complexity of the problem, scientists need to develop new methods that can cover this rapid extension. In this context, scientists are working to formulate any problem as an optimization problem to be easily tackled by optimization algorithms, as they provide better solutions than other traditional methods. In different fields, a great number of problems are formulated as an optimization problem, such as genetic algorithms used to automatically find and classify solitary lung nodules \cite{de2014automatic}, perform a classification for web pages, mining the web content, and dynamic organizing of the web content by ant colony optimization [80], In [81] Hussein et al., use the HHO to discover and design the drug through chemical descriptor selection and chemical compound activities. Applying HHO in microchannel heat sinks to minimize entropy generation [82], COVID-19 prediction [83], finally applying image segmentation and thresholding in [84, 85].

4.1 Basic Structure of Optimization Problems

In this section, we will try to support the readers who may not be familiar with optimization methods with the basic definitions and terms related to the optimization field. The process of solving an optimization problem using a metaheuristic algorithm starts with identifying the real-world problem, after that we move to the problem description stage in which we define the characteristics of the problem, determining the functional requirement in addition to analysis of nonfunctional requirements. After completing the problem description stage, we move to the research stage, in which the researcher first concentrates on how to mathematically formulate the problem in a mathematical form. To formulate the problem, we need to determine the design variables and parameters, formulating the objective function, determining the basic constraints on the variables, analyzing the complexity of the problem, and finally justifying the use of a metaheuristic algorithm. In the following paragraphs, the three main components which exist in any optimization problem are the objective function, the decision variables, and a set of constraints on these variables are discussed in detail.

Optimization model: Every system can be considered as a set of inputs producing one or more input, The system uses the set of constraints to minimize the number of inputs, in other words, we consider only the inputs that obey the constraint (i.e., valid inputs) and discard the other which does not match with the constraints (i.e., invalid inputs). The system performs processing on the valid inputs to produce the optimal solution, which can be evaluated using the objective function to obtain the minimum or maximum output value. In fact, the optimization algorithm will not find the optimal values of constraints; instead, it uses the constraints to produce the optimal solutions and construct the feasible solution area. The feasible solution area can be considered as the area which contains an infinite number of feasible solutions and one or more can be classified as the optimal one.

Formulating the optimization model as an optimization problem: When we solve the problem using the optimization algorithm, we look for all possible combinations of inputs. For example, if we have 3 inputs each with 10 discrete values, then we get 1000 combinations of inputs. The initial test to solve and evaluate the input is to use brute-force techniques. The brute force techniques will do better to obtain the optimal solution, but what about the large sized problems. Certainly, we will find a big problem in handling these problems using the brute-force techniques; hence searching all possible combinations for most real-world problems is impossible.

To avoid confusion for non-familiar people with the area of optimization, in this study, we will introduce the basic and most frequent terminologies used in the field:

- The search space: it is the area in which all possible combinations of inputs are located.
- The search landscape: it is the set of all possible combinations of inputs with their corresponding objective values.
- Decision variables: it is the unknown quantities that need to be determined by assigning values to them. It is also known as the design variables. All possible values that can be assigned to these variables are named variable scope or domain. It can be mathematically as $\boldsymbol{X_i}$ where $\boldsymbol{i=1,2,3\ldots N}$.
- The objective function: This is the equation of the decision variables. In which all the decision variables exist with different parameters. It is used to judge the quality of the solution obtained for the problem being handled. In other words, after calculating the values of the decision variables, we substitute them in the objective function to obtain the objective value. The minimum objective value is the optimal solution for minimization problems, and the maximum is the optimal solution for the maximization problem.

Mathematically the single objective optimization problem can be formulated as Eq. (1) while the Multi objective optimization problem can be formulated as Eq. (2).

The optimization problems can be categorized in different ways. Categorizing optimization problems is an important step in choosing the algorithm that provides the optimal solution. It is not easy to introduce a rigorous or comprehensive taxonomy for optimization problems. This is due to the multiplicity of the classification term. But due to the important role of this taxonomy, in this paper we present a simplified and summarized version of the available taxonomies, illustrated in Fig. 8. In the following subsections, the different subcategories of the optimization problem are discussed in detail.

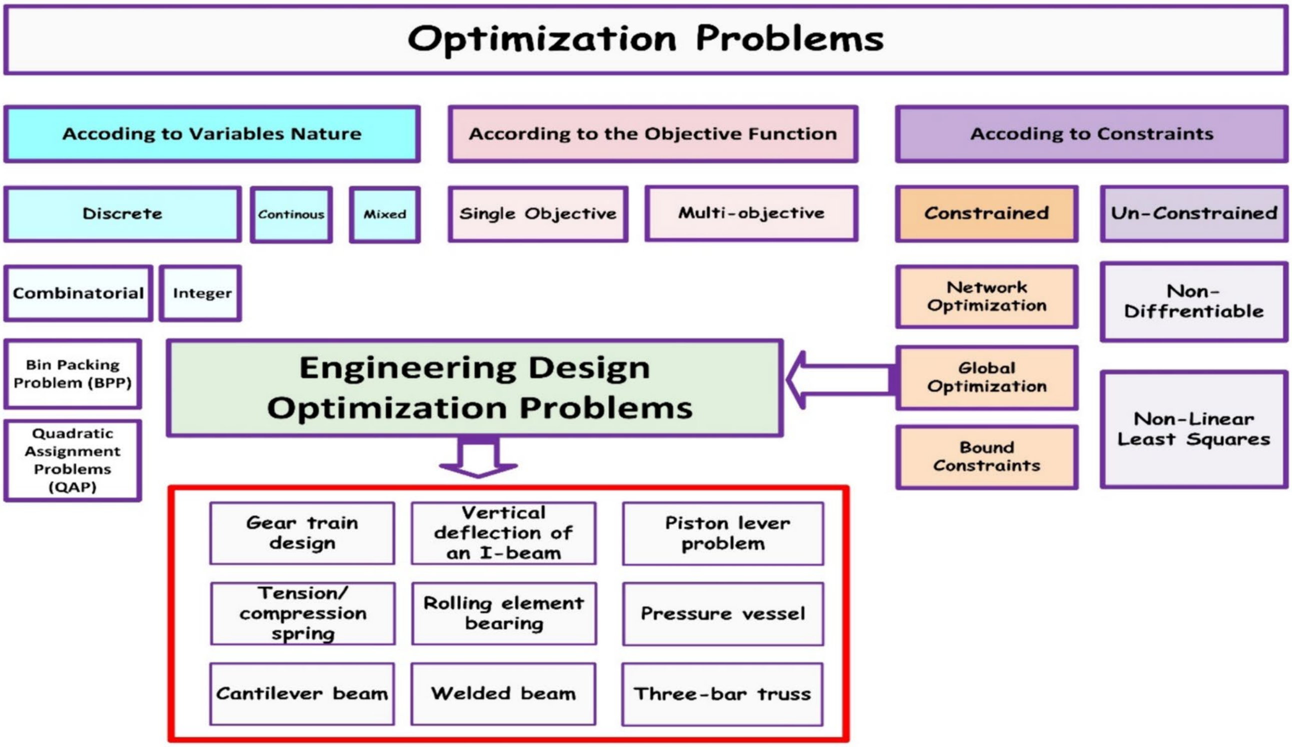


Fig. 8 Optimization problem taxonomy

4.2 Taxonomy According to the Objective Function

In terms of the number of objectives, there are two types. If the number of objectives is greater than one, the problem is called a multi-objective optimization problem; otherwise, the problem is named a single-objective optimization problem. Usually, real-world optimization problems are multi-objective. For example, if we need to design a table, we will consider two objectives, for example, minimizing the weight and the price of the table.

Single-objective optimization Only one global optimal solution exists in single-objective optimization. The objective function only considers one objective; therefore, the best optimal solution can be easily determined by comparing the obtained solutions using basic comparison operators $<, >, \leq, \geq$, and $=$, the nature of this type allows the algorithm to easily tackle optimization problems. Without loss of generality, Eq. (1) is used to determine the mathematical structure of a single-objective optimization problem.

$$
\begin{aligned}
&Minimize : f(x_1, x_2, x_3, \ldots, x_{n-1}, x_n)\\
&Subject\ to :\\
&g_1(x_1, x_2, x_3, \ldots, x_{n-1}, x_n) \geq 0, \quad i = 1, 2, 3 \ldots m\\
&h_1(x_1, x_2, x_3, \ldots, x_{n-1}, x_n) = 0, \quad i = 1, 2, 3 \ldots p\\
&lb_i \leq x_i \leq ub_i, \quad i = 1, 2, 3 \ldots n
\end{aligned}
\tag{1}
$$

where the problem decision variable is symbolized by ***n***, ***m*** and ***P*** exist to represent the number of inequality and equality constraints, respectively. For the i^{th} variable, ub_i and lb_i are used to represent the upper and lower boundaries, respectively.

Multi-objective optimization In contrast to single objective optimization, A set (more than one) of objectives need to be optimized simultaneously in the multi-objective optimization problems. Usually, these objectives are a conflict with each other, so most of work in this type is paid to achieving a trade-off between these objectives. The set of solutions in this type is called a Pareto optimal solution. The Pareto optimal dominance is employed to compare the solutions obtained in order to determine the optimal solutions. Extra storage is needed to hold the Pareto optimal solutions. Without loss of generality, Eq. (2) is used to determine the mathematical structure of a single objective optimization problem.

$$
\begin{aligned}
&Minimize :\\
&F(x) = \left[f_1(x), f_2(x), f_3(x), \ldots, f_o(x)\right]\\
&Subject\ to :\\
&g_1(x) \geq 0, \quad i = 1, 2, 3 \ldots m\\
&h_1(x) = 0, \quad i = 1, 2, 3 \ldots p\\
&l_i \leq x_i \leq u_i, \quad i = 1, 2, 3 \ldots n
\end{aligned}
\tag{2}
$$

where the problem decision variable is symbolized by $\boldsymbol{n}$, $\boldsymbol{m}$ and $\boldsymbol{P}$ exist to represent the number of inequality and equality constraints respectively. For the variable i^{th}, U_i and L_i are used to represent the upper and lower limits, respectively. The number of objectives is denoted by o, and the g_i and h_i are the i^{th} inequality and equality constraints, respectively. In general, the clash among objectives enforces the problem designer to consider more than one criterion in the comparison of obtained solutions and therefore the classical comparison operator does not perform better, instead, the Pareto dominance Eq. (3) is used to define the best optimal solutions.

$$\begin{aligned} &\forall i \in \{1, 2, \ldots, O\} : f_i(x) \leq f_i(y) \Lambda \exists i \in \{1, 2, \ldots, k\} : f_i(x) < f_i(y) \\ &Where\ \vec{x} = \left(x_1, x_2, \ldots, x_k\right),\ \text{and}\ \vec{y} = \left(y_1, y_2, \ldots, y_k\right) \end{aligned} \tag{3}$$

Here the two solutions are represented by the vectors x and y. The x is said to dominate y denoted as $(x \leq y)$ if x has at least one better value in all objectives.

4.3 Taxonomy According to Function Form

From another angle, classification can be done according to function form. If we have a real-world optimization problem, the constraints are linear qualities and inequalities and the objective function formed as linear then the problem is said to be a linear optimization problem. In nonlinear optimization, one or both of the objective functions and constraints are nonlinear, and this is the realistic and complex one [86].

4.4 Taxonomy According to the Design Variable

According to the nature of the design variables, we can present three different types of optimization problems, as detailed in the following points.

Discrete optimization problems In discrete optimization problems the values of the design variables are discrete and in which there is a finite set of values. The shortest path problem and the minimum spanning tree problem are two instances of this type. For more details, we can mention that the discrete optimization consists of integer programming and combinatorial optimization. *Integer programming* deals with the formulation and solution of discrete integers (or binary integers) valued in the design variables. On the other hand, *combinatorial optimization* emphasizes the combinatorial origin, formulation, or solution of a problem. Mainly it seeks to achieve pairs (i.e., Assignments, groupings, orderings) of discrete and finite values under the influence of specific constraints. These pairs involve a component of solutions of potential combinatorial problem solutions [87]. In Bioinformatics, Artificial intelligence and other fields combinatorial optimization can be applied such as identifying propositional formula models or defining the 3D structure of protein, finding the shortest path in graphs, the travelling salesman problem, the knapsack problem in addition to the *pin packing problem, the quadratic assignment problem* which has been tackled in this study.

Continuous optimization problems In continuous optimization problems, A range of values is assigned to the design variables, so every design variable has an infinite set of values. These problems have two types constrained continuous optimization problems which there is a constant on the variables. For unconstrained continuous optimization problems there is an absence of these constraints maximization the general yield for differential amplifiers, optimization of the mechanical system of shock absorption are two examples of this type [88].

Mixed discrete–Continuous optimization problems In many problems a design variable has a mixture of discrete and continuous values, in this case we call the problem mixed discrete–continuous type. This type is the most widely used one, where numerous real-world problems are complex and possess a mixed quantitative and qualitative input. In [89], a set of instances is addressed using black-box optimization techniques.

4.5 Taxonomy According to Constraints

Furthermore, the classification can be according to the restrictions on the design variables.

Unconstrained Optimization Problem If there are no constraints on the design variables we call this type as unconstrained optimization problem, the unconstrained optimization can be viewed as iterative methods stating with initial estimation for the optimal solution then a set of iteration is used to reach for the optimal solution. Usually, the solutions were reduced iteratively to an optimal solution. In [90], Fletcher and Roger mentioned that the unconstrained optimization methods differ according to how much information the user provides, such as the gradient method, the second derivative method, and the non-derivative method.

Constrained optimization problem If there is one or more constraints, the optimization problem falls under the constrained optimization problem class. Furthermore, there are two subclasses of this type. The first subclass is *equality constraint problem* in which the values of the design variables are restricted to be equal to the specific value. The second subclass is *inequality-constrained problem* the design variables are restricted to greater / smaller than a specific value. From a formulation perspective, every equality constraint can be mathematically transformed to two inequality constraints. For example, $\phi(x) = 0$ is equivalent to $\phi(x) < 0$ and $\phi(x) > 0$ [86]. Mainly, the constrained optimization covers three types of optimizations which are network optimization, bound constraints optimization, and the global optimization. Global optimization includes one of the most widely used problems, which is engineering design optimization problems.

5 Performance Assessment of Optimization Algorithms

First of all, we must refer to an important term, the efficiency of the algorithm, which means how the algorithm responds against finding the optimal solutions for the problem to be solved. Achieving an optimal solution is not the only purpose of a good optimization algorithm; instead, the algorithm must be high quality and achieve a better situation in the applicability process on different classes of problems. To judge the quality and applicability of the algorithm, the algorithm must be compared against a set of qualitative and quantitative measures. The good quality algorithm performs better and achieves better results when tested against qualitative and quantitative measures. In this section, we will present the whole assessment environment used to test the quality and applicability of the algorithm.

CEC Test suite CEC stands for Congress on evolutionary computation. Mainly the CEC holds a different class of problems, which may be uni-model, multi-modal, fixed-dimension multi-modal, and composite. CEC is usually used to test the performance of the algorithm and its ability to solve different classes of problems. In the art of optimization almost all studies perform the CEC function as a fitness function to test algorithm's performance itself and to compare the algorithm's performance against other algorithms.

Statistical Measures In this metric, the Best, Worst, Mean, and standard deviation are computed to the obtained solutions to judge about the quality of all solutions obtained together. The best solution is the one with a minimum value of fitness function in minimization optimization and the opposite is right for maximization optimization. The Worst is the solution which has a maximum value of fitness function on the minimization optimization and the opposite is right for maximization optimization. while the mean is used to compute the average value of all obtained solutions (obtained from executing the algorithm many times), and the small value of the mean means that the algorithm is doing better. Finally, the standard deviation or STD is the statistical measure that gives the reader insight into the differences among the obtained solutions, and the algorithm with small STD value is also better than the other with large value. There is also an important statistical measure, which is capable of measuring the whole performance of algorithms for any number of functions. This measure uses the mean rank sum value to rank the algorithms. Ranking these values in ascending order enables us to say that the algorithm with the lowest value is the best among all algorithms participating in comparison for all functions together.

Convergence curve Drawing a relation between the solutions scored by the algorithm and the number of iterations or number of function evaluations is the primary goal of the convergence curve. To summarize the behavior of the algorithm, the convergence curve is drawn to judge the speed of the algorithm in reaching the global optimal solution. For the minimization problem and to compare the performance of many algorithms. The lower convergence curve is better than the upper one. Also, we can compute how fast the algorithm converges towards the optimal solution through the rate of convergence measure.

Diversity Diversity measure is one of the measures related to the algorithm's convergence behavior. In simple words, diversity means how the search agents of the algorithm are distributed in the search space. A high diversity value of the algorithm can be translated into a great exploration ability of the algorithm, and a low value can be translated into a great exploitation ability of the algorithm. Hence the diversity values of the algorithm must be smoothly transited from high value in the first iterations of the algorithm to low value in the rest of iterations of the algorithm. In this context, we can say that the good diversity of the algorithm leads to avoiding premature convergence and achieve a good speed in achieving the optimal solution hence score a high level of efficiency.

Trajectory diagram In order to test the behavior of a specific agent of the algorithm over the curse of iterations the trajectory diagram is used. The fluctuations of the curve are an indication of the better performance of that agent and its ability to explore and exploit the search space better.

Search history diagram To visualize the history of positions scored by the search agent during the process of optimization, the search history curve is drawn.

Exploration and exploitation The exploration and exploitation (EXPL-EXPT) curves are used to visualize the exploitative and explorative capabilities of the algorithm. Usually, the overlaps between the two curves exist to tell us about the shifting between exploration and exploitation, and therefore an EXPL-EXPT balance.

Real-world problems To test the ability of the algorithm in solving different classes of problem the real-world problem is tackled. Engineering design problems are the most widely used problems as many algorithms use the (pressure vessel, welded beam, 15/3/25/52-bar truss system, tension/compression spring…etc.) classical design problems to quiz the algorithm performance.

Operation platforms Alongside the previous measures, the algorithm quality can be affected by the environment setup in which the algorithm is executed. The good environment in both software and hardware capabilities leads to good behavior of the algorithm. In this context, we must mention that when we compare more than one algorithm to judge which is better, we must execute the algorithms in the same environment to achieve a fair comparison.

6 Metaheuristics Applications

As mentioned above, MAs have a great degree of applicability, as they operate better in solving different problems that involve a computation time restriction, a high-dimensional problem, and other kinds of problems. Specifically, MAs are capable of dealing with different classes of optimization problems in different fields. In the following subsections, the applicability of MAs in some of these fields are illustrated in detail.

6.1 IEEE Congress on Evolutionary Computation (IEEE CEC)

CEC stands for Congress on evolutionary computation. Mainly the CEC holds a different class of problems, which may be uni-model, multi-modal, fixed-dimension multimodal, and composite. CEC is usually used to test the performance of the algorithm and is considered as an indication of the capabilities of the algorithm to solve different classes of problems. In the art of optimization, almost all studies perform the CEC function as fitness functions to test the algorithm's performance itself and to compare the algorithm's performance against other algorithms. Almost a different version of the CEC test suite is introduced every year. Tables 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and 12 are presented below to provide the reader with the basic information on each version of the CEC benchmark function and how metaheuristic algorithms are applied to solve these benchmark functions.

6.2 Engineering Design Problems

It is easy to provide an optimal design for a simple problem that contains a small number of design variables with a small range of values. In contrast to complex problems with many components, algorithms consume a huge amount of time to develop an optimal design. For example, the mechanical

Table 2 Metaheuristics optimization algorithms for solving CEC 2005 benchmark functions

Basic information	Number of problems	25
	Dimensions	2, 10, 30, and 50
Problem types	Uni-model	F1, F2, F3, F4
	Multi Model	F7, F8, F9, F10, F11, F12, F13, F14, F15, F16, F17, F18, F19, F20, F21, F22, F23, F24, F25
Meta-heuristic algorithms	Single objective	1) Light spectrum optimizer (LSO) [91] 2) IOrchard Algorithm (IOA) [92] 3) K-means optimizer (KO) [93] 4) Criminal search optimization (CSO)[94] 5) Geyser inspired algorithm (GIO) [95] 6) The cheetah optimizer (CO) [96] Cooperation search algorithm [97]
	Multi objective	1) Multi-objective bonobo optimizer (MOBO) [98] 2) Multi-hybrid algorithm (MHA) [99] 3) Multi objective enhanced Harris Hawks optimizer (MO-EHHO) [100] 4) Chemical reaction partial swarm optimization [101] 5) Multi-objective bonobo optimizer (MOBO) [98]

Table 3 Metaheuristics optimization algorithms for solving CEC2009 test function

Basic information	Number of problems	30
	Dimensions	2, 10, and 30
Problem types	Two-objective functions	UF1—UF2—UF3—UF4 – UF5—UF6—UF7
	Three-objective functions	UF8—UF9—UF10
Multi objective metaheuristics algorithms	1) Multi-objective sunflower optimization [102] 2) Multi-strategy genetic algorithm [103] 3) Multi-objective whale optimization algorithm [104] 4) Multi-objective invasive weed optimization [105] 5) Multi-objective stochastic fractal search [106] 6) Multi-objective modified symbiotic organisms search algorithm [107] 7) Multi-objective equilibrium optimizer algorithm [108] 8) Multi-objective hybrid CSA-PSO optimization algorithm [109]	

Table 4 Metaheuristics optimization algorithms for solving CEC 2013 test functions

Basic information	Number of problems	28
	Dimensions	10, 30, and 50
Problem types	Uni-modal functions	F1, F2, F3, F4, and F5
	Basic multimodal functions	F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16, F17, F18, F19, F20
	Composition functions	F21, F22, F23, F24, F25, F26, F27, F28
Meta-heuristics algorithms	1) success history-based adaptive differential evolution hybrid with modified Whale optimization algorithm [110] 2) comprehensive learning JAYA algorithm (CLJAYA) [111] 3) Hybrid variable mesh optimization and DE algorithm [112] 4) Search and Rescue Optimization Algorithm (SAR) [113] 5) Elephant Herding Optimization Algorithm (EHA) [114] 6) Hybrid firefly and particle swarm optimization algorithm [115] 7) Harmony Search Algorithm (HSA) [116] 8) Dual Population Genetic Algorithm [117] 9) Five Phases Algorithm (FPA) [118] 10) Improved African vultures optimization algorithm [119]	

Table 5 Metaheuristics optimization algorithms for solving CEC 2014 test functions

Basic information	Number of problems	30
	Dimensions	10, 30, 50, and 100
Problem types	Uni-modal functions	F1, F2, and F3
	Basic multimodal functions	F4,F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, and F16
	Hybrid functions	F17, F18, F19, F20, F21, and F22
	composition functions	F23, F24, F25, F26, F27, F28, F29, and F30
Meta-heuristics algorithms	1) The solar system algorithm [120] 2) Young's double-slit experiment optimizer [121] 3) Queuing search algorithm [122] 4) Geyser inspired algorithm [95] 5) Enhanced particle swarm optimization algorithm [123] 6) Nutcracker optimizer [124] 7) Kepler optimization algorithm [125] 8) Simulated Kalman filter [126] 9) Electromagnetic field optimization [127] 10) Light spectrum optimizer [32]	

problem with different components and multiple objectives and constraints. Another example for complex problems is the engineering design problems in which the design process starts with exploiting the experience of designers to guess an optimal design for any problem, but this is not the optimal direction. In order to treat this poor thinking, we need systematic work that guarantees achieving an optimal design that is better than any other human design. Automatic techniques or, in other words, metaheuristics algorithms (MAs) are used to effectively diversify the search space with large parameters, minimizing the cost, and improving the product life cycle. Mainly, the MAs tune the parameters of the problem to produce the best optimal values of the design variables, hence achieving the optimal design. Table 13 is used to highlight the work of single-objective metaheuristics optimization algorithms in solving engineering design problems; also, Table 14 is introduced to clarify the tries of multi-objective optimization algorithms in tackling engineering design problems.

6.3 NP-Hard Problems

In NP-hard problems the NP stands for nondeterministic polynomial time where the nondeterministic refers to nondeterministic Turing machines which apply the idea of brute-force search method. On the other hand, the polynomial is used to refer to the amount of time required to apply the quick search to get the single solution of the deterministic

Table 6 Metaheuristics optimization algorithms for solving CEC 2015 test functions

Basic information	Number of problems	15
	Dimensions	10, 30, 50, and 100
Problem types	Uni-modal functions	F1, F2
	Basic multimodal functions	F3, F4, F5
	Hybrid functions	F6, F7, F8
	Composition functions	F9, F10, F11, F12, F13, F14, F15
Meta-heuristics algorithms	1) Self-defense Techniques of Natural Plants Algorithm [128] 2) Smart Flower Optimization Algorithm [129] 3) Differential Evolution algorithm combined with fuzzy logic [130] 4) Numeric Crunch Algorithm [131] 5) Walrus Optimization Algorithm (WaOA) [132] 6) hybrid firefly particle swarm optimization algorithm [115] 7) Group teaching optimization algorithm [133] 8) Enhanced Jaya algorithm [134] 9) Spherical search optimizer [135] 10) Clouded Leopard Optimization [136]	

Table 7 Metaheuristics optimization algorithms for solving CEC 2017 test functions

Basic information	Number of problems	30
	Dimensions	10, 30, 50, and 100
Problem types	Uni-modal functions	F1, F2, and F3
	Basic multimodal functions	F4, F5, F6, F7, F8, F9, F10
	Hybrid functions	F11, F12, F13, F14, F15, F16, F17, F18, F19, F20
	Composition functions	F21, F22, F23, F24, F25, F26, F27, F28, F29, F30
Meta-heuristics algorithms	1) Marine Predators Algorithm [137] 2) Emperor Penguin and Salp Swarm Algorithm [138] 3) Atomic orbital search [139] 4) Chaos game optimization [140] 5) Archimedes optimization algorithm [31] 6) Coati Optimization Algorithm [141] 7) Mother optimization algorithm [142] 8) One-to-One-Based Optimizer [143] 9) Lévy flight distribution [40] 10) Wild horse optimizer [144]	

algorithm, or the time consumed by the nondeterministic Turing machines to perform extensive search. P is the set of all decision problems solvable in polynomial time. Specifically, the decision problem has two answers YES and No. Consequently, if all YES answers are checked in polynomial time, then the problem belongs to set of NP problems; on the other hand, co-NP is used for NO answer. If the polynomial-time solution obtained for a specific problem leads to a polynomial-time solution for all problems in the NP in this case the problem is said to be NP-hard. Also, a problem is NP-Complete iff it is NP-Hard, and it is in NP itself. Due to the high computational complexity, the exhaustive search methods do not have the ability of getting the best solution.

Quadratic Assignment Problem (QAP) As mentioned in [195] the QAP is NP-hard, as the polynomial time is not sufficient to obtain the approximate solution from optimal solution. QAP was first introduced by Koopmans and Beckmann in 1957 [196] as an extension of the linear assignment problem. QAP is considered a combinatorial optimization problem that has been considered and tackled by many research studies in the last three decades. However, the good results obtained in these studies but until now the QAP is not well solved as there is no exact algorithm capable of solving problems with more than 20 input sizes in a reasonable amount of computational time [197]. In QAP we seek to locate the facilities in its appropriate location under the condition that it is an exact-one-to-exact-one problem, that is, each site can only grasp only one facility and each facility must be placed in only one site where the distances between facilities and sites are determined. The main optimization goal of QAP is to minimize the distance and flow between each pair of facilities delegated to their relevant sites. Recently, QAP is addressed by many books, studies and reviews, as listed in

Table 8 Metaheuristics optimization algorithms for solving CEC 2018 test functions

Basic information	Number of problems	30
	Dimensions	10, 30, 50, and 100
Problem types	Uni-modal Functions	TF1, TF2, and TF3
	Basic multimodal functions	TF4, TF5, TF6, TF7, TF8, TF9, TF10
	Hybrid functions	TF11, TF12, TF13, TF14, TF15, TF16, TF17, TF18, TF19, F20
	Composition functions	TF21, TF22, TF23, TF24, TF25, TF26, TF27, TF28, TF29, TF30
Meta-heuristics algorithms	1) Cooperative based hyper-heuristic algorithm [145] 2) An improved moth-flame optimization [146] 3) multi-trial vector-based differential evolution (MTDE) [147] 4) The Quantum-based Avian Navigation Optimizer Algorithm (QANA) [148] 5) Enhanced moth-flame optimization algorithm (MFO-SFR) [149] 6) Multi-trial vector-based monkey king evolution algorithm (MMKE) [150] 7) Modified LSHADE algorithm with a rank-based selective pressure strategy [151] 8) Hybrid Mean–Variance Mapping Optimization (MVMO-PH) [152] 9) multi-objective evolutionary algorithm based on 10) decomposition based on information feedback model (MOEA/D-IFM) [153] 10) Adaptive Fox Optimization (AFOX) Algorithm [154]	

Table 9 Metaheuristics optimization algorithms for solving CEC2019 test functions

Basic information	Number of problems	10
	Dimensions	9, 10, 16, 18
Problem types	Multi model functions	
Meta-heuristics algorithms	1) Electrical search algorithm [155] 2) Green Anaconda Optimization 3) Enhanced Lévy Arithmetic Algorithm [156] 4) Election-Based Optimization Algorithm (EBOA) [157] 5) An improved binary grey wolf optimizer [158] 6) Electrical search algorithm [155] 7) Water and Salt Transport Optimization [159] 8) Serval optimization algorithm [160] 9) Dynamic Cat Swarm Optimization algorithm [161] 10) The Bedbug Meta-heuristic Algorithm [162]	

Table 15. From another angle, there are several problems that are considered as special types of QAP (Table 16).

The Bin Packing Problem (BPP) The Bin Packing Problem (BPP) is one of most familiar combinatorial problems that is considered as strongly NP-hard problem [218]. In BPP we need to pack a set of items m into n bins with the aim of minimizing the number of bins required to hold all items. The BPP can be mathematically formulated as in Eq. 4 [219].

$$\begin{aligned} & Min : \ z(n) = \sum_{j=1}^{n} y_j \\ & \text{Subject to :} \\ & \sum_{j=1}^{n} W_i X_{ij} \le C_{yj} \quad \forall \text{j} \in \{1 \text{ up to n }\} \\ & \sum_{j=1}^{n} X_{ij} = 1 \quad \forall \text{i} \in \{1 \text{ up to n }\} \\ & Y_j, X_{ij} = \begin{cases} 1, & \text{if item } i \text{ is located in bin } j \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{4}$$

where capacity of the bin y_j is symbolized by C_{yj}.

The BPP benchmark data sets consist of three different types that are commonly classified as Easy, Medium, and Hard class as mentioned in [220]. Also, the BPP appears in one, two, three, and multi-dimensional form (Table 17).

Travelling Salesman Problem (TSP) One of the most familiar combinatorial optimization NP-Hard is the TSP in which we need to minimize the route as possible consumed to visit all cites precisely once and return to the initial city given a list of cities and distances among them. For example, in the TSP of *20* city, we have a huge number of feasible solutions (approx.1.22 × 1017). Guess how much time is required to perform this task using exhaustive search? the answer is very long. Therefore, exhaustive searches have disabilities in tackling such problems. The use of MAs destroys this disability, as it was used to find near optimal solutions in a reasonable amount of time [233]. Vehicle routing problems (VRPs) is the general form of TSP and is a multi-objective real-world problem tackled by many MAs such as genetic algorithm (GA), particle

Table 10 Metaheuristics optimization algorithms for solving CEC 2020 test functions

Basic information	Number of problems	10
	Dimensions	5, 10, 15, and 20
Problem types	Uni-modal functions	CEC01
	Basic multimodal functions	CEC02, CEC03, CEC04
	Hybrid functions	CEC05, CEC06, CEC07
	Composition functions	CEC08, CEC09, CEC10
Meta-heuristics algorithms	1) Fire Hawk Optimizer [163] 2) Teaching learning based artificial bee colony (TLABC) [164] 3) The solar system algorithm [120] 4) Numeric Crunch Algorithm [131] 5) Atomic orbital search [139] 6) Light spectrum optimizer [32] 7) The archerfish hunting optimizer [165] 8) Hybrid Salp swarm-Harris hawks optimization algorithm [166] 9) Energy valley optimizer [167] 10) Squid Game Optimizer (SGO) [168]	

Table 11 Metaheuristics optimization algorithms for solving CEC 2021 test functions

Basic information	Number of problems	10
	Dimensions	10 and 20
Problem types	Uni-modal functions	TF01
	Basic multimodal functions	TF02, TF03, TF04
	Hybrid functions	TF05, TF06, TF07
	Composition functions	TF08, TF09, TF10
Meta-heuristics algorithms	1) Hierarchical learning particle swarm optimization [169] 2) Walrus optimizer [170] 3) An improved remora optimization algorithm [171] 4) An improved wild horse optimizer [171] 5) Multi-strategy enhanced dung beetle optimizer [172] 6) Self-organizing migrating algorithm [173] 7) Self-adaptive differential evolution algorithm [174] 8) Battlefield optimization algorithm [175] 9) Multi-objective sunflower optimization [102] 10) Multi-objective arithmetic optimization algorithm [102]	

Table 12 Metaheuristics optimization algorithms for solving CEC 2022 test functions

Basic information	Number of problems	12
	Dimensions	2, 5, 10 and 20
Problem types	Uni-modal functions	TF01
	Basic multimodal functions	TF02, TF03, TF04, TF05
	Hybrid functions	TF06, TF07, TF08
	Composition functions	TF09, TF10, TF11, TF12
Meta-heuristics Algorithms	1) Young's double-slit experiment optimizer [121] 2) Light spectrum optimizer [32] 3) T cell immune algorithm [176] 4) Kepler optimization algorithm [125] 5) An adaptive differential evolution [177] 6) Hierarchical learning particle swarm optimization [169] 7) EWSO: boosting white shark optimizer [178] 8) Sand cat arithmetic optimization algorithm [179] 9) Modified bald eagle search algorithm [180] 10) Red-tailed hawk algorithm [181]	

Table 13 Single objective algorithms for solving engineering design problems

Single objective optimization algorithm and reference	Engineering design problems
Gradient-Based Optimizer (GBO) [72]	Three-bar truss design
	Cantilever beam design
	Rolling element bearing
	Speed reducer
	I-beam vertical deflection
	Tension–compression spring design
Marine Predators Algorithm (MPA) [137]	Welded beam design
	Pressure vessel design
	Tension–compression spring design
Sine cosine-grey wolf optimizer (SC-GWO) [182]	Three-bar truss design
	Pressure vessel
	Gear train design
	Speed reducer
	Tension–compression spring design
Barnacles Mating Optimizer (BMO) [183]	Optimal reactive power dispatch (ORPD) problem
Tunicate Swarm Algorithm (TSA) [184]	25-bar truss design
	Displacement of loaded structure
	Rolling element bearing
	Speed reducer
	Pressure Vessel
	Welded beam design
	Tension–compression spring design
Enhanced flower pollination algorithm [185]	Welded beam design
	Tension–compression spring design
Slime mould algorithm (SMA) [186]	cantilever beam design
	I-beam vertical deflection design
	Pressure Vessel
	Welded beam design
Chaotic gravitational search algorithm (CGSA) [187]	Tension–compression Spring design
	Pressure Vessel
	Welded beam design
	Rolling element bearing
	Speed reducer
	Multiple disc clutch brake
	Hydro-static thrust bearing design
Artificial rabbits' optimization (ARO) [188]	Tension–compression spring design
	Pressure Vessel
	Rolling element bearing
	Gear train design
	Cantilever beam design

swarm optimization (PSO) and colony optimization (ACO) as in [234].

Job Shop Scheduling (JSS) JSS is a NP-Hard problem in which the algorithm seeks to consume a polynomial time to solve it. In JSS we need to process a finite set of jobs using a limited set of machines. JSS is a general type of scheduling problem. JSS is addressed by many MAs in [235–237], and [238].

6.4 Medical Science

Most of medical activities (i.e., Diagnosing, imaging, treatment, and monitoring) depends in its work on the computer or electronic device that is operate using an algorithm-based software [68]. Several researchers have used GAs for edge detection of images acquired using different imaging modalities, including magnetic resonance imaging, CT,

Table 14 Multi objective algorithms for solving engineering design problems

Single objective optimization algorithm and reference	Engineering design problems
Multi-Objective Ant Lion Optimizer (MOALO) [189]	Disk brake design
	Cantilever beam design
	Brush-less dc wheel motor
	Speed reducer
	Safety isolating transformer design
	Welded beam design
	4-bar truss design
Multi-Objective Stochastic Fractal Search (MOSFS) [106]	Welded beam engineering design problem
Novel variable-fidelity (VF) optimization integrated with multi-objective genetic algorithms (MOGAs). [190]	Torque arm optimization design
	Micro-aerial vehicle fuselage engineering design problem
Multi-objective Spotted Hyena Optimizer (MOSHO) [191]	Multiple-disk clutch brake
	Pressure vessel design
	Gear train design
	Welded beam design
	25-bar truss design
Multi-objective modified adaptive symbiotic organisms search (MOMASOS) [192]	Multiple-disk clutch brake
	Speed reducer design
	cantilever beam design
	Welded beam design
Multi-objective sine–cosine algorithm (MO-SCA) [193]	Multiple-disk clutch brake
	Spring design problem
	Gear train design
	Welded beam design
	Speed reducer problem
	Four-bar truss design
Multi-objective marine predator algorithm (MOMPA) [194]	Multiple-disk clutch brake
	I-beam design problem
	Gear train design
	Welded beam design
	Speed reducer problem
	Pressure vessel problem
	Tool spindle problem
	Car crash problem
	Brush-less dc wheel motor
	Safety txmer
	Four-bar truss design
	10-bar truss design
	25-bar truss design

and ultrasound [239]. In [240], Pereira et al., applied a set of computational tools for mammogram segmentation to improve the detection of breast cancer using GA combined with wavelet analysis to allow the detection and segmentation of suspicious areas with 95% sensitivity. GA has been applied for feature selection to identify a region of interest in mammograms as normal or containing a mass [241]. Also GA is combined with a support vector machine to differentiate benign and malignant breast tumours in ultrasound images [242], GA is combined with diversity index to discover lung nodules by developing an automatic threshold clustering method [80]. In [243] electroencephalography signals were used to detect hypoglycemia in patients with type ***1*** diabetes. Depending on neural networks in conjunction with ant colony optimization (ACO) and particle swarm optimization (PSO) Suganthi and Madheswaran use a more advanced computer-aided decision support system and mammogram to group tumours and detect breast cancer stages as described in [244]. Based on artificial bee colony (ABC) algorithm Kockanat and et al., Develop a technique

Table 15 Meta-heuristics optimization algorithms for solving the QAP

Meta-heuristic optimization algorithms for solving the QAP	References	Year
adaptive large neighborhood search algorithm (ALNS)	[198]	2023
Improved hunting search algorithm (I-HSA)	[199]	2019
Iterative local search (ILS) algorithm	[200]	2020
Hybrid teaching–learning-based optimization algorithms	[201]	2015
Hybrid simulated annealing (SA) and tabu search (TS)	[201]	2015
Particle Swarm Optimization (PSO)	[202]	2011
Greedy Genetic Algorithm (GGA)	[203]	2000
Artificial Bee Colony (ABC)	[204]	2019
Improved Ant Colony Optimization Algorithm (I-ACO)	[205]	2008
Genetic Approach Optimization (GAO)	[206]	1995
Tournament selection-based antlion optimization algorithm (ALO)	[207]	2019
A biogeography-based hybrid with tabu search optimization algorithm	[208]	2016
Migrating birds optimization (MBO	[209]	2012

Table 16 Special types of quadratic assignment problem

Quadratic assignment problem instance	Brief note	References	Year
The Quadratic Bottleneck Assignment Problem (QBAP)	Instead of minimizing the overall cost of facilities network we seek to minimize the distance between two facilities	[210]	2012
Generalized QAP (GQAP)	The main idea of this type is to allow multiple facilities to be located in the site in case of resource availability	[211]	2004
Bi-quadratic Assignment Problem (BiQAP)	This essential type of QAP that is defined by allowing quadruple of facilities to be interacted simultaneously instead of pairs	[212]	2004
The Quadratic Semi-Assignment Problem (QSAP)	The main idea of this instance is to provide the flexibility in the number of facilities and sites where different number of facilities and sites may be exist	[213]	1969
The Multi-Story Space Assignment Problem (MSAP)	MSAP is an innovative type of the multi-story facility assignment problem in which department locations are unequal in size	[214]	2010
The quadratic three-dimensional assignment problem	The Q3AP is mainly used to optimize the process of data bits mapping to modulation symbols by hybridizing the automatic repeat request scheme	[215]	2008
Multi-objective quadratic assignment problem (MOQAP)	In this study the authors generate a two instance of MOQAP to allow sitting number of instance parameters	[216]	2003
The bipartite quadratic assignment problem and extensions	In this study, authors introduce a three different neighborhood structures. the best among them can be identified in polynomial time even though two of these neighborhoods are of exponential size	[217]	2016

for demonising images using *2D* impulse response digital filter as illustrated in [245].

6.5 Robotics

Robotics is a vital active research field that owns some challenges that needs to be optimized such as task performance, Decrease the robotics cost, achieve a better reliability, in addition to minimize the unit complexity over other traditional robot systems. In this context, metaheuristics can be used to tune machine learning methods to enhance the collaborative behavior of robotics. One of the most active problems in robotics is the redundant humanoid manipulator issue. The complexity of this problem comes from the existence of multiple number of degrees of freedom and complex joint structure. This problem causes difficulty in achieving an inverse kinematics solution. Scientists make an effort to formulate this problem as a minimization problem, hence the MAs can perform better in solving this problem. In [246], the multilayer perceptron neural network is trained by the exploitative and explorative capabilities of the bee's algorithm to learn the inverse kinematics of a robot manipulator arm. To conquer the problem of multi-solution, the GA is used to achieve a global optimal solution for inverse kinematics of 7-DOF (seven degree of freedom) manipulator [247]. Also, the inverse kinematics of the

Table 17 Applying the meta-heuristics optimization for the bin packing problem

Optimization algorithms	BPP type	References	Year
A modified squirrel search algorithm	1-Dim BPP	[221]	2019
Scheduling optimization method	1-Dim BPP	[222]	2021
Deterministic greedy algorithm	rectangular BPP	[223]	2019
A genetic algorithm	3-Dim BPP	[224]	2014
Hybrid harmony search algorithm	large scale BPP	[225]	2021
Hybrid evolutionary algorithm	large scale BPP	[226]	2013
An enhanced grasshopper optimization algorithm	1-Dim BPP	[227]	2020
Branch-and-price algorithm	ordered open-end BPP	[228]	2008
Algorithm of Changes (AOC)	binpack 1, 4 and 8	[229]	2010
First-Fit and Best-Fit Decreasing algorithm	variable sized BPP	[230]	2003
Improved Lévy-based whale optimization algorithm	1-Dim BPP	[231]	2018
Evolutionary particle swarm optimization	Multi-objective BPP	[232]	2008

seven-degree-of-freedom (7-DOF) manipulator is perfectly tackled by the particle swarm optimization algorithm (PSO) by exploiting the strong intelligent scene and collaborative behavior among particles [248]. Biogeography-based optimization (BBO) is hybrid with differential evolution (DE) and uses the merits of the hybrid migration operator and the adapted Gaussian mutation operator to solve the inverse kinematics problem of the 8-DOF redundant humanoid manipulator [249].

6.6 Finance

Metaheuristics algorithms can be one of the most promising techniques used to solve different types of problem that occur in the finance and banking activities. In the following points, we will introduce a list of the most familiar problems and how the metaheuristics used to solve these problems.

Portfolio optimization and selection problem (POSP) in this problem, investors seek to assign optimal weights to the assets of the portfolio to achieve a minimal risk of investment. In [250], the authors provide a survey to solve POSP using metaheuristics and examples. Furthermore, the three GA, TS, and SA metaheuristic algorithms are used to solve POSP. The authors of [251] use the PSO algorithm to solve the POSP version with a cardinality constraint.

Index tracking problem (ITP) The ITP is a trading strategy that can depend mainly on two processes (hold and buy). In ITP we want to simulate the behavior of the index of the stock market using a minimum number of stocks. In other words, the ITP is developed to passively simulate the performance of the stock market index. For the specific German index, the authors of [252] use the SA to minimize tracking errors. The combinatorial search is hybrid with the DE for solving the ITP. The authors of [253] compare the performance of GA with quadratic programming and propose a solution approach to minimize the returns on the index using data from the FTSE100 index. Finally, in [254] the authors conducted a set of experiments to solve a special type of ITP and noticed that there was an improvement in an index.

Options pricing problem (OPP) Speculative activities are one of the most familiar tasks in financial markets, and the option can be one of the tools for speculative activities. Due to the fast dynamic motion of the financial market, it is difficult to guess the price of the option using traditional methods, so metaheuristic algorithms can be a promising choice in that case. In order to find parameters that achieve consistency between the model and market prices, Gilli and Schumann [255] use the PSO and DE to study the pricing of the calibration option. Finally, the authors of [256] have shown that the pricing of option operations can be enhanced compared to the traditional binomial lattice method when we use the ACO algorithm.

6.7 Telecommunications Networks

The recently needs for developing complex and large computer systems lead to an urgent demand for designing and developing high quality and more extensive network design and routing techniques and optimally solving problems in such an area. Also, we can notice that most problems in telecommunications are complex and hard to solve using traditional techniques and approximate algorithms, so there is urgent need to employ metaheuristic algorithms to solve network design and routing problems. A set of nodes (i.e., computers, databases, equipment, or radio transmitters) can be connected together using a transmission link (i.e., optical fiber, copper cable, radio, or satellite links) to construct communication networks. Under a set of constraints such as reliability, throughput, delay and link capacity, we seek to achieve a minimum cost of configurations as an objective function for these networks, and many problems can be appeared such as number of nodes, number of routing paths, the frequency assignment, and the capacity installation. A large number of studies using metaheuristics in solving telecommunications problems such as Kim et al. [257] employ a SA algorithm in the mobile radio system to allocate the nominal cells of channels. To minimize the installation cost and maximize traffic, the authors in [258] use the tabu search algorithm with randomized greedy procedures to find the location of the base stations of the universal mobile-based communication system. Specifically, good approximate solutions for large and medium-sized instances are obtained by the randomized greedy procedures, and these solutions were improved by using the tabu search algorithm. Finally, a new metaheuristic algorithm developed based on the Genetic Algorithm and Ant System was proposed to achieve better and efficient solutions for real-life transportation network design problems in large real networks located in two different places (Canada, city of Winnipeg) [259].

6.8 Food Manufacturing Industry

Recently, the metaheuristics can be considered as one of the most widely used efficient decision-making techniques that can be used to solve problems in different disciplines. In this section, we will present brief information about using metaheuristics in one of these disciplines, which is the food manufacturing industry. Specifically, metaheuristics can be applied in many food processes such as thermal drying, fermentation, and distillation. In [260], the authors develop a new hybrid method based on artificial bee colony (ABC) and the record-to-record travel algorithm (RRT) for Optimizing the Traceability in the Food Industry. The proposed method is employed to solve and provide the optimal minimal solution for the batch dispersion manufacturing problem. The hybrid RRT-ABC is used in the French food industry to carry out real-world experiments (that is, sausage manufacturing) to obtain high-performance results compared to traditional methods. The Artificial Bee Colony Algorithm (ABCA) used in the development of a delivery route optimization method to achieve a fresh food distribution without decreasing the quality of the food [261]. Finally, in [262], the Simulated Annealing (SA) is hybrid with the Virus Colony Search Algorithm (VCS) to improve the quality of the result of a sustainable Closed-Loop Supply Chain Network (CLSCN) design in the olive industry.

7 Open Issues and Challenges

However, the good features and abilities of the MAs in solving a wide range of problems, like other techniques, suffers from a set of problems in the following points, we will refer to these problems.

The stochastic nature and near optimal solution As we know, generating an optimal solution is one of the main features of deterministic algorithms such as simplex method. On the contrary to that, the metaheuristics algorithms (as it is a stochastic algorithm in nature) does not guarantee optimality of the obtained solution, but it provides an acceptable solution. This is one of the significant disadvantages of MAs. It is worth mentioning that the deterministic methods (unlike the stochastic methods) face difficulty when dealing with high-complex problems (that is, high-dimensionality and non-differentiable problems). Practically, when we decide to use one of the previous two methods, we choose to gain something and give the other.

The scale-ability and expensive computational cost Practically, the MAs score great promising results in solving problems in different natures such as discrete, continuous and combinatorial problems that contain a large number of decision variables. However, when solving large-scale global optimization problems (LSGOP) the MAs consume an expensive amount of computational cost. This scalability, challenge is one of the most important challenges that researchers must consider in the future due to the great growth in the size of the optimization problems when dealing with high-dimensional machine learning and large-scale engineering problems. In this context, many strategies are developed by the researchers to cover this problem such as the parallelization, approximation and surrogate modelling, hybridization of local search and memetic algorithms, decomposing the big problems into sub-problems, and befit from the sampling techniques.

The weakness of theoretical and mathematical analysis In most sciences such as chemistry, Biology, physics and others, the mathematical analysis of a method can be computed accurately to specify how much the method costs in terms of computational cost. Unlike those sciences, in metaheuristics we encounter a challenge in computing the exact computational cost of the algorithm, the reason behind this difficulty is from mathematical perspective it is difficult to analyze why the metaheuristics algorithms are so successful. Also, researchers need to pay attention to solving problems in determining the convergence analysis of many metaheuristics' optimization algorithms. Finally, researchers also need to develop innovative methods that allow researchers to easily analyze and compute the algorithm's cost in the case of modification and scaling up the algorithm.

Intensification and diversification trade-off The algorithm's degree of effectiveness is measured by the ability of the algorithm to transit smoothly between the exploration (that is, explore as much as possible the feasible area) and the exploitation (that is, achieving good steps towards the optimal solution's area) stages. Achieving a high degree of intensification and diversification balance is one of the most important challenges or issues in most MAs. However, some algorithms achieve an acceptable degree of trade-off between exploration and exploitation; the vast majority of MAs need to address this challenge by scoring a high level of global diversification and local intensification [263].

Large-scale real-world problem formulation Nowadays the vast majority of problems in recent fields such as data science and big data analysis tasks are considered as large-scale real-world problem (LSRP) that is due to the large number of problem components and problem dimensions. Formulating a large-scale real-world problem (LSRP) is one of the crucial issues in metaheuristic algorithms. The issue comes from the large number of optimization variables (decision variables) included in the problem, how these variables interact with each other, how much the variables or components are related to each other, and what is the effect of one variable on the other variables. Also, it is worth mentioning that the large number of variables is translated as the problem size, which affects the computational cost of the algorithm that deals with this problem.

The limitations of the No-Free-Lunch theorem One of the most fundamental theories in the field of optimization is the No-Free-Lunch theorem [264] which states that there is no universal optimizer for all kinds of problems that is the algorithm may do better in some kinds of problems and do no better for the other kinds. We cannot generalize this theory, as it has been proved for the type of single-objective optimization, but it does not hold yet for problems with continuous and infinite domains in addition to multi-objective optimization [265, 266]. In this context, the researchers in the field of metaheuristics must answer how to apply the NFL in terms of several dimensions?

Comparing different algorithms Comparing similar algorithms through the absolute value of the objective function or number of function evaluations is a possible task. On the other hand, we encounter a problem in comparing different algorithms with different objectives through a formal theoretical analysis. Practically no fair/honest or rigorous comparisons exist in this field [267].

Parameter tuning and control The algorithm's parameter plays the most vital role in determining the performance of any optimization algorithms. The algorithm's designer can change the performance of the algorithms by applying the parameter tuning process of the algorithm. Specifically, we can say that poor tuning leads to poor performance, and the opposite is true. As mentioned in [268], it is practically not an easy task to tune the algorithm parameter and control it by changing its values. Another point we must refer to is that, for well-tuned parameters, there are no clear reasons for unchanging the values of these parameters during the optimization process. Until now, the process of parameter tuning has been implemented by applying parametric tests, while parameter control can be implemented stochastically in which the values of the parameters are picked randomly within a prespecified range. Therefore, there is an urgent need to develop automatic systematic methods to control and tune the parameters. The authors in [269] and [270] proposed a self-tuning method as a trial to encounter problems of parameter tuning and control, but with this trial, the computational cost is still expensive. Based on the previous notes, there is an urgent need to develop an automatic method that applies an adaptive change of the parameters in addition to less effect on the computational cost of the algorithm.

The lack of big data applicability Dealing with big data and developing a big data algorithm has turned into an urgent demand today as the data volume has increased dramatically with the help of automatic data collection methods. In this context, we noticed that there is no more concentration on the application of metaheuristics on big data in the current literature. There are no more studies on how to benefit from applying metaheuristics along with big data algorithms. Consequently, in this review, we inform the researchers to spend more effort and trials in developing new reliable methodologies and algorithms to solve big data problems with the help of metaheuristics.

The lack of machine learning and metaheuristics combination One of the most powerful and influential methods for making a decision and performing a predictions task is the machine learning (ML). Recently, very helpful results have been achieved by the ML techniques. So, researchers in the metaheuristics field must pay an attention to methods that benefit from the ML techniques in optimizing the work of current MAs algorithms or developing a new ML-based metaheuristics algorithms. The following points may be helpful and promising with regard to this point.

- Using the new advances in reinforcement, ensemble, and deep learning in applying an automatic choice of specific problems to be handled by existing and new optimization algorithms [271].
- Benefit from the capabilities of machine learning techniques in optimizing the work of the optimization field by generating an automatic model for representing the optimization problems, adjusting the analysis techniques for analyzing the search space, in addition to beating large and complex problems by decomposing them into smaller size problems [272]. In another prospective, we can use the ML capabilities in applying automatic configurations of the algorithms by allowing the ML algorithms to choose the appropriate values for the algorithm's operators, especially for metaheuristic algorithms due to a large number of parameters [273].

Shortened the gap between the metaheuristic's algorithms and the problem domain knowledge Treating the problem as a black box is a double-edged weapon. However, this can be considered as a strength of the metaheuristic's algorithms over other algorithms, but it also a challenge. Considering and integrating the domain knowledge of the problem with the designed algorithm will dramatically increase its performance. For example, a problem-orientated research direction can be obtained by designing the algorithm's operator and search mechanisms based on the characteristics of the problem which also can be benefit in reducing the complexity of the algorithm by considering the optimality conditions of the problem being considered [273].

In summary, the following observations from the experiment are:

- Apply the MAs on parallel computing and combine the metaheuristic techniques with the modern parallel computing technologies to generate a powerful method matched with the future generation of computing.
- Exploit the benefits of artificial intelligence and machine learning techniques to provide new algorithms that have the ability to automatically adjust the parameters and automatically analyze the algorithms.
- Developing new methods directed towards strengthens the ability of MAs in addressing the large-scale global optimization (LSGO) problems.
- A great effort must be paid for the hybridization process to allow the algorithms to use the Powers of many algorithms, also generating intelligent techniques that can provide the researcher with insights about what algorithms best suited to be hybridize together?

7.1 Emerging Technologies

After discussing the open issues and challenges, we see that there is much future work in the field of metaheuristics, a set of guidelines must be declared to help the future researcher in the field to address these challenges. In this section, the guidelines used to dive deeper into potential future research directions are introduced. Specifically, we will concentrate on two emerging technologies which are machine learning and quantum computing and how these technologies enhance the optimization process.

7.1.1 Quantum-Inspired Metaheuristics

Metaheuristics can be employed to obtain a global optimal solution for a wide range of different problems in different computational aspects. These methods can benefit from the concept of quantum computing (QC) to enhance the solutions obtained. Hybridizing the quantum computing with the metaheuristics will produce a quantum-inspired metaheuristic algorithm (QIMAs). QIMAs can be considered as an alternative approach to classical optimization methods for solving the optimization algorithm [274]. The main idea behind the QIMAs is to better use the quantum computing principles with the metaheuristics in order to boost the performance of the classical optimization algorithms by scoring a higher-performing results than traditional metaheuristic algorithms. Specifically, the use of QC in metaheuristics will accelerate convergence, enhance exploration, enhance exploitation, and provide a good balance between the two capabilities of the algorithm. The most promising merit that affects the performance of the algorithm is the parallel processing feature in QC [275]. Finally, QIMAs can be used in different disciplines such as engineering and science.

7.1.2 Intelligent Optimization

In this section we will introduce a new type of optimization that is considered as one of the most promising topics in the future of the metaheuristic field. Intelligent optimization (IO) is developed as a test to intelligently adjust the set of inputs and their values to achieve an optimal output(s). In other words, IO cost minimal consumption in determining and choosing the optimal solution among all possible solutions of the problem. The importance of using the IO is dramatically increased when solving complex and NP-hard problems in which the selection of the optimal solution through an exhaustive search is considered impossible or practically difficult. In addition, IO can be used as an important solution for the time-consuming problem of many optimization algorithms. IO can be used in all steps of the optimization process, such as defining the problem, handling, and formulating the objective function(s) and constraints.

7.1.3 Hybrid Metaheuristics and Mathematical Programming

In the last years, hybrid optimization algorithms have achieved promising results compared to classical optimization algorithms. The main aim behind the hybrid metaheuristics is to provide a reliable and high-performance solutions for the large and complex problems. One of the most widely used combinations is hybrid metaheuristics with mathematical programming approaches. This combination will increase the quality of the solution, as it benefits from the two methods in determining an exact solution in a reasonable amount of time. The following points define the mathematical programming approaches that can be used with metaheuristics to increase the quality of the solutions obtained [276].

- Enumerative algorithms: in this approach we can use one of the well-known tree search methods such as dynamic programming and branch and bound. These methods follow the divide-and-conquer strategy where the search space can be divided into smaller search spaces, and then in each

sub area we apply the optimization separately. By applying this strategy, the quality of the solution will increase, and the time consumed will decrease.
- Decomposition and Relaxation methods: in this approach we can decompose the large problem using the Bender's decomposition method or apply the Lagrangian relaxation method to convert the problem into smaller problems.
- Pricing and Cutting plane algorithms: in this approach, we prune using polyhedral combinatorics.

8 Conclusion

In this review, a comprehensive study of metaheuristic algorithms is introduced that involves defining the concept of optimization. Studying the appearance of metaheuristic term. Introducing an explanation of the features of the MAs more than other techniques; Different taxonomies of the MAs according to different aspects such as inspiration source, number of search agents, population updating mechanisms, and number of parameters. Studying the metrics used in the Performance Evaluation of the algorithm. A great effort is paid to clarify the optimization problem in detail, concentrating on different classification techniques, and, moreover, the study reviews the use of metaheuristics in different application areas such as engineering design problems, NP hard problems, medical science, and robotics. Finally, we introduce some of the issues that exist in the MAs literature and the future directions of this important field.

Author Contributions Essam H. Houssein: Supervision, Software, Methodology, Conceptualization, Formal analysis, Investigation, Visualization, Writing—review and editing. Mahmoud Khalaf Saeed: Conceptualization, Visualization, Software, Data curation, Resources, Writing—original draft. Gang Hu: Formal analysis, Resources, Validation, Writing—review and editing. Mustafa M. Al-Sayed: Conceptualization, Formal analysis, Visualization, Resources, Validation, Writing—review and editing. All authors read and approved the final paper.

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

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

Declarations

Conflict of interest The authors declare that there is no conflict of interest.

Ethical Approval This article does not contain studies with human participants or animals carried out by any of the authors.

References

1. Talbi EG (2009) Metaheuristics: from design to implementation. Wiley, Berlin
2. Schneider J, Kirkpatrick S (2007) Stochastic optimization. Springer, Berlin
3. Sörensen K, Sevaux M, Glover F (2018) A history of metaheuristics. In: Handbook of Heuristics. Springer, Berlin, pp 791–808
4. Glover F, Laguna M, Marti R (2003) Scatter search and path relinking: advances and applications. Handbook of metaheuristics. Springer, Berlin, pp 1–35
5. Fister Jr I, Yang X-S, Fister I, Brest J, Fister D (2013) A brief review of nature-inspired algorithms for optimization. arXiv:1307.4186
6. Siddique N, Adeli H (2015) Nature inspired computing: an overview and some future directions. Cogn Comput 7:706–714
7. Molina D, Poyatos J, Del Ser J, García S, Hussain A, Herrera F (2020) Comprehensive taxonomies of nature-and bio-inspired optimization: inspiration versus algorithmic behavior, critical analysis recommendations. Cogn Comput 12:897–939
8. Baykasoğlu A, Ozsoydan FB (2017) Evolutionary and population-based methods versus constructive search strategies in dynamic combinatorial optimization. Inf Sci 420:159–183
9. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks, vol 4, IEEE. pp 1942–1948
10. Mucherino A, Seref O (2007) Monkey search: a novel metaheuristic search for global optimization. In: AIP conference proceedings. American Institute of Physics. pp 162–173
11. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
12. Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. IEEE Comput Intell Mag 1(4):28–39
13. Gandomi AH, Yang X-S, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 29:17–35
14. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98
15. Hashim FA, Houssein EH, Hussain K, Mabrouk MS, Al-Atabany W (2022) Honey badger algorithm: new metaheuristic algorithm for solving optimization problems. Math Comput Simul 192:84–110
16. Holland JH (1992) Genetic algorithms. Sci Am 267(1):66–73
17. Price KV (2013) Differential evolution. Handbook of optimization: from classical to modern approach. Springer, Berlin, pp 187–214
18. Sette S, Boullart L (2001) Genetic programming: principles and applications. Eng Appl Artif Intell 14(6):727–736
19. Khalid AM, Hamza HM, Mirjalili S, Hosny KM (2023) MOCOVIDOA: a novel multi-objective coronavirus disease optimization algorithm for solving multi-objective optimization problems. Neural Comput Appl 1–29

20. Houssein EH, Oliva D, Samee NA, Mahmoud NF, Emam MM (2023) Liver cancer algorithm: a novel bio-inspired optimizer. Comput Biol Med 165:107389
21. Fard AF, Hajiaghaei-Keshteli M (2016) Red deer algorithm (RDA); a new optimization algorithm inspired by red deers' mating. Int Conf Ind Eng 12:331–342
22. Rao RV, Rao V (2016) Teaching-learning-based optimization algorithm. Springer, Berlin
23. Laguna M (2018) Tabu search. Handbook of heuristics. Springer, Berlin, pp 741–758
24. Daskin A, Kais S (2011) Group leaders optimization algorithm. Mol Phys 109(5):761–772
25. Emami H (2022) Stock exchange trading optimization algorithm: a human-inspired method for global optimization. J Supercomput 78(2):2125–2174
26. Xu Y, Cui Z, Zeng J (2010) Social emotional optimization algorithm for nonlinear constrained optimization problems. In: Swarm, evolutionary, and memetic computing: first international conference on swarm, evolutionary, and memetic computing, SEMCCO 2010, Chennai, India, December 16–18, 2010. Proceedings 1, Springer, New York. pp 583–590
27. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: A gravitational search algorithm. Inf Sci 179(13):2232–2248
28. Erol OK, Eksin I (2006) A new optimization method: big bang–big crunch. Adv Eng Softw 37(2):106–111
29. Patel VK, Savsani VJ (2015) Heat transfer search (HTS): a novel optimization algorithm. Inf Sci 324:217–246
30. Hashim FA, Houssein EH, Mabrouk MS, Al-Atabany W, Mirjalili S (2019) Henry gas solubility optimization: a novel physics-based algorithm. Futur Gener Comput Syst 101:646–667
31. Hashim FA, Hussain K, Houssein EH, Mabrouk MS, Al-Atabany W (2021) Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. Appl Intell 51:1531–1551
32. Abdel-Basset M, Mohamed R, Sallam KM, Chakrabortty RK (2022) Light spectrum optimizer: a novel physics-inspired metaheuristic optimization algorithm. Mathematics 10(19):3466
33. Siddique N, Adeli H (2017) Nature-inspired chemical reaction optimisation algorithms. Cogn Comput 9:411–422
34. Abdechiri M, Meybodi MR, Bahrami H (2013) Gases Brownian motion optimization: an algorithm for optimization (GBMO). Appl Soft Comput 13(5):2932–2946
35. Irizarry R (2004) LARES: An artificial chemical process approach for optimization. Evol Comput 12(4):435–459
36. Javidy B, Hatamlou A, Mirjalili S (2015) Ions motion algorithm for solving optimization problems. Appl Soft Comput 32:72–79
37. Kaveh A, Dadras A (2017) A novel meta-heuristic optimization algorithm: thermal exchange optimization. Adv Eng Softw 110:69–84
38. Huang D, Yang J, Xiang D, Xu G (2022) Gaussian swarm optimization: a math-inspired metaheuristic algorithm for solving optimization problems. SSRN 4313360
39. Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. Knowl-Based Syst 96:120–133
40. Houssein EH, Saad MR, Hashim FA, Shaban H, Hassaballah M (2020) Lévy flight distribution: a new metaheuristic algorithm for solving engineering optimization problems. Eng Appl Artif Intell 94:103731
41. Abdel-Basset M, El-Shahat D, Jameel M, Abouhawwash M (2023) Exponential distribution optimizer (EDO): a novel math-inspired algorithm for global optimization and engineering problems. Artif Intell Rev 1–72
42. Tanyildizi E, Demir G (2017) Golden sine algorithm: a novel math-inspired algorithm. Adv Electr Comput Eng 17(2)
43. Yang X-S (2012) Flower pollination algorithm for global optimization. In: International conference on unconventional computing and natural computation. Springer, Berlin. pp 240–249
44. Xing B, Gao W-J, Xing B, Gao W-J (2014) Invasive weed optimization algorithm. In: Innovative computational intelligence: a rough guide to 134 clever algorithms. pp 177–181
45. Premaratne U, Samarabandu J, Sidhu T (2009) A new biologically inspired optimization algorithm. In: 2009 International conference on industrial and information systems (ICIIS). IEEE. pp 279–284
46. Zhao Z, Cui Z, Zeng J, Yue X (2011) Artificial plant optimization algorithm for constrained optimization problems. In: 2011 Second international conference on innovations in bio-inspired computing and applications. IEEE. pp 120–123
47. Cai W, Yang W, Chen X (2008) A global optimization algorithm based on plant growth theory: plant growth optimization. In: 2008 International conference on intelligent computation technology and automation (ICICTA), vol 1. IEEE. pp 1194–1199
48. Zhang H, Zhu Y, Chen H (2014) Root growth model: a novel approach to numerical function optimization and simulation of plant root system. Soft Comput 18:521–537
49. Labbi Y, Attous DB, Gabbar HA, Mahdad B, Zidan A (2016) A new rooted tree optimization algorithm for economic dispatch with valve-point effect. Int J Electr Power Energy Syst 79:298–311
50. Singh PR, Elaziz MA, Xiong S (2019) Ludo game-based metaheuristics for global and engineering optimization. Appl Soft Comput 84:105723
51. Mahmoodabadi MJ, Rasekh M, Zohari T (2018) TGA: team game algorithm. Future Comput Inform J 3(2):191–199
52. Fadakar E, Ebrahimi M (2016) A new metaheuristic football game inspired algorithm. In: 2016 1st Conference on swarm intelligence and evolutionary computation (CSIEC), IEEE. pp 6–11
53. Razmjooy N, Khalilpour M, Ramezani M (2016) A new metaheuristic optimization algorithm inspired by FIFA world cup competitions: theory and its application in PID designing for AVR system. J Control Autom Electr Syst 27:419–440
54. Moosavian N, Roodsari BK (2014) Soccer league competition algorithm: a novel meta-heuristic algorithm for optimal design of water distribution networks. Swarm Evol Comput 17:14–24
55. Kashan AH (2014) League championship algorithm (LCA): an algorithm for global optimization inspired by sport championships. Appl Soft Comput 16:171–200
56. Gao-Wei Y, Zhanju H (2012) A novel atmosphere clouds model optimization algorithm. In: 2012 International conference on computing, measurement, control and sensor network, IEEE. pp 217–220
57. Ghorbani N, Babaei E (2014) Exchange market algorithm. Appl Soft Comput 19:177–187
58. Ahrari A, Atai AA (2010) Grenade explosion method—a novel tool for optimization of multimodal functions. Appl Soft Comput 10(4):1132–1140
59. Savsani P, Savsani V (2016) Passing vehicle search (PVS): a novel metaheuristic algorithm. Appl Math Model 40(5–6):3951–3978
60. Yao P, Gupta SM (2022) Small world optimization algorithm for solving multi-objective u-shaped sequence-dependent disassembly line balancing problem. Small 7(05):15–28
61. Tang H-K, Cai Q, Goh SK (2022) Meta-heuristic optimizer inspired by the philosophy of Yi Jing. Philosophy. https://doi.org/10.21203/rs.3.rs-1259241/v1
62. Askari Q, Younas I, Saeed M (2020) Political optimizer: a novel socio-inspired meta-heuristic for global optimization. Knowl-Based Syst 195:105709

63. Dueck G (1993) New optimization heuristics: The great deluge algorithm and the record-to-record travel. J Comput Phys 104(1):86–92
64. Kirkpatrick S, Gelatt CD Jr, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680
65. Chinnasamy S, Ramachandran M, Amudha M, Ramu K (2022) A review on hill climbing optimization methodology. Recent Trends Manag Commer. https://doi.org/10.46632/rmc/3/1/1
66. Lourenço HR, Martin OC, Stützle T (2003) Iterated local search. Handbook of metaheuristics. Springer, Berlin, pp 320–353
67. Resende MGC, Ribeiro C (1998) Greedy randomized adaptive search procedures (GRASP). AT&T Labs Res Tech Rep 98(1):1–11
68. Rajwar K, Deep K, Das S (2023) An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges. Artif Intell Rev 56:1–71
69. Hatamlou A (2013) Black hole: a new heuristic optimization approach for data clustering. Inf Sci 222:175–184
70. da Luz EFP, Becceneri JC, de Campos Velho HF (2008) A new multi-particle collision algorithm for optimization in a high performance environment. J Comput Interdiscip Sci 1(1):3–10
71. Cheng M-Y, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. Comput Struct 139:98–112
72. Doğan B, Ölmez T (2015) A new metaheuristic for numerical function optimization: vortex search algorithm. Inf Sci 293:125–145
73. Shaheen AM, Ginidi AR, El-Sehiemy RA, Ghoneim SSM (2020) A forensic-based investigation algorithm for parameter extraction of solar cell models. IEEE Access 9:1–20
74. Nematollahi AF, Rahiminejad A, Vahidi B (2017) A novel physical based meta-heuristic optimization method known as lightning attachment procedure optimization. Appl Soft Comput 59:596–621
75. Karaboga D, Basturk B (2007) Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. International fuzzy systems association world congress. Springer, New York, pp 789–798
76. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw 114:163–191
77. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris Hawks optimization: algorithm and applications. Futur Gener Comput Syst 97:849–872
78. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
79. Camacho-Villalón CL, Dorigo M, Stützle T (2023) Exposing the grey wolf, moth-flame, whale, firefly, bat, and antlion algorithms: six misleading optimization techniques inspired by bestial metaphors. Int Trans Oper Res 30(6):2945–2971
80. de Carvalho Filho AO, de Sampaio WB, Silva AC, de Paiva AC, Nunes RA, Gattass M (2014) Automatic detection of solitary lung nodules using quality threshold clustering, genetic algorithm and diversity index. Artif Intell Med 60(3):165–177
81. Wang L, Shen J, Yong J (2012) A survey on bio-inspired algorithms for web service composition. In: Proceedings of the 2012 IEEE 16th international conference on computer supported cooperative work in design (CSCWD), IEEE. pp 569–574
82. Houssein EH, Hosney ME, Oliva D, Mohamed WM, Hassaballah M (2020) A novel hybrid Harris Hawks optimization and support vector machines for drug design and discovery. Comput Chem Eng 133:106656
83. Abbasi A, Firouzi B, Sendur P (2021) On the application of Harris Hawks optimization (HHO) algorithm to the design of microchannel heat sinks. Eng Comput 37:1409–1428
84. Houssein EH, Abohashima Z, Elhoseny M, Mohamed WM (2022) Hybrid quantum-classical convolutional neural network model for COVID-19 prediction using chest X-ray images. J Comput Des Eng 9(2):343–363
85. Houssein EH, Emam MM, Ali AA (2021) Improved manta ray foraging optimization for multi-level thresholding using COVID-19 CT images. Neural Comput Appl 33(24):16899–16919
86. Probert M (2012) Engineering optimisation: an introduction with metaheuristic applications, by Xin-She Yang: scope: reference. Level: undergraduate, postgraduate, early career researcher, researcher, specialist, scientist, engineers.
87. Hoos HH, Stützle T (2004) Stochastic local search: foundations and applications. Elsevier, New York
88. Mockus J, Eddy W, Reklaitis G (2013) Bayesian heuristic approach to discrete and global optimization: algorithms, visualization, software, and applications, vol 17. Springer, Berlin
89. Halstrup M (2016) Black-box optimization of mixed discrete-continuous optimization problems. Black-Box Optim. https://doi.org/10.17877/DE290R-17800
90. Andrei N (2008) An unconstrained optimization test functions collection. Adv Model Optim 10(1):147–161
91. Chen D, Ge Y, Wan Y, Deng Y, Chen Y, Zou F (2022) Poplar optimization algorithm: a new meta-heuristic optimization technique for numerical optimization and image segmentation. Expert Syst Appl 200:117118
92. Kaveh M, Mesgari MS, Saeidian B (2023) Orchard algorithm (OA): a new meta-heuristic algorithm for solving discrete and continuous optimization problems. Math Comput Simul 208:95–135
93. Minh H-L, Sang-To T, Wahab MA, Cuong-Le T (2022) A new metaheuristic optimization based on k-means clustering algorithm and its application to structural damage identification. Knowl-Based Syst 251:109189
94. Srivastava A, Das DK (2022) Criminal search optimization algorithm: a population-based meta-heuristic optimization technique to solve real-world optimization problems. Arab J Sci Eng 47(3):3551–3571
95. Ghasemi M, Zare M, Zahedi A, Akbari M-A, Mirjalili S, Abualigah L (2023) Geyser inspired algorithm: a new geological-inspired meta-heuristic for real-parameter and constrained engineering optimization. J Bionic Eng 1–35
96. Akbari MA, Zare M, Azizipanah-Abarghooee R, Mirjalili S, Deriche M (2022) The cheetah optimizer: a nature-inspired metaheuristic algorithm for large-scale optimization problems. Sci Rep 12(1):10953
97. Feng Z-K, Niu W-J, Liu S (2021) Cooperation search algorithm: a novel metaheuristic evolutionary intelligence algorithm for numerical optimization and engineering optimization problems. Appl Soft Comput 98:106734
98. Das AK, Nikum AK, Krishnan SV, Pratihar DK (2020) Multi-objective bonobo optimizer (MOBO): an intelligent heuristic for multi-criteria optimization. Knowl Inf Syst 62:4407–4444
99. Salgotra R, Singh U, Singh S, Mittal N (2021) A hybridized multi-algorithm strategy for engineering optimization problems. Knowl-Based Syst 217:106790
100. Choo YH, Cai Z, Le V, Johnstone M, Creighton D, Lim CP (2023) Enhancing the Harris' hawk optimiser for single- and multi-objective optimisation. Soft Comput 27(22):16675–16715
101. Dubey K, Sharma SC (2021) A novel multi-objective CR-PSO task scheduling algorithm with deadline constraint in cloud computing. Sustain Comput 32:100605

102. Pereira JLJ, Gomes GF (2023) Multi-objective sunflower optimization: a new hyper-cubic meta-heuristic for constrained engineering problems. Expert Syst e13331
103. Sopov E (2015) A self-configuring metaheuristic for control of multi-strategy evolutionary search. In: Advances in Swarm and computational intelligence: 6th international conference, ICSI 2015 held in conjunction with the Second BRICS Congress, CCI 2015, Beijing, China, June 25–28, 2015, proceedings, Part III 6. Springer 2015:29–37
104. Kumawat IR, Nanda SJ, Maddila RK (2017) Multi-objective whale optimization. In: Tencon 2017–2017 IEEE Region 10 Conference, IEEE. pp 2747–2752
105. Kundu D, Suresh K, Ghosh S, Das S, Panigrahi BK, Das S (2011) Multi-objective optimization with artificial weed colonies. Inf Sci 181(12):2441–2454
106. Khalilpourazari S, Naderi B, Khalilpourazary S (2020) Multi-objective stochastic fractal search: a powerful algorithm for solving complex multi-objective optimization problems. Soft Comput 24:3037–3066
107. Saha S, Mukherjee V (2021) A novel multi-objective modified symbiotic organisms search algorithm for optimal allocation of distributed generation in radial distribution system. Neural Comput Appl 33:1751–1771
108. Abdel-Basset M, Mohamed R, Abouhawwash M (2021) Balanced multi-objective optimization algorithm using improvement based reference points approach. Swarm Evol Comput 60:100791
109. Jain S, Ramesh D, Bhattacharya D (2021) A multi-objective algorithm for crop pattern optimization in agriculture. Appl Soft Comput 112:107772
110. Chakraborty S, Sharma S, Saha AK, Chakraborty S (2021) Shade–WOA: a metaheuristic algorithm for global optimization. Appl Soft Comput 113:107866
111. Zhang Y, Jin Z (2022) Comprehensive learning JAYA algorithm for engineering design optimization problems. J Intell Manuf 1–25
112. López ED, Puris A, Bello RR (2015) VMODE: a hybrid metaheuristic for the solution of large scale optimization problems. Investig Oper 36(3)
113. Shabani A, Asgarian B, Gharebaghi SA, Salido MA, Giret A (2019) A new optimization algorithm based on search and rescue operations. Math Probl Eng 2019:1–23
114. Tuba V, Beko M, Tuba M (2017) Performance of elephant herding optimization algorithm on CEC 2013 real parameter single objective optimization. WSEAS Trans Syst 16:100–105
115. Aydilek IB, Karaçizmeli IH, Tenekeci ME, Kaya S, Gümüşçü A (2021) Using chaos enhanced hybrid firefly particle swarm optimization algorithm for solving continuous optimization problems. Sādhanā 46(2):65
116. Umbarkar AJ, Sheth PD, Chinchanikar AM (2022) Solving IEEE CEC-2013 real parameter optimization problems using harmony search algorithm. In: 2022 3rd international conference for emerging technology (INCET). IEEE 2022:1–7
117. Umbarkar AJ, Moon LR, Sheth PD (2018) Comparative study of CEC'2013 problem using dual population genetic algorithm. Int J Inf Eng Electron Bus 10(5):40–45
118. Wang X, Chu S-C, Pan J-S (2021) Five phases algorithm for global optimization. In: Advances in intelligent information hiding and multimedia signal processing: proceeding of the IIH-MSP 2021 & FITAT 2021, Kaohsiung, Taiwan, vol 1. Springer 2022:81–97
119. Fan J, Li Y, Wang T (2021) An improved African vultures optimization algorithm based on tent chaotic mapping and time-varying mechanism. PLoS ONE 16(11):e0260725
120. Zitouni F, Harous S, Maamri R (2020) The solar system algorithm: a novel metaheuristic method for global optimization. IEEE Access 9:4542–4565
121. Abdel-Basset M, El-Shahat D, Jameel M, Abouhawwash M (2023) Young's double-slit experiment optimizer: a novel metaheuristic optimization algorithm for global and constraint optimization problems. Comput Methods Appl Mech Eng 403:115652
122. Zhang J, Xiao M, Gao L, Pan Q (2018) Queuing search algorithm: a novel metaheuristic algorithm for solving engineering optimization problems. Appl Math Model 63:464–490
123. Tolabi HB, Ara AL, Hosseini R (2021) An enhanced particle swarm optimization algorithm to solve probabilistic load flow problem in a micro-grid. Appl Intell 51:1645–1668
124. Abdel-Basset M, Mohamed R, Jameel M, Abouhawwash M (2023) Nutcracker optimizer: a novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems. Knowl-Based Syst 262:110248
125. Abdel-Basset M, Mohamed R, Azeem SAA, Jameel M, Abouhawwash M (2023) Kepler optimization algorithm: a new metaheuristic algorithm inspired by Kepler's laws of planetary motion. Knowl-Based Syst 268:110454
126. Ibrahim Z, Aziz NHA, Aziz NAA, Razali S, Mohamad MS (2016) Simulated Kalman filter: a novel estimation-based metaheuristic optimization algorithm. Adv Sci Lett 22(10):2941–2946
127. Abedinpourshotorban H, Shamsuddin SM, Beheshti Z, Jawawi DNA (2016) Electromagnetic field optimization: a physics-inspired metaheuristic optimization algorithm. Swarm Evol Comput 26:8–22
128. Caraveo C, Valdez F, Castillo O (2018) A new optimization metaheuristic based on the self-defense techniques of natural plants applied to the CEC 2015 benchmark functions. In: Advances in fuzzy logic and technology 2017: proceedings of: EUSFLAT-2017—the 10th conference of the European Society for Fuzzy Logic and Technology, September 11–15, 2017, Warsaw, Poland IWIFSGN'2017—the sixteenth international workshop on intuitionistic fuzzy sets and generalized nets, September 13–15, 2017, Warsaw, Poland, vol 1, p 10. Springer
129. Sattar D, Salim R (2021) A smart metaheuristic algorithm for solving engineering problems. Eng Comput 37(3):2389–2417
130. Ochoa P, Castillo O, Soria J (2015) Differential evolution algorithm using a dynamic crossover parameter with fuzzy logic applied for the CEC 2015 benchmark functions. In: Fuzzy information processing: 37th conference of the North American Fuzzy Information Processing Society, NAFIPS 2018, Fortaleza, Brazil, July 4–6, 2018, Proceedings 37. Springer 2018:580–591
131. Thapliyal S, Kumar N (2023) Numeric crunch algorithm: a new metaheuristic algorithm for solving global and engineering optimization problems. Soft Comput 27(22):1–47
132. Trojovský P, Dehghani M (2023) A new bio-inspired metaheuristic algorithm for solving optimization problems based on walruses behavior. Sci Rep 13(1):8775
133. Zhang Y, Chi A (2021) Group teaching optimization algorithm with information sharing for numerical optimization and engineering optimization. J Intell Manuf 34(4):1–25
134. Zhang Y, Chi A, Mirjalili S (2021) Enhanced JAYA algorithm: a simple but efficient optimization method for constrained engineering design problems. Knowl-Based Syst 233:107555
135. Zhao J, Tang D, Liu Z, Cai Y, Dong S (2020) Spherical search optimizer: a simple yet efficient meta-heuristic approach. Neural Comput Appl 32:9777–9808
136. Trojovská E, Dehghani M (2022) Clouded leopard optimization: a new nature-inspired optimization algorithm. IEEE Access 10:102876–102906
137. Faramarzi A, Heidarinejad M, Mirjalili S, Gandomi AH (2020) Marine predators algorithm: A nature-inspired metaheuristic. Expert Syst Appl 152:113377

138. Dhiman G (2021) ESA: a hybrid bio-inspired metaheuristic optimization approach for engineering problems. Eng Comput 37:323–353
139. Azizi M (2021) Atomic orbital search: a novel metaheuristic algorithm. Appl Math Model 93:657–683
140. Talatahari S, Azizi M (2021) Chaos game optimization: a novel metaheuristic algorithm. Artif Intell Rev 54:917–1004
141. Dehghani M, Montazeri Z, Trojovská E, Trojovský P (2023) Coati optimization algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems. Knowl-Based Syst 259:110011
142. Matoušová I, Trojovský P, Dehghani M, Trojovská E, Kostra J (2023) Mother optimization algorithm: a new human-based metaheuristic approach for solving engineering optimization. Sci Rep 13(1):10312
143. Dehghani M, Trojovská E, Trojovský P, Malik OP (2023) OOBO: a new metaheuristic algorithm for solving optimization problems. Biomimetics 8(6):468
144. Naruei I, Keynia F (2022) Wild horse optimizer: a new meta-heuristic algorithm for solving engineering optimization problems. Eng Comput 38(suppl. 4):3025–3056
145. Fritsche G, Pozo A (2019) Cooperative based hyper-heuristic for many-objective optimization. In: Proceedings of the genetic and evolutionary computation conference. pp 550–558
146. Nadimi-Shahraki MH, Fatahi A, Zamani H, Mirjalili S, Abualigah L (2021) An improved moth-flame optimization algorithm with adaptation mechanism to solve numerical and mechanical engineering problems. Entropy 23(12):1637
147. Nadimi-Shahraki MH, Taghian S, Mirjalili S, Faris H (2020) MTDE: an effective multi-trial vector-based differential evolution algorithm and its applications for engineering design problems. Appl Soft Comput 97:106761
148. Nadimi-Shahraki MH (2023) An effective hybridization of quantum-based avian navigation and bonobo optimizers to solve numerical and mechanical engineering problems. J Bionic Eng 20(3):1361–1385
149. Nadimi-Shahraki MH, Zamani H, Fatahi A, Mirjalili S (2023) MFO-SFR: an enhanced moth-flame optimization algorithm using an effective stagnation finding and replacing strategy. Mathematics 11(4):862
150. Nadimi-Shahraki M-H, Taghian S, Zamani H, Mirjalili S, Abd Elaziz M (2023) MMKE: multi-trial vector-based monkey king evolution algorithm and its applications for engineering optimization problems. PLOS ONE 18(1):e0280006
151. Stanovov V, Akhmedova S, Semenkin E (2018) Lshade algorithm with rank-based selective pressure strategy for solving CEC 2017 benchmark problems. In: 2018 IEEE congress on evolutionary computation (CEC). IEEE. pp 1–8
152. Rueda J, Erlich I (2018) Hybrid population based MVMO for solving CEC 2018 test bed of single-objective problems. In: 2018 IEEE congress on evolutionary computation (CEC). IEEE. pp 1–8
153. Zhang Y, Wang G-G, Li K, Yeh W-C, Jian M, Dong J (2020) Enhancing MOEA/D with information feedback models for large-scale many-objective optimization. Inf Sci 522:1–16
154. ALRahhal H, Jamous R (2023) Afox: a new adaptive nature-inspired optimization algorithm. Artif Intell Rev 56:1–44
155. Demirci H, Yurtay N, Yurtay Y, Zaimoğlu EA (2023) Electrical search algorithm: a new metaheuristic algorithm for clustering problem. Arab J Sci Eng 48(8):10153–10172
156. Barua S, Merabet A (2023) Lévy arithmetic algorithm: an enhanced metaheuristic algorithm and its application to engineering optimization. Expert Syst Appl 241:122335
157. Trojovský P, Dehghani M (2022) A new optimization algorithm based on mimicking the voting process for leader selection. PeerJ Comput Sci 8:e976
158. Majumdar P, Bhattacharya D, Mitra S, Rodrigues LR, Oliva D (2020) An improved binary grey wolf optimizer for constrained engineering design problems. Expert Syst 41:e13458
159. Guan Z, Cao Z, Wang P et al (2023) Water and salt transport optimization: new metaheuristic algorithm for engineer design. vol 2
160. Dehghani M, Trojovský P (2022) Serval optimization algorithm: a new bio-inspired approach for solving optimization problems. Biomimetics 7(4):204
161. Ahmed AM, Rashid TA, Saeed SAM (2021) Dynamic cat swarm optimization algorithm for back-board wiring problem. Neural Comput Appl 33(20):13981–13997
162. Rezvani K, Gaffari A, Dishabi MRE (2023) The bedbug meta-heuristic algorithm to solve optimization problems. J Bionic Eng 1–21:2465–2485
163. Azizi M, Talatahari S, Gandomi AH (2023) Fire hawk optimizer: a novel metaheuristic algorithm. Artif Intell Rev 56(1):287–363
164. Duman S, Kahraman HT, Sonmez Y, Guvenc U, Kati M, Aras S (2022) A powerful meta-heuristic search algorithm for solving global optimization and real-world solar photovoltaic parameter estimation problems. Eng Appl Artif Intell 111:104763
165. Zitouni F, Harous S, Belkeram A, Hammou LEB (2022) The archerfish hunting optimizer: a novel metaheuristic algorithm for global optimization. Arab J Sci Eng 47(2):2513–2553
166. Singh N, Houssein EH, Singh SB, Dhiman G (2023) Hssahho: a novel hybrid salp swarm-harris hawks optimization algorithm for complex engineering problems. J Ambient Intell Humaniz Comput 14(9):11569–11605
167. Azizi M, Aickelin U, Khorshidi HA, Shishehgarkhaneh MB (2023) Energy valley optimizer: a novel metaheuristic algorithm for global and engineering optimization. Sci Rep 13(1):226
168. Azizi M, Shishehgarkhaneh MB, Basiri M, Moehler RC (2023) Squid game optimizer (SGO): a novel metaheuristic algorithm. Sci Rep 13(1):5373
169. Wang Y, Wang Z, Wang G-G (2023) Hierarchical learning particle swarm optimization using fuzzy logic. Expert Syst Appl 120759
170. Han M, Du Z, Yuen K, Zhu H, Li Y, Yuan Q (2023) Walrus optimizer: a novel nature-inspired metaheuristic algorithm. Expert Syst Appl 122413
171. Zheng R, Jia H, Abualigah L, Wang S, Wu D (2022) An improved remora optimization algorithm with autonomous foraging mechanism for global optimization problems. Math Biosci Eng 19:3994–4037
172. Shen Q, Zhang D, Xie M, He Q (2023) Multi-strategy enhanced dung beetle optimizer and its application in three-dimensional UAV path planning. Symmetry 15(7):1432
173. Kadavy T, Pluhacek M, Viktorin A, Senkerik R (2021) Self-organizing migrating algorithm with clustering-aided migration and adaptive perturbation vector control. In: Proceedings of the genetic and evolutionary computation conference companion. pp 1916–1922
174. Brest J, Maučec MS, Bošković B (2021) Self-adaptive differential evolution algorithm with population size reduction for single objective bound-constrained optimization: Algorithm j21. In: 2021 IEEE congress on evolutionary computation (CEC). IEEE. pp 817–824
175. Setiawan D, Suyanto S, Erfianto B, Gozali AA (2023) Battlefield optimization algorithm. SSRN 4585054
176. Zhang H, Zhang Y, Niu Y, He K, Wang Y (2023) T cell immune algorithm: a novel nature-inspired algorithm for engineering applications. IEEE Access. https://doi.org/10.1109/ACCESS.2023.3311271
177. Salgotra R, Singh S, Singh U, Kundu K, Gandomi AH (2022) An adaptive version of differential evolution for solving CEC2014, CEC 2017 and CEC 2022 test suites. In: 2022 IEEE

symposium series on computational intelligence (SSCI). IEEE. pp 1644–1649
178. Houssein EH, Saeed MK, Al-Sayed MM (2023) Ewso: boosting white shark optimizer for solving engineering design and combinatorial problems. Math Comput Simul. https://doi.org/10.1016/j.matcom.2023.11.019
179. Chen S, Zheng J (2023) Sand cat arithmetic optimization algorithm for global optimization engineering design problems. J Comput Des Eng 10(6):2122–2146
180. Ferahtia S, Rezk H, Djerioui A, Houari A, Motahhir S, Zeghlache S (2023) Modified bald eagle search algorithm for lithium-ion battery model parameters extraction. ISA Trans 134:357–379
181. Ferahtia S, Houari A, Rezk H, Djerioui A, Machmoum M, Motahhir S, Ait-Ahmed M (2023) Red-tailed hawk algorithm for numerical optimization and real-world problems. Sci Rep 13(1):12950
182. Gupta S, Deep K, Moayedi H, Foong LK, Assad A (2021) Sine cosine grey wolf optimizer to solve engineering design problems. Eng Comput 37:3123–3149
183. Sulaiman MH, Mustaffa Z, Saari MM, Daniyal H (2020) Barnacles mating optimizer: a new bio-inspired algorithm for solving engineering optimization problems. Eng Appl Artif Intell 87:103330
184. Kaur S, Awasthi LK, Sangal AL, Dhiman G (2020) Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. Eng Appl Artif Intell 90:103541
185. Lei M, Zhou Y, Luo Q (2019) Enhanced metaheuristic optimization: wind-driven flower pollination algorithm. IEEE Access 7:111439–111465
186. Li S, Chen H, Wang M, Heidari AA, Mirjalili S (2020) Slime mould algorithm: a new method for stochastic optimization. Futur Gener Comput Syst 111:300–323
187. Rather SA, Bala PS (2020) Swarm-based chaotic gravitational search algorithm for solving mechanical engineering design problems. World J Eng 17(1):97–114
188. Wang L, Cao Q, Zhang Z, Mirjalili S, Zhao W (2022) Artificial rabbits optimization: a new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. Eng Appl Artif Intell 114:105082
189. Abualigah L, Shehab M, Alshinwan M, Mirjalili S, Elaziz MA (2021) Ant lion optimizer: a comprehensive survey of its variants and applications. Archiv Comput Methods Eng 28:1397–1416
190. Shu L, Jiang P, Zhou Q, Xie T (2019) An online variable-fidelity optimization approach for multi-objective design optimization. Struct Multidiscip Optim 60:1059–1077
191. Dhiman G, Garg M (2020) Mosse: a novel hybrid multi-objective meta-heuristic algorithm for engineering design problems. Soft Comput 24(24):18379–18398
192. Tejani GG, Pholdee N, Bureerat S, Prayogo D, Gandomi AH (2019) Structural optimization using multi-objective modified adaptive symbiotic organisms search. Expert Syst Appl 125:425–441
193. Tawhid MA, Savsani V (2019) Multi-objective sine-cosine algorithm (MO-SCA) for multi-objective engineering design problems. Neural Comput Appl 31:915–929
194. Jangir P, Buch H, Mirjalili S, Manoharan P (2023) Mompa: multi-objective marine predator algorithm for solving multi-objective optimization problems. Evol Intel 16(1):169–195
195. Sahni S, Gonzalez T (1976) P-complete approximation problems. J ACM 23(3):555–565
196. Lawler EL (1975) The quadratic assignment problem: a brief review. In: Combinatorial programming: methods and applications: proceedings of the NATO advanced study institute held at the Palais des Congres, Versailles, France, 2–13 September, pp 351–360
197. Pitsoulis L (2001) Quadratic assignment problem. Quadratic assignment problem, ch 3. Springer, Boston, pp 2075–2107
198. Fathollahi-Fard AM, Wong KY, Aljuaid M (2023) An efficient adaptive large neighborhood search algorithm based on heuristics and reformulations for the generalized quadratic assignment problem. Eng Appl Artif Intell 126:106802
199. Agharghor A, Riffi ME, Chebihi F (2019) Improved hunting search algorithm for the quadratic assignment problem. Indonesian J Electr Eng Comput Sci 14(1):143–154
200. Shah S (2020) Implementation of iterative local search (ILS) for the quadratic assignment problem. Authorea Preprints
201. Dokeroglu T (2015) Hybrid teaching–learning-based optimization algorithms for the quadratic assignment problem. Comput Ind Eng 85:86–101
202. Lv C, Zhao H, Yang X (2011) Particle swarm optimization algorithm for quadratic assignment problem. In: Proceedings of 2011 international conference on computer science and network technology. IEEE. vol 3, pp 1728–1731
203. Ahuja RK, Orlin JB, Tiwari A (2000) A greedy genetic algorithm for the quadratic assignment problem. Comput Oper Res 27(10):917–934
204. Dokeroglu T, Sevinc E, Cosar A (2019) Artificial bee colony optimization for the quadratic assignment problem. Appl Soft Comput 76:595–606
205. Mouhoub M, Wang Z (2008) Improving the ant colony optimization algorithm for the quadratic assignment problem. In: 2008 IEEE congress on evolutionary computation (IEEE World congress on computational intelligence). IEEE. pp 250–257
206. Tate DM, Smith AE (1995) A genetic approach to the quadratic assignment problem. Comput Oper Res 22(1):73–83
207. Kılıç H, Yüzgeç U (2019) Tournament selection based antlion optimization algorithm for solving quadratic assignment problem. Eng Sci Technol Int J 22(2):673–691
208. Lim WL, Wibowo A, Desa MI, Haron H (2016) A biogeography-based optimization algorithm hybridized with tabu search for the quadratic assignment problem. Comput Intell Neurosci 2016:27–27
209. Duman E, Uysal M, Alkaya AF (2012) Migrating birds optimization: a new metaheuristic approach and its performance on quadratic assignment problem. Inf Sci 217:65–77
210. Steinberg L (1961) The backboard wiring problem: a placement algorithm. SIAM Rev 3(1):37–50
211. Lee C-G, Ma Z (2004) The generalized quadratic assignment problem. Research Report, Dept. Mechanical Industrial Eng., Univ. Toronto, Canada, p. M5S
212. Lawler EL (1963) The quadratic assignment problem. Manage Sci 9(4):586–599
213. Greenberg H (1969) A quadratic assignment problem without column constraints. Naval Res Logist Q 16(3):417–421
214. Hahn P, Smith JM, Zhu Y-R (2010) The multi-story space assignment problem. Ann Oper Res 179:77–103
215. Hahn PM et al (2008) The quadratic three-dimensional assignment problem: exact and approximate solution methods. Eur J Oper Res 184(2):416–428
216. Knowles J, Corne D (2003) Instance generators and test suites for the multiobjective quadratic assignment problem. International conference on evolutionary multi-criterion optimization. Springer, Berlin, pp 295–310
217. Punnen AP, Wang Y (2016) The bipartite quadratic assignment problem and extensions. Eur J Oper Res 250(3):715–725
218. Gary MR, Johnson DS (1979) Computers and intractability: a guide to the theory of NP-completeness. WH Freeman and Company, New York
219. Zendaoui Z, Layeb A (2016) Adaptive cuckoo search algorithm for the bin packing problem. Modeling and implementation of complex systems. Springer, Berlin, pp 107–120

220. Scholl A, Klein R (2007) Bin packing. http://www.wiwi.uni-jena.de/Entscheidung/binpp/. Accessed Oct 2007
221. El-Ashmawi WH, Elminnaam DSA (2019) A modified squirrel search algorithm based on improved best fit heuristic and operator strategy for bin packing problem. Appl Soft Comput 82:105565
222. Zhao F et al (2021) A novel cooperative multi-stage hyper-heuristic for combination optimization problems. Complex Syst Model Simul 1(2):91–108
223. Chen M et al (2019) An efficient deterministic heuristic algorithm for the rectangular packing problem. Comput Ind Eng 137:106097
224. Li X, Zhao Z, Zhang K (2014) A genetic algorithm for the three-dimensional bin packing problem with heterogeneous bins. In: IIE annual conference. proceedings. Institute of Industrial and Systems Engineers (IISE), p 2039
225. Adamuthe A, Nitave T (2021) Optimizing large scale bin packing problem with hybrid harmony search algorithm. Int J Ind Eng Comput 12(2):205–220
226. Blum C, Schmid V (2013) Solving the 2D bin packing problem by means of a hybrid evolutionary algorithm. Procedia Comput Sci 18:899–908
227. Feng H et al (2020) An enhanced grasshopper optimization algorithm to the bin packing problem. J Control Sci Eng 2020:1–19
228. Ceselli A, Righini G (2008) An optimization algorithm for the ordered open-end bin-packing problem. Oper Res 56(2):425–436
229. Tam S-C et al (2010) A new optimization method, the algorithm of changes, for bin packing problem. In: 2010 IEEE fifth international conference on bio-inspired computing: theories and applications (BIC-TA). IEEE, pp 994–999
230. Kang J, Park S (2003) Algorithms for the variable sized bin packing problem. Eur J Oper Res 147(2):365–372
231. Abdel-Basset M et al (2018) An improved nature inspired meta-heuristic algorithm for 1-D bin packing problems. Pers Ubiquit Comput 22:1117–1132
232. Liu DS et al (2008) On solving multiobjective bin packing problems using evolutionary particle swarm optimization. Eur J Oper Res 190(2):357–382
233. Panwar K, Deep K (2021) Discrete grey wolf optimizer for symmetric travelling salesman problem. Appl Soft Comput 105:107298
234. Jozefowiez N, Semet F, Talbi E-G (2008) Multi-objective vehicle routing problems. Eur J Oper Res 189(2):293–309
235. Kahraman C (2006) Metaheuristic techniques for job shop scheduling problem and a fuzzy ant colony optimization algorithm. Fuzzy applications in industrial engineering. Springer, Berlin, pp 401–425
236. Liu SQ, Ong HL (2004) Metaheuristics for the mixed shop scheduling problem. Asia-Pac J Oper Res 21(1):97–115
237. Mejtsky GJ (2007) A metaheuristic algorithm for simultaneous simulation optimization and applications to traveling salesman and job shop scheduling with due dates. In: 2007 winter simulation conference. IEEE, pp 1835–1843
238. Mejtsky GJ (2008) The improved sweep metaheuristic for simulation optimization and application to job shop scheduling. In: 2008 winter simulation conference. IEEE, pp 731–739
239. Gudmundsson M, El-Kwae EA, Kabuka MR (1998) Edge detection in medical images using a genetic algorithm. IEEE Trans Med Imaging 17(3):469–474
240. Pereira DC, Ramos RP, Do Nascimento MZ (2014) Segmentation and detection of breast cancer in mammograms combining wavelet analysis and genetic algorithm. Comput Methods Program Biomed 114(1):88–101
241. Sahiner B et al (1996) Image feature selection by a genetic algorithm: application to classification of mass and normal breast tissue. Med Phys 23(10):1671–1684
242. Wu W-J, Lin S-W, Moon WK (2012) Combining support vector machine with genetic algorithm to classify ultrasound breast tumor images. Comput Med Imaging Graph 36(8):627–633
243. Nguyen BL (2014) Non-invasive detection of hypoglycemia in patients with type 1 diabetes using electroencephalography signals. PhD thesis, etection of hypo
244. Suganthi M, Madheswaran M (2012) An improved medical decision support system to identify the breast cancer using mammogram. J Med Syst 36:79–91
245. Kockanat S, Karaboga N, Koza T (2012) Image denoising with 2-D FIR filter by using artificial bee colony algorithm. In: 2012 International symposium on innovations in intelligent systems and applications. pp 1–4. IEEE
246. Pham DT, Castellani M, Fahmy A (2008) Learning the inverse kinematics of a robot manipulator using the bees algorithm.
247. Yang Y et al (2007) A new solution for inverse kinematics of 7-DOF manipulator based on genetic algorithm. In: 2007 IEEE international conference on automation and logistics. pp. 1947–1951. IEEE
248. Huang H-C, Chen C-P, Wang P-R (2012) Particle swarm optimization for solving the inverse kinematics of 7-DOF robotic manipulators. In: 2012 IEEE international conference on systems, man, and cybernetics (SMC). pp 3105–3110
249. Ren Z-W, Wang Z-H, Sun L-N (2015) A hybrid biogeography-based optimization method for the inverse kinematics problem of an 8-DOF redundant humanoid manipulator. Front Inf Technol Electron Eng 16(7):607–616
250. Doering J, Juan AA, Kizys R, Fito A, Calvet L (2016) Solving realistic portfolio optimization problems via metaheuristics: a survey and an example. In: Modeling and simulation in engineering, economics and management: international conference, MS 2016, Teruel, Spain, July 4–5, pp 22–30. Springer, Berlin
251. Thomaidis NS, Angelidis T, Vassiliadis V, Dounias G (2009) Active portfolio management with cardinality constraints: an application of particle swarm optimization. N Math Nat Comput 5(03):535–555
252. Derigs U, Nickel N-H (2003) Meta-heuristic based decision support for portfolio optimization with a case study on tracking error minimization in passive portfolio management. OR Spectrum 25:345–378
253. Rafaely B, Bennell JA (2006) Optimisation of FTSE 100 tracker funds: a comparison of genetic algorithms and quadratic programming. Manag Financ 32(6):477–492
254. Oh KJ, Kim TY, Min S (2005) Using genetic algorithm to support portfolio optimization for index fund management. Expert Syst Appl 28(2):371–379
255. Gilli M, Schumann E (2011) Calibrating option pricing models with heuristics. Natural computing in computational finance, vol 4. Springer, New York, pp 9–37
256. Kumar S, Thulasiram RK, Thulasiraman P (2008) A bioinspired algorithm to price options. In: Proceedings of the 2008 C3S2E conference, pp 11–22
257. Kim S-H, Chang K-N, Kim S (2000) A channel allocation for cellular mobile radio systems using simulated annealing. Telecommun Syst 14:95–106
258. Amaldi E, Capone A, Malucelli F (2003) Planning UMTS base station location: optimization models with power control and algorithms. IEEE Trans Wirel Commun 2(5):939–952
259. Bagloee SA et al (2013) A hybrid meta-heuristic algorithm for solving real-life transportation network design problems. Int J Logist Syst Manag 16(1):41–66
260. Dhouib S (2021) Hybrid metaheuristic to optimize traceability in the food industry. Int J Strateg Eng 4(2):14–27
261. Katiyar S, Khan R, Kumar S (2021) Artificial bee colony algorithm for fresh food distribution without quality loss by delivery route optimization. J Food Qual 2021:1–9

262. Seydanlou P et al (2022) A multi-objective optimization framework for a sustainable closed-loop supply chain network in the olive industry: hybrid meta-heuristic algorithms. Expert Syst Appl 203:117566
263. Blum C, Roli A (2003) Metaheuristics in combinatorial optimization: overview and conceptual comparison. ACM Comput Surv 35(3):268–308
264. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82
265. Auger A, Doerr B (2011) Theory of randomized search heuristics: foundations and recent developments. Algorithmica 60(1):1–26
266. Auger A, Teytaud O (2010) Continuous lunches are free plus the design of optimal optimization algorithms. Algorithmica 57(1):121–146
267. Yang X-S (2020) Nature-inspired optimization algorithms: challenges and open problems. J Comput Sci 46:101104
268. Eiben AE, Smit SK (2011) Parameter tuning for configuring and analyzing evolutionary algorithms. Swarm Evol Comput 1(1):19–31
269. Mohamed AW, Mohamed AK (2019) Adaptive guided differential evolution algorithm with novel mutation for numerical optimization. Int J Mach Learn Cybern 10:253–277
270. Yang X-S, Karamanoglu M, He X (2014) Flower pollination algorithm: a novel approach for multiobjective optimization. Eng Optim 46(9):1222–1237
271. Karimi-Mamaghan M et al (2022) Machine learning at the service of metaheuristics for solving combinatorial optimization problems: a state-of-the-art. Eur J Oper Res 296(2):393–422
272. Ma J et al (2022) A comprehensive comparison among metaheuristics (MHs) for geo-hazard modeling using machine learning: insights from a case study of landslide displacement prediction. Eng Appl Artif Intell 114:105150
273. Chou J-S, Nguyen T-K (2018) Forward forecast of stock price using sliding-window metaheuristic-optimized machine-learning regression. IEEE Trans Industr Inf 14(7):3132–3142
274. Karmakar S, Dey A, Saha I (2017) Use of quantum-inspired metaheuristics during last two decades. In: 2017 7th International conference on communication systems and network technologies (CSNT). IEEE. pp 272–278
275. Gharehchopogh FS (2023) Quantum-inspired metaheuristic algorithms: comprehensive survey and classification. Artif Intell Rev 56(6):5479–5543
276. Talbi E-G (2016) Combining metaheuristics with mathematical programming, constraint programming and machine learning. Ann Oper Res 240(1):171–215