Research Paper

Metaheuristic algorithms for complex optimization: a critical review of foundations, hybrid strategies, and surrogate modeling

Awaz Ahmed Shaban[1*], Saman M. Almufti[2], Renas Rajab Asaad[3], Rasan Ismael Ali[4]

[1*,2,3]Information Technology Department, Technical College of Informatics-Akre, Akre University for Applied Sciences, Duhok, Iraq.

[4]Department of Computer Science, College of Science, Knowledge University, Erbil, Iraq.

Article Info

Article History:
Received: 15 July 2025
Revised: 25 September 2025
Accepted: 03 October 2025
Published: 17 November 2025

Keywords:
Metaheuristic Algorithms
Hybrid Optimization
Surrogate Models
Combinatorial Optimization
Swarm Intelligence

ABSTRACT

Metaheuristic algorithms have emerged as indispensable tools for solving complex optimization problems across engineering, logistics, and data-driven decision-making domains. Unlike traditional methods that require strict assumptions or derivative information, metaheuristics provide flexible frameworks capable of delivering high-quality, near-optimal solutions in high-dimensional, multimodal, and constrained search spaces. This paper presents a comprehensive and critical survey of metaheuristic algorithms, offering a structured taxonomy that spans single-solution and population-based methods as well as classical and emerging approaches. Special attention is devoted to hybridization strategies and surrogate-assisted techniques, which are increasingly employed to enhance convergence speed, robustness, and scalability in computationally expensive environments. Drawing upon recent studies, we analyze strengths, limitations, and parameterization challenges while identifying persistent gaps in reproducibility, adaptive control, and large-scale applicability. Finally, we propose a forward-looking roadmap emphasizing adaptive parameter tuning, ensemble surrogates, and parallel or distributed computing to guide the next generation of metaheuristic optimization frameworks. This synthesis provides researchers and practitioners with an authoritative reference for selecting, adapting, and innovating metaheuristic algorithms in diverse application contexts.

Corresponding Author:
Awaz Ahmed Shaban
Information Technology Department, Technical College of Informatics-Akre, Akre University for Applied Sciences, Duhok, Iraq.
Email: Awaz.Amedy1@gmail.com

1. INTRODUCTION

Optimization plays a pivotal role in modern science, engineering, and decision-making, encompassing an extraordinarily wide range of domains such as structural and mechanical design, transportation and logistics, finance, energy systems, and artificial intelligence. The main aim in such areas is to find the optimal setting of the decision variables, which might mean either minimizing costs, having the highest performance, or getting a good compromise among the conflicting objectives [1]. Traditional optimization methods, which are based on gradient techniques, linear or nonlinear programming, and exact combinatorial techniques, have been the primary tools for these kinds of problems for a long time. However, they often suffer from major limitations when they are brought into the real world with its complexities [2], [3]. The presence of high-dimensional decision spaces, non-smooth and non-convex objective functions, noisy or costly evaluations, and very tight constraints are some of the reasons why such methods are sometimes considered impractical or they spend a lot of time and resources.

Metaheuristics, which have been inspired by nature, physics, and human cognition, are now seen as an effective and flexible way of problem-solving that can provide good and almost optimal answers without the need for derivative information or strong assumptions about the search area. Metaheuristics include a wide range of strategies from simulated annealing and tabu search to swarm intelligence, evolutionary algorithms, and the newest bio inspired methods. Their power to smoothly transition between exploration (broadening the search to steer clear of local optimum) and exploitation (concentrating the search around the most promising solutions) is what makes them very suitable for optimization problems that are both expensive and of large scale [4].

Even though they are widely used, and their development continues, there are still some issues that need to be resolved. A number of the existing metaheuristics depend on fixed or problem-specific parameter settings, which results in variability and unpredictability in their performance. Hybridization (mixing various algorithms or coupling local search heuristics) and surrogate modeling (estimation of expensive objective functions through cheap predictive models) have proved to be very powerful yet lacking a systematic integration needed for a unified framework. This causes practitioners to be unclear as to which method or hybrid strategy is renderable depending on the problem, while at the same time making it difficult for the metaheuristics to be applied to really large and complicated areas of the real world [5].

In response to these problems, this article puts forth a detailed and critical review of metaheuristic algorithms, their theoretical foundations, and practical usage. The citation of our contributions is made threefold. The first one is the provision of a categorized taxonomy of the preceding heuristic methods that points out their operation mechanisms, parameterization demands, and convergence characteristics across single-objective, multi-objective, and constrained optimization problems. The second is the discussion of hybridization and surrogate modeling roles being the twin strategies for improving efficiency, robustness, and scalability with recent research developments in ensemble and adaptive surrogate-assisted optimization. The third is the delineation of a looking ahead roadmap for metaheuristic development where adaptive parameter control, real-time optimization, and distributed computing are marked as key facilitators of next-generation optimization frameworks.

2. RELATED WORK

The swift changes in the metaheuristic algorithms have resulted in a plethora of literature that is not only vast but also varied. This review article presents the key advancements, classes of algorithms identified, comparative evaluations done, and the research gaps yet to be filled.We emphasize how our work situates within and advances this landscape.

2.1. Classification and Taxonomy of Metaheuristics

Tomar presents a comprehensive summary by classifying metaheuristics into evolutionary, swarm, physics-inspired, human-inspired and hybrid groups, and through comparing their main operators, mathematical models and typical domains (e.g. combinatorial vs. continuous). Tomar further ranks the algorithms with their strengths and weaknesses according to the problem types of scheduling, structural optimization, and data mining [6].

Elaborate on that taxonomy by concentrating on the recently suggested metaheuristics during 2019-2024. Their review of the survey is over 150 novel algorithms, which are then classified according to the nature of their inspirations, novelty of operators introduced, adaptivity of parameter control, and the diversity of benchmark problems they addressed. Maintain that the overwhelming number of metaphor-based algorithms has obscured the distinction between substantial innovation and minor rebranding [7], [8]. Velasco, Guerrero, and Hospitaler conduct a critical examination on a group of 111 so-called new or improved metaheuristics. They observe that 65 % of the proposals in question are nothing more than variants of the existing algorithms, and only 43 % cite the No Free Lunch theorem as part of their reasoning. This indicates a pattern: the release of metaheuristic variants without thorough theoretical backing or empirical validation has turned into a deluge [9].

Provide a more specific historical perspective by reviewing the 20 years of GRASP (Greedy Randomized Adaptive Search Procedure) combined with Path Relinking. Their research demonstrates how the blending of methods can result in a more consistent performance in the search, and also shows the progress in memory structures and path-guided intensification [6], [10]. Focus on the Biased Random-Key Genetic Algorithm (BRKGA). Their review not only analyzes but also provides a comprehensive synthesis of more than 150 studies which have utilized BRKGA in the areas of scheduling, network design, and industrial optimization. Nonetheless, the authors bring to light the cases in which local search or domain-specific heuristics are resorted to for hybridization of BRKGA and thus the improvement of convergence [11], [12], [13].

2.2. Comparative Studies and Application Surveys

Present a meta-analysis of the total of 1,676 articles from the period 1994 to 2023, in which the application of metaheuristics in different areas is analyzed and the evolution of the methodological rigor is discussed. The applications are grouped, the most popular algorithm families are marked out, and the publication trends are detected. According to Li, although swarm-based and evolutionary methods take the lead, hybrid and surrogate-assisted techniques are gradually appearing more frequently [9]. Perform a thorough review of hybrid metaheuristics with applications in feature selection in machine learning covering the period from 2019 to 2023. Apart from the review of hybridization strategies (e.g. two-stage methods, memetic integration), the authors also investigate fitness functions, transfer mechanisms, evaluation metrics, statistical tests, and dataset types. One of the main findings of their review is the overwhelming preference for single-objective hybrids and the call for more multi-objective hybrid explorations in future research [14].

Discuss metaheuristic algorithms used in a variety of healthcare applications which include diagnosis, scheduling, image segmentation, and treatment planning. They present a summary of the popular metaheuristic algorithms (GA, PSO, DE), and the approaches to handling medical constraints are pointed out along with the discussion of the interpretability and validation problems that are specific to the clinical settings. Agrawal perform a systematic review in path planning issues, examining used strategies, trade-offs (real-time vs. optimal paths), and performance metrics in the contexts of both robot and autonomous vehicle navigation [6]. The research literature across different fields always points out two main trends: one is the mixture of global exploration with local refinement through hybridization, and the second is using surrogate or approximation models to reduce the computational cost incurred in expensive evaluations.

2.3. Hybrid Metaheuristics and Surrogate-Assisted Methods

Hybrid metaheuristics combining two or more algorithms or embedding local heuristics have become a dominant theme in recent research. An instance is Othman and Ku-Mahamud who claim that most hybrid feature selection approaches follow, to a very loose extent, the typical “global search + local optimization” pattern. There are many hybrids among those which take and modify conventional algorithms (e.g. GA, PSO) by introducing the specific domain-inspired perturbations or applying the local search heuristics on them [15], [16], [17]. The paper of sharply criticizes the number of hybrid proposals that are excessively and completely unoriginal at the same time; moreover, the authors suggest that some of the new hybrids come from the recombination of known algorithmic blocks without the slightest support of theory. The authors also call for the implementation of stringent ablation studies, comparisons with algorithmic baselines, and sharing of code to enhance reproducibility.

Metaheuristics supported by surrogates deal with the costly evaluation issue by substituting a portion of actual evaluations with approximate models like Kriging, radial basis functions, polynomial regression, or neural networks. The use of these surrogate models makes it possible for the algorithms to very quickly explore the candidate solutions. Some hybrid strategies alternate between surrogate-guided prediction and true evaluation, while others use ensemble surrogates or active learning to query the true function selectively [18], [19]. Nonetheless, few application-oriented comparative studies rigorously compare surrogate-assisted hybrids against pure metaheuristics under equal computational budgets. This is a notable gap: researchers often report absolute improvements but omit clarity about overheads of surrogate training or model management.

2.4. Critiques, Gaps, and Methodological Challenges

The metaheuristic literature is not without its criticisms. Velasco and they also refer to older debates which maintain that a large proportion of the “new” metaheuristics are nothing but metaphorical renamings or slight adaptations of the existing paradigms. Theoretical justification, strict empirical benchmarks, or open code, the booming of metaheuristic variants risks limiting the field’s [20], [21] progress. Li argue that a lot of studies present "best-of-run" results without any statistical testing, running the process several times, or analyzing robustness under noise. This weakens the claims of superiority of the algorithms, especially considering the stochastic characteristics of these methods.

Another gap lies in scalability and dimensionality. Most validation remains on low-to-moderate dimensional benchmarks. Real-world problems with hundreds or thousands of decision variables remain sparsely explored. Saman notes the need for both scalable algorithms and efficient surrogate modeling to approach those domains [22], [23].

Furthermore, repeatability and transparency remain weak. Even fewer coders, parameter settings, or detailed workflows are published in papers which causes a lack of reproducibility. Previous papers urge that reviewers and editors raise standards to demand substantive novelty, reproducibility, and clarity [24], [25], [26]. Finally, adaptive parameter control and self-adjusting metaheuristics remain underdeveloped. Find that many “novel” algorithms still use fixed coefficients or schedules. More research is needed on integrating reinforcement learning, machine learning, and online adaptation to modulate search dynamics [27], [28].

3. METAHEURISTIC ALGORITHMS: BACKGROUND AND FUNDAMENTALS

Metaheuristics are strategies ranked at a very high level that solve problems and offer operators that are flexible and of general purpose, so that they could be used for complicated optimization problems. Metaheuristics are not like exact optimization methods, which can only guarantee global optimality under very strict conditions, as they are practicality and scalability oriented and can still produce high-quality solutions even for very large, nonlinear, and non-convex search spaces [29] within a reasonable time of computer power. The reason for these algorithms' triumph is their ability to manage the two opposites of exploration diversification of the search across new regions of the solution space and exploitation the intensification of the search around promising areas to refine candidate solutions simultaneously. This very balancing act enables it to be the best through metaheuristics to avoid premature convergence and, at the same time, keep a large diversity of solutions throughout the process of optimization [30].

3.1. Core Principles of Metaheuristics

Such diverse inspirations as natural selection to physical annealing and collective animal behavior lie behind most metaheuristics, yet they all share three basic principles:

- **Exploration (Diversification):** Systematic probing of previously unvisited or underexplored regions of the search space. Such effective exploration helps the algorithms to escape the local optima and also enhances their ability to discover the globally competitive solutions [31].
- **Exploitation (Intensification):** The focused refinement of candidate solutions in the promising regions, which enhances the speed of convergence as well as the quality of the solution.
- **Memory Structures (Learning):** A variety of mechanisms can be utilized for recording information about past solutions or search routes and then reusing it. Among the examples are tabu lists in Tabu Search, pheromone trails in Ant Colony Optimization, and archives in multi-objective evolutionary algorithms [32].

Most of the time these principles are put into practice with parameters and thus through cooling schedules, inertia weights, mutation rates, or the like, allowing the practitioners to adjust the balance of the two opposite forces, namely exploration and exploitation, according to the characteristics of the problem.

3.2. Classification of Metaheuristic Algorithms

A general classification of metaheuristic methods can be made according to the nature (population structure) and the source of inspiration:

- **Single-Solution-Based Metaheuristics:** The sequence of operations of these algorithms involves one candidate solution at a time, and gradually it is improved through moves to the neighborhood or stochastic perturbations. Simulated Annealing (SA) and Tabu Search (TS) are typical representatives that are particularly suitable for discrete or combinatorial optimization [21], [33].
- **Population-Based Metaheuristics:** These techniques work with a population of potential solutions taking advantage of the inherent diversity and exploring simultaneously. Genetic Algorithms (GA) [34], Particle Swarm Optimization (PSO) [35], Differential Evolution (DE) [36], [37], and Ant Colony Optimization (ACO) [38], [39], [40], [41] are among the most significant ones.
- **Hybrid and Adaptive Metaheuristics:** In today's algorithms it is a common practice to mix the advantages of different methods e.g., local search committed to population-based algorithms or surrogate models used for quicker convergence and wider scalability [42], [43], [44].

3.3. Advantages Over Traditional Methods

The application of a metaheuristic method offers the following benefits compared to gradient-based or deterministic methods [45]:

- **Derivative-Free Operation:** The information of the gradient or Hessian is not needed, thus making the methods applicable to black-box and discontinuous functions [46].
- **Global Search Capability:** Their stochastic nature and built-in diversification mechanisms assist them in escaping local minima [47].
- **Scalability and Flexibility:** They could be tailored to solve a variety of problems, such as single-objective, multi-objective, constrained, and dynamic optimization [48].

Nevertheless, these metaheuristic advantages come with a caveat: they usually do not assure global optimality and their results can heavily depend on the settings of the parameters. This, in turn, advocates for the adoption of mechanisms such as adaptive parameter control, surrogate-assisted evaluation, and hybridization, which will be discussed in later sections of this document. Metaheuristics are powerful and flexible problem-solving strategies, which operate as high-level approaches allowing through their very nature and phasing to handle difficult optimization computations. Unlike classical optimization methods that are limited to a few conditions and thus, guarantee global optimum in those scenarios, metaheuristics have emphasized, besides, an aspect of the very nature of the algorithm that is practical and scalable thus yielding high-quality solutions, in the period of time, though even for large and nonlinear, nonconvex search spaces [1]. The power of these algorithms comes from the capability of treating their very nature as a duality of exploration the diversification of the search across the new regions of the solution space and exploitation the intensification of the search around the promising areas to refine candidate solutions. This equilibrium allows metaheuristics to dodge getting confined in one point too early and at the same time sustain a good variety of solutions during the whole process of optimization [2].

3.4. Core Principles of Metaheuristics

Even though their inspirations are varied coming from nature like selection to physical annealing or collective animal behavior, most metaheuristic approaches still hold the same three principles:

- **Exploration (Diversification):** A methodical search in the previously unvisited or less explored areas of the search space. Good exploration makes it easier for the algorithms to get out of local optima and thus, their ability to discover globally winning solutions is improved [3].
- **Exploitation (Intensification):** The concentrated improvement of candidate solutions in such regions that they are already promising, which leads to faster convergence and better quality of the solutions.
- **Memory Structures (Learning):** Ways of recording and reusing the information about past solutions or search trajectories. Such mechanisms are found in tabu lists of Tabu Search, pheromone trails of Ant Colony Optimization, and archives of multi-objective evolutionary algorithms, for instance [4].

These principles are frequently applied in a manner that is parameterized through cooling schedules, inertia weights, mutation rates, or pheromone evaporation factors allowing the practitioners to regulate the balance of power between exploration and exploitation depending on the characteristics of the problem.

3.5. Classification of Metaheuristic Algorithms

Metaheuristics can be sorted out in large categories based on their population structure and the source of inspiration:

- **Single-Solution-Based Metaheuristics:** The algorithms in this category work only on one candidate solution at a time and improve it iteratively through neighborhood moves or stochastic perturbations. Typical cases are Simulated Annealing (SA) and Tabu Search (TS), which are very good for discrete or combinatorial optimization problems [5], [6].
- **Population-Based Metaheuristics:** The procedures under this technique work on a population of candidate solutions all at once, thus, allowing the existence of natural diversity and simultaneous exploration. The most significant ones include Genetic Algorithms (GA) [7], Particle Swarm Optimization (PSO) [8], Differential Evolution (DE) [9], and Ant Colony Optimization (ACO) [10], [11].
- **Hybrid and Adaptive Metaheuristics:** The trend nowadays in developing the algorithms is to use the strengths of more than one method—like combining local search with population-based algorithms or embedding surrogate models in order to get faster convergence and better scalability.

3.6. Advantages Over Traditional Methods

The use of metaheuristic optimization techniques has certain advantages over gradient-based or deterministic methods, as mentioned in [12].

- **Derivative-Free Operation:** They are able to work without the need for gradient or Hessian information and hence are applicable for black-box and discontinuous functions [13].
- **Global Search Capability:** The random aspect of these methods together with their built-in diversification strategies will surely assist them in getting out of local minima [14].
- **Scalability and Flexibility:** They can be tailored to fit different kinds of problems such as single-objective, multi-objective, constrained, and dynamic optimization etc. [15].

At the same time, these advantages are accompanied by a major drawback: global optimality is usually not guaranteed and the performance of metaheuristics can be heavily dependent on parameter settings. Thus, the need for adaptive parameter control, surrogate-assisted evaluation, and hybridization is evident, which are the topics that will be discussed in the subsequent sections of this paper.

4. DETAILED ALGORITHM DESCRIPTIONS

4.1. Simulated Annealing (SA)

Simulated Annealing is modeled after the annealing process that is used in metallurgy, where a material is heated and slowly cooled so that a strong crystalline structure is obtained14. The algorithm begins with an initial solution and subsequently explores the neighborhood iteratively, accepting better solutions and occasionally worse solutions based on a probability which is dictated by a temperature parameter [16].

Algorithm Steps

1. **Initialization:** Begin with an initial solution x and a high temperature T.
2. **Neighbor Generation:** Create a neighbor x' for the current solution x.
3. **Evaluation:** Find out the change in the objective function value: Δ f = f(x') - f(x).
4. **Acceptance Criterion**
   - If Δ f <0, the new solution x=x^' is accepted.
   - If Δ f ≥0, the new solution is accepted with probability $p = e^{-\frac{\Delta f}{T}}$.
5. **Temperature Update:** The temperature T is reduced by the cooling factor α (e.g., T=α·T, where 0<α<1).
6. **Termination:** Conduct steps 2-5 again and again until a stopping criterion is fulfilled (e.g., max number of iterations or temperature is low enough).

4.2. Tabu Search (TS)

Tabu Search is a metaheuristic optimization algorithm that enhances the local search process by using memory structures which not only avoid the repetition of previous solutions but also make new areas of the solution space available34. Moving from one solution to another in the neighborhood and maintaining a tabu list of solutions that have been visited recently to avoid revisiting those solutions for a certain number of iterations is how it works [17].

Algorithm Steps

1. **Initialization:** Start with an initial solution x and an empty tabu list.
2. **Neighborhood Generation:** Generate a set of candidate neighbor solutions $N(x)$.
3. **Selection:** Choose the best neighbor x' from $N(x)$ that is not in the tabu list (or meets an aspiration criterion).
4. **Tabu Update:** Add the current solution x to the tabu list and remove the oldest entry if the list is full.
5. **Move:** Move to the selected neighbor $x = x'$.
6. **Termination:** Repeat steps 2-5 until a stopping criterion is met.

4.3. Genetic Algorithms (GA)

Genetic Algorithms are nature's way of doing things through selection and genetics, which then applies crossover, mutation, and selection as their mechanisms to transform the population of solutions into those with better fitness24. Every single solution is represented by a chromosome and takes an active part in the genetic operations to produce new offspring [18], [19].

Algorithm Steps

1. **Initialization:** Create an initial population of solutions (chromosomes).
2. **Evaluation:** Evaluate the fitness of each solution in the population.
3. **Selection:** Choose parent solutions according to their fitness.
4. **Crossover:** The two parents will share their genetic traits, and thus new offspring will be created.
5. **Mutation:** The offspring will undergo random changes to maintain diversity among them.
6. **Replacement:** The new generation will consist of only the infants while the adults will be eliminated.
7. **Termination:** Carry out steps 2 to 6 until a stopping condition is satisfied.

4.4. Particle Swarm Optimization (PSO)

Particle Swarm Optimization is a nature-like metaheuristic that works with a lot of people (particles) and is modeled after the social behavior of birds flocking or fish schooling34. A population of particles, each one a potential solution, is in constant motion, and their positions and velocities are updated iteratively based on two factors: the best-known position for that particle and the best-known position for the entire swarm [20].

Algorithm Steps

1. **Initialization:** The particles in the swarm will initially have random positions and speeds.
2. **Evaluation:** Determine the performance of each particle with respect to the objective.
3. **Update Personal Best:** If a particle's current position is better than its personal best, then it will be marked as the new personal best.
4. **Update Global Best:** The best personal best in the swarm will define the new global best position.
5. **Update Velocity and Position:** Each particle's velocity and position are updated based on the equations below:

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot (p_{best,i} - x_i(t)) + c_2 \cdot r_2 \cdot (g_{best} - x_i(t))$$

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

Where v_i is the velocity of particle i, x_i is the position of particle i, w is the inertia weight, c_1 and c_2 are the acceleration coefficients, r_1 and r_2 are the random numbers between 0 and 1, $p_{best,i}$ is the personal best position of particle i, and g_{best} is the global best position.

6. **Termination:** Repeat steps 2-5 until a stopping criterion is met.

4.5. Ant Colony Optimization (ACO)

Ant Colony Optimization imitates the way ants search for food using pheromone trails to lead them to the best paths [3]. The artificial ants carry out their solution-building tasks, putting pheromones on the parts they are using, thus affecting the choices of the following ants [21].

Algorithm Steps

1. **Initialization:** Initialize pheromone trails on all solution components.
2. **Ant Construction:** Each ant constructs a solution by probabilistically choosing components based on pheromone levels and heuristic information.
3. **Pheromone Update:** Level of pherromone on the solution is updated components based on the quality of the solutions constructed by the ants. This typically involves:
   - Pheromone Evaporation: Reduce pheromone levels to avoid premature convergence.

     $$\tau_{ij} = (1-\rho) \cdot \tau_{ij}$$

     Where $tau_{ij}$the pheromone is level on component (i, j) and rho is the evaporation rate.
   - Pheromone Deposition: Increase pheromone levels on components used by good solutions.

     $$\tau_{ij} = \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$$

     Where m is the number of ants and $\Delta\tau_{ij}^{k}$ is the amount of pheromone deposited by ant k on component (i, j).
4. **Termination:** Repeat steps 2-3 until a stopping criterion is met.

4.6. Differential Evolution (DE)

Differential Evolution is an algorithm that works with a population and uses the differences between solutions as a source for creating new candidates 34. It is the best choice when dealing with continuous optimization problems, as it can efficiently cope with non-differentiable and multi-modal functions [22].

Algorithm Steps

1. **Initialization:** Create an initial population of solutions.
2. **Mutation:** For each solution x_i, create a mutant vector v_i using the following equation:
$$v_i = x_{r1} + F \cdot (x_{r2} - x_{r3})$$
Where x_{r1}, x_{r2}, and x_{r3} are randomly selected solutions from the population, and F is a scaling factor.
3. **Crossover:** Create a trial vector u_i by combining the components of the target vector x_i and the mutant vector v_i using a crossover Probability CR.
4. **Selection:** If the trial vector u_i has better fitness than the target vector x_i, replace x_i with u_i.
5. **Termination:** Repeat steps 2-4 until a stopping criterion is met.

4.7. Cuckoo Search (CS)

Cuckoo Search takes its cue from cuckoo birds' brood parasitism, which involves laying their eggs in the nests of other bird species5. The algorithm calls for the use of Lévy flights to come up with new solutions, and the less fit ones are replaced by the new ones [23], [24].

Algorithm Steps

1. **Initialization:** Initialize a population of host nests.
2. **Cuckoo Laying:** Each cuckoo lays an egg (new solution) and places it in a randomly chosen nest.
3. **Host Bird Discovery:** Some host birds discover alien eggs and either throw them away or abandon the nest.
4. **New Nests:** Abandoned nests are replaced by new nests (new solutions).
5. **Termination:** Repeat steps 2-4 until a stopping criterion is met.

4.8. Evolution Strategies (ES)

Evolution Strategies are a group of evolutionary algorithms that depend on mutation and selection as the primary means to breed a population of solutions34. A vast majority of these algorithms incorporate a mechanism for self-adaptive mutations to carry out the process of searching for good solution points [25].

Algorithm Steps

1. **Initialization:** Create an initial population of solutions, each with associated strategy parameters (e.g., mutation step sizes).
2. **Mutation:** Mutate each solution by adding a random value drawn from a normal distribution with a standard deviation determined by the strategy parameters.
3. **Evaluation:** Determine the fitness of the mutated solutions.
4. **Selection:** Pick the top solutions that will be used to create the next generation.
5. **Adaptation:** Modify the strategy parameters depending on the effectiveness of the mutations.
6. **Termination:** Go through steps 2-5 repeatedly until a stopping rule is reached.

4.9. Variable Neighborhood Search (VNS)

Variable Neighborhood Search is a strategy for optimization that traverses the neighborhoods of the current solution in a systematic way, to get rid of local optima13. The approach involves the current solution being shaken up so as to leap into a different neighborhood and then the application of local search to determine a better solution [26].

Algorithm Steps

1. **Initialization:** Start with an initial solution x and a set of neighborhood structures N_k for $k = 1, \dots, k_{max}$.
2. **Shaking:** Generate a solution x' randomly from the k-th neighborhood of x, $x' \in N_k(x)$.
3. **Local Search:** Apply a local search method starting from x' to obtain a local optimum x''.
4. **Move or Not:** If x'' is better than x, move to x'' and set $k = 1$. Otherwise, set $k = k + 1$.
5. **Neighborhood Change:** If $k > k_{max}$, set $k = 1$.

6. **Termination:** Repeat steps 2-5 until a stopping criterion is met.

4.10. Greedy Randomized Adaptive Search Procedure (GRASP)

Greedy Randomized Adaptive Search Procedure is a multi-start meta-heuristic that combines a greedy construction with a local search to provide solutions of good quality. It comprised two main stages: one for generating an initial solution and then local search for refining it [27], [28].

Algorithm Steps

1. **Construction Phase**
   - Prepare a limited candidate list (RCL) of solution components guided by a greedy criterion.
   - Choose a component at random from the RCL and include it into the incomplete solution.
   - Keeps on doing this until a full solution is built up.
2. **Local Search Phase:** Apply a local search method to improve the constructed solution.
3. **Termination:** Repeat steps 1-2 until a stopping criterion is met, and return the best solution found.

4.11. Lion Optimization Algorithm (LOA)

The Lion Optimization Algorithm is a metaheuristic that takes its cues from the territorial defense and social structure of lions in the wild, their hunting and pride dynamics, etc. It takes on such activities as marking territory, breeding, and natural selection to cover and take advantage of the search space [29], [30].

Algorithm Steps

1. **Initialization:** Generate an initial population of lions, divided into nomadic lions (explorers) and resident lions (exploiters). Randomly assign each lion its initial position within the search space.
2. **Fitness Evaluation:** Fitness of each lion (solution) is calculated, and thus the evaluation of the fitness function of the respective lion is done accordingly.
3. **Territorial Defense (Exploitation):** The resident lions, by updating their positions in the vicinity of their current territory, signify a local search.
$$\vec{x}_i^{new} = \vec{x}_i^{current} + \alpha \cdot r \cdot (\vec{x}_{best} - \vec{x}_i^{current})$$
   Where α denotes a scaling factor and r denotes a random vector.
4. **Nomadic lions (Exploration):** Nomadic lions entice local optima by going on a random movement to explore the new regions of the search space.
$$\vec{x}_i^{new} = \vec{x}_i^{current} + \beta \cdot \text{LevyFlight}()$$
   Where β controls the step size and Levy flights encourage large, sporadic jumps.
5. **Mating and Offspring Generation:** The fittest male and female lions are selected and a crossover operation is to take place to produce offspring. The weakest lions in the population are replaced by the offspring.
6. **Termination:** Steps 2-5 are repeated until a stopping criterion is reached (for example, the maximum number of iterations). The position of the fittest lion is considered the optimal solution.

4.12. Social Spider Optimization (SSO)

Social Spider Optimization draws on the interaction among social spiders for its algorithm. The algorithm represents the population as a colony of spiders, which are connected through a communal web and therefore can easily interact by using vibrations to locate food sources (the optimal solutions) [31], [32].

Algorithm Steps

1. **Initialization:** Start with a population of spiders (solutions) where each one has a random position and weight. The weight corresponds to the solution's fitness.
2. **Vibration Calculation:** Each spider detects the vibrations from the other spiders in the colony. The vibration intensity is a factor of the spider's weight and the distance between the two.

3. **Position Update (Movement):** Spiders change their positions according to the vibrations coming from a superior spider (attraction) and random movement.
$$\vec{x}_i^{new} = \vec{x}_i^{current} + r \cdot (\vec{x}_k - \vec{x}_i^{current}) \cdot \text{vibration}_{ik} + (r - 0.5)$$
Where $\vec{x}_k$ is a spider with a higher weight, and r is a random number.
4. **Weight Update:** The spider's weight is updated according to its new fitness value.
5. **Cooperative Interaction:** The best (heaviest) spiders, thus, through their influence, make others move simulating the flow of information among the colony.
6. **Termination:** Repeat steps 2-5 until a stopping criterion is being met.

4.13. Vibrating Particles System (VPS)

The Vibration Particles System (VPS) method is a physics-based heuristic that imitates suspension of the one-degree-of-freedom systems. The particles (solutions) are attracted to three leaders: the historically best solution, a good solution, and a bad solution, thus creating a balance between exploration and exploitation [33], [34].

Algorithm Steps

1. **Initialization:** Create a flock of randomly positioned particles as the first step in the search space.
2. **Guide Selection:** For every particle, choose three guides:
   - HP (Historic Best Position): The optimal solution discovered until now.
   - GP (Good Particle): A random selection from the top half of the population.
   - BP (Bad Particle): A random selection from the bottom half of the population.
3. **Position Update:** Each particle's position is modified through the equation of vibration which depicts the process of damped free vibration.
$$\vec{x}_i^{new} = w_1 \cdot D \cdot r_1 + w_2 \cdot D \cdot r_2 + w_3 \cdot D \cdot r_3$$
Where $D = A \cdot (\overrightarrow{\text{Guide}} - \vec{x}_i^{current})$, A is an acceleration coefficient that decreases over time, r_1, r_2, r_3 are random numbers, and w_1, w_2, w_3 are weights corresponding to HP, GP, and BP, respectively.
4. **Fitness Evaluation and Update:** The positions are evaluated. The historic best solution (HP) is updated if a better solution is discovered.
5. **Termination:** Steps 2-4 are repeated until the predetermined maximum number of iterations is reached.

4.14. Cat Swarm Optimization (CSO)

Cat Swarm Optimization, on the contrary, is based on the character of cats. It portrays the two modes of cats, which are sleeping and watching (the seeking mode) and chasing the prey (the tracing mode). This method, which with its dual mode, effectively offers the strengths of both methods [35].

Algorithm Steps

1. **Initialization:** Set up a cat population that represents the solution. Each cat is assigned a position and a velocity. Decide on the Mixture Ratio (MR) that specifies the number of cats in the tracing mode. Fitness Evaluation: Each cat's fitness level is evaluated.
2. **Behavior Mode Assignment:** Following the MR, the cats will be randomly assigned to either the seeking or tracing mode.
3. **Seeking Mode (Exploration):** This mode exemplifies a cat that is sleeping but attentive at the same time. The main operations include:
   - **Copying:** Produce various copies of the present cat.
   - **Mutation:** Slight displacement of each copy's position will be done along one dimension.
   - **Selection:** The copies will be evaluated and the best one will be chosen to take the current cat's position.
4. **Tracing Mode (Exploitation):** This mode represents a cat that is actively hunting its food. The cat's velocity and position will be updated in the same way as in PSO:
$$\vec{v}_k^{new} = \vec{v}_k^{current} + c \cdot r \cdot (\vec{x}_{best} - \vec{x}_k^{current})$$

$$\vec{x}_k^{new} = \vec{x}_k^{current} + \vec{v}_k^{new}$$

Where c is a constant and r is a random number.

5. **Termination:** Repeat steps 2-5 until the termination criterion is satisfied.

4.15. Elephant Herding Optimization (EHO)

One model of Elephant Herding Optimization is the herd based on the behavior of elephants. This algorithm consists of two primary operations: clan updating, which signifies local exploitation, and separating, which opens up exploration by allowing some elephants to leave their clan [36], [37], [38].

Algorithm Steps

1. **Initialization:** Initialize a population of elephants randomly. The population is divided into several clans.
2. **Clan Updating Operator (Exploitation):** For each clan, the position of each elephant is influenced by the matriarch (the fittest elephant in the clan).
$$\vec{x}_{new,ci} = \vec{x}_{ci} + \alpha \cdot (\vec{x}_{best,c} - \vec{x}_{ci}) \cdot r$$
Where $\vec{x}_{ci}$ is the position of elephant i in clan c, $\vec{x}_{best,c}$ is the matriarch of the clan, α is a scaling factor, and r is a random number. The matriarch's position is updated with a centralizing factor β.
$$\vec{x}_{new,best,c} = \beta \cdot \vec{x}_{center,c}$$
3. **Separating Operator (Exploration):** In each generation, the worst elephant in each clan is replaced by a new elephant generated randomly within the search space. This mimics young elephants leaving their natal clan and helps explore new areas.
$$\vec{x}_{worst,c} = \vec{x}_{min} + (\vec{x}_{max} - \vec{x}_{min} + 1) \cdot rand$$
4. **Fitness Evaluation:** Evaluate the new positions of all elephants.
5. **Termination:** Repeat steps 2-4 until the stopping criterion is met.

5. MATHEMATICAL FORMULATIONS

5.1. Single-Objective Optimization

The optimization process with a single objective function means that the decision space Ω would contain a solution x* that would minimize the single objective function f (x), where f is the objective function.

$$\min_{x \in \Omega} f(x)$$

Where $f: \Omega \to \mathbb{R}$ is the objective function.

5.2. Multi-Objective Optimization

For multi-objective optimization problems, the objective is to minimize a vector of functions:

$$\min_{x \in \Omega} F\,(x) = \big(f_1(x), f_2(x), \dots, f_k(x)\big)$$

Subject to constraints:

$$g_i(x) \leq 0, i = 1, \dots, m; h_j(x) = 0, j = 1, \dots, p$$

Where $f_i(x)$ are the individual objective functions, $g_i(x)$ are inequality constraints, and $h_j(x)$ are equality constraints. The solution x^* is considered Pareto optimal if no other solution exists that improves some objectives without worsening at least one other [39].

5.3. Constraint Handling Techniques

Metaheuristic algorithms would apply various methods to manage constraints, among which are the following.

- **Penalty Functions:** Imposing a penalty on the objective function for non-conforming solutions to the constraints.

$$f'(x) = f(x) + P \cdot \sum_{i=1}^{m} \max(0, g_i(x)) + P \cdot \sum_{j=1}^{p} | h_j(x) |$$

Where P is a penalty parameter.

- **Repair Methods:** Changing the non-conforming solutions to make them conforming ones;
- **Feasibility Rules:** Making the feasibility considerations part of the selection and replacement stages.

Surrogate Models and Hybridization

5.4. Types of Surrogate Models

Surrogate models approximate the objective function, thus lessening the computational cost of evaluating candidate solutions [40]. The most common surrogate models include:

- **Polynomial Regression:** The use of polynomial functions to represent the objective function;
- **Radial Basis Functions (RBF):** The interpolation of the known data points via radial basis functions;
- **Kriging:** A geostatistical method that provides predictions alongside the confidence estimates;
- **Support Vector Machines (SVM):** A machine learning technique that is capable of handling both regression and classification tasks.

5.5. Hybrid Metaheuristic Approaches

Hybrid metaheuristics combine various optimization methods to exploit their strengths and compensate for their weaknesses. The following cases are among them.

Memetic Algorithms: A combination of genetic algorithms and local search results in a hybrid.

Hybrid ACO with Local Search: The integration of ant colony optimization and local search has been done to enhance the quality of the solution.

Surrogate-Assisted Evolutionary Algorithms: Surrogate models help to reduce the number of expensive objective function evaluations.

6. APPLICATIONS AND CASE STUDIES

6.1. Engineering Design Problems

Metaheuristic algorithms are considered a major tool in the process of engineering design as they are capable of optimizing parameters such as the dimensions of structures, material characteristics, and control system settings. One can find various example applications like [41].

- **Structural Optimization:** The established stress and displacement conditions are met, while the structure's weight is reduced as much as possible.
- **Aerospace Engineering:** The airfoil's shape alteration is such that it obtains the highest lift and loses the least drag.
- **Control Systems Design:** The tuning of the parameters of the PID controller is done in such a way that the desired performance characteristics are achieved.

6.2. Feature Selection

Metaheuristic algorithms can be employed to perform a search for the most suitable features for the machine learning model.

- **Wrapper Methods:** The performance of the chosen machine learning model is the metric used to assess feature subsets.
- **Filter Methods:** The features are ranked through statistical metrics, and only those with the highest ranks are chosen.
- **Embedded Methods:** Feature selection is not only allowed but is also a part of the training process of the machine learning model.

6.3. Combinatorial Optimization

Combinatorial optimization problems are defined as problems where one has to search the whole set of possible solutions and choose the best one.The main reason for the suitability of metaheuristic algorithms for these problems is their flexibility in dealing with the discrete characteristic of the search space. The following are some of them [43].

- **Traveling Salesman Problem (TSP):** The objective is to find the shortest round trip that covers all the cities and goes back to the starting one [44].
- **Vehicle Routing Problem (VRP):** It consists of finding the most efficient routes for a group of delivery vehicles that need to serve so many customers [45].
- **Scheduling Problems:** Resources and tasks distribution are optimized to lessen the time or cost of completion [46].

7. PERFORMANCE EVALUATION AND COMPARISON

7.1. Benchmark Functions

The test functions for benchmarking are a specific set of problems that are considered standard for analyzing the performance of optimization algorithms. The attributes of these functions are very specific, and they include unimodality, multi-modality, separability, and scalability, thus allowing for a systematic comparison between different algorithms [47], [48].

7.2. Statistical Analysis

Statistical analysis is used to assess the significance of performance differences between algorithms. Common statistical tests include [49], [50].

- **T-tests:** Compare the means of two groups.
- **ANOVA:** Compare the means of multiple groups.
- **Non-parametric Tests:** Used when the data does not meet the assumptions of parametric tests.

8. CHALLENGES AND FUTURE DIRECTIONS

Metaheuristic algorithms face several challenges

- **Parameter Tuning:** Finding the optimal parameter settings for an algorithm can be time-consuming and problem-dependent.
- **Premature Convergence:** Local optima might be reached by algorithms even before the whole search space is investigated.
- **Scalability:** A few algorithms could be incapable of coping with large-scale problems.
- Research directions of the future are as follows:
- **Adaptive Parameter Control:** Creating systems that automatically change algorithm parameters during the search process3.
- **Hybridization Strategies:** Merging distinct metaheuristics and accurate methods to use their joint strengths.
- **Parallelization:** The application of parallel computation in making the search process faster.

9. CONCLUSION

Metaheuristic algorithms are the go-to tools for overcoming complex optimization issues in different areas. They say that recent breakthroughs, especially the use of surrogate models, and hybridization can greatly cut down the costs and at the same time, increase the quality of the solutions. Proper tuning of parameters, adaptive sampling, and the combination of ensemble or hierarchical surrogate models are needed to get better performance. Future research should not only be concerned with model management but also with parallelization to improve the overall results of optimization. The continuous creation and refinement of metaheuristic algorithms will still have a decisive influence in the fight against complex problems of a real-world nature.

Acknowledgements

The authors have no specific acknowledgments to make for this research.

Funding Information

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Author Contributions Statement

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Awaz Ahmed Shaban	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓	
Saman M. Almufti		✓	✓		✓				✓		✓		✓	✓
Renas Rajab Asaad	✓		✓				✓			✓	✓		✓	
Rasan Ismael Ali			✓	✓		✓		✓		✓		✓		✓

C: **C**onceptualization
M: **M**ethodology
So: **So**ftware
Va: **Va**lidation
Fo: **Fo**rmal analysis
I: **I**nvestigation
R: Resources
D: Data Curation
O: Writing- **O**riginal Draft
E: Writing- Review& **E**diting
Vi: **Vi**sualization
Su: **Su**pervision
P: **P**roject administration
Fu: Funding acquisition

Conflict of Interest Statement

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Informed Consent

All participants were informed about the purpose of the study, and their voluntary consent was obtained prior to data collection.

Ethical Approval

The study was conducted in compliance with the ethical principles outlined in the Declaration of Helsinki and approved by the relevant institutional authorities.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] S. M. Almufti, 'Historical survey on metaheuristics algorithms', Int. J. Sci. World, vol. 7, no. 1, p. 1, Nov. 2019. doi.org/10.14419/ijsw.v7i1.29497

[2] S. M. Almufti, 'Vibrating Particles System Algorithm performance in solving Constrained Optimization Problem', Acad. J. Nawroz Univ., vol. 11, no. 3, pp. 231-242, Aug. 2022. doi.org/10.25007/ajnu.v11n3a1499

[3] B. Shojaei, H. Dashti Naserabadi, and M. J. Taheri Amiri, 'Optimizing competency-based human resource allocation in construction project scheduling: A multi-objective meta-heuristic approach', Qubahan Academic Journal, vol. 4, no. 3, pp. 861-881, Sept. 2024. doi.org/10.48161/qaj.v4n3a1027

[4] M. Dehghani, Z. Montazeri, G. Bektemyssova, O. P. Malik, G. Dhiman, and A. E. M. Ahmed, 'Kookaburra Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving

optimization problems', Biomimetics (Basel), vol. 8, no. 6, p. 470, Oct. 2023. doi.org/10.3390/biomimetics8060470

[5] S. M. Almufti, R. B. Marqas, P. S. Othman, and A. B. Sallow, 'Single-based and population-based metaheuristics for solving np-hard problems', Iraqi Journal of Science, vol. 62, no. 5, pp. 1710-1720, 2021. doi.org/10.24996/10.24996/ijs.2021.62.5.34

[6] J. Kudela and R. Matousek, 'New benchmark functions for single-objective optimization based on a zigzag pattern', IEEE Access, vol. 10, pp. 8262-8278, 2022. doi.org/10.1109/ACCESS.2022.3144067

[7] K. Deb, 'GeneAS: A robust optimal design technique for mechanical component design', in Evolutionary Algorithms in Engineering Applications, Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 497-514. doi.org/10.1007/978-3-662-03423-1_27

[8] K. E. Parsopoulos and M. N. Vrahatis, 'Unified particle swarm optimization for solving constrained engineering optimization problems', in Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 582-591. doi.org/10.1007/11539902_71

[9] H. T. Sadeeq, A. M. Abdulazeez, N. A. Kako, D. A. Zebari, and D. Q. Zeebaree, 'A new hybrid method for global optimization based on the bird mating optimizer and the differential evolution', in 2021 7th International Engineering Conference "Research & Innovation amid Global Pandemic" (IEC), Erbil, Iraq, 2021. doi.org/10.1109/IEC52205.2021.9476147

[10] S. Mohammed Almufti, R. P. Maribojoc, and A. V. Pahuriray, 'Ant based system: Overview, modifications and applications from 1992 to 2022', Polaris Global Journal of Scholarly Research and Trends, vol. 1, no. 1, pp. 29-37, Oct. 2022. doi.org/10.58429/pgjsrt.v1n1a85

[11] S. M. Almufti, U-Turning Ant Colony Algorithm powered by Great Deluge Algorithm for the solution of TSP Problem. 2015.

[12] A. Kaveh and T. Bakhshpoori, Metaheuristics: Outlines, MATLAB codes and examples, 2019th edn. Cham, Switzerland: Springer Nature, 2019. doi.org/10.1007/978-3-030-04067-3

[13] A.-R. Hedar and M. Fukushima, 'Derivative-free filter simulated annealing method for constrained continuous global optimization', J. Glob. Optim., vol. 35, no. 4, pp. 521-549, Aug. 2006. doi.org/10.1007/s10898-005-3693-z

[14] Nawroz University, R. Asaad, N. Abdulnabi, and University of Duhok, 'Using local searches algorithms with ant colony optimization for the solution of TSP problems', Acad. J. Nawroz Univ., vol. 7, no. 3, pp. 1-6, 2018. doi.org/10.25007/ajnu.v7n3a193

[15] S. M. Almufti, R. Boya Marqas, and V. Ashqi Saeed, 'Taxonomy of bio-inspired optimization algorithms', J. Adv. Comput. Sci. Technol., vol. 8, no. 2, pp. 23-31, Aug. 2019. doi.org/10.14419/jacst.v8i2.29402

[16] A.-R. Hedar and M. Fukushima, 'Derivative-free filter simulated annealing method for constrained continuous global optimization', J. Glob. Optim., vol. 35, no. 4, pp. 521-549, Aug. 2006. doi.org/10.1007/s10898-005-3693-z

[17] A. Acan and A. Ünveren, 'A great deluge and tabu search hybrid with two-stage memory support for quadratic assignment problem', Appl. Soft Comput., vol. 36, pp. 185-203, Nov. 2015. doi.org/10.1016/j.asoc.2015.06.061

[18] D. Q. Zeebaree, H. Haron, A. Mohsin Abdulazeez, and S. R. M. Zeebaree, "Combination of K-means clustering with Genetic Algorithm: A review," 2017.

[19] A. Acan, H. Altincay, Y. Tekol, and A. Unveren, 'A genetic algorithm with multiple crossover operators for optimal frequency assignment problem', in The 2003 Congress on Evolutionary Computation, 2003. CEC '03, Canberra, ACT, Australia, 2004.

[20] L. dos S. Coelho, 'Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems', Expert Syst. Appl., vol. 37, no. 2, pp. 1676-1683, Mar. 2010. doi.org/10.1016/j.eswa.2009.06.044

[21] S. M. Almufti and A. A. Shaban, 'U-Turning Ant Colony algorithm for solving symmetric Traveling Salesman Problem', Acad. J. Nawroz Univ., vol. 7, no. 4, p. 45, Dec. 2018. doi.org/10.25007/ajnu.v7n4a270

[22] Y. Y. Ao and H. Q. Chi, 'An adaptive differential evolution algorithm to solve constrained optimization problems in engineering design', Engineering, vol. 02, no. 01, pp. 65-77, 2010. doi.org/10.4236/eng.2010.21009

[23] A. S. Joshi, O. Kulkarni, G. M. Kakandikar, and V. M. Nandedkar, 'Cuckoo search optimization- A review', Mater. Today, vol. 4, no. 8, pp. 7262-7269, 2017. doi.org/10.1016/j.matpr.2017.07.055

[24] S. M. Almufti, R. B. Marqas, R. R. Asaad, and A. A. Shaban, "Cuckoo search algorithm: overview, modifications, and applications," 2025. doi.org/10.14419/efkvvd44

[25] Y. Sun, D. Wierstra, T. Schaul, and J. Schmidhuber, 'Efficient natural evolution strategies', in Proceedings of the 11th Annual conference on Genetic and evolutionary computation, Montreal Québec Canada, 2009. doi.org/10.1145/1569901.1569976

[26] P. Hansen and N. Mladenović, 'Variable Neighborhood Search', in Search Methodologies, Boston, MA: Springer US, 2006, pp. 211-238. doi.org/10.1007/0-387-28356-0_8

[27] T. A. Feo and M. G. C. Resende, 'Greedy randomized adaptive search procedures', J. Glob. Optim., vol. 6, no. 2, pp. 109-133, Mar. 1995. doi.org/10.1007/BF01096763

[28] M. Laguna, R. Martí, A. Martínez-Gavara, S. Pérez-Peló, and M. G. C. Resende, 'Greedy Randomized Adaptive Search Procedures with Path Relinking. An analytical review of designs and implementations', Eur. J. Oper. Res., vol. 327, no. 3, pp. 717-734, Dec. 2025. doi.org/10.1016/j.ejor.2025.02.022

[29] S. M. Almufti, 'Lion algorithm: Overview, modifications and applications', 2022. doi.org/10.46291/ICONTECHvol6iss3pp1-11

[30] L. Velasco et al., “Comparative study on the performance of metaheuristics in the optimization of seismic designs,” 2024.

[31] S. Feng, Y. Hu, Y. Chen, and M. Huang, 'An improved spider optimization algorithm coordinated by pheromones', Sci. Rep., vol. 12, no. 1, p. 5962, Apr. 2022. doi.org/10.1038/s41598-022-09800-x

[32] S. Almufti, 'The novel Social Spider Optimization algorithm: Overview, modifications, and applications', iij, vol. 5, no. 2, pp. 32-51, June 2021. doi.org/10.46291/ICONTECHvol5iss2pp32-51

[33] M. Wedyan, O. Elshaweesh, E. Ramadan, and R. Alturki, 'Vibrating particles system algorithm for solving classification problems', Comput. Syst. Sci. Eng., vol. 43, no. 3, pp. 1189-1206, 2022. doi.org/10.32604/csse.2022.024210

[34] S. Almufti, 'Vibrating Particles System algorithm: Overview, modifications and Applications', iij, vol. 6, no. 3, pp. 1-11, Sept. 2022. doi.org/10.46291/ICONTECHvol6iss3pp1-11

[35] R. R. Ihsan, S. M. Almufti, B. M. S. Ormani, R. R. Asaad, and R. B. Marqas, 'A survey on Cat Swarm Optimization algorithm', AJRCoS, pp. 22-32, June 2021. doi.org/10.9734/ajrcos/2021/v10i230237

[36] M. Rajasekaran, M. S. Thanabal, and A. Meenakshi, 'Association rule hiding using enhanced elephant herding optimization algorithm', Automatika, vol. 65, no. 1, pp. 98-107, Jan. 2024. doi.org/10.1080/00051144.2023.2277998

[37] K. Naicker and V. M.Srivastava, 'Realization and fabrication of 3rd order high pass filter with CMOS', Int. J. Eng. Technol., vol. 7, no. 4, pp. 5967-5976, Apr. 2019. doi.org/10.14419/ijet.v7i4.23127

[38] D. Praveena Anjelin and S. Ganesh Kumar, 'An effective classification using enhanced elephant herding optimization with convolution neural network for intrusion detection in IoMT architecture', Cluster Comput., June 2024. doi.org/10.1007/s10586-024-04512-5

[39] Z. N. Rashid, S. R. M. Zeebaree, M. A. M. Sadeeq, R. R. Zebari, H. M. Shukur, and A. Alkhayyat, 'Cloud-based parallel computing system via single-client multi-hash single-server multi-thread', in 2021 International Conference on Advance of Sustainable Engineering and its Application (ICASEA), Wasit, Iraq, 2021. doi.org/10.1109/ICASEA53739.2021.9733077

[40] S. Jueyendah and C. H. Martins, 'Computational Engineering and Physical Modeling Optimal Design of Welded Structure Using SVM ARTICLE INFO ABSTRACT, " Optimal Design of Welded Structure Using SVM', Optimal Design of Welded Structure Using SVM. Computational Engineering and Physical Modeling, vol. 7, no. 3, pp. 84-107, 2024.

[41] S. M. Almufti, A. Ahmad Shaban, Z. Arif Ali, R. Ismael Ali, and J. A. Dela Fuente, 'Overview of metaheuristic algorithms', Polaris Global Journal of Scholarly Research and Trends, vol. 2, no. 2, pp. 10-32, Apr. 2023. doi.org/10.58429/pgjsrt.v2n2a144

[42] J. G. Dy and C. E. Brodley, Feature Selection for Unsupervised Learning. 2004.

[43] A. Ahmed Shaban, S. M. Almufti, and R. R. Asaad, 'Metaheuristic algorithms for engineering and CombinatorialOptimization: A comparative study across problems categories and benchmarks', Int. J. Sci. World, vol. 11, no. 2, pp. 38-49, Aug. 2025. doi.org/10.14419/0hndc578

[44] S. M. Almufti and A. A. Shaban, 'Comparative analysis of metaheuristic algorithms for solving the travelling salesman problems', Int. J. Sci. World, vol. 11, no. 2, pp. 26-30, July 2025. doi.org/10.14419/7fk7k945

[45] Z. Sherinov, A. Unveren, and A. Acan, 'An evolutionary multi-objective modeling and solution approach for fuzzy vehicle routing problem', in 2011 International Symposium on Innovations in Intelligent Systems and Applications, Istanbul, Turkey, 2011. doi.org/10.1109/INISTA.2011.5946143

[46] I. M. Ibrahim, Task Scheduling Algorithms in Cloud Computing: A Review. 2021.

[47] K. Hussain, M. N. Mohd Salleh, S. Cheng, and R. Naseem, 'Common benchmark functions for metaheuristic evaluation: A review', JOIV Int. J. Inform. Vis., vol. 1, no. 4-2, pp. 218-223, Nov. 2017. doi.org/10.30630/joiv.1.4-2.65

[48] S. K. Azad, O. Hasançebi, S. K. Azad, and O. K. Erol, "Evaluating efficiency of big-bang big-crunch algorithm in benchmark engineering optimization problems," 2011.

[49] A. Ahmed Shaban and I. Mahmood Ibrahim, "International Journal of Scientific World Swarm intelligence algorithms: a survey of modifications and applications," 2025.

[50] S. M. Almufti and A. A. Shaban, 'A deep dive into the artificial bee colony algorithm: theory, improvements, and real-world applications', Int. J. Sci. World, vol. 11, no. 1, pp. 178-187, May 2025. doi.org/10.14419/v9d3s339

How to Cite: Awaz Ahmed Shaban, Saman M. Almufti, Renas Rajab Asaad, Rasan Ismael Ali. (2025). Metaheuristic algorithms for complex optimization: a critical review of foundations, hybrid strategies, and surrogate modeling. International Journal of Information Technology and Computer Engineering(IJITC), 5(2), 28–46. https://doi.org/10.55529/ijitc.52.28.46

BIOGRAPHIES OF AUTHORS

	**Awaz Ahmed Shaban**, is a Researcher at the Information Technology Department, Technical College of Informatics–Akre, Akre University for Applied Sciences, Duhok, Iraq. Her research interests include artificial intelligence, optimization algorithms, and data-driven decision systems. She has contributed to several studies in computational intelligence and applied informatics. Email: Awaz.Amedy1@gmail.com
	**Saman M. Almufti**, is a researcher at both the Technical College of Informatics–Akre, Akre University for Applied Sciences, Duhok, Iraq, and the Department of Computer Science, College of Science, Knowledge University, Erbil. His research focuses on metaheuristic algorithms, swarm intelligence, and applied machine learning. He has published numerous papers in high-impact international journals. Email: Saman.Almofty@gmail.com

	**Renas Rajab Asaad, ** is a researcher at the Technical College of Informatics–Akre, Akre University for Applied Sciences, and the Department of Computer Science, College of Science, Knowledge University, Erbil. His primary research areas include optimization, computer vision, and intelligent computing systems.
	**Rasan Ismael Ali,** is a researcher at the Department of Computer Science, Knowledge University in the College of Science, Erbil. Artificial Intelligence, Information Systems, and Computational Optimization are some of her research interests. He is working on several projects related to intelligent system development as well as pure computer science applications.