



Paper Type: Original Article

An Overview of Metaheuristic and Hyper-Heuristic Algorithms

Loghman Rashidi*

Department of Computer Engineering, Imam Reza International University, Mashhad, Iran; loghman.rashidy67@gmail.com.

Citation:

Received: 20 October 2024
Revised: 14 January 2025
Accepted: 12 March 2025

Rashidi, L. (2025). An overview of metaheuristic and hyper-heuristic algorithms. *Metaheuristic algorithms with applications*, 2(2), 191-208.

Abstract

A group of algorithms used to solve NP-hard problems is called a metaheuristic or hyper-heuristic. Problems with a large number of answers and that take too long to find the best one are. The use of metaheuristic & hyper-heuristic algorithms to solve difficult problems yields acceptable solutions in a short time. These methods fall into the category of optimization algorithms. In optimization algorithms, problems that do not have a definite solution converge to an optimal solution in a very short time. Various algorithms have been introduced so far that are based on the intelligence of the events around us. Each of these methods has been used to solve complex problems that have not yielded acceptable results from heuristic algorithms. According to the National Football League (NFL) theory, none of the algorithms can solve all problems. Each of these algorithms produces more optimal solutions to specific problems than the others. For this reason, efforts to design new methods continue to address a broader range of issues. This article examines new metaheuristic algorithms and their classification. Many metaheuristic algorithms have been introduced today, each capable of achieving optimal solutions to specific problems. This potential, along with new techniques and Machine Learning (ML), has led to the production of a new generation of these algorithms, known as hyper-heuristic algorithms. These types of algorithms aim to create hybrid algorithms that solve more problems with a single

Keywords: NP-hard, Metaheuristic algorithms, Hyper-heuristic algorithms, Machine learning, optimization.

1 | Introduction

In various fields of engineering, the basic sciences, medicine, and even the humanities, some problems cannot be solved with mathematical methods and classical algorithms within a tolerable time. These issues fall into the category of hardships. The only way to solve these problems so far is through an innovative algorithm known as a metaheuristic or hyper-heuristic. Metaheuristic and hyper-heuristic algorithms, by varying heuristics, explore the vast search space to find an optimal solution. The primary purpose of designing these

✉ Corresponding Author: loghman.rashidy67@gmail.com

doi <https://doi.org/10.48313/maa.v2i2.45>



Licensee System Analytics. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0>).

algorithms is to achieve the best possible solution from a vast area. Today, metaheuristic and hyper-heuristic algorithms are widely used in various applications.

In most cases, metaheuristic and hyper-heuristic algorithms give the desired answer. So far, many algorithms have been proposed in this field. There are two main reasons for the proliferation of metaheuristic algorithms. The first and most important reason is that each metaheuristic and hyper-heuristic algorithm is efficient in a specific area of critical problems. It means that one or more algorithms provide optimal solutions for each problem. The second reason is the widespread source of inspiration for metaheuristic and hyper-heuristic algorithms. The nature and events of life inspire these algorithms, and this vast resource contains many ideas for new algorithms. Any phenomenon that has an intelligent initiative is a source of inspiration for the new algorithm. In general, metaheuristic and hyper-heuristic algorithms are divided into four categories: evolutionary, physical, human, and swarm algorithms, each of which refers to the source of inspiration for its sub-algorithms. While their metaheuristic algorithms are compelling, they have a significant drawback. This argument is the disadvantage of not solving all problems with a particular algorithm. That is, each algorithm solves only a limited number of problems well. Today, metaheuristic and hyper-heuristic algorithms are combined with Machine Learning (ML), and by combining their capabilities, they try to solve a wider range of problems. With the increasing growth of science in various fields, the need for metaheuristic and hyper-heuristic algorithms is felt more than ever. The primary origins of these algorithms lie in artificial intelligence and operations research [1], [2]. The larger the problem data, the more difficult it is to solve. As we know, there is no solution to polynomial-time problems; they are called hard [3]. So far, no algorithm has been introduced that can promptly provide a definitive answer to complex problems [4]. Optimization methods are an excellent solution to solve these problems [5]. These methods can find optimal or near-optimal answers to solve difficult problems in a short time [6]. Optimization methods play an essential role in industry, scientific development, management, and problem-solving that can be modeled in this field.

Multidimensional, discontinuous models and noise-containing data cannot be solved by traditional methods [7]. In general, optimization methods are divided into two categories: single-objective and multi-objective. There are usually compromises between multi-objective methods. In other words, improving one goal is likely to weaken at least one other goal [8]. Large-scale Optimization (LSGO) refers to optimization problems with a large number of variables and many decision-making conditions. Many of the problems that exist today across various engineering and planning fields can be categorized as LSGOs that require new solutions [9]. For example, inversion problems in biological systems and industrial design as a decision-making problem are among the most time-consuming and significant problems in the category of very complex problems [9], [10].

Metaheuristic and hyper-heuristic algorithms are used to approximate optimal solutions to a wide range of problems, including engineering, basic sciences, medicine, and other disciplines concerned with solving them. These algorithms are more capable than innovations. This essential feature of avoiding falling into the local optimum is the most critical difference [11]. The main problem with innovative algorithms is the production of local, non-optimal (weak) answers [12]. Historically, metaheuristic and hyper-heuristic algorithms have been classified into both classical and modern categories. The algorithms of the 90s and earlier are called classical, and those of the 2000s and later are called modern [13]. Also, "metaheuristic" algorithms are divided into two types of population-based path-axis based on the kind of response they generate [14], [15]. Implementing these algorithms and extending them to different problems across different scientific disciplines are other advantages of this type of algorithm. If the right strategy is adopted to address the issues and the correct parameters are selected, these algorithms will yield the desired answer. The value of this becomes even more striking when we know that, in robust problems, as the number of parameters increases, response time increases exponentially, which, even with the most powerful computers, may take several years to obtain. Metaheuristic and hyper-heuristic algorithms solve this problem and are an excellent alternative to traditional techniques. Metaheuristic and hyper-heuristic algorithms are a good choice for optimizing robust issues [16]. Four essential features make metaheuristic and hyper-heuristic algorithms very popular: 1) they have a simple idea and implementation, 2) they are flexible; that is, they quickly change shape for all issues, 3)

they act as a black box, so these algorithms' input and output are essential, and 4) they do not fall into the trap of local optimism [17]. The fourth case, which is the advantage of metaheuristic and hyper-heuristic over innovations, is due to the possibility of proper adjustment between two essential factors in them. The exploration and exploitation phases are two critical components of these algorithms [16]. The intensity phase seeks the optimal answer in the existing candidates, and the exploration phase seeks the solution in the spaces that have not been searched so far. The metaheuristic and hyper-heuristic algorithm include iterative and production processes associated with internal learning [18].

The power of these algorithms comes from abstracting the features of nature in the simplest possible way. It draws on millions of years of natural selection and adaptation to the environment in the context of the problem [19]. The primary source of the creation of metaheuristic and hyper-heuristic algorithms is nature. Among the theories that inspired them are evolution, physical phenomena, social phenomena, and collective intelligence, which will be explained in the next section. These algorithms follow anything that can be achieved with a new algorithm, including music, agriculture, or beam ideas [20-23]. Today, these algorithms are widely used across a range of sciences, including the application of the Wall Algorithm for error control in wind generators [24], the interpretation of Photoplethysmography (PPG) signals for the detection of cardiovascular disease, and the management of cardiovascular disease. Automated with the manufacturer of fuzzy Proportional–Integral–Derivative (PID) (controller coefficients) [25], [26], store planning [27], application in intelligent devices [28], and many engineering applications, including autonomy, optimal control, resource management, security diagnostics stated [29-32]. Each metaheuristic and hyper-heuristic algorithm is suitable for solving a specific set of problems, and each has its advantages and disadvantages. For example, noise tolerance can be an advantage not shared by everyone. According to the National Football League (NFL) theory, it is impossible to solve all problems with a single algorithm [33], [34]. Evolutionary algorithms, for example, are famous for solving global optimization problems in non-convex spaces [35].

Along with these new algorithms, their combination and the use of each are also considered [36-38]. This approach has led to the emergence of various algorithms. The emergence of new problems and the efficiency of new metaheuristic and hyper-heuristic algorithms underscore the importance of designing new algorithms. According to NFL theory, the need to model nature and the environment, and to develop new algorithms, is essential to overcoming future problems, solving today's issues, and meeting needs.

2 | Metaheuristic Algorithms

The world around us inspires metaheuristic algorithms. Nature and biology are two of these sources. In addition to biology, the laws of nature and physics, the human cognitive and behavioral sciences, and the rules of the worlds of sports and politics, the collective life of animals and even plants is another discipline whose insights have inspired the development of metaheuristic algorithms. There is a group intelligence that solves problems and prevents ruin in all these cases. In general, metaheuristic algorithms can be divided into four broad categories. The first category is evolutionary-based algorithms. The second category is called physics-based algorithms. The third category is human-based algorithms, and finally, the fourth category is swarm-based. *Fig. 1* provides a general classification of metaheuristic algorithms [34].

2.1 | Evolutionary-Based Algorithms

Evolutionary-based methods include algorithms based on Charles Darwin's theory of evolution. These algorithms are population-based, inspired by evolution and the generation of successive generations. Often, the first generation arises by chance, and subsequent generations arise from parent-child integration and the birth of new children.

2.2 | Genetic Algorithm

The first metaheuristic algorithm in this category was proposed by Holland [39]. This algorithm uses two leading operators, recombination and mutation, to generate new responses. In most cases, the population size of each generation is constant. Genetic Algorithms (GAs) have been used in many fields, including optimization in standard functions [39]. Other algorithms inspired by this category include genetic programming. In genetic programming, the answers produced are the same programs. This algorithm is used in autonomous applications [40]. The location-based algorithm is another metaheuristic that improves a unit's responses through repetition over many generations and uses mutations [41]. In this algorithm, the answers are factors in the environment, and instead of jumping, the components are assigned random values [7]. The incremental learning algorithm is an estimating algorithm and, in many cases, has performed better than the classical GA. Generation in this algorithm follows a probability vector and uses mutation, as in other algorithms in this category [42]. The immune system algorithm is a metaheuristic based on the structure of the human immune system. This algorithm is used to detect intrusion into the system, up to classifying data and other items. In this type of algorithm, there is a concept called transcendence, which refers to a variable value [43]. The weed algorithm is an algorithm inspired by weed survival and adaptability. This type of algorithm is used in various engineering problems and types of packages [44]. Other algorithms such as Photosynthetic Algorithm (PA) [45], Sapling Growing Up (SGUA) [46], Plant Growth (PG) Optimization [47], Rooted Tree Optimization (RTO) [48], Runner Root Algorithm (RRA) [49], and Paddy Field Algorithm (PFA) [50], are in this category.

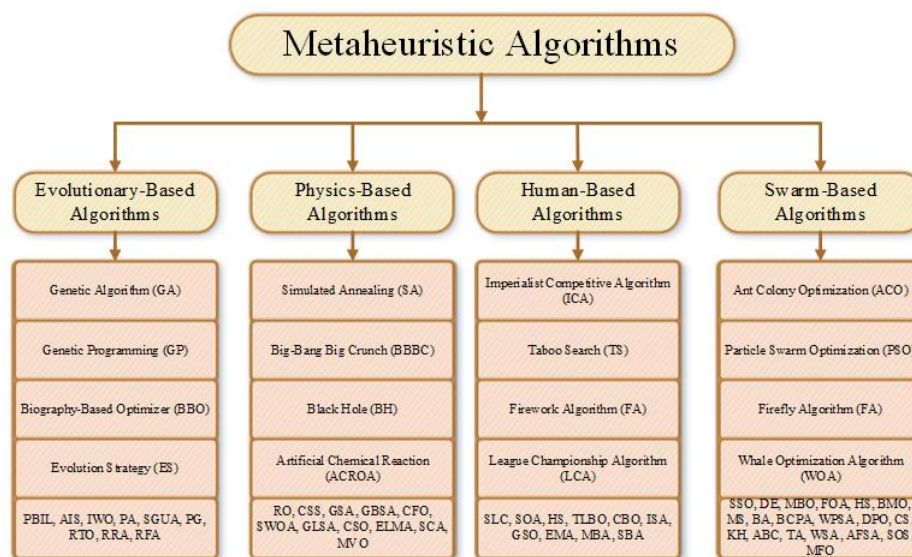


Fig. 1. The metaheuristic classification.

2.3 | Physics-Based Algorithms

The second category of metaheuristic algorithms is algorithms based on physics. Algorithms are inspired by the laws of physics and the phenomena they describe. These sources are the inspiration for the laws of matter and its changes, as well as for the laws of astrophysics and everything defined in physics. The most famous physical batch algorithm is Simulated Annealing (SA). In this algorithm, the crystallization process of metal particles is simulated when the molten metal cools slowly. This algorithm is applied in the search space to achieve the optimal answer. In the mountaineering algorithm, the algorithm continues along the path until the solutions improve. The mountaineering algorithm is satisfied with the first greedy answer it achieves. But this greedily best answer may not be the best answer across the answer space. The refrigeration simulation algorithm has a mechanism for searching across the response space and simultaneously locally and greedily exploring, increasing the likelihood of exploring unknown areas. The acceptance function guides the

movement between neighbors and avoids purely greedy choices. In this algorithm, a mechanism is used to escape from local optima.

A refrigeration simulation algorithm is suitable for searching vast spaces [51]. The Big Bang algorithm is another algorithm in a group of physics-based algorithms. The scientific theory of the Big Bang inspires this algorithm. The Big Bang algorithm consists of two main parts. In the first part, the Big Bang is simulated, producing a large amount of data. A large dataset is generated randomly in this section. Then the time of contraction is given for this massive volume, and in this part, as the world shrinks, the answer approaches. The Big Bang algorithm is used for optimal data mining and search [52]. The black hole metaheuristic algorithm is another astrophysics-related algorithm.

A black hole is a population-based algorithm, and stars are evaluated as answers by a fitting function. In fact, each solution in this algorithm is assigned a weight. The best star in each stage is the black hole that the rest of the stars are attracted to. The black hole metaheuristic algorithm is suitable for addressing environmental factors and issues [53]. The Chemical Reaction Optimization Algorithm is an algorithm based on chemical reactions. The basis of this algorithm is based on chemical structures and state changes in matter. This algorithm exploits the chemical changes that occur during state changes. This algorithm defines these changes mathematically and uses this process to solve problems. Applications of this algorithm include applications in multi-sequence levels and data mining [54]. The radiation optimization algorithm is another widely used physical algorithm. The phenomenon of light physics inspires this algorithm. The laws governing the refraction of light have given rise to the idea of solving complex problems. The sequence of light and its radius during normal motion, along with the changes that occur due to light refraction, form the basis of this algorithm. One application of this algorithm is solving engineering problems [21]. Other algorithms such as, Charged System Search (CSS) [55], Gravitational Search Algorithm (GSA) [56], Galaxy-Based Search (GBSA) [57], Central Force Optimization (CFO) [58], Small World Optimization (SWOA) [59], Gravitational Local Search (GLSA) [60], Curved Space Optimization (CSO) [61], and Electromagnetism-Like Mechanism Algorithm (ELMA) [62] Are in this category.

2.4 | Human-Based Algorithms

The third category of algorithms is based on social and human laws. Humans have set some of these laws, and some have been made throughout history and are unwritten. Laws have been enacted and have shaped the customs of a community, thereby constituting cultural algorithms. Some phenomena that have occurred in history, such as colonialism. Events took place in a society based on human standards, such as revolutions and laws passed to regulate the assembly and establish rules, such as football laws. These are excellent sources of inspiration for metaheuristic algorithms on which socio-humanistic algorithms are human-based. The colonial competition algorithm is the most common in this category. This algorithm is based on a historical rule and the relations between the colonial and colonial countries. This algorithm depicts the imperialist countries' approach to their colonies as mathematical laws. In this algorithm, the answers are the same countries. In competition, these countries try to influence the rest of the nations. The best solution for any colonial region and its surrounding areas is to maintain its colonies. The colonizers compete with each other until they get a good response.

The colonial competition algorithm has been used in many cases and can be applied to systems proposing array antennas and fuzzy control [63]. The Fireworks Algorithm is based on random initial answers and metric searches around the same points, in the hope of finding the optimal response. This algorithm considers the explosion and scattering of fire sparks as a model for solving problems. The Fireworks Optimization Algorithm is used to solve multi-objective problems [64]. The sports competition league algorithm simulates the rules set by humans. In fact, in this algorithm, teams are supposed to compete with each other. The winner of this contest is the best answer. In other words, this algorithm analyzes groups using a strategy and identifies the most potent team by simulating a competition among them. This algorithm works well in situations involving uncertainty [65]. The neighbor's Taboo or forbidden search algorithm is looking for the optimal

answer. This algorithm is inspired by a story about people in the Pacific who should not touch the natives' sacred objects. The algorithm's general procedure is to start with a random answer and find the best neighbor. Uniquely, this algorithm also gives duplicate and weak solutions a chance to be reselected, helping avoid being trapped in local minima. There is a forbidden list of answers. The answer is selected if it is not in the banned list, and if it is in the banned list, it is chosen provided that it is the best answer ever seen [66], [67]. Soccer League Competition (SLC) [68], Seeker Optimization Algorithm (SOA) [69], Harmony Search (HS) [22], Teaching Learning Based Optimization (TLBO) [70], Colliding Bodies Optimization (CBO) [71], Interior Search Algorithm (ISA) [72], Group Search Optimizer (GSO) [73], Exchange Market Algorithm (EMA) [74], Mine Blast Algorithm (MBA) [75], and Social-Based Algorithm (SBA) [76] are other algorithms in this category.

2.5 | Swarm-Based Algorithms

The fourth category of algorithms is based on the swarm intelligence of living things and particles that, alone, do not have intelligence and even have limitations. One of the newest algorithms in this category is the Trees Social Relations Optimization Algorithm. This algorithm is based on the social life of trees and their coordination in response to their environment. TSR is a general-purpose algorithm for solving continuous and discrete problems. This algorithm works like a jungle. Each jungle consists of several smaller sub-jungles managed by a mother tree. In addition to the mother tree, each sub-jungle consists of other trees that follow the instructions of the mother tree. Answers that are prone to growth are known as seedlings in any sub-jungle. This algorithm gives special priority to these answers, which are scored by A Growth Parameter (GP). This algorithm can be implemented in parallel with multiple processors. Each sub-jungle can be assigned to one processor [34]. Another of these algorithms is the Ant Colony (ACO) algorithm proposed by Dorigo [77], inspired by ants' cooperation to reach the food source. There is a particular order and harmony in the ACO. Ants can do nothing on their own and face many limitations in dealing with their environment. But the cooperation and interaction among these creatures have created a unique organization that has served as the inspiration for the ACO algorithm. The goal of the ants is to obtain the source of energy at the lowest cost, which they achieve through chance and purpose. In this way, first the ants randomly look for food and, along the way, release a substance called pheromone. If this path leads to food, pheromone depletion will result. Otherwise, it will only be used to return to the nest. In this way, short paths to food are marked with more pheromones, attracting more ants and creating an attractive path for them. For this pathway not to become permanent, the absorption factor and the pheromones evaporate. This algorithm is an excellent example of congestion intelligence. Ants that lack individual intelligence do a great job of navigating. This algorithm solves discrete problems and finds the path of least cost [77]. A Particle Swarm Optimization (PSO) optimization algorithm is inspired by the collective movement of birds to reach food. In this algorithm, in each iteration, there is a collective experience that finds the best possible answer, and a personal experience that, based on the mathematical relations of each particle, takes a path between these two points. The PSO algorithm is based on the way a group of living things finds food. In this algorithm, the best personal experience and the best group experience are the two principles for achieving the answer. This algorithm has particular engineering applications and is used for optimization in engineering problems [78]. The Wall Algorithm mentioned earlier is based on whaling. Whales are one of the most intelligent creatures under the sea. Wolves can learn to hunt from previous generations. The general pattern of whale hunting is similar to that of a PSO. But in this pattern, the whales try to move away from the best response to explore more of the sea. In fact, after finding a food source, they circle and create a spiral in the shape of 9 and do their hunting. Then each wall tries to move away.

In this algorithm, two phases of exploitation and exploration are used to leverage existing solutions and achieve better ones. This algorithm is also used in engineering optimizations. The initial tests of this algorithm were on mechanical engineering problems [79]. The Firewall Algorithm seeks the optimal solution to the problem by assigning random locations and quantities based on the location fit of each firewall, serving as a model for the number of glow pigments. The glowworm site is then updated in successive iterations of the

algorithm to find an optimal or near-optimal solution. This algorithm has been used in work related to resource management [80]. The spider community algorithm is another swarm intelligence algorithm. This algorithm treats spiders on the net as answers; each creates a vibration when food reaches the net and notifies the other agents that it has received a response. Meanwhile, other factors are aware of each other's progress toward the answer and monitor each other's actions. This algorithm has been used in security and network applications [81]. Dolphin Echolocation (DE) [82], Marriage in Honey Bees (MBO) [83], Fruit Fly Optimization (FOA) [84], Hunting Search (HS) [85], Bird Mating Optimizer (BMO) [86], Monkey Search (MS) [87], Bat Inspired Algorithm (BA) [88], Bee Collecting Pollen (BCPA) [89], Wolf Pack Search (WPSA) [90], Dolphin Partner Optimization (DPO) [91], Cuckoo Search (CS) [92], Krill Herd (KH) [23], Artificial Bee Colony (ABC) [93], Termite Algorithm (TA) [94], Wasp Swarm Algorithm (WSA) [95], Artificial Fish-Swarm (AFSA) [96], and Symbiotic Organisms Search (SOS) [97] are other algorithms in this category.

2.6 | Hyper-Heuristic Algorithm

Hyper-heuristic algorithms are the next level after metaheuristic algorithms. These algorithms result from combining and selecting metaheuristic algorithms [98]. Hyper-heuristic algorithms are divided into two general types. The first type results from combining several innovative or metaheuristic algorithms. The second type of algorithm is to select metaheuristic or innovative methods [99]. In 1963, authors hypothesized the composition of planning laws. This hypothesis represented the production of a new algorithm by several algorithms. In the early 2000s, the Hyper-heuristic algorithm was first used to arrange several innovative algorithms. The working process of Hyper-heuristic algorithms is to try different sequences of innovative algorithms and select the best sequence as the answer [98]. Many multi-objective algorithms do this by mathematical modeling. Such algorithms include multi-purpose, multi-objective algorithms, such as the non-dominated Sorting Genetic Algorithm II (NSGA-II) for optimizing machining process parameters [100] and multiple-objective PSO optimization [101]. Hyper-heuristic algorithms in scientific papers are divided into four practical categories [98].

This classification is based on the performance of these algorithms. The first category is the logical combination of algorithms, including higher-level algorithms developed through innovation and metaheuristic algorithms. Some of them generate a new high-level algorithm by combining innovative algorithms with intelligent metaheuristic algorithms. These supra-heuristic algorithms are called automated discovery sequences [102]. There are algorithms called automated programming systems in the second category of hyper-heuristic algorithms. This model was developed from algorithms for satellite systems. The first experiment focused on the routing process in mountainous and rugged terrain. This field is related to artificial intelligence and ML [89]. The third category is the first and most essential part of parameter control in hyper-heuristic algorithms, based on algorithmic changes. The title of this category is "Automatic Parameter Control in Evolutionary Algorithms." Over time, as evolutionary algorithms such as genetic algorithms evolved, innovative algorithms were added that automatically adjusted several specific parameters, making parameter selection automatic [103]. The title of the fourth category is "automated learning of heuristic-innovative methods." The super-innovations that the algorithm automatically generated did not make much progress. But in recent years, significant progress has been made in this area. Multi-objective algorithms and hyper-heuristic algorithms are a combination. The artificial intelligence methods used in these algorithms include ML, deep learning, and reinforcement learning. The most widely used super-innovative algorithm has recently been reinforcement learning [104]. Hyper-heuristic algorithms based on reinforcement learning work on the Q-Learning Algorithm [105].

Q-learning is one of the Reinforcement Learning Algorithms that follow a specific policy, learning an action-value function to select actions in different situations. One of the strengths of this method is the ability to understand this function without having a particular model of the environment [106], [108]. This algorithm communicates with the Q-learning model between low-level and motion algorithms, thereby making it multi-functional [105], [107]. Automated ML and many hybrid and super-innovative algorithms classify and separate the various parts of algorithms. This classification has been simplified using ML. Common ML algorithms

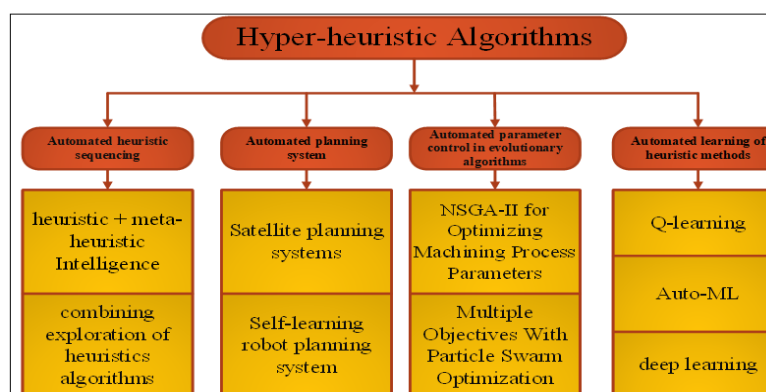
include decision trees, neural networks, support vector machines, and more. The use of these algorithms in ML science has led to significant progress in developing super-innovative algorithms [108]. Since no single algorithm has been developed to address all complex problems, hyper-heuristic algorithms aim to address them. Algorithms that use automated ML and classify problems must perform two operations optimally. These actions adjust data input properties and parameters, resulting in optimal classification. One of the best ways to learn ML is using the Auto-weka and Auto-sklearn systems, which have helped solve such problems [105], [106], [108].

In-depth knowledge can be used in various fields, some of which include web search, content filtering on social networks, cameras and smartphones, speech-to-text conversion, and much more. All this is done through deep learning [109]. Deep learning is divided into several categories. One model is a neural network that updates its weights in each layer on a recurrent basis. This model is called Backpropagation [110]. Evolutionary neural networks are another type of these models. The evolutionary neural network has a function similar to that of the early perceptron neural networks, which process images in deep layers [111]. The third model, called repetitive neural networks, is dynamic, unattended, and requires no training [112]. Reinforcement learning is another of these models. Problems encountered in the real world must be solved without training, which can be achieved by repeating correct answers [113]. Using deep learning systems, we have presented a way to build a system for designing super-innovative algorithms, which is fully automated with the help of ML, and neural networks and ML are also used in this system. Fig. 2 shows an overview of the basis of the work of hyper-heuristic algorithms.

Fig. 2. The hyper-heuristic classification.

3 | National Football League Theory

NP-Hard problems are divided into discrete and continuous problems. As mentioned earlier in the Introduction, according to the NFL theory, no algorithm can solve all NP-Hard problems. According to this theory, the development of metaheuristic and hyper-heuristic algorithms is necessary. An example is given in



this section to prove this theory. Here are the discrete (GA) [39], Gray Wolf (GWO) [17], Imperialist Algorithm (ICA) [64], SA [51], Taboo [67], [68], and ACO [78] methods applied to both the classic and well-known vendor problems.

In the Traveling Salesman (TS) problem, algorithms seek the shortest route between cities. Random coordinates for a thousand cities have been used to implement this problem. A TS must travel through all these cities, returning to the starting point after visiting each city exactly once. The implementation diagram of this problem is shown in Fig. 3. Table 1 also shows the numerical values of this implementation. This experiment was performed for each algorithm in 100 iterations.

The same algorithms are tested on the backpack problem. In this case, we are looking to fill a bag with a capacity of 625 per unit weight with the highest value. In this case, the initial population is 50, and the number of repetitions is 100. Also, the maximum allowable number of each object is 5. In this case, the goal is to

maximize points. Fig. 4 shows the results of this experiment. Also, to better understand the answers, the numerical values of the results are provided in the Table. 2.

In the following, Table 3 presents the rankings of the algorithms according to the best answer and allows comparison. According to the results of these two experiments, the ACO algorithm has obtained the best results in the TS. Still, it is in the backpack after the gray wolves and the Taboo. As shown in Table 3, the results in these two experiments are quite different. According to this proven NFL theory, the development of metaheuristic algorithms and their combination is essential to achieving ultraheuristic algorithms.

Robot path planning problems and welded beams have been used to demonstrate this theory for continuous problems. Discrete (GAs), GWO, Imperialist Competitive Algorithm (ICA), Harmony (HS) [22], PSO [79], and Whale Optimization (WOA) [80] have been used. In the robot path planning problem, the robot must move from the start point to the goal, which is full of obstacles. The target is to find the shortest way. The implementation diagram of this problem is shown in Fig. 5. Table 3 also shows the numerical values of this implementation. This experiment was performed for each algorithm in 100 iterations.

In a welded beam, the objective is to minimize the fabrication cost. A rigid member is welded to a beam, and a load is applied to its end. The total cost of making is equal to the effort costs plus the cost of the weld and beam material. So, this problem aims to minimize the fabrication cost of a welded beam [34]. For the welded beam, the number of repetitions is 100. Fig. 6 shows the results of this experiment. Also, to better understand the answers, the numerical values of the results are given in Table 4.

Fig. 3. TS problem results from some algorithms.

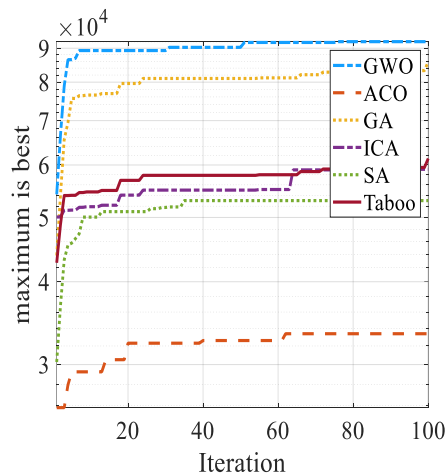
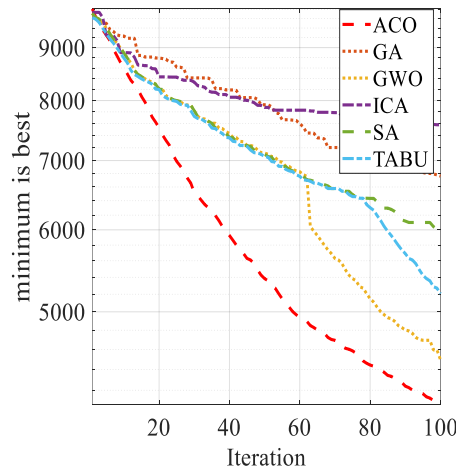


Fig. 4. Knapsack results by some algorithms.

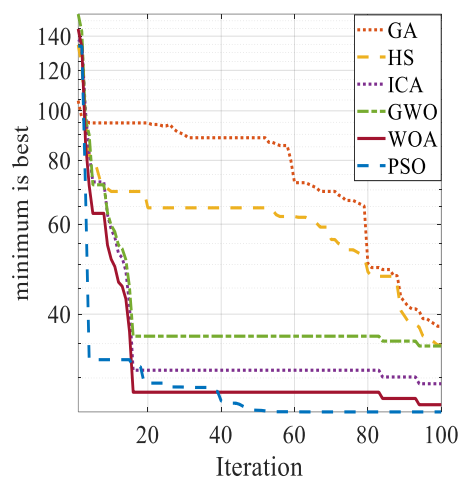


Fig. 5. Robot path planning results by some algorithms.

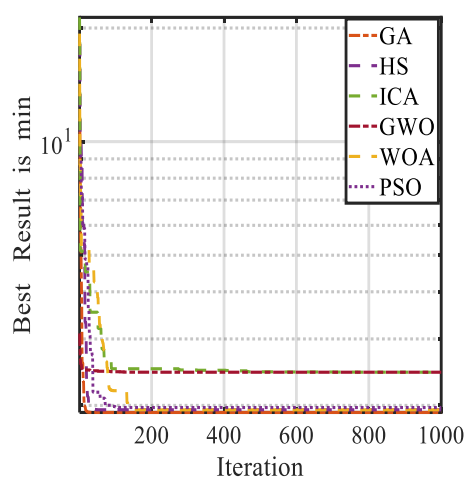


Fig. 6. Welded beam results by some algorithms.

Table 1. TSP results.

Algorithms	Worse	Best	Mean	STD
ACO	852. e+03	3.41 e+03	5.11 e+03	1.43 e+03
GA	9.65 e+03	6.76 e+03	7.93 e+03	1.84 e+03
GWO	9.73 e+03	4.41 e+03	7.52 e+03	1.35 e+03
ICA	9.73 e+03	7.57 e+03	8.11 e+03	536.8624
SA	9.71 e+03	5.98 e+03	7.29 e+03	1.00 e+03
Taboo	9.70 e+03	5.10 e+03	7.88 e+03	891.18

Table 2. Knapsack results.

Algorithms	Worse	Best	Mean	STD
ACO	25830	33390	32174	17265
GA	53264	78376	69236	29378
GWO	43808	85904	80126	53472
ICA	50084	59100	54758	82332
SA	30264	53016	51734	31952
Taboo	42720	61360	57563	23584

Table 3. Discrete algorithms ranking.

Algorithms	TSP Rank	Knapsack Rank
ACO	1	6
GA	5	2
GWO	2	1
ICA	6	4
SA	4	5
Taboo	3	3

Table 4. Robot path planning results.

Algorithms	Worse	Best	Mean	STD
GA	104.69	39.69	76.83	19.48
HS	134.68	39.80	65.90	14.90
ICA	137.83	27.06	84.32	27.07
GWO	135.03	36.89	62.71	22.42
WOA	125.24	25.72	42.84	18.23
PSO	114.78	25.01	84.47	19.70

Table 5. Welded beam results.

Algorithms	Worse	Best	Mean	STD
GA	14.14	1.45	2.05	0.98
HS	16.22	1.35	2.08	0.94
ICA	15.67	2.09	1.92	0.87
GWO	18.32	2.02	1.97	0.86
WOA	19.79	1.19	1.89	0.94
PSO	14.40	1.23	2.07	0.91

Table 6. Continues the algorithms ranking.

Algorithms	Robot Path Rank	Welded Beam Rank
GA	5	4
HS	6	3
ICA	3	6
GWO	4	5
WOA	2	1
PSO	1	2

4 | Metaheuristic and Hyper-Heuristic Development

Since the first metaheuristic algorithm called genetics, countless algorithms have been introduced and are being developed and improved. Metaheuristic algorithms are divided into two categories, classical and modern, mentioned in the Introduction. *Fig. 7* shows the number of articles related to several classical algorithms in Google Scholar. *Fig. 8* also shows the number of articles on new (modern) algorithms [115]. According to *Figs. 7* and *8*: Most research and development has been done on GAs, which is natural given their antiquity. One question always arises in the context of metaheuristic algorithms. Which metaheuristic algorithm is more robust? This question has no definite answer.

On the other hand, according to the NFL theory, each algorithm answers several problems better than others. It may be an algorithm that is weaker in all respects than different existing algorithms. All these are reasons for the development of existing algorithms and the Introduction of new metaheuristic and hyper-heuristic algorithms. An algorithm for a particular problem is best when it gives the best results in the experiment. On the other hand, we face an abundance of algorithms for this experiment. The problem structure, the types of

algorithms, and previous experience help us reduce the number of selected algorithms. Ultimately, the investigation determines which algorithm is best suited to achieving the optimal solution.

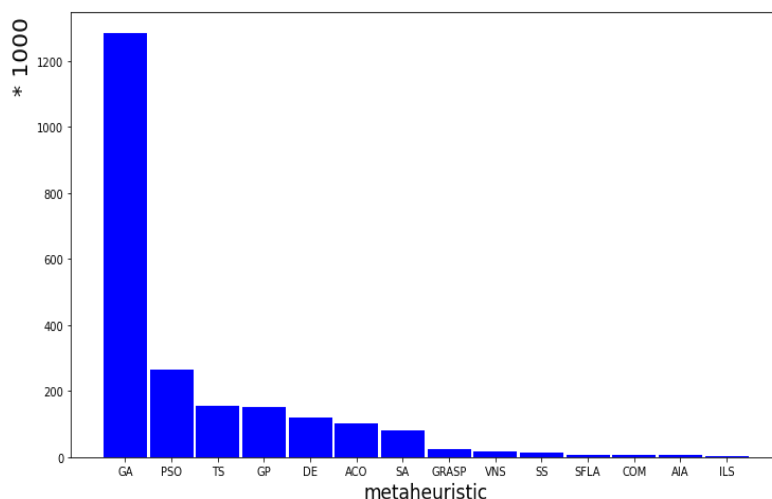


Fig. 7. Number of articles related to classic algorithms.

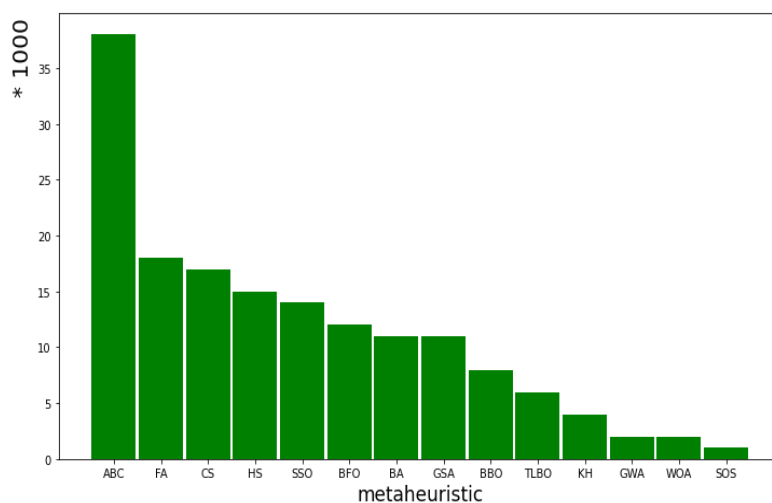


Fig. 8. Number of articles related to modern algorithms.

5 | Conclusion

Metaheuristic algorithms solve problems that do not have a known solution in a reasonable time. Sufficient time is the usual limit for solving a problem. In these problems, if the size and amount of data are doubled, the problem's complexity increases exponentially. If we take the issue of the TS as an example, it may take several years for urban metro systems like Tokyo's to resolve it. So these problems cannot be solved in classical terms with classical algorithms. Using metaphors to solve complex issues yields acceptable, relative answers quickly. These algorithms fall into the category of optimization algorithms. In other words, in these algorithms, the priority is to achieve the correct answer quickly rather than the best solution over a very long time. This long time becomes so great that, as things get more significant, it becomes practically unattainable. Various metaheuristic algorithms have been introduced, each belonging to a specific category: biological, physical, congestion, or social methods. These methods have been used to solve problems that have not been addressed promptly by traditional means. However, none of them can be superior to the other techniques for solving all issues. For this reason, efforts to design metaheuristic methods continue to address a broader range of topics.

Authors' Contributions

The author carried out all aspects of the research and manuscript preparation. The author has read and approved the final version of the manuscript.

Funding

This study did not receive any specific funding from public, commercial, or non-profit funding agencies.

Data Availability

All data are included in the text.

Conflict of Interest

The author declares that he has no conflicts of interest.

Consent for Publication

The author has given consent for the publication of this manuscript.

Ethics Approval and Consent to Participate

This study does not involve any research conducted on human participants or animals.

References

- [1] Martí, R., & Reinelt, G. (2022). *Exact and heuristic methods in combinatorial optimization*. Springer Berlin, Heidelberg. <https://doi.org/10.1007/978-3-662-64877-3>
- [3] Neumann, F., & Witt, C. (2010). Combinatorial optimization and computational complexity. In *Bioinspired computation in combinatorial optimization: Algorithms and their computational complexity* (pp. 9-19). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-16544-3_2
- [4] Dokeroglu, T., Sevinc, E., Kucukyilmaz, T., & Cosar, A. (2019). A survey on new generation metaheuristic algorithms. *Computers & industrial engineering*, 137, 106040. <https://doi.org/10.1016/j.cie.2019.106040>
- [5] Boussaid, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Information sciences*, 237, 82-117. <https://doi.org/10.1016/j.ins.2013.02.041>
- [7] Wu, G., Mallipeddi, R., & Suganthan, P. N. (2019). Ensemble strategies for population-based optimization algorithms—A survey. *Swarm and evolutionary computation*, 44, 695-711. <https://doi.org/10.1016/j.swevo.2018.08.015>
- [8] Chugh, T., Sindhya, K., Hakanen, J., & Miettinen, K. (2019). A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. *Soft computing*, 23(9), 3137-3166. <https://doi.org/10.1007/s00500-017-2965-0>
- [9] Mahdavi, S., Shiri, M. E., & Rahnamayan, S. (2015). Metaheuristics in large-scale global continues optimization: A survey. *Information sciences*, 295, 407-428. <https://doi.org/10.1016/j.ins.2014.10.042>
- [10] Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Computers & structures*, 169, 1-12. <https://doi.org/10.1016/j.compstruc.2016.03.001>
- [11] Birattari, M., Paquete, L., Stützle, T., & Varrentrapp, K. (2001). *Classification of metaheuristics and design of experiments for the analysis of components*. <https://eden.dei.uc.pt/~paquete/papers/AIDA-01-05.pdf>
- [12] Shukla, A. K., Tripathi, D., Reddy, B. R., & Chandramohan, D. (2020). A study on metaheuristics approaches for gene selection in microarray data: Algorithms, applications and open challenges. *Evolutionary intelligence*, 13(3), 309-329. <https://doi.org/10.1007/s12065-019-00306-6>
- [13] Gomes, W. J., Beck, A. T., Lopez, R. H., & Miguel, L. F. (2018). A probabilistic metric for comparing metaheuristic optimization algorithms. *Structural safety*, 70, 59-70. <https://doi.org/10.1016/j.strusafe.2017.10.006>

- [14] Kazemzadeh Azad, S. (2018). Seeding the initial population with feasible solutions in metaheuristic optimization of steel trusses. *Engineering optimization*, 50(1), 89-105. <https://doi.org/10.1080/0305215X.2017.1284833>
- [15] Mousavirad, S. J., & Ebrahimpour-Komleh, H. (2017). Multilevel image thresholding using entropy of histogram and recently developed population-based metaheuristic algorithms. *Evolutionary intelligence*, 10(1), 45-75. <https://doi.org/10.1007/s12065-017-0152-y>
- [16] Talbi, E. G. (2009). *Metaheuristics: From design to implementation*. John Wiley & Sons. https://zeus.inf.ucv.cl/~bcrawford/DiplomadoIA_2021/Cap1_Metaheuristics_Talbi.pdf
- [17] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf optimizer. *Advances in engineering software*, 69, 46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
- [18] Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys (CSUR)*, 35(3), 268–308. <https://doi.org/10.1145/937503.937505>
- [19] Osman, I. H., & Laporte, G. (1996). Metaheuristics: A bibliography. *Annals of operations research*, 63(5), 511–623. <https://doi.org/10.1007/BF02125421>
- [20] Premaratne, U., Samarabandu, J., & Sidhu, T. (2009). A new biologically inspired optimization algorithm. *2009 international conference on industrial and information systems (ICIIS)* (pp. 279-284). IEEE. <https://doi.org/10.1109/ICIINF5.2009.5429852>
- [21] Kaveh, A., & Khayatizad, M. (2012). A new meta-heuristic method: Ray optimization. *Computers & structures*, 112, 283–294. <https://doi.org/10.1016/j.compstruc.2012.09.003>
- [22] Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: Harmony search. *Simulation*, 76(2), 60–68. <https://doi.org/10.1177/003754970107600201>
- [23] Gandomi, A. H., & Alavi, A. H. (2012). Krill herd: A new bio-inspired optimization algorithm. *Communications in nonlinear science and numerical simulation*, 17(12), 4831–4845. <https://doi.org/10.1016/j.cnsns.2012.05.010>
- [24] Qais, M. H., Hasanien, H. M., & Alghuwainem, S. (2020). Whale optimization algorithm-based Sugeno fuzzy logic controller for fault ride-through improvement of grid-connected variable speed wind generators. *Engineering applications of artificial intelligence*, 87, 103328. <https://doi.org/10.1016/j.engappai.2019.103328>
- [25] Prabhakar, S. K., Rajaguru, H., & Lee, S. W. (2019). Metaheuristic-based dimensionality reduction and classification analysis of PPG signals for interpreting cardiovascular disease. *IEEE access*, 7, 165181-165206. <https://scispace.com/pdf/metaheuristic-based-dimensionality-reduction-and-2ey5x77lb6.pdf>
- [26] Pilla, R., Azar, A. T., & Gorripotu, T. S. (2019). Impact of flexible AC transmission system devices on automatic generation control with a metaheuristic based fuzzy PID controller. *Energies*, 12(21), 4193. <https://doi.org/10.3390/en12214193>
- [27] Bissoli, D. C., Zufferey, N., & Amaral, A. R. S. (2021). Lexicographic optimization-based clustering search metaheuristic for the multiobjective flexible job shop scheduling problem. *International transactions in operational research*, 28(5), 2733–2758. <https://doi.org/10.1111/itor.12745>
- [28] Reda, H. T., Mahmood, A., Diro, A., Chilamkurti, N., & Kallam, S. (2020). Firefly-inspired stochastic resonance for spectrum sensing in CR-based IoT communications. *Neural computing and applications*, 32(20), 16011–16023. <https://doi.org/10.1007/s00521-019-04584-0>
- [29] Yang, X. S. (2010). *Engineering optimization: An introduction with metaheuristic applications*. John Wiley & Sons. <https://doi.org/10.1080/00107514.2012.661773>
- [30] Essaid, M., Idoumghar, L., Lepagnot, J., & Brévilliers, M. (2019). GPU parallelization strategies for metaheuristics: A survey. *International journal of parallel, emergent and distributed systems*, 34(5), 497-522. <https://doi.org/10.1080/17445760.2018.1428969>
- [31] Zhang, J., Xiao, M., Gao, L., & Pan, Q. (2018). Queuing search algorithm: A novel metaheuristic algorithm for solving engineering optimization problems. *Applied mathematical modelling*, 63, 464–490. <https://doi.org/10.1016/j.apm.2018.06.036>
- [32] Fathollahi-Fard, A. M., Hajiaghahi-Keshteli, M., & Tavakkoli-Moghaddam, R. (2018). The social engineering optimizer (SEO). *Engineering applications of artificial intelligence*, 72, 267-293. <https://doi.org/10.1016/j.engappai.2018.04.009>
- [33] Wolpert, D. H., & Macready, W. G. (2002). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1), 67–82. <https://doi.org/10.1109/4235.585893>
- [34] Alimoradi, M., Azgomi, H., & Asghari, A. (2022). Trees social relations optimization algorithm: A new Swarm-based metaheuristic technique to solve continuous and discrete optimization problems. *Mathematics and computers in simulation*, 194, 629–664. <https://doi.org/10.1016/j.matcom.2021.12.010>

- [35] Gomes, W. J., Beck, A. T., & Haukaas, T. (2013). Optimal inspection planning for onshore pipelines subject to external corrosion. *Reliability engineering & system safety*, 118, 18-27. <https://doi.org/10.1016/j.res.2013.04.011>
- [36] Pellerin, R., Perrier, N., & Berthaut, F. (2020). A survey of hybrid metaheuristics for the resource-constrained project scheduling problem. *European journal of operational research*, 280(2), 395-416. <https://doi.org/10.1016/j.ejor.2019.01.063>
- [37] Peres, W., Júnior, I. C. S., & Passos Filho, J. A. (2018). Gradient based hybrid metaheuristics for robust tuning of power system stabilizers. *International journal of electrical power & energy systems*, 95, 47-72. <https://doi.org/10.1016/j.ijepes.2017.08.014>
- [38] Silva, M. A. L., de Souza, S. R., Souza, M. J. F., & de Franca Filho, M. F. (2018). Hybrid metaheuristics and multi-agent systems for solving optimization problems: A review of frameworks and a comparative analysis. *Applied soft computing*, 71, 433-459. <https://doi.org/10.1016/j.asoc.2018.06.050>
- [39] Holland, J. H. (1992). Genetic algorithms. *Scientific american*, 267(1), 66-73. <https://www.jstor.org/stable/24939139>
- [40] Koza, J. (1992). On the programming of computers by means of natural selection. In *Genetic programming* (pp. 1-35). MIT Press. <https://cir.nii.ac.jp/crid/1573105975012577408>
- [41] Simon, D. (2009). Biogeography-based optimization. *Evolutionary computation, iee transactions on*, 12, 702-713. <https://doi.org/10.1109/TEVC.2008.919004>
- [42] Mezura-Montes, E., & Coello, C. A. C. (2005). Useful infeasible solutions in engineering optimization with evolutionary algorithms. *Mexican international conference on artificial intelligence* (pp. 652-662). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/11579427_66
- [43] De Castro, L. N., & Timmis, J. (2002). *Artificial immune systems: A new computational intelligence approach*. Springer Science & Business Media. <https://kar.kent.ac.uk/13726/>
- [44] Mehrabian, A. R., & Lucas, C. (2006). A novel numerical optimization algorithm inspired from weed colonization. *Ecological informatics*, 1(4), 355-366. <https://doi.org/10.1016/j.ecoinf.2006.07.003>
- [45] Murase, H. (2000). Finite element inverse analysis using a photosynthetic algorithm. *Computers and electronics in agriculture*, 29(1-2), 115-123. [https://doi.org/10.1016/S0168-1699\(00\)00139-3](https://doi.org/10.1016/S0168-1699(00)00139-3)
- [46] Karci, A., & Alatas, B. (2006). Thinking capability of saplings growing up algorithm. *International conference on intelligent data engineering and automated learning* (pp. 386-393). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/11875581_47
- [47] Zhou, Y., Wang, Y., Chen, X., Zhang, L., & Wu, K. (2017). A novel path planning algorithm based on plant growth mechanism. *Soft computing*, 21(2), 435-445. <https://doi.org/10.1007/s00500-016-2045-x>
- [48] Labbi, Y., Attous, D. B., Gabbar, H. A., Mahdad, B., & Zidan, A. (2016). A new rooted tree optimization algorithm for economic dispatch with valve-point effect. *International journal of electrical power & energy systems*, 79, 298-311. <https://doi.org/10.1016/j.ijepes.2016.01.028>
- [49] Merrikh-Bayat, F. (2015). The runner-root algorithm: A metaheuristic for solving unimodal and multimodal optimization problems inspired by runners and roots of plants in nature. *Applied soft computing*, 33, 292-303. <https://doi.org/10.1016/j.asoc.2015.04.048>
- [50] Kirkpatrick, S. (1983). Improvement of reliabilities of regulations using a hierarchical structure in a genetic network. *Science*, 220, 671-680. <https://www.science.org/doi/abs/10.1126/science.220.4598.671>
- [51] Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of optimization theory and applications*, 45(1), 41-51. <https://doi.org/10.1007/BF00940812>
- [52] Erol, O. K., & Eksin, I. (2006). A new optimization method: Big bang-big crunch. *Advances in engineering software*, 37(2), 106-111. <https://doi.org/10.1016/j.advengsoft.2005.04.005>
- [53] Hatamlou, A. (2013). Black hole: A new heuristic optimization approach for data clustering. *Information sciences*, 222, 175-184. <https://doi.org/10.1016/j.ins.2012.08.023>
- [54] Alatas, B. (2011). ACROA: Artificial chemical reaction optimization algorithm for global optimization. *Expert systems with applications*, 38(10), 13170-13180. <https://doi.org/10.1016/j.eswa.2011.04.126>
- [55] Kaveh, A., & Talatahari, S. (2010). A novel heuristic optimization method: Charged system search. *Acta mechanica*, 213(3), 267-289. <https://doi.org/10.1007/s00707-009-0270-4>
- [56] Rashedi, E., Rashedi, E., & Nezamabadi-pour, H. (2018). A comprehensive survey on gravitational search algorithm. *Swarm and evolutionary computation*, 41, 141-158. <https://doi.org/10.1016/j.swevo.2018.02.018>
- [57] Shah-Hosseini, H. (2011). Principal components analysis by the galaxy-based search algorithm: A novel metaheuristic for continuous optimisation. *International journal of computational science and engineering*, 6(1-2), 132-140. <https://doi.org/10.1504/IJCSE.2011.041221>

- [58] Formato, R. (2007). Central force optimization: A new metaheuristic with applications in applied electromagnetics. *Progress in electromagnetics research-pier - prog electromagn res*, 77, 425–491. <https://doi.org/10.2528/PIER07082403>
- [59] Du, H., Wu, X., & Zhuang, J. (2006). Small-world optimization algorithm for function optimization. *International conference on natural computation* (pp. 264–273). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/11881223_33
- [60] Webster, B., & Bernhard, P. J. (2003). *A local search optimization algorithm based on natural principles of gravitation*. https://repository.fit.edu/ces_faculty/192
- [61] Moghaddam, F. F., Moghaddam, R. F., & Cheriet, M. (2012). *Curved space optimization: A random search based on general relativity theory*. *arXiv preprint arXiv:1208.2214*. <https://doi.org/10.48550/arXiv.1208.2214>
- [62] Filipović, V., Kartelj, A., & Matić, D. (2013). An electromagnetism metaheuristic for solving the maximum betweenness problem. *Applied soft computing*, 13(2), 1303–1313. <https://doi.org/10.1016/j.asoc.2012.10.015>
- [63] Atashpaz-Gargari, E., & Lucas, C. (2007). Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. *2007 IEEE congress on evolutionary computation* (pp. 4661–4667). IEEE. <https://doi.org/10.1109/CEC.2007.4425083>
- [64] Tan, Y., & Zhu, Y. (2010). Fireworks algorithm for optimization. *International conference in swarm intelligence* (pp. 355–364). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-13495-1_44
- [65] Kashan, A. H. (2011). An efficient algorithm for constrained global optimization and application to mechanical engineering design: League championship algorithm (LCA). *Computer-aided design*, 43(12), 1769–1792. <https://doi.org/10.1016/j.cad.2011.07.003>
- [66] Glover, F. (1989). Tabu search—part I. *ORSA journal on computing*, 1(3), 190–206. <https://doi.org/10.1287/ijoc.1.3.190>
- [67] Glover, F. (1990). Tabu search—part II. *ORSA journal on computing*, 2(1), 4–32. <https://doi.org/10.1287/ijoc.2.1.4>
- [68] Moosavian, N., & Roodsari, B. K. (2014). Soccer league competition algorithm: A novel meta-heuristic algorithm for optimal design of water distribution networks. *Swarm and evolutionary computation*, 17, 14–24. <https://doi.org/10.1016/j.swevo.2014.02.002>
- [69] Dai, C., Chen, W., Zhu, Y., & Zhang, X. (2009). Seeker optimization algorithm for optimal reactive power dispatch. *IEEE transactions on power systems*, 24(3), 1218–1231. <https://doi.org/10.1109/TPWRS.2009.2021226>
- [70] Kiziloz, H. E., Deniz, A., Dokeroglu, T., & Cosar, A. (2018). Novel multiobjective TLBO algorithms for the feature subset selection problem. *Neurocomputing*, 306, 94–107. <https://doi.org/10.1016/j.neucom.2018.04.020>
- [71] Kaveh, A., & Mahdavi, V. R. (2014). Colliding bodies optimization: A novel meta-heuristic method. *Computers & structures*, 139, 18–27. <https://doi.org/10.1016/j.compstruc.2014.04.005>
- [72] Gandomi, A. H. (2014). Interior search algorithm (ISA): A novel approach for global optimization. *ISA transactions*, 53(4), 1168–1183. <https://doi.org/10.1016/j.isatra.2014.03.018>
- [73] He, S., Wu, Q. H., & Saunders, J. R. (2009). Group search optimizer: An optimization algorithm inspired by animal searching behavior. *IEEE transactions on evolutionary computation*, 13(5), 973–990. <https://doi.org/10.1109/TEVC.2009.2011992>
- [74] Ghorbani, N., & Babaei, E. (2014). Exchange market algorithm. *Applied soft computing*, 19, 177–187. <https://doi.org/10.1016/j.asoc.2014.02.006>
- [75] Sadollah, A., Bahreininejad, A., Eskandar, H., & Hamdi, M. (2013). Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Applied soft computing*, 13(5), 2592–2612. <https://doi.org/10.1016/j.asoc.2012.11.026>
- [76] Ramezani, F., & Lotfi, S. (2013). Social-based algorithm (SBA). *Applied soft computing*, 13(5), 2837–2856. <https://doi.org/10.1016/j.asoc.2012.05.018>
- [77] Dorigo, M., Birattari, M., & Stutzle, T. (2007). Ant colony optimization. *IEEE computational intelligence magazine*, 1(4), 28–39. <https://doi.org/10.1109/MCI.2006.329691>
- [78] Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95-international conference on neural networks* (Vol. 4, pp. 1942–1948). IEEE. <https://doi.org/10.1109/ICNN.1995.488968>
- [79] Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in engineering software*, 95, 51–67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
- [80] Yang, X. S. (2010). Firefly algorithm, stochastic test functions and design optimisation. *International journal of bio-inspired computation*, 2(2), 78–84. <https://doi.org/10.1504/IJBIC.2010.032124>

- [81] Zhou, Y., Zhou, Y., Luo, Q., & Abdel-Basset, M. (2017). A simplex method-based social spider optimization algorithm for clustering analysis. *Engineering applications of artificial intelligence*, 64, 67–82. <https://doi.org/10.1016/j.engappai.2017.06.004>
- [82] Kaveh, A., & Farhoudi, N. (2013). A new optimization method: Dolphin echolocation. *Advances in engineering software*, 59, 53–70. <https://doi.org/10.1016/j.advengsoft.2013.03.004>
- [83] Abbass, H. A. (2001). MBO: Marriage in honey bees optimization-A haplometrosis polygynous swarming approach. *Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546)* (Vol. 1, pp. 207–214). IEEE. <https://doi.org/10.1109/CEC.2001.934391>
- [84] Pan, W. T. (2012). A new fruit fly optimization algorithm: Taking the financial distress model as an example. *Knowledge-based systems*, 26, 69–74. <https://doi.org/10.1016/j.knosys.2011.07.001>
- [85] Oftadeh, R., Mahjoob, M. J., & Shariatpanahi, M. (2010). A novel meta-heuristic optimization algorithm inspired by group hunting of animals: Hunting search. *Computers & mathematics with applications*, 60(7), 2087–2098. <https://doi.org/10.1016/j.camwa.2010.07.049>
- [86] Askarzadeh, A., & Rezaazadeh, A. (2013). A new heuristic optimization algorithm for modeling of proton exchange membrane fuel cell: Bird mating optimizer. *International journal of energy research*, 37(10), 1196–1204. <https://doi.org/10.1002/er.2915>
- [87] Mucherino, A., & Seref, O. (2007). Monkey search: A novel metaheuristic search for global optimization. *AIP conference proceedings* (Vol. 953, No. 1, pp. 162–173). American Institute of Physics. <https://doi.org/10.1063/1.2817338>
- [88] Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010)* (pp. 65–74). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-12538-6_6
- [89] Lu, X., & Zhou, Y. (2008). A novel global convergence algorithm: Bee collecting pollen algorithm. *International conference on intelligent computing* (pp. 518–525). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-85984-0_62
- [90] Yang, C., Tu, X., & Chen, J. (2007). Algorithm of marriage in honey bees optimization based on the wolf pack search. *The 2007 international conference on intelligent pervasive computing (IPC 2007)* (pp. 462–467). IEEE. <https://doi.org/10.1109/IPC.2007.104>
- [91] Shiqin, Y., Jianjun, J., & Guangxing, Y. (2009). A dolphin partner optimization. *2009 WRI global congress on intelligent systems* (Vol. 1, pp. 124–128). IEEE. <https://doi.org/10.1109/GCIS.2009.464>
- [92] Yang, X. S., & Deb, S. (2009). Cuckoo search via Lévy flights. *2009 world congress on nature & biologically inspired computing (NaBIC)* (pp. 210–214). IEEE. <https://doi.org/10.1109/NABIC.2009.5393690>
- [93] Karaboga, D., & Basturk, B. (2007). Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. *International fuzzy systems association world congress* (pp. 789–798). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-72950-1_77
- [94] Martin, R., & Stephen, W. (2006). Termite: A swarm intelligent routing algorithm for mobilewireless Ad-Hoc networks. In *Stigmergic optimization* (pp. 155–184). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-34690-6_7
- [95] Pinto, P. C., Runkler, T. A., & Sousa, J. M. (2007). Wasp swarm algorithm for dynamic MAX-SAT problems. *International conference on adaptive and natural computing algorithms* (pp. 350–357). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-71618-1_39
- [96] Farzi, S. (2009). Efficient job scheduling in grid computing with modified artificial fish swarm algorithm. *International journal of computer theory and engineering*, 1(1), 13. <https://doi.org/10.7763/IJCTE.2009.V1.3>
- [97] Cheng, M. Y., & Prayogo, D. (2014). Symbiotic organisms search: A new metaheuristic optimization algorithm. *Computers & structures*, 139, 98–112. <https://doi.org/10.1016/j.compstruc.2014.03.007>
- [98] Burke, E. K., Hyde, M. R., Kendall, G., Ochoa, G., Özcan, E., & Woodward, J. R. (2018). A classification of hyper-heuristic approaches: Revisited. In *Handbook of metaheuristics* (pp. 453–477). Springer. https://doi.org/10.1007/978-3-319-91086-4_14
- [99] Burke, E. K., Petrovic, S., & Qu, R. (2006). Case-based heuristic selection for timetabling problems. *Journal of scheduling*, 9(2), 115–132. <https://doi.org/10.1007/s10951-006-6775-y>
- [100] Yusoff, Y., Ngadiman, M. S., & Zain, A. M. (2011). Overview of NSGA-II for optimizing machining process parameters. *Procedia engineering*, 15, 3978–3983. <https://doi.org/10.1016/j.proeng.2011.08.745>

- [101] Coello, C. C., & Lechuga, M. S. (2002). MOPSO: A proposal for multiple objective particle swarm optimization. *Proceedings of the 2002 congress on evolutionary computation. CEC'02 (Cat. No. 02TH8600)* (Vol. 2, pp. 1051-1056). IEEE. <https://doi.org/10.1109/CEC.2002.1004388>
- [102] Gratch, J., Chien, S., & DeJong, G. (1993). Learning search control knowledge for deep space network scheduling. *Proceedings of the tenth international conference on machine learning* (pp. 135-142). Morgan Kaufmann Publishers. <https://doi.org/10.1016/b978-1-55860-307-3.50024-1>
- [103] Bäck, T., Fogel, D. B., & Michalewicz, Z. (1997). Handbook of evolutionary computation. *Release, 97(1)*, B1. <https://doi.org/10.1201/9780367802486>
- [104] Jaton, J. C., Huser, H., Blatt, Y., & Pecht, I. (1975). Circular dichroism and fluorescence studies of homogeneous antibodies to type III pneumococcal polysaccharide. *Biochemistry, 14(24)*, 5308-5311. <https://doi.org/10.1021/bi00695a013>
- [105] Melo, F. S. (2001). *Convergence of Q-learning: A simple proof*. https://www.academia.edu/36081666/Convergence_of_Q_learning_a_simple_proof
- [106] Choong, S. S., Wong, L. P., & Lim, C. P. (2018). Automatic design of hyper-heuristic based on reinforcement learning. *Information sciences, 436*, 89-107. <https://doi.org/10.1016/j.ins.2018.01.005>
- [107] Jiang, K., Yao, J., & Tan, X. (2023). *Contextual conservative q-learning for offline reinforcement learning*. *arXiv preprint arXiv:2301.01298*. <https://doi.org/10.48550/arXiv.2301.01298>
- [108] Xavier-Junior, J. C., Freitas, A. A., Ludermir, T. B., Feitosa-Neto, A., & Barreto, C. A. (2020). An evolutionary algorithm for automated machine learning focusing on classifier ensembles: An improved algorithm and extended results. *Theoretical computer science, 805*, 1-18. <https://doi.org/10.1016/j.tcs.2019.12.002>
- [109] Thornton, C. (2014). *Auto-WEKA: Combined selection and hyperparameter optimization of supervised machine learning algorithms* [Thesis]. <https://www.cs.ubc.ca/~hoos/Publ/ThoEtAl13.pdf>
- [110] Thornton, C., Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2013). Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. *Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 847-855). Association for Computing Machinery (ACM). <https://doi.org/10.1145/2487575.2487629>
- [111] Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1). MIT Press. <https://doi.org/10.1007/s10710-017-9314-z>
- [112] Kim, Y. (2014). Convolutional neural networks for sentence classification. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1746-1751). Association for Computational Linguistics. <https://doi.org/10.3115/v1/D14-1181>
- [113] Zaremba, W., Sutskever, I., & Vinyals, O. (2014). *Recurrent neural network regularization*. <https://doi.org/10.48550/arXiv.1409.2329>