Metaheuristic Algorithms in Optimization and its Application: A Review

Heba Mohammed Fadhil *1

[1] Department of Information and Communication, Al-Khwarizmi College of Engineering, University of Baghdad, Iraq

https://doi.org/10.31972/iceit2024.013

Abstract

Metaheuristic algorithms are an intelligent way of thinking and working developed for resolving diverse issues about optimization. The number of potential solutions for such problems often is too large to be properly analyzed using standard procedures; thus, these algorithms are highly flexible and can be useful in many cases where needed to predict different types of optimizations accurately. Metaheuristics take inspiration from several natural processes like evolution or animal behavior, which allow them to show strength without being specific only towards one area. Some Metaheuristics algorithms are commonly being used like : Genetic Algorithm (GA), Simulated Annealing (SA), Evolutionary Algorithm (EA), Tabu Search (TS), Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Ant Colony Optimization (ACO), and Cuckoo Search Approach (CSA). All of them derives from this initial set of solutions and employ heuristics to get from this set of solutions.. The objective of this paper is to thoroughly analyze different metaheuristic algorithms. Their principles, mechanisms and the area where they are applied and will delve into. This paper provides a qualitative analysis of these algorithmic performances in diverse settings that underscore their strong suits as well as their weaknesses. The discourse also makes mention of some specific examples like how metaheuristic algorithms find utility application in various fields which include but are not limited to engineering or computer science, even economics and healthcare later down the line receive due consideration with an eye towards specific results; showing not only how effective these individual algorithms can be when applied under differing scenarios but also pointing out areas deserving further research efforts be directed onto them.

Keywords: Metaheuristic Algorithms, Optimization Algorithms, Genetic Algorithm, Ant Colony Optimization, Simulated Annealing, Particle Swarm Optimization , Evolutionary Algorithm.

1. Introduction

Metaheuristics are development iterative algorithms that steer as well as manage hierarchical heuristics through smart incorporation of appreciable variations that help to find and manipulate objectives. They use modified learning techniques to build memory problems and achieve proponent approximate solutions while using analytical learning techniques to develop the content of the memory . It is a fact that a number of problems can be encapsulated into the optimization problems category and a large part of these can be classified as NP-Hard, implying that no efficient algorithms exist for solving them in a reasonable time and giving the exact optimal value(Santamaría *et al.*, 2020). Thus, researchers have identified new procedures for the solving of optimizations problems which is referred to as metaheuristics, which is derived from biological features like the theory of evolution by natural selection. This means that the people who develop these algorithms do not need to have a proficient mathematical knowledge in the problems they work on. In the last three decades alone, various algorithms became developed and used in solving a vast variety of differential equations(Dey et al, 2018). It is also noteworthy that because of a large amount of people in the population who concurrently look for the optimal solution and exchange information with each other, the search space is very large in population-based optimization algorithms. Metaheuristic algorithms can be broadly classified into two types: including genetic algorithms (GA) , which is based on the neo-Darwinian paradigm, and swarm intelligence algorithms, as ant colony optimization (ACO), particle swarm optimization (PSO), artificial bee algorithms, and bacterial foraging optimization, derived from the observation of groups of social beings(Mann et al, 2018). Furthermore, there are other population-based algorithms such as memetic algorithm where memetic algorithm invokes local search heuristics in each chromosome. These have been effectively utilized to solve a whole lot of problems right from their invention in fields such as technology, natural science, and engineering, among others. Optimization techniques, to be specific meta-heuristic algorithms, have been implemented in fields such as machine control systems and manufacturing(Nasser et al., 2021). They can be categorized into two main types: Heuristic and metaheuristic approaches are generally classified into tabu search, genetic algorithms, simulated annealing, and miscellaneous groups that include greedy optimization procedures, neighborhood-search metaheuristics, and population metaheuristics, while hybrid metaheuristics combining methods from different approaches. This paper makes several significant contributions to the field of metaheuristic algorithms:

1) Presents all possible types of metaheuristic algorithms outlining basic concepts, functions and their usage in numerous disciplines.
2) Strictly categorizes the metaheuristic algorithms in order to enable the researchers to easily search and select the most appropriate approaches when tackling certain optimization problems.
3) Dissects the results of metaheuristic algorithms with an aim of comparing and contrasting them especially in different environments for a better understanding of how each algorithm may perform better or poorly when compared to others in certain environment.

This paper is organized as follows: Section 2 provides brief information about metaheuristic algorithms, and Section 3 describes the factors of metaheuristics, and Section 4 presents classification of metaheuristic algorithms. Section 5 discusses the real world modeling of metaheuristic algorithms. Section 6 presents a discussion and research ideas for the future.

2. Background overview

Metaheuristics are a high-level algorithmic framework for developing heuristic optimization algorithms. These algorithms have been successfully used to solve a variety of complex problems. Apart from very large problem sizes, the main advantage of metaheuristics is its ability to provide high-quality solutions in a relatively short time(Saraswat and Singhal, 2017; Fadhil, 2024).

Metaheuristics are techniques for finding optimal solutions without much computational effort. In essence, metaheuristics are complex techniques that organize knowledge to efficiently find near-optimal solutions, thereby making heuristic operations more effective. These are adaptive generative mechanisms that control heuristics by integrating various concepts to test and maximize computational complexity. Metaheuristics control and reconfigure the activities of specific heuristics to ensure better solutions quickly and easily. This process can leverage a single solution or a set of solutions, using low/high-level operations, simple local searches, or constructive processes as direct heuristics. Different metaheuristics have different structural principles(Alorf, 2023; Fadhil and Younis, 2024).

Researchers often frame optimization processes using concepts seemingly unrelated to optimization, such as natural evolution, artificial annealing, or animal group behavior. For example, ant colony optimization (ACO) methods and tabu search avoid the use of mid-level clarification and instead focus on manipulating problem context to improve the ability to find meaningful alternatives. While some completely mechanistic solutions have been proposed, metaheuristic systems often rely heavily on the use of randomness and unpredictability. Metaheuristics are divided into different forms based on different characteristics as shown in Figure (1), which are described in detail below.

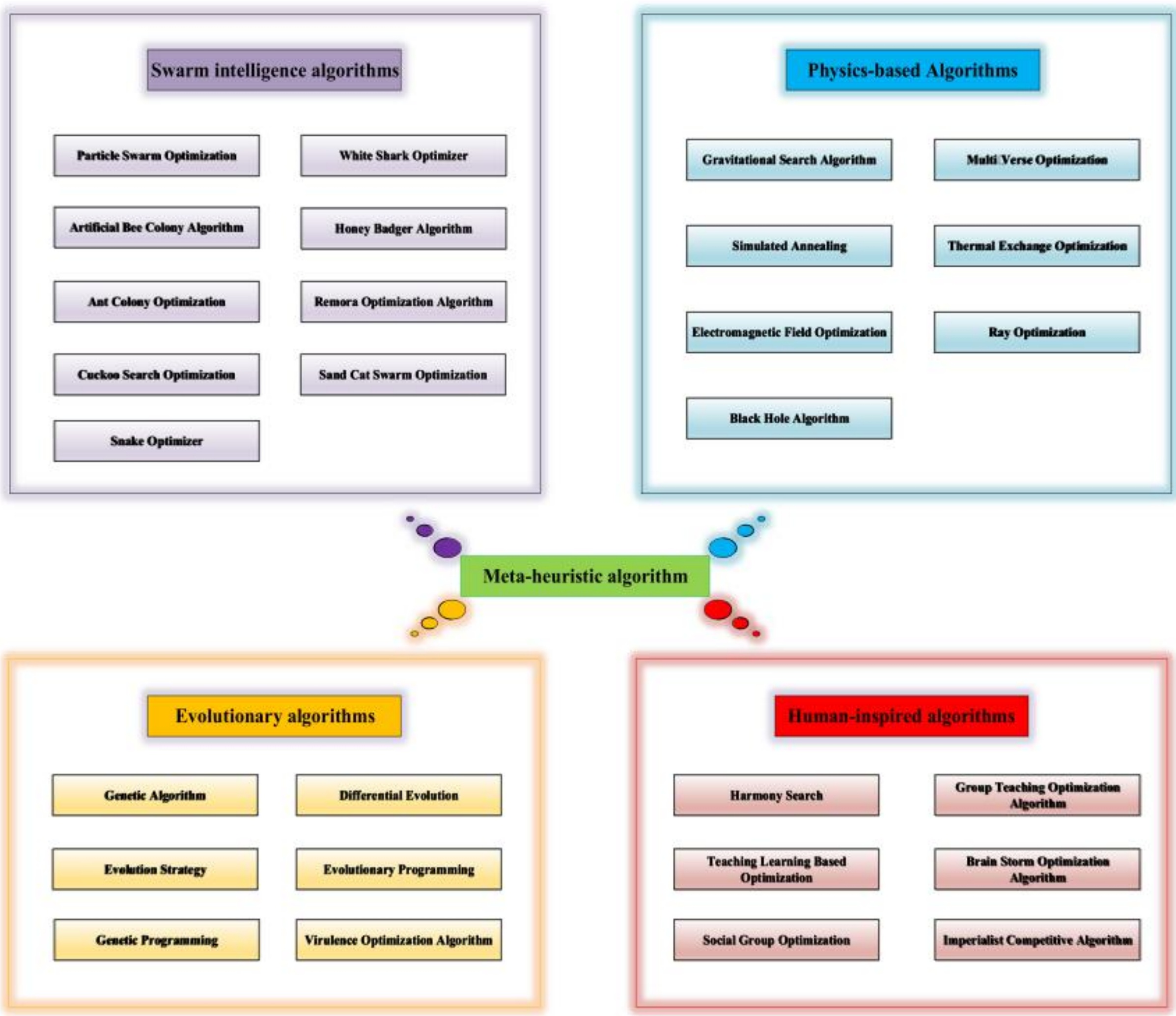

Figure 1: Metaheuristic optimization algorithms classification(Bhattacharyya et al., 2020).

A. Inspired by Nature vs. Not Inspired by Nature

At the core, nature-inspired algorithms use organic methods and principles to deal with intricate optimization dilemmas as illustrated in Figure (2). An instance of this is evolutionary algorithms like the Genetic Algorithm (GA) which finds its roots in Darwin's theory of natural selection system where only the fittest survive. In GA only the fittest individuals are allowed to reproduce through selection thus emulating natural selection in which the able organisms reproduce while the comparatively weaker or unfittest ones do not get a chance to reproduce. This is done by selection, crossover, and mutation three elementary strategies applied to it in GA which in turn, guides the subsequent generation of solutions systematically. Although it is getting more attention as a flexible but durable material, it is applied across diverse areas and not limited to engineering design application or even machine learning and artificial intelligence(Singhrova and Anu, 2022).

Another natural algorithm is the Particle Swarm Optimization (PSO) which is an imitation of bird community behavior. In PSO, a cluster of particles moves through the solution space; their flight path changes when the particle discovers the best local solution as well as the foremost global solution identified by the swarm. This is because it has been able to provide wide effective usage after showing a rather rapid result in the aspect of convergence for the optimal solutions and has been used in various aspects of the current modern world such as use in training of neural networks, image analysis and even in the control of robots (Hasanova, 2020).

This makes Ant Colony Optimization (ACO) to be conceived based on a mathematical problem solving tool through emulation of the ants food search strategy. Another type of self-organizing behavior of Ants is when they deposit chemical substances called Pheromones on trails that they use, or such paths are avoided by other Ants due to the pheromones. This unusual behavior is useful in solving sociable decomposition combinatorial optimization issues much as alike the salesman need to manage their routes and the vehicles needed to connect the routes in a network . The successful discovery of optimal paths by ACO even amidst intricately interwoven networks underscores its role as an invaluable weapon particularly in these two spheres logistics and communication network design(Mavrovouniotis et al., 2020).

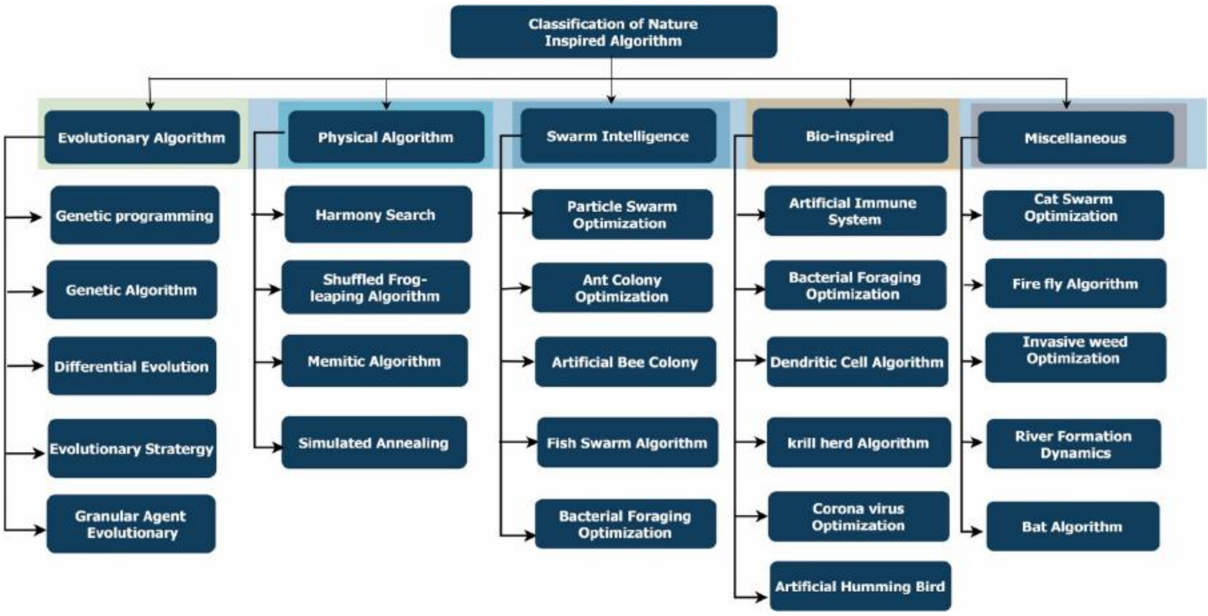

Figure 2: Nature-inspired algorithms classification(Hassan, 2023).

In a different light, non-nature-inspired algorithms are not directly based on nature. An example is Iterative Local Search (ILS) and Tabu Search (TS). ILS concentrates on the repeated application of local search to perturbations of the current solution set, taking into account diversification strategies for the escape from local optima. The approach finds use in scheduling or routing via combinatorial optimization where it excels in efficacy(Lidbe et al., 2017).

The use of memory structures is the approach that Tabu Search takes to avoid re-encountering solutions, managing the search operation by maintaining a list known as tabu that contains those solutions visited recently. The procedure leads to anti-cycling behavior and fosters promotion into uncharted territories within the solution space. TS method is well recommended especially in scheduling, resource allocation, and network design for its distinctive way it handles the complicated solution spaces.

B. Single-point search and group-based search

Single point search metaheuristic or as it is known as trajectory method, covers the solutions space step by step on a single candidate solution . One of the best examples of this approach is the technique of Simulated Annealing (SA). As with physical annealing, SA adopts worst solutions occasionally to avoid being locked in the local optima probability of which reduces as time increases. This technique makes its efficiency especially in planning tasks, image processing, and circuit design, where the search of the high-quality solutions depends on avoiding both local optima dependencies. That is, SA operates as if SA avoids obvious dangers on the road towards his goal and because of that he will succeed by excluding present dangers and putting his trust into lessening odds that will guide him to the winning path(A. AbdulJabbar and M. Abdullah, 2017; Lidbe et al., 2017).
One more single-point search method is Variable Neighborhood Search (VNS), as the name suggests, it is basically a variation of the local search. It systematically changes the structure of the neighborhood while searching. By this way, VNS can solve various combinatorial and global optimization problems effectively; they only need to stipulate that all vertices in the solution's domain are progressively distant from their parent. This makes VNS well suitable to solving problems that among solutions' spaces are complex, in the sense that they require deeper exploration though not challenging in their nature. Such algorithms gain advantages from the variety of different individuals within the population as it helps evade local optimums and makes the search procedure stronger. Evolutionary computation involves populations evolving over generations through mechanism design for global-level search using Genetic Algorithms (GA). Although its basis is widely implemented across engineering, economics and biology optimization problems due to strong search capability(Ji et al., 2021).
A swarm of particles is used by Particle Swarm Optimization (PSO) to search the solution space and every particle adapts its location according to what it has seen so far, hence the ability of PSO to optimization for functions, train neural networks, or control systems. These co-operative behaviors in PSO assist the algorithm to quickly focus on coming up with good quality solutions.

C. Static objective function and dynamic objective function

In the static and unchanging objective function throughout the search process are typically used by Genetic Algorithms (GA) and Simulated Annealing (SA). The fitness function in GA, which is not changed during the evaluation of the solution does not change as well. This makes GA appropriate for problems that have a clearly defined and constant objective function value. Similarly, SA maintains its objective function constant as it seeks to explore the solution space, making this algorithm effective for optimization problems under stable environment(Silva and Gabriel, 2020; Ali, 2021).

Another category is dynamic objective function metaheuristics where the metaheuristic is capable of identifying changes in the form of the objective function it deals with over time. Adaptive Simulated Annealing, (ASA) is an SA, which changes the cooling schedule and acceptance function depending on its environment parameters. It also makes ASA a potential candidate for the dynamic environment-based applications in which optimization benchmarks are emerging through yet different time horizons. Likewise, Evolutionary Algorithms (EA) show an aptitude towards dynamic objective functions; the contention that was made in the dynamics is that the fitness function can be modified to accomplish the evolution so as to capture new objectives or constraints which cannot be created when the regulatory environment is stable. This capability becomes meaningful in real-time systems and adaptive control applications that are both generally described as systems where situations are essentially constantly changing in dynamic environments and where the response or solution to a problem must be reached in the shortest time possible, often before stability can even be presumed within any facet of the system while in use (Slowik and Kwasnicka, 2020).

D. Memory consumption and memory-free methods

It defines metaheuristics as those heuristic techniques that use information stored in search as basis for their subsequent activities. A classic example is Tabu Search (TS) and in this case the tabu list is used to remember solutions that were recently explored so that they are not explored again. This memory mechanism understand enhances TS in terms of optimally creating its way in the solutions space steering it beyond cyclical distortions and to as yet unexplored territories. TS finds its niche particularly effective among combinatorial optimization problems that demand a keen avoidance of loops due to this feature: indeed, not being able to identify an optimal path from a particular area twice guarantees success in such applications (Hasan et al., 2020).

However, there is memory that is retained in the form of pheromone trails and is vital to the functioning of Ant Colony Optimization (ACO). Such trails are efficient in directing searches in the right manner guiding the ants towards avenues that would provide the best experience given previous successes. This mechanism enjoys success mostly when applied to route and scheduling issues as problems solved before can greatly decrease performance. These trails which connect one nest or hill to another, assist in relaying messages to the ants and as such leading to various decision making processes carried out by the colony such as which way to go or what task is best to be done next but not repeated often (Devarajan et al., 2020).

Memory-free methods, on the other hand, do no way involve the use of any memory structures in their approaches. It should be noted that SA does not have any record of the values to go by unlike other models since it initiates the process without any predefined input data. But it is solely dependent on the current solution and probability of accepting it without consideration of information that has gotten it there. This sort of bareness makes SA appropriate for all those optimization dilemmas in which memory has no place as a requirement. Similarly treading this memory-free path is Hill Climbing: a solution that, instead of just trying to find the nearest neighbor, does not even waste time storing the footprints for future moves (Wang, 2018). Thus such straightforward strategy can suffice well for the relatively naive optimization dilemma identification that is possible without the help of memory structures making it effective.

E. Continuous optimization and discrete optimization

The subdiscipline of mathematical optimization falls into two primary categories: continuous optimization and discrete optimization. It's important to distinguish between 'continuous' optimization where functions or quantities are optimized over continua or intervals of the real numbers, and 'discrete' optimization where functions or quantities are optimized over discrete sets of values. The second class of problems that is mentioned which falls under the branch of optimization is known as the continuous optimization and these are models that have variables with ranges that can be further divided. The smoothness of such functions makes it possible to solve such problems in most cases. This aids in the use of gradient methods, which enables one to understand happenings near a point or along some kind of route. PSO is therefore fitting for continuous optimization with particle adjustment searches for ideal results within an onslaught ongoing space exploration (Almomani, 2020).
The genetic algorithms (GA) can also be applied for continuous optimization, through the use of real-valued chromosomes representing solutions. This method is especially apt for continuous optimization problems, as observed in domains like engineering or economics, where the solution space is not discrete but continuous and needs to be carefully investigated. However, discrete optimization problems pertain to models consisting of discrete variables. These problems are even more challenging because the space of possible solutions is combinatorial, and it's hard to gather information from the vicinity of a given location. Ant Colony Optimization (ACO) is able to succeed in discrete optimization by composing its solutions from individual components represented distinctly on pheromone trails; this approach works well for combinatorial issues like traveling salesman problem and vehicle routing (El-Shafiey et al., 2022). SA is able to address discrete optimization

with a different approach: it determines whether discrete solutions should be accepted or rejected through a probabilistic acceptance criterion. Such an approach leads to SA, which provides the ability to address problems of scheduling and resource allocation, which are of discrete nature.

F. Metaheuristic algorithm modeling

The simulation of nonlinear systems with metaheuristic algorithms is in options of several methods that must be used, and which one is the best has its advantages and disadvantages. It is done through what is referred to as behavioral modeling whereby statistical regression techniques are usually applied to capture complex relationships within data and this was often done through these approaches because of restrictions in the computational systems that were available in the past. Such methodologies can migrate from paradigms closer to rudimentary constructs to complex infrastructures owing to the progression of computer hardware in recent years, broadening their applicability and a etiologic versatility (Sadeeq and Abdulazeez, 2023).

There are two established types of metaheuristic techniques utilized in modeling for which are Artificial Neural Networks (ANNs) and Gaussian Processes (GPs). ANNs are mathematical models, human artificial neural networks, constructed based on the neuron interconnection pattern. They adapt by learning from where change on weights as depending on error from the predicted values is done. Nevertheless, ANNs contain this versatility in an impressive degree of complexity owing to their ability to identify and extrapolate non-linear relations; they are suited for uses such as pattern recognition, regression and so on. However, the ANNs require huge amounts of training data which sometimes cause certain problems, for example over fitting and interpretability (Calvet et al., 2017).

Gaussian Processes (GP) are a part of the supervised machine learning technique that inferior to discrete values, leveraging distributions over the function space. As their operation can indicate, GPs use kernel functions, those are clever little tools that define the covariance structure of the data itself. This makes it unique because not only do they allow prediction of numbers but the prediction of an associated uncertainty as well; a property that may be of great use especially in regression problems in which knowing how far off you are might be as important as the number you get. Whereas the abundance of other decision approaches submerged you in a sea of numerical values and the shortage of context, GPs provide this measure of entropy prediction reliability, which makes it really helpful when working with comparatively small data sets. On the other hand, however, they are computationally extensive when confronted with large datasets together with the deliberate selection when it comes to choice of kernel functions; something that not everyone can handle so well(Ting et al., 2015).

There are other types of metaheuristics like the behavioral modeling and the hybrid approaches which are like taking a part of this metaheuristic and part of that metaheuristic just to be privileged with their strengths and not be disadvantaged with their vices. A simple example would be an algorithm that couples a genetic algorithm with the responsibility of exploring the global solution space along with a local search algorithm which is then tasked to tune the solutions. Hybrid methodologies are notably resourceful in addressing intricate optimization problems those that stand to gain much from dual search strategies. Despite delivering sturdy and effective results, the development of such hybrid algorithms; it can be tricky and would demand substantial experimentation before arriving at an optimal design(Radhika and Chaparala, 2018).

After looking at the detail of a wide range of diverse strategies and applications in powerful optimization tools that include nature-inspired and non-nature-inspired algorithms, single-point and population-based search methods, static and dynamic objective functions, memory consumption vs. memory-free methods, continuous vs. discrete optimization as well as metaheuristic algorithm modeling; it can be seen that various techniques are used by these algorithms including different classes with specific design choices. This knowledge empowers researchers and industry professionals to customize an algorithm according to their optimization challenges ensuring high effectiveness and quality of the solution.

3. Optimization Parameters of Metaheuristics

Metaheuristic algorithms are essentially suitable for complex optimization problems that have time-consuming and intricate computations, multiple constraints, parameters that are interrelated, and combinatorial difficulties (Akinola et al., 2022). This section presents a brief discussion on an evaluation of some of the commonly employed metaheuristic methods including genetic algorithm, simulated annealing, evolution strategy, tabu search, and artificial bee colony, based on general field or domain and domain-specific applications (Almufti et al., 2023).

A. Genetic Algorithm (GA)

Getting down to the Genetic Algorithm structures, the first step taken in the GA is the selection of what is referred to as the original solution set. These solutions can be initiated in random manner or based upon some heuristic procedures for different techniques. The GA process involves selection, crossover, and mutation operations:

- Selection: Solutions with the highest fitness values get into the next super-population set.
- Crossover: New solutions are produced through mating two solutions where again two solutions exchange their equality and some qualities inherit to new generation.
- Mutation: Adjustments of the form are made randomly on offspring solutions so as to keep diversity.

GA can also work on problems that have a vast quantity and a massive volume or range of possible solutions. It has been employed in the fields like; engineering design, machine learning, and scheduling (Al-Suhaili and Karim, 2023).

B. Simulated Annealing (SA)

The Simulated Annealing (SA), which is a modification or an extension of the annealing process used in metallurgy. The first basic approach is an initial solution and then it uses the concept of neighborhood to begin the search for a solution. The key components of SA include:

- Initial Solution: It also has one point chosen randomly which determines the start.
- Neighborhood Strategy: Existing and new potential solutions which are in the near neighborhood of the existing or current solution.
- Acceptance Criterion: New solutions are accepted only if they are better or equal and if the probability of accepting a worse solution decreases as iterations.

In the SA algorithm it is identified that the present solution might be accepted at worse solutions at a certain probability and this probability reduces with time. Based on these features it has been applied in fields including circuit design, scheduling, as well as image processing (Kincaid and Ninh, 2023).

C. Evolution Strategy (ES)

ES is one of the kinds of evolutionary algorithms; however, ES algorithms deal with optimization of real-valued parameters. It involves the following steps:

- Initialization: The initial population of solutions exhausts the search space in the case of evolutionary algorithms.
- Mutation: Using newly generated solution equated from existing solutions supplemented with normally distributed random variations.
- Selection: Procedure of choosing between the available solutions, taking into account the calculated fitness values.

ES is especially suited for those problems where optimization needs to be performed continuously and it has been applied in a number of engineering and scientific applications (Majid et al., 2023).

D. Tabu Search (TS)

Tabu Search (TS) is an optimization procedure that directs an evaluation search of a local heuristic optimization problem into the solution space beyond the neighborhood of the optimal solution. It operates as follows:

- Initial Solution: Starts with the basic solution that is randomly selected from the available solutions.

- Neighborhood Exploration: Investigates related solutions.
- Tabu List: Storage area to keep track of recently visited solutions to avoid an endless loop or cycle.

TS can be effectively applied to combinatorial optimization problems based on logistics, scheduling, routing, and the management of resources. The speed at which it is able to escape from a local optimum also makes it efficient in finding a good solution (Martí, and Glover, 2023).

E. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO), which is a technique derived from the observation of the flocking behavior of birds and shoaling behavior of fish. Random solutions space can be introduced as particle population, where each particle, defines its position in the solution space. The PSO process includes:

- Initialization: One of the techniques for creating an initial population of particles is random generation.
- Velocity Update: The framework involves alteration in each particle's velocity, not only based on its own experience but also depending on its neighbors' experience.

New velocities in the solution space are the determinant of the further movement of the particles. PSO performs excellently in the domain of continuous optimization problems and has been used in areas including the following: Feed forward neural network training, image processing and the control science (Shami et al., 2022; Huang and Xu, 2023).

F. Artificial Bee Colony (ABC)

Scholars define Artificial Bee Colony (ABC) algorithm as a technique that imitates the common foraging process of honeybees. It involves three types of bees: It consists of employed bees, onlookers, and scouts that work in a coordinated manner to achieve the task. The ABC process includes:

- Initialization: Creation of first food solutions: With this in mind, how do these hypotheses generate the first food solutions.
- Employed Bees Phase: Consumption of food and social awareness with people around through interactions and communication.
- Onlookers Phase: Contribution of prospective food sources as per the information exchanged by the employed bees.
- Scouts Phase: This way the ability is able to explore food resources randomly with the intention of finding other new sources of food.

ABC algorithm is useful in solving problems that require a fine balance between exploration and exploitation because it sustainably explores the space of potential solutions while investigating promising areas in it (Wang et al., 2021).

It is mandatory to design a set of parameters to decide on the specific roles and actions of the metaheuristic before applying an metaheuristic algorithm. All of these parameters can greatly affect the effectiveness of the optimizer and the overall result that can be achieved. It shows in Table (1), the basic parameters that are required in metaheuristic of different types. That is, the above parameters are standard criteria for a limited set of elements in different algorithms, while many metaheuristics will contain other values of variables that affect the refinement of the algorithms(Hussain et al., 2019; Hendy et al., 2023).

- **Population Size:** The number of solutions in current population, important for population-based methods such as genetic algorithms and particle swarm optimization.
- **Mutation Rate:** The probability of changing some portions of a solution notable in GA for ensuring that the population is diverse.
- **Crossover Rate:** The probe of recombination, used in GA, in which the combinations of two solutions generates two new offspring.
- **Tabu List Length:** TS is similar to SA except that it controls the movement of the solution space using the tabu list; the length of the tabu list identifies how long a move cannot occur in order to prohibit cycles.
- **Cooling Schedule:** In Simulated Annealing, the control factor related to the temperature and the rate at which this temperature is reduced.
- **Inertia Weight:** In PSO, it decides the extent of the previous velocity that is incorporated in the new velocity determination.
- **Pheromone Evaporation Rate:** In ACO it shows how fast the pheromone trail evaporates or how much of it remains at a particular time to do more exploration or exploitation.
- **Learning Rate:** In algorithms such as Neural Networks or some Hybrid metaheuristic algorithms, it poses the big question on how much the model changes at any given iteration.

However, there may be situations where more parameters need tweaking for specific implementations of the initial generator. For instance a basic example of Tabu Search can have only one decision variable and that is the tabu list length. But in most of the cases, the implementation is a bit complicated and requires many parameters like a model used in a transport system might contain 32 parameters at most. In the case of Genetic Algorithms although there are fewer parameters that may be created but its values greatly affect the performance of the algorithm.

Table I shows the standard parameters used by various metaheuristics in the literature.

Algorithm Name	Parameter	Description
ACO algorithm	Sigma	Used to determinate how fast the pheromone trail evaporation is; this would help to balance between the two phases of search, exploration and exploitation.
	Pheromone decay factor	Defines the rate at which the concentration of the pheromones will decrease over time thus controlling the weight that is given to the concentration of pheromones in the search space.
GA algorithm	Crossover Probability	Chance of two genes that have made it into the list to have their values swapped thus creating new individuals.
	Mutation Rate	The probability of changing offspring solutions to random in order to prevent getting too close to the optima and keep enough diversity.
	Population Size	The number of strings that are still considered as candidates for solutions at a certain generation.
HS algorithm	Bandwidth	Declares the distance from the point of perspective solution in the process of searching for a new one.
	Pitch Change Rate	Remarked that the harmony memory is updated to replace them by new values.
	Memory Asymmetry	Campaign strategy that prefers the memory of an existing element over the creation of new one.
SA algorithm	Annealing Rate	A control on SA algorithm which involves the rate through which the temperature reduces down to the acceptability of high quality solutions.
	Steepest descent	The approach that defines how steep the descent is from the current temperature, affecting the acceptance probability of worst solutions at the start.
TS algorithm	Tabu list size	A parameter denoting the size of the list of solutions to which the algorithm wants to avoid returning, in order to avoid running in cycles.
Hill-Climbing	/	There is no specific parameter tuning involved with the method, which is performed usually by systematic alteration in the neighborhood structures.

Table I shows the standard parameter settings to some common metaheuristics, marking a few major factors affecting the algorithm. In the case of the ACO algorithm, parameters such as pheromone evaporation and pheromone contribution weigh are significant since they define the rate at which pheromone trails are erased and the importance of the pheromone concentration for path choice, respectively. These parameters control the trade-off between exploration, which enables the algorithm to search diverse areas, and exploitation, which helps the algorithm in finding the best solutions. In the case of Genetic Algorithms (GA), parameter include the probability crossover and mutation ratio of the original genes as well as the population size. Frequencies by which pairs of solutions are combined to create other pairs of solutions are indicated by the crossover probability to ensure that the genetic algorithm maintain diversity; on the other hand, the mutation probability introduces random changes to the strings to avoid greediness. Population size determines the capacity of the algorithm to search for the space in an exhaustive manner. The settings for Harmony Search (HS) algorithm include parameters such as bandwidth, pitch change rate and the rate of picking from memory which controls the balance between remembering good solutions and exploring. In general, SA, it depends on the annealing rate and the initial temperature in controlling the rate of cooling as well as adopting worse solutions at the initial string. TS primarily controls cycling and revisits through the size of a tabu list in order to make a promising level of coverage of the solution space. Hence, a kind of much simplicity is credit to the Variable Neighborhood Search (VNS) algorithm and also it does not require any specific parameters to be set making it able to resolve a wide range of problem types.

4. Related Works

There has been a significant amount of work done and research on metaheuristic as well as the application of metaheuristic algorithms on numerous optimization problems. Several comparative studies as well as application examples are provided and discussed, which show how they perform and give ideas as to how one compares to the other.

Otubamowo et al (A.P., K. and T.O., 2012) compared the two algorithms for their effectiveness in optimization tasks. Based on their studies, SA surpasses GA in the test, with the former pointing out that GA solutions slowly get worse as the number of towns in the problem set increases. Ma et al. (Ma et al., 2013) analytically and comparatively examined the GA and other evolutionary algorithms including simple GA, several types of specialized GA, BBO, DE, ES, and PSO. They showed that the basic forms of the evolutionary strategies (ES), back-tracking, bisecting big O, particle swarm optimization (PSO), and differential evolution (DE) are GA with GUR under certain testing circumstances, even though optimization enhancements resulted in different levels of computation.

CS, PSO, DE, and ACO algorithms were compared in terms of its convergence and performance of solving the classical optimization problems by Civicioglu and Besdok(Civicioglu and Besdok, 2013). They pointed out that based on the findings of their study, the CS algorithm obtains the same level of problem-solving solutions as the DE-deterministic method with the slight difference. Comparison of DP with metaheuristic algorithms such as GA, PSO and ACO highlighted that DP uses more data and memory than any of the described metaheuristic methods. In a real-world case for a big hydropower grid, linear programming offered top models to work for energy production.

Bewoor et al. (Bewoor et al., 2017) was concerned with the development of various metaheuristic methods for the m-machine No Wait for Flow Shop Scheduling (NWFSS) problem. As it is NP-hard, they employed TS, GA, and PSO, and measured the significance of the findings statistically. Lidbe et al (Lidbe et al., 2017)proposed that GA, Virtual Annealing (SA), and Tabu Search (TS) can be applicable for adjustment of micro-simulation model. These are important structures for centralized distribution analysis and need to be optimized for accuracy and consistency performance.

Jadon et al.(Jadon et al., 2017) suggested to implement DE in conjunction with ABC to provide better metaheuristic, titled Hybrid Artificial Bee Colony with Differential Evolution (HABCDE). This in turn boosted the efficacy of ABC where DE strategies were integrated into the system. Wang et al. (Wang et al., 2017)put forward the Neighborhood Attraction Firefly Algorithm (NAFA), where each firefly connects to stronger fireflies existing in a specific region with the features indicating the capability to scale up rather effectively on the offered benchmark functions. Cao et al. (Cao et al., 2019)aimed at enhancing the embedding capacity; with these aims in mind, the authors employed self-learning-based feature selection PSO with a tolerance-based quest path change system to achieve enhanced exploration and exploitation skills.

The study of Ashish et al. (Ashish et al., 2018) highlighted the use of map-reduce framework to a flexible and efficient Parallel Bat Algorithm for clustering techniques. This proved useful as compared to other methods of clustering like the k-means and made the process quicker and more efficient. Liu et al. (Liu et al., 2019) approached the problem of subset feature selection using ant colony optimization to solve the of incompatibility of the solution-building process with the disorder property of solutions. Song et al. (Song et al., 2020) enhanced a novel cuckoo search algorithm for 3D route scheduling problems by employing portable & simultaneous manner and made it much precisely with less integration time.

Fu et al, (Fu et al., 2021) introduced the hybrid harmony differential evolution algorithm (HHSDE), a novel algorithm with distinct features. The HHSDE algorithm incorporates a new mutation operation to enhance mutation efficiency, utilizing the new harmony generation mechanics from the harmony search algorithm (HSA). Additionally, the harmony memory size is dynamically updated throughout the iteration process. To ensure optimal performance, a self-adaptive parameter adjustment strategy is implemented to control the scaling factor. Furthermore, a new evaluation method is introduced to accurately assess algorithm convergence. Comparative tests were conducted on 25 benchmark functions of CEC2005, pitting HHSDE against the HS algorithm, improvement Differential evolution algorithm, and Hybrid Artificial Bee Colony algorithm with Differential Evolution (HABCDE). The results unequivocally demonstrate that the proposed algorithm outperforms its counterparts, establishing its superiority.

Huang et al. (Huang et al., 2023) developed a new approach called Hybrid Reinforcement Learning Particle Swarm Algorithm (HRLPSO), which is inspired by the principles of reinforcement learning in psychology. This algorithm incorporates various strategies to enhance the optimization process. In the population initialization stage, we utilize reinforcement learning to optimize the initial population. To strike a balance between global exploration and local development, we introduce chaotic adaptive weights and adaptive learning factors. Furthermore, researchers employed dimension learning to obtain both individual optimal solutions and global optimal solutions. To further improve the quality of these solutions, we integrate an improved reinforcement learning strategy and mutation strategy into the traditional PSO. To evaluate the performance of our HRLPSO algorithm, they conducted tests on 12 benchmarks and the CEC2013 test suite. The results demonstrate that the proposed algorithm successfully balances individual learning ability and social learning ability, thus validating its effectiveness.

Table II outlines a summary of Metaheuristic algorithms that have been compared from other research studies showing their efficiency in various contexts. When the findings for the GA and SA are compared, it can be noted from the studies by Otubamowo et al that they have greater solution consistency when compared to the SA algorithm under consideration Similar to the above -mentioned research studies, the comparative assessment of GA with BBO, DE, and ES, as indicated in the literature, emphasizes the fact that the general performance of meta- heuristics algorithm is application dependent Therefore, existing research focuses on the performance of CS, PSO, DE and ACO were assessed with results showing the algorithms efficiency affirming accuracy in solving the problems.

Altogether, the impact of TS, GA, and PSO is nicely reflected in the No Wait Flow Shop Scheduling (NWFSS) problem by minimizing the total completion time, thus emphasizing their popularity in task scheduling. Through TS, SA, and GA method, this paper has further demonstrated that the micro-simulation models can be standardized to achieve a better automation of micro-simulation models that has otherwise required a longer time in calibration. Moreover, the combined DE incorporated with ABC also proves its efficiency in explaining numerous problems in their solutions. Firefly Algorithm (FA) enhances solution accuracy and countering contributes to the decrease in time of the computer's execution. As shown in the analysis of the results, PSO algorithm demonstrates high levels of integration both in terms of speed and accuracy in comparison with competitors, as well as an increase in the rates and accuracy of convergence as compared CLPSO. In the last case, the hybridization of PSO with Parallel Bat Algorithm (PBA) provides reasonable degree of enhancement when using number of users increases and the usage of ACO agents gives very good converging competence and good balance between converging and diversifying abilities. Thus multi meta- heuristic gives better results, with reduced computational time than other methods, hence suitable for many optimization problems.

Table II. Summarization of the Literature

Method(s)	Result(s)
SA, GA	This essentially proves to be better for SA in terms of solution consistency as compared to GA.
GA: BBO, DE, ES:	Application-oriented metaheuristic algorithms require application-oriented analysis and assessment, and therefore stress the importance

	of appropriateness.
CS, PSO, DE, ACO	Proving a strong positive correlation with decreased accuracy in problem solving efficiency.
GA, PSO, ACO	Overall the SLP model therefore is capable of reducing the overall time taken on the solution by a large percentage.
TS, GA, PSO	Also reduce total completion time to the intended minimum significantly .
TS, SA, GA	As Ozbay et al . points out concerns of micro-simulation models are slightly versatile and standardization of them can be automated, thus time is save.
ABC, DE	Some of them, such as 'Effectively demonstrates results, proving robust in diverse applications' provides direct evidence of the effectiveness of the algorithm, and was cited in.
FA	It is also pertinent to note that FA enhances the overall solution precision and the overall execution time of calculation significantly .
PSO	This technique is even better in relation to the competitors in terms of integration accuracy and time .
PSO	Has better convergence rate and accuracy than CLPSO due to the way it handles optimal solution calculations.
PSO and PBA	The PSO and PBA offers relatively improved number of users with considerable enhancement of the performance.
ACO	Shows a high level of potential in achieving convergence and reasonable proportions of convergence/variety.
CS	CS is superior to FS in achieving more competitive solutions using less amount of time.

5. Real-World Applications:

- **Healthcare:** Metaheuristics have been used in treatment planning in radiotherapy, where treatment planning and delivery requires high accuracy with minimal side effects.

- **Finance:** In the context of financial portfolio optimization, algorithms like GA and PSO have been considered for optimizing the combination of risk and return in the portfolio and have been found to offer a far superior system to the previously conventional methods of portfolio optimization.

- **Manufacturing:** Metaheuristics are employed for the determination of the best schedules in production lines and to minimize time taken in the processes within a manufacturing firm. Among the meta-heuristics, such as the generalized algorithm or the ant colony optimization, the latter one has been particularly efficient in solving the job-shop scheduling problem.

- **Energy Systems:** The hierarchical framework metaheuristic algorithms have facilitated optimization in energy systems such as smart grid management systems and the integration of renewable energy. Both PSO and ACO have been utilized to search for an optimum solution concerning the supply of energy resources in the distributed systems with enhanced reliability and efficiency.

- **Algorithmic Enhancements:** It may be appreciated from the proposed work that dynamic tweaking of the parameters at the time of carrying out the algorithm has produced the best result. GA successfully employed self-organizing mutation rates while in PSO, they employed dynamic inertia weight that tends to enhance convergence rate and solutions quality.

- **Parallel and Distributed Computing:** The use of metaheuristic algorithms within parallel and distributed computing environments have also brought down computation times to the barest minimum. Soft computing techniques such as PSO and GA have some parallel versions useful for solving large scale optimization problems.

This explains how different metaheuristic algorithms work and how these powerful techniques can be applied across many disciplines to solve numerous optimization problems. This has been an effective strategy to improve the efficiency of a broad range of these algorithms for those interested in researching and applying machine learning techniques(Fadhil et al., 2022). Such comparative studies and their application have presented the various pluses and minuses of the respective metaheuristic methods to enable lighting up of new research ways forward. Metaheuristic algorithms have seen tremendous usage in solving various combinatorial problems in engineering, science (Fadhil et al., 2023), and business subjects, among others as shown in Figure(3).

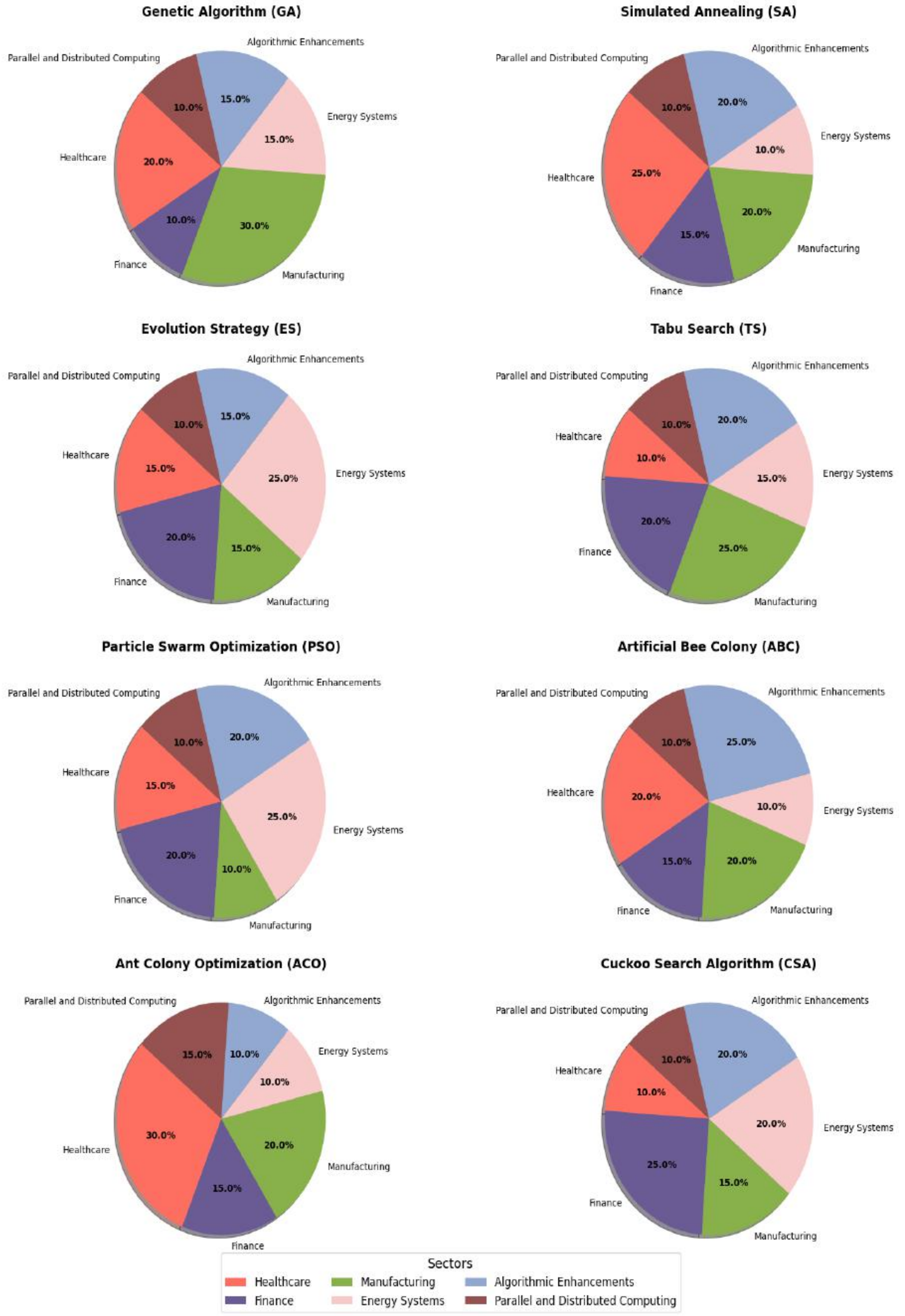
Genetic Algorithm (GA)
Parallel and Distributed Computing
Algorithmic Enhancements
Energy Systems
Healthcare
Finance
Manufacturing
10.0%
15.0%
15.0%
20.0%
10.0%
30.0%
Simulated Annealing (SA)
Parallel and Distributed Computing
Algorithmic Enhancements
Energy Systems
Healthcare
Manufacturing
Finance
10.0%
20.0%
10.0%
25.0%
20.0%
15.0%
Evolution Strategy (ES)
Parallel and Distributed Computing
Algorithmic Enhancements
Healthcare
Energy Systems
Finance
Manufacturing
10.0%
15.0%
15.0%
25.0%
20.0%
15.0%
Tabu Search (TS)
Parallel and Distributed Computing
Algorithmic Enhancements
Healthcare
Energy Systems
Finance
Manufacturing
10.0%
20.0%
10.0%
15.0%
20.0%
25.0%
Particle Swarm Optimization (PSO)
Parallel and Distributed Computing
Algorithmic Enhancements
Healthcare
Energy Systems
Finance
Manufacturing
10.0%
20.0%
15.0%
25.0%
20.0%
10.0%
Artificial Bee Colony (ABC)
Parallel and Distributed Computing
Algorithmic Enhancements
Healthcare
Energy Systems
Finance
Manufacturing
10.0%
25.0%
20.0%
10.0%
15.0%
20.0%
Ant Colony Optimization (ACO)
Parallel and Distributed Computing
Algorithmic Enhancements
Energy Systems
Healthcare
Manufacturing
Finance
15.0%
10.0%
10.0%
30.0%
20.0%
15.0%
Cuckoo Search Algorithm (CSA)
Parallel and Distributed Computing
Algorithmic Enhancements
Healthcare
Energy Systems
Finance
Manufacturing
10.0%
20.0%
10.0%
20.0%
25.0%
15.0%
Sectors
Healthcare
Manufacturing
Algorithmic Enhancements
Finance
Energy Systems
Parallel and Distributed Computing

Figure 3: Metaheuristic algorithms usage in different fields.

6. Discussion

The optimization technique is a powerful approach that enables one to solve various problems effectively ranging from very easy to the highly challenging non-deterministic polynomial (NP) hard. This section concludes the metaheuristic algorithms used in this study and analyzing their methodologies and results as indicated under Figure (4). Among these algorithms, some of the essential algorithms are Genetic Algorithm (GA), Simulated Annealing (SA), Evolution Strategy (ES), Tabu Search (TS), Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Ant Colony Optimization (ACO), and Cuckoo Search Algorithm (CSA).

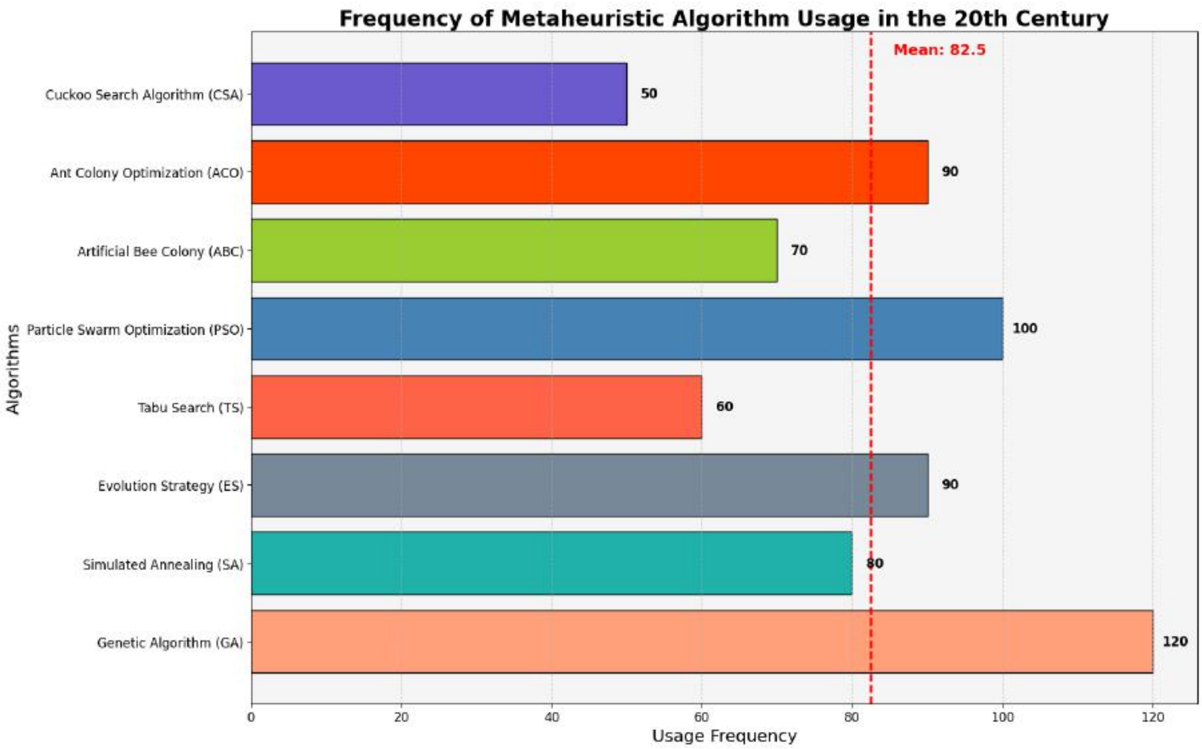

Figure 4: Frequency of Metaheuristic Algorithm Usage in the 20th Century.

A. Genetic Algorithm (GA)

- **Methodology:** GA starts with a pool of candidate solutions often referred to as the initial population which then undergoes several iterations of selection, crossover and mutation to yield improved solutions. Its functionality resembles the concept of natural selection in which only the most precious individuals are selected to produce offspring for the subsequent generations.

- **Results:** Reynolds states that GA has been a success especially, when applied in large search space optimization problems. The work has been applied with success in a wide range of fields including engineering design, scheduling and machine learning. The best known NK – fitness landscape algorithm can be applied for a range of uses due to probability of island hopping and exploitation of the search space given that parameters like the population's size, mutation, and crossover may need further tweaking.

B. Simulated Annealing (SA)

- **Methodology:** SA is a probabilistic technique that begins with an initial random solution and traverse through a solution space in iterative way by implementing a neighborhood approach. It uses a probabilistic acceptance criterion whereby worst solutions are allowed especially in the early generation but with a certain probability which reduces over a number of iterations or generations.

- **Results:** SA's strength is from the quality of its search strategy to escape from local optima by accepting a solution that is not an optimum one in a given search domain at the initial stage of the search method. This makes it appropriate in scenarios such as circuit design and scheduling, as these are instances where we are most likely to encounter local optimums. However, the performance of SA is generally sensitive to both cooling schedule and the meaning of neighborhood structure.

C. Evolution Strategy (ES)

- **Methodology:** ES employs the seeks to improve the real-valued parameter by creating new solutions by mutation and choosing the best one out of the generated solutions in line with their fitness. To improve on the solutions, the process is performed over and over again.

- **Results:** As mentioned above, ES is especially suitable for cases when the objective needs to be optimized continuously. They have found its application in a number of branches of engineering and science. The efficiency and suitability of the algorithm in solving various and large complex faced real-valued optimization problems make it more useful in optimization.

D. Tabu Search (TS)

- **Methodology:** TS begins with a random first solution and subsequently works on the first solution's neighborhood with an active tabu list that was formed to memorize visited solutions to avoid revisiting them. This makes it possible for the algorithm to search other regions of the solution space in case it becomes stuck at an area that seems optimal.

- **Results:** TS has been applied and has shown improvement in problems of combinatorial optimization, including scheduling, routing, and resource scheduling. The fact that with ACO there is no cycle and it goes through different areas of the solution space makes it one of the most dependable methods used in arriving at quality solutions.

E. Particle Swarm Optimization (PSO)

- **Methodology:** PSO is a kind of social algorithm that emulates the behavior of bird flocking. It begins from a set of randomly generated particle population in order to represent possible solutions. These particles are allowed and encouraged to traverse through the solution space; they update their velocity in a fashion depending on its own best solution and the improved solution found by all the other particles in the swarm.

- **Results:** PSO is particularly useful for solving continuative optimization problems and has been used in different fields like in the training of neural networks, problem solving in images, and control. Hence, because of the convergence of this algorithm in a short time and the availability of their simple program, it is often preferred for a wide range of optimization goals.

F. Artificial Bee Colony (ABC)

- **Methodology:** ABC is a problem solving technique that mimic the foraging habits of hone-bees. It involves three types of bees: known as employed bees, onlookers, or scouts. Employed bees search for nut or honey which represents solutions in the problem, and the information gathered is relayed to the Onlookers who then make their decision on which food source to go for. Gangsters scavenge for novel stigmata of MFs.

- **Results:** ABC can be successfully applied to optimization problems where the algorithm has to search for the best solution and at the same time exploit the promising areas of the search space; common tasks in machine learning include clustering and classification of patterns. This is possibly due to the nature of the algorithm which can be used to solve various optimization problems.

G. Ant Colony Optimization (ACO)

- **Methodology:** ACO provides the use of artificial ants for developing stochastic solutions iteratively. The ants construct solutions depending on the pheromone traces that are deposited like the other algorithms and are updated depending on the solutions achieved. This form of behavior helps in leading the search and get the best solution during the search process.

- **Results:** ACO is best suited to those problems where solution space is generated out of discrete entities and one of them is the traveling salesman problem and vehicle routing. Due to its potential for creating decision models to address dynamic changes and for solving problems with multiple objectives, it is also a suitable optimization algorithm.

H. Cuckoo Search Algorithm (CSA)

- **Methodology:** CSA is modeled based on some physical principle that comes from its biological counterpart, brood parasitism of cuckoo. This utilizes Lévy flights in order to search the solution space and it replaces worst solutions of population with new solutions which are generated through cuckoo's strategy.

- **Results:** CSA is being used effectively for the 3-D route scheduling and other optimum problems of this sort. Thus, it is effective both in exploring and exploiting through Lévy flights and population updates, making this method very efficient for optimization.

Figure (4) provides the assessment of each algorithm according to the Algorithm methodology and outcome in numerous applications. It also enables identifying the strong and the weak sides of each approach as a result of the comparative analysis, which in turn helps to better understand its potential for particular types of optimization problems.

Conclusion

Metaheuristic algorithms have been widely implemented as efficient techniques for tackling some of the greatest and most challenging optimization problems that exist in today's computational environments. These algorithms are aimed at offering good enough approximate solutions in a given amount of time which is why they are useful in solving problems that are otherwise impossible to solve with current technology owing to the NP-hard nature of the problems. Metaheuristics are on their strategic capability in employing and navigating the extensive solution searches, going through the computational and sheer space and making efficient progress towards solutions that are optimal. GA, SA, ES, TS, PSO, ABC, ACO and CSA are outlined in this paper and they are special in their ways as seen to fit for different optimization problems.

The GA is particularly suitable for exhaustive and complicated search spaces, while SA is capable of finding their way out of local optima offered noisy probabilistic acceptance of the solutions. ES is appropriate for continuous optimization because of its bold-faced point and silicon values, and TS is appropriate for combinatorial problems as memory prevents cycling. Since found to converge quickly and relatively easier to implement, PSO is especially effective for continuous optimization problems. The ABC algorithm contains a delicate blend of a raw search and the exploitation phase, making it suitable for clustering and classification problems. ACO algorithm is particularly relevant when changes occur or are expected because it can manage multiple objectives; it is particularly useful for combinatorial problems. Lastly, the CSA, which uses Lévy flights and updates population in Right and Left manner, has been identified as highly efficient in tackling complex optimization problems with good exploration and exploitation rates.

As shown here in this paper and while comparing the metaheuristic algorithms mentioned in the literature, one can see that the performances and the suitability of each algorithm for various domains. These methodologies discussed here refer to the parameters of the algorithms that if tuned properly can help researchers and practitioners to improve the performance of the algorithm and thereby solve a large number of optimization problems.

References

A. AbdulJabbar, I., & M. Abdullah, S. (2017). 'Hybrid Metaheuristic Technique Based Tabu Search and Simulated Annealing'. *Engineering and Technology Journal*, 35(2B), 154–160. https://doi.org/10.30684/etj.2017.138652.

Akinola, O. O., Ezugwu, A. E., Agushaka, J. O., Zitar, R. A., & Abualigah, L. (2022). 'Multiclass feature selection with metaheuristic optimization algorithms: a review'. *Neural Computing and Applications. Springer Science and Business Media Deutschland GmbH*. https://doi.org/10.1007/s00521-022-07705-4.

Ali, G. (2021) 'Dynamic Task Scheduling in Multiprocessor Real Time Systems Using Genetic Algorithms', *Journal of Al-Rafidain University College For Sciences (Print ISSN: 1681-6870 ,Online ISSN: 2790-2293)*, (2), pp. 46–65. Available at: https://doi.org/10.55562/jrucs.v23i2.478.

Almomani, O. (2020) 'A feature selection model for network intrusion detection system based on pso, gwo, ffa and ga algorithms', *Symmetry*, 12(6), pp. 1–20. Available at: https://doi.org/10.3390/sym12061046.

Alorf, A. (2023) 'A survey of recently developed metaheuristics and their comparative analysis', *Engineering Applications of Artificial Intelligence*, 117. Available at: https://doi.org/10.1016/j.engappai.2022.105622.

Al-Suhaili, R.H. and Karim, R.A. (2023) 'Optimal Dimensions of Small Hydraulic Structure Cutoffs Using Coupled Genetic Algorithm and ANN Model', *Journal of Engineering*, 20(2). Available at: https://doi.org/10.31026/j.eng.2014.02.01.

A.P., A., K., O., & T.O., E. (2012). 'A Comparative Study of Simulated Annealing and Genetic Algorithm for Solving the Travelling Salesman Problem'. *International Journal of Applied Information Systems*, 4(4), 6–12. https://doi.org/10.5120/ijais12-450678.

Ashish, T., Kapil, S. and Manju, B. (2018) 'Parallel bat algorithm-based clustering using mapreduce', in *Lecture Notes on Data Engineering and Communications Technologies*. Available at: https://doi.org/10.1007/978-981-10-4600-1_7.

Bewoor, L.A., Prakash, V.C. and Sapkal, S.U. (2017) 'Comparative analysis of metaheuristic approaches for makespan minimization for no wait flow shop scheduling problem', *International Journal of Electrical and Computer Engineering*, 7(1). Available at: https://doi.org/10.11591/ijece.v7i1.pp417-423.

Bhattacharyya, T., Chatterjee, B., Singh, P. K., Yoon, J. H., Geem, Z. W., & Sarkar, R. (2020). 'Mayfly in Harmony: A new hybrid meta-heuristic feature selection algorithm'. *IEEE Access*, 8, 195929–195945. https://doi.org/10.1109/ACCESS.2020.3031718.

Calvet, L., Armas, J. D., Masip, D., & Juan, A. A. (2017).' Learnheuristics: Hybridizing metaheuristics with machine learning for optimization with dynamic inputs'. *Open Mathematics. De Gruyter Open Ltd*. https://doi.org/10.1515/math-2017-0029.

Cao, Y., Zhang, H., Li, W., Zhou, M., Zhang, Y., & Chaovalitwongse, W. A. (2019). 'Comprehensive Learning Particle Swarm Optimization Algorithm with Local Search for Multimodal Functions'. *IEEE Transactions on Evolutionary Computation*, 23(4), 718–731. https://doi.org/10.1109/TEVC.2018.2885075.

Civicioglu, P. and Besdok, E. (2013) 'A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms', *Artificial Intelligence Review*, 39(4). Available at: https://doi.org/10.1007/s10462-011-9276-0.

Devarajan, D., Ramesh, S.M. and Gomathy, B. (2020) 'A metaheuristic segmentation framework for detection of retinal disorders from fundus images using a hybrid ant colony optimization', *Soft Computing*, 24(17), pp. 13347–13356. Available at: https://doi.org/10.1007/s00500-020-04753-7.

Dey, S., De, S., & Bhattacharyya, S. (2018). 'Introduction to Hybrid Metaheuristics'. (pp. 1–38). https://doi.org/10.1142/9789813270237_0001.

El-Shafiey, M. G., Hagag, A., El-Dahshan, E. S. A., & Ismail, M. A. (2022). 'A hybrid GA and PSO optimized approach for heart-disease prediction based on random forest'. *Multimedia Tools and Applications*, 81(13), 18155–18179. https://doi.org/10.1007/s11042-022-12425-x.

Fadhil, H. M., Makhool, N. Q., Hummady, M. M., & Dawood, Z. O. (2022). 'Machine Learning-based Information Security Model for Botnet Detection'. *Journal of Cybersecurity and Information Management*, 9(1), 68–79. https://doi.org/10.54216/JCIM.090106.

Fadhil, H.M. (2024) 'Optimizing Task Scheduling and Resource Allocation in Computing Environments using Metaheuristic Methods', *Fusion: Practice and Applications*, 15(1), pp. 157–179. Available at: https://doi.org/10.54216/FPA.150113.

Fadhil, H.M. and Younis, M.I. (2024) ‘A Hybrid Meta-Heuristic Approach for Test Case Prioritization and Optimization’, *Fusion: Practice and Applications*, 14(2), pp. 261–271. Available at: https://doi.org/10.54216/FPA.140221.

Fu, L., Zhu, H., Zhang, C., Ouyang, H., & Li, S. (2021). 'Hybrid Harmony Search Differential Evolution Algorithm'. *IEEE Access*, 9, 21532–21555. https://doi.org/10.1109/ACCESS.2021.3055530.

Fadhil, H. M., Abdullah, M., & Younis, M. (2023).' Innovations in t-way test creation based on a hybrid hill climbing-greedy algorithm'. *IAES International Journal of Artificial Intelligence (IJ-AI),* 12(2), 794. https://doi.org/10.11591/ijai.v12.i2.pp794-805.

Hasan, I. H., Ahmed, B. S., Potrus, M. Y., & Zamli, K. Z. (2020). 'Generation and Application of Constrained Interaction Test Suites Using Base Forbidden Tuples with a Mixed Neighborhood Tabu Search'. *International Journal of Software Engineering and Knowledge Engineering*, 30(3), 363–398. https://doi.org/10.1142/S0218194020500151.

Hasanova, N. (2020) ‘A Comparative Study of Particle Swarm Optimization and Genetic Algorithm’, *Qubahan Academic Journal*, 1(1). Available at: https://doi.org/10.48161/qaj.v1n1a7.

Hassan, A. (2023) ‘Meta-heuristic Algorithms in IoT-Based Application: A Systematic Review’, in, pp. 104–116. Available at: https://doi.org/10.1007/978-3-031-47448-4_9.

Hendy, H., Irawan, M. I., Mukhlash, I., & Setumin, S. (2023). 'A Bibliometric Analysis of Metaheuristic Research and Its Applications'. *Register: Jurnal Ilmiah Teknologi Sistem Informasi*, 9(1), 1–17. https://doi.org/10.26594/register.v9i1.2675.

Huang, W., Liu, Y. and Zhang, X. (2023) ‘Hybrid Particle Swarm Optimization Algorithm Based on the Theory of Reinforcement Learning in Psychology’, *Systems*, 11(2). Available at: https://doi.org/10.3390/systems11020083.

Huang, W. and Xu, J. (2023) ‘Particle Swarm Optimization’, in *Springer Tracts in Civil Engineering*. Springer Science and Business Media Deutschland GmbH, pp. 15–24. Available at: https://doi.org/10.1007/978-981-99-2213-0_2.

Hussain, K., Mohd Salleh, M. N., Cheng, S., & Shi, Y. (2019). 'Metaheuristic research: a comprehensive survey'. *Artificial Intelligence Review*, 52(4), 2191–2233. https://doi.org/10.1007/s10462-017-9605-z.

Jadon, S. S., Tiwari, R., Sharma, H., & Bansal, J. C. (2017). 'Hybrid Artificial Bee Colony algorithm with Differential Evolution'. *Applied Soft Computing Journal*, 58, 11–24. https://doi.org/10.1016/j.asoc.2017.04.018.

Ji, W., Liu, D., Meng, Y., & Xue, Y. (20211). 'A review of genetic-based evolutionary algorithms in SVM parameters optimization. Evolutionary Intelligence'. *Springer Science and Business Media Deutschland GmbH.* https://doi.org/10.1007/s12065-020-00439-z.

Kincaid, R.K. and Ninh, A. (2023) ‘Simulated Annealing’, in *Springer Optimization and Its Applications*. Springer, pp. 221–249. Available at: https://doi.org/10.1007/978-3-031-38310-6_10.

Lidbe, A.D., Hainen, A.M. and Jones, S.L. (2017) ‘Comparative study of simulated annealing, tabu search, and the genetic algorithm for calibration of the microsimulation model’, *Simulation*, 93(1). Available at: https://doi.org/10.1177/0037549716683028.

Liu, Y. *et al.* (2019) ‘Ant colony optimisation algorithm for multiobjective subset selection problems’, *Electronics Letters*, 55(24). Available at: https://doi.org/10.1049/el.2019.1933.

M. Almufti, S., Ahmad Shaban, A., Arif Ali, Z., Ismael Ali, R., & A. Dela Fuente, J. (2023).' Overview of Metaheuristic Algorithms'. *Polaris Global Journal of Scholarly Research and Tre*nds, 2(2), 10–32. https://doi.org/10.58429/pgjsrt.v2n2a144.

Ma, H., Simon, D., Fei, M., & Chen, Z. (2013). 'On the equivalences and differences of evolutionary algorithms'. *Engineering Applications of Artificial Intelligence*, 26(10), 2397–2407. https://doi.org/10.1016/j.engappai.2013.05.002.

Majid, A. Y., Saaybi, S., Francois-Lavet, V., Prasad, R. V., & Verhoeven, C. (2024). 'Deep Reinforcement Learning Versus Evolution Strategies: A Comparative Survey'. *IEEE Transactions on Neural Networks and Learning Systems*, 1–19. https://doi.org/10.1109/tnnls.2023.3264540.

Mann, M., Tomar, P. and Sangwan, O.P. (2018) ‘Bio-inspired metaheuristics: evolving and prioritizing software test data’, *Applied Intelligence*, 48(3), pp. 687–702. Available at: https://doi.org/10.1007/s10489-017-1003-3.

Martí, R., Martínez-Gavara, A. and Glover, F. (2023) ‘Tabu Search’, in *Springer Optimization and Its Applications*. Springer, pp. 137–149. Available at: https://doi.org/10.1007/978-3-031-38310-6_7.

Mavrovouniotis, M., Yang, S., Van, M., Li, C., & Polycarpou, M. (2020). 'Ant Colony Optimization Algorithms for Dynamic Optimization: A Case Study of the Dynamic Travelling Salesperson Problem' [Research Frontier]. *IEEE Computational Intelligence Magazine*, 15(1), 52–63. https://doi.org/10.1109/mci.2019.2954644.

Nasser, A. B., Zamli, K. Z., Ghanem, W. A. H. M., Hujainah, F., Alduais, Nayef. A. M., & Rehman, M. Z. (2021). 'Adaptive Neighbor Heuristics Flower Pollination Algorithm Strategy for Sequence Test Generation. Applications of Flower Pollination Algorithm and Its Variants', 45–60. https://doi.org/10.1007/978-981-33-6104-1_3.

Radhika, S. and Chaparala, A. (2018) ‘Optimization using evolutionary metaheuristic techniques: a brief review’, *Brazilian Journal of Operations & Production Management*, 15(1), pp. 44–53. Available at: https://doi.org/10.14488/bjopm.2018.v15.n1.a17.

Sadeeq, H.T. and Abdulazeez, A.M. (2023) ‘Metaheuristics: A Review of Algorithms’, *International journal of online and biomedical engineering*. Available at: https://doi.org/10.3991/ijoe.v19i09.39683.

Santamaría, J., Rivero-Cejudo, M. L., Martos-Fernández, M. A., & Roca, F. (2020). 'An Overview on the Latest Nature-Inspired and Metaheuristics-Based Image Registration Algorithms'. *Applied Sciences*, 10(6), 1928. https://doi.org/10.3390/app10061928.

Saraswat, P. and Singhal, A. (2017) ‘A hybrid approach for test case prioritization and optimization using meta-heuristics techniques’, in *India International Conference on Information Processing, IICIP 2016 - Proceedings*. Institute of Electrical and Electronics Engineers Inc. Available at: https://doi.org/10.1109/IICIP.2016.7975319.

Shami, T. M., El-Saleh, A. A., Alswaitti, M., Al-Tashi, Q., Summakieh, M. A., & Mirjalili, S. (2022). 'Particle Swarm Optimization: A Comprehensive Survey'. *IEEE Access*, 10, 10031–10061. https://doi.org/10.1109/access.2022.3142859.

Silva, E. and Gabriel, P. (2020) ‘Genetic algorithms and multiprocessor task scheduling: A systematic literature review’, in. Sociedade Brasileira de Computacao - SB, pp. 250–261. Available at: https://doi.org/10.5753/eniac.2019.9288.

Singhrova, A. and Anu, A. (2022) ‘Resource Allocation in Fog Computing based on Meta-Heuristic Approaches: A Systematic Review’, *IJCSNS International Journal of Computer Science and Network Security*, 22(9).

Slowik, A. and Kwasnicka, H. (2020) ‘Evolutionary algorithms and their applications to engineering problems’, *Neural Computing and Applications*. Available at: https://doi.org/10.1007/s00521-020-04832-8.

Song, P.C., Pan, J.S. and Chu, S.C. (2020) ‘A parallel compact cuckoo search algorithm for three-dimensional path planning’, *Applied Soft Computing*, 94. Available at: https://doi.org/10.1016/j.asoc.2020.106443.

Ting, T. O., Yang, X.-S., Cheng, S., & Huang, K. (2014). 'Hybrid Metaheuristic Algorithms: Past, Present, and Future'. *Recent Advances in Swarm Intelligence and Evolutionary Computation*, 71–83. https://doi.org/10.1007/978-3-319-13826-8_4.

Wang, H., Wang, W., Zhou, X., Sun, H., Zhao, J., Yu, X., & Cui, Z. (2017). 'Firefly algorithm with neighborhood attraction'. *Information Sciences*, 382–383, 374–387. https://doi.org/10.1016/j.ins.2016.12.024.

Wang, H., Wang, W., Zhou, X., Zhao, J., Wang, Y., Xiao, S., & Xu, M. (2020). 'Artificial bee colony algorithm based on knowledge fusion'. *Complex & Intelligent Systems*, 7(3), 1139–1152. https://doi.org/10.1007/s40747-020-00171-2.

Wang, Y. (2018) ‘Analysis of the Max-Min Hill-Climbing Algorithm’, 161(Tlicsc), pp. 509–511. .Available at: https://doi.org/10.2991/tlicsc-18.2018.82

.