Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)

[1] Volume 7- Issue 1, January 2024
[2] Paper : 91

A survey on Dragonfly heuristic optimization algorithm, applications, and challenges

Mrs.R.Priya
Associate Professor, Information Technology,Anjalai Ammal Mahalingam Engineering College,Kovilvenni,Tamilnadu, India 1 ,
rpriya.it2017@gmail.comline 1: 4th

Dr.S.Sujatha
Professor and Head,Department of Computer Applications,BIT Campus, Anna University,Tiruchirappalli, Tamilnadu, India,
sujathaaut@gmail.com

Abstract— One of the most current heuristic optimization methods is the Dragonfly Algorithm. Additionally, it has demonstrated its ability to optimize several real-world problems. The static and dynamic swarming behaviour of dragonflies in the wild served as the basis for the Dragonfly Algorithm. In the Dragonfly Algorithm, the potential solutions to an optimization problem are represented as a swarm of dragonflies. Each dragonfly represents a candidate solution, and they interact with each other to search for the optimal solution. The algorithm uses both local search and global search strategies to balance exploration and exploitation of the search space. This review study provides an in-depth analysis of the dragonfly method, as well as its uses and difficulties.

An overview of the algorithm is discussed first. In addition, a survey of engineering applications that utilized the dragonfly algorithm is also provided. The algorithm is compared to other metaheuristic methods to see how well it can handle different challenges. In addition, the technique's congestion facts and some potential future works are discussed. The purpose of this study was to provide assistance to other researchers interested in studying the algorithm and using it to optimize engineering problems.

Keywords: Dragonfly algorithm · heuristic optimization method

1. Introduction

Swarm intelligence (SI) is utilized by numerous researchers in various fields. The ability of natural swarm systems to study the behaviours of swarms and creatures astonished natural scientists and biologists. The family of nature-inspired population-based algorithms includes swarm-based algorithms. As to true issues, these calculations produce great outcomes with regard to cost, speed, and strength [1]. Swarm intelligence systems are made up of a group of agents who live together as a population. They are made up of a collection of decentralized and self-organized intellectual performance systems. SI-based techniques include building nests, foraging groups of social insects, and collective clustering and sorting [3]. SI has recently been used to solve a variety of problems in robotics, telecommunications, continuous and combinatory optimization, and other fields. and excellent outcomes were frequently achieved [5]. The researchers have recently proposed some novel approaches. Particle Multitude Streamlining (PSO) was proposed by Kennedy furthermore, Eberhart [6]. Dragonfly Optimization Algorithm (DA) was proposed by Mirjalili. DA basically impersonates the ways of behaving of hunting and relocation of dragonflies. In [14], the Harmony Search Algorithm (HA) is proposed. It imitates the musician's process of improving music. The musician strives to improve harmony based on previous experiences. According to [15], the Donkey and Smuggler Optimization (DSO) algorithm was suggested. DSO imitates the mentalities of jackasses to choose and look through courses. Yazdani and co fostered one more illustration of nature-roused calculations, which is known as the Lion Advancement Calculation (LOA) [16]. The LOA imitates the cooperative behavior and unique lifestyle of lions. The lions can be divided into residents and nomads according to their social organization. The residents are known as pride

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)
[1] Volume 7- Issue 1, January 2024
[2] Paper : 91

because they consist of several lions living together. On the other hand, most people only see nomads in groups of two or more. Lions can switch between living as nomads and residents or vice versa. A brand-new Learner Performance-Based Behavior (LPB) algorithm has also been proposed by Rahman, C., and Rashid [17]. Metaheuristic algorithms in their multi-objective variants are also used to optimize multi-objective problems, as shown in reference [18].

DA has been utilized in a variety of applications by various researchers, and the outcomes have been satisfactory. The dragonfly algorithm was cited in nearly 300 different works throughout the entirety of this review paper, which was completed in March 2019. In almost every application, it resulted in satisfying outcomes. Additionally, another review paper on the DA and its applications in applied science was published by the authors of this one [19]. The authors of that review paper provided an overview of the applied science field (such as image processing, machine learning, wireless, and networking) in their review.

The Dragonfly Algorithm was propelled by the static and dynamic amassing conduct of dragonflies in nature. In fact, the static and dynamic swarming behaviors represent the two essential phases of optimization exploration and exploitation. Additionally, in a static swarm, dragonflies fly over various locations in sub-swarms. Similar to exploration, this helps the algorithm locate appropriate locations in the search space. Dragonflies, on the other hand, fly in the same direction as a larger swarm in a dynamic swarm. In addition, using an algorithm to help it converge on the global best solution is the same as this type of swarming.

2. Dragonfly Algorithm

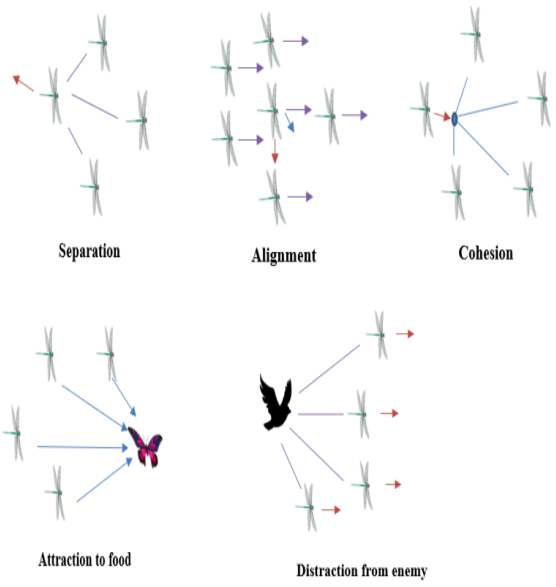
Swarms of the Dragonfly Algorithm generally act in accordance with three fundamental principles:

Separation: which indicates the avoidance of neighboring static collisions

Alignment: which returns the speed of individuals paired with neighboring individuals

Cohesion: which indicates the individual tendency toward the center of the herd

Because survival is the ultimate goal of any swarm, all members should be drawn to food sources and away from potential attackers. When considering these two behaviors, there are five primary factors in position updating of individuals in swarms: As depicted in the figure below, separation, alignment, cohesion, attraction, and distraction:



Dragonfly Algorithm Mathematical Model

The positioning movement of dragonflies consists of the five behaviors outlined in the preceding section. In this way, we should characterize the tasks:

2.1 Separation

We can calculate the Separation between two adjacent dragonflies as follows:

$$S_i = -\sum_{j=1}^{N} X - X_j \qquad (1)$$

X signifies the current individual's position.

X_j specifies the j^{th} dragonfly's position in the neighborhood.

N designates the dragonflies' number in the neighboring.

S signifies the i^{th} dragonfly's separation motion.

2.2 Alignment

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)
[1] Volume 7- Issue 1, January 2024
[2] Paper : 91

The equation is used to determine the alignment of dragonflies:

$$A_i = \frac{\sum_{j=1}^{N} V_j}{N} \qquad (2)$$

where V_j describes the velocity of the $j^{(th)}$ neighbouring individual.

2.3 Cohesion

We can derive the Cohesion as follows:

$$C_i = \frac{\sum_{j=1}^{N} X_j}{N} - X \qquad (3)$$

where X shows the position of the current individual, $j^{(th)}$ indicates the neighboring individual, and N is the number of neighboring individuals.

2.4 Attraction

We compute the Attraction operation toward the source of food using this equation:

$$F_i = X^+ - X \qquad (4)$$

Where X presents the position of the current individual, and X^+ describes the position of the food source.

2.5 Distraction

We can determine the Distraction from the enemy as shown here:

$$R_i = X^- - X \qquad (5)$$

Where *X*, *shows* the position of the current individual and X^- denotes the natural enemy.

To update the position of artificial dragonflies in a search space and replicate their motions, two vectors are considered: step vector (ΔX) and position vector (X).

So, the step vector presents the direction of the movement of the dragonflies. We can define this vector by the equation:

$$\Delta X_t + 1 = (sS_i + aA_i + cC_i + fF_i + rR_i) + w\Delta X_t \qquad (6)$$

Where s, a, c, and f are weights for the phases: separation, alignment, cohesion, attraction, and distraction. Also, w presents the inertia weight, and t is the iteration counter.

Exploration and exploitation stages can be achieved by changing the values of these weights.

The position vector is easily determined with the help of the step vector, as shown in the equation:

$$X_t + 1 = X_t + \Delta X_t + 1 \qquad (7)$$

Low cohesion weight (c) and high alignment weight (a) are allocated to the exploration phase. Similarly, low alignment weight (a) and high cohesion weight (c) are assigned to the exploitation phase.

The position vector indicates the dragonfly's location; however, if no surrounding solutions exist, the dragonflies must fly in a random search space, and their position must be updated using the adjusted equation:

$$X_t + 1 = X_t + X_t.levy(d) \qquad (8)$$

where $levy(d)$ is:

$$levy(d) = 0.01 \frac{r_1.\sigma}{|r_2|^{\frac{1}{\beta}}} \qquad (9)$$

where we range r1,r2 in [0,1], β is constant,

σ is defined as follows:

$$\sigma = \left(\frac{(\beta! \times sin(\frac{\pi\beta}{2}))}{(\frac{\beta-1}{2})! \times \beta \times 2^{(\frac{\beta-1}{2})}}\right)^{\frac{1}{\beta}} \qquad (10)$$

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)
[1] Volume 7- Issue 1, January 2024
[2] Paper : 91

A. Pseudocode of the Dragonfly Algorithm

Algorithm 1: Pseudocode of the Dragonfly Algorithm

```
Initialize the dragonflies population X_i (i = 1, 2, ..., n)
Initialize step vectors ΔX_i (i = 1, 2, ..., n)
while the end condition is not satisfied do
    Calculate the fitness functions for each dragonfly;
    Update food sources and natural enemy;
    Calculate Separation (S), Alignment (A), Cohesion (C),
     Attraction (F) and Distraction (R);
    Using their equations;
    if Neighbor(dragonfly) ≥ 1 then
        Update velocity vector (ΔX) using equation (6);
        Update position vector (X) using equation (7);
    else
        Update position vector using equation (8);
    end
end
```

A set of solutions to specific optimization problems is generated at random by the algorithm to begin the optimization process. Random values defined within the lower and higher ranges of the variables are used to initialize the position and step vectors of dragonflies.

Every dragonfly's position and step were updated in each cycle. Additionally, the Euclidean distance between all dragonflies is calculated and N of them is selected as the neighborhood for updating (X) and (Delta X) vectors. After that, we carried on iteratively updating the positions until the final requirement was met.

3. Variants of the dragonfly algorithm

The Dragonfly Algorithm is a nature-inspired optimization algorithm that is inspired by the behaviour of dragonflies in nature. While the original Dragonfly Algorithm was proposed by Xingming Sun in 2013, there have been several variations and extensions of the algorithm that have been proposed by researchers. Here are a few notable variants of the Dragonfly Algorithm:

1. Enhanced Dragonfly Algorithm (EDA): This variant of the Dragonfly Algorithm incorporates additional strategies to improve exploration and exploitation abilities. It introduces a mutation operator to enhance diversity in the population and a local search operator to exploit promising solutions. The EDA has been shown to improve the performance of the original Dragonfly Algorithm in solving optimization problems.

2. Self-Adaptive Dragonfly Algorithm (SADA): The SADA incorporates a self-adaptive mechanism that allows the algorithm to dynamically adjust its parameters during the optimization process. By adapting its parameters based on the problem characteristics and the evolution of the population, the SADA aims to improve convergence speed and overall optimization performance.

3. Hybrid Dragonfly Algorithm (HDA): The HDA combines the Dragonfly Algorithm with other optimization algorithms to take advantage of their complementary strengths. For example, the HDA may integrate the particle swarm optimization (PSO) algorithm with the Dragonfly Algorithm to enhance exploration and exploitation abilities. The hybridization of different algorithms can lead to improved optimization performance and robustness.

4. Multi-Objective Dragonfly Algorithm (MODA): The MODA extends the Dragonfly Algorithm to handle multi-objective optimization problems. It incorporates mechanisms to maintain a diverse set of solutions along the Pareto front, which represents the optimal trade-off between multiple conflicting objectives. The MODA allows for the exploration of the solution space and provides a set of solutions representing different trade-offs between objectives.

These are just a few examples of the variants and extensions of the Dragonfly Algorithm. Each variant aims to address

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)
[1] Volume 7- Issue 1, January 2024
[2] Paper : 91

specific limitations or improve specific aspects of the original algorithm to make it more effective in solving different types of optimization problems.

4. Applications of Dragonfly Algorithm in Engineering

The Dragonfly Algorithm (DA) and its variants have been applied to various engineering fields for solving optimization problems. Here are some applications of the Dragonfly Algorithm in engineering:

1. Structural Design Optimization: The DA has been employed in structural design optimization problems, such as finding the optimal shape and dimensions of structures. It can be used to minimize the weight of structures while ensuring they meet design constraints and performance requirements.

2. Manufacturing Process Optimization: The DA has been applied to optimize manufacturing processes, including parameters tuning in machining operations, scheduling of production processes, and supply chain management. By optimizing these processes, the DA can improve efficiency, reduce costs, and enhance productivity in manufacturing industries.

3. Power System Optimization: The DA has been utilized in power system optimization, such as optimal power flow, economic dispatch, and renewable energy integration. It helps in finding the optimal configuration and operation of power systems, considering factors like load demand, generation capacity, transmission losses, and renewable energy sources.

4. Water Resources Management: The DA has been applied to water resources management problems, including reservoir operation, water allocation, and flood control. By optimizing the operation of water systems, the DA can help in maximizing water availability, minimizing flood risks, and improving water distribution efficiency.

5. Communication Network Optimization: The DA has been used in optimizing the design and configuration of communication networks, such as wireless sensor networks and mobile ad hoc networks. It helps in optimizing network coverage, energy efficiency, data routing, and resource allocation in communication systems.

6. Image Processing and Computer Vision: The DA has been employed in image processing and computer vision applications, such as image enhancement, feature extraction, object recognition, and image segmentation. It helps in optimizing image processing algorithms and improving the performance of computer vision systems.

5. A comparison between the dragonfly algorithm and other algorithms

Comparing the Dragonfly Algorithm (DA) with other optimization algorithms can provide insights into their relative strengths and weaknesses. Here, I'll compare the DA with two popular algorithms: Particle Swarm Optimization (PSO) and Genetic Algorithm (GA). Please note that the performance of these algorithms can vary depending on the specific problem and its characteristics.

1. Dragonfly Algorithm vs. Particle Swarm Optimization (PSO):

- Exploration vs. Exploitation: Both the DA and PSO aim to strike a balance between exploration and exploitation. However, the DA often provides better exploration capabilities due to its population-based approach and the introduction of mutation and local search operators. PSO tends to emphasize exploitation,

focusing on the social interaction and velocity update mechanisms.

- Population Structure: The DA utilizes a population structure where each dragonfly represents a potential solution. PSO, on the other hand, maintains a swarm of particles. This population-based approach in the DA allows for a wider exploration of the search space.

- Convergence Speed: PSO often exhibits faster convergence speed compared to the DA. PSO's velocity update mechanism allows particles to quickly converge towards promising regions in the search space. The DA, on the other hand, may take longer to converge, especially in complex and multimodal problems.

    - Premature Convergence: The DA is less prone to premature convergence due to its population-based nature and the inclusion of diversity-promoting mechanisms like mutation. PSO can be more susceptible to premature convergence, especially if the particles converge to local optima.

2. Dragonfly Algorithm vs. Genetic Algorithm (GA):

    - Representation and Operators: Both the DA and GA utilize a population-based approach, but they differ in their representation and genetic operators. The DA uses real-valued vectors to represent solutions, while GA typically uses binary strings or permutations. GAs employ genetic operators such as crossover and mutation, while the DA employs attraction and mutation operators to explore and exploit the search space.

    - Diversity Maintenance: The DA tends to maintain diversity in its population by incorporating attraction mechanisms and mutation operators, which can help prevent premature convergence. GAs also maintain diversity through genetic operators but may require additional techniques like elitism or niche methods to preserve diversity.

    - Convergence Speed: The convergence speed of both algorithms depends on the problem characteristics. GAs often require a large number of generations to converge due to their population-based nature and genetic operations. The DA may also require a substantial number of iterations to converge, especially in complex problems.

    - Scalability: Both the DA and GA face scalability challenges when dealing with high-dimensional and large-scale problems. However, GAs are known to have better scalability due to the binary nature of their representation, allowing them to handle large search spaces more efficiently.

It's important to note that the performance of these algorithms can vary depending on the problem domain, problem characteristics, and parameter settings. It's recommended to perform thorough experimentation and benchmarking to determine the most suitable algorithm for a specific optimization problem.

6. Challenges of the Dragon Fly Algorithm

While the Dragonfly Algorithm (DA) and its variants have shown promise in solving optimization problems, they also face several challenges. Some of the key challenges associated with the Dragonfly Algorithm are:

1. Parameter Selection: Like many optimization algorithms, the Dragonfly Algorithm relies on a set of parameters that need to be carefully chosen to achieve good performance. Selecting the appropriate values for parameters such as population size, step size, and attraction coefficient can be a non-trivial task. The performance of the algorithm can be sensitive to the parameter settings, and finding optimal parameter values often requires extensive experimentation and tuning.

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)
[1] Volume 7- Issue 1, January 2024
[2] Paper : 91

2. Premature Convergence: Premature convergence occurs when the algorithm converges to a suboptimal solution before thoroughly exploring the search space. The Dragonfly Algorithm may suffer from premature convergence, especially when dealing with complex and multimodal optimization problems. The balance between exploration and exploitation is crucial to mitigate this issue, and the design of effective strategies for maintaining diversity within the population is essential.

3. Scalability: The scalability of the Dragonfly Algorithm can be a challenge when dealing with high-dimensional and large-scale optimization problems. As the dimensionality of the problem increases, the search space grows exponentially, making it more difficult for the algorithm to explore and find optimal solutions efficiently. Addressing the scalability challenge often requires the development of specialized techniques or problem-specific adaptations.

4. Convergence Speed: The convergence speed of the Dragonfly Algorithm may vary depending on the problem characteristics and the efficiency of exploration and exploitation mechanisms. In some cases, the algorithm may take a considerable number of iterations to converge to a satisfactory solution. Enhancing the convergence speed is an ongoing research area, and various techniques, such as hybridization with other algorithms or adaptive mechanisms, have been proposed to address this challenge.

5. Benchmarking and Comparisons: Proper benchmarking and performance evaluation of the Dragonfly Algorithm can be challenging. The algorithm's performance can vary depending on the problem type, problem dimensionality, and problem-specific characteristics. Ensuring fair and comprehensive comparisons with other optimization algorithms requires careful selection of benchmark functions, standardized evaluation metrics, and statistically rigorous experimental procedures.

Addressing these challenges requires continuous research and development efforts. Researchers are actively working on advancing the Dragonfly Algorithm and its variants to overcome these limitations and make them more robust, efficient, and applicable to a wide range of optimization problems.

7. Advantages and Disadvantages of the Dragon Fly Algorithm

The Dragonfly Algorithm (DA) and its variants offer several advantages and disadvantages in the context of optimization problem-solving. Let's explore them:

Advantages of Dragonfly Algorithm:

1. Nature-Inspired Approach: The Dragonfly Algorithm is inspired by the behavior of dragonflies in nature, which allows it to mimic their adaptive and intelligent characteristics. This nature-inspired approach can help in solving complex optimization problems by leveraging the inherent efficiency and effectiveness of natural systems.

2. Exploration and Exploitation Balance: The DA strikes a balance between exploration (searching for new solutions) and exploitation (refining promising solutions). This balance allows the algorithm to efficiently explore the solution space, avoiding getting stuck in local optima while also exploiting good solutions to converge towards the global optimum.

3. Population-Based Optimization: The DA employs a population of solutions, which promotes diversity

and allows for parallel search in the solution space. The population-based nature of the algorithm enables it to explore multiple regions simultaneously, increasing the chances of finding better solutions.

4. Robustness and Flexibility: The DA exhibits robustness and flexibility in handling different types of optimization problems. It can be applied to a wide range of domains and problem complexities, making it suitable for various engineering, scientific, and real-world applications.

5. Global Optimality: The Dragonfly Algorithm aims to find the global optimum solution rather than settling for local optima. By employing exploration mechanisms and maintaining diversity in the population, the algorithm has the potential to discover better solutions that are globally optimal.

Disadvantages of Dragonfly Algorithm:

1. Parameter Tuning: The performance of the Dragonfly Algorithm is sensitive to the selection of its parameters, such as population size, step size, and attraction coefficient. Finding optimal parameter values for different problem domains can be challenging and time-consuming.

2. Convergence Speed: The convergence speed of the Dragonfly Algorithm can be slower compared to some other optimization algorithms. It may require a large number of iterations to converge to an optimal solution, particularly for complex and multimodal problems.

3. Premature Convergence: Premature convergence occurs when the algorithm converges to a suboptimal solution before fully exploring the search space. The Dragonfly Algorithm is not immune to this issue, especially when dealing with challenging optimization problems. Ensuring a proper balance between exploration and exploitation is crucial to mitigate premature convergence.

4. Scalability: The scalability of the Dragonfly Algorithm can be a concern when applied to high-dimensional and large-scale optimization problems. As the problem dimensionality increases, the search space grows exponentially, making it more challenging for the algorithm to explore and find optimal solutions efficiently.

5. Benchmarking and Comparison: Proper benchmarking and fair comparison of the Dragonfly Algorithm with other optimization algorithms can be complex. The performance of the algorithm can vary depending on the problem characteristics, making it challenging to draw definitive conclusions about its superiority over other approaches.

8. Results and evaluations of DA

The performance evaluation and results of the Dragonfly Algorithm (DA) have been assessed through various studies and comparisons with other optimization algorithms. Here are some key findings and evaluations of the DA:

1. Optimization Performance: The DA has shown competitive performance in solving a wide range of optimization problems, including both single-objective and multi-objective problems. It has been applied to various domains such as engineering, manufacturing, power systems, water resources management, and image processing, among others.

2. Comparison with Other Algorithms: The DA has been compared with popular optimization algorithms like Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and Differential Evolution (DE) in benchmark functions and real-world problems. The results of these comparisons

vary depending on the problem characteristics, but the DA has demonstrated comparable or even superior performance in terms of solution quality and convergence speed in several cases.

3. Robustness: The DA has shown robustness in dealing with noisy and dynamic optimization problems. Its ability to maintain diversity within the population and balance exploration and exploitation has been beneficial in handling uncertain and changing environments.

4. Scalability: The scalability of the DA has been examined in studies involving high-dimensional optimization problems. While the algorithm's performance can be affected by the curse of dimensionality, the DA has shown reasonable scalability and has been able to handle moderate to high-dimensional problems.

5. Hybridization: Hybridization of the DA with other optimization techniques has been explored to improve its performance further. For example, combining the DA with algorithms like PSO or GA has led to hybrid algorithms that harness the strengths of both approaches, resulting in enhanced optimization performance and convergence speed.

6. Real-World Applications: The DA has been successfully applied to various real-world optimization problems. Examples include optimal design of structures, power system optimization, manufacturing process optimization, and image processing tasks. In these applications, the DA has demonstrated its effectiveness in finding high-quality solutions and improving system performance.

It's important to note that the performance of the DA can vary depending on the specific problem, problem formulation, and parameter settings. Proper parameter tuning and adaptation to the problem domain are crucial for achieving optimal results. Additionally, benchmarking and evaluation studies provide valuable insights into the algorithm's performance and help identify its strengths and limitations in different contexts.

9. Future Works

The Dragonfly Algorithm (DA) and its variants have shown promise in solving optimization problems, but there are still several avenues for future research and development. Hybrid approaches that combine the strengths of the DA with other optimization algorithms can be explored in the future to achieve better convergence, solution quality, and robustness. Extending the DA to handle constraint optimization problems and dynamic optimization problem is an important direction for future research. Investigating parallel and distributed implementations of the DA can improve its scalability and efficiency, particularly for large-scale optimization problems. These future research directions can help further advance the Dragonfly Algorithm and enhance its capabilities in solving complex optimization problems

References

1.Dorigo., M. Optimization, Learning and Natural Algorithms,1992. PhD thesis [in Italian], Dipartimento di Elettronica, Politec-nico di Milano, Milan, Italy.

2. Bonabeau E, Dorigo M, Theraulaz G (1999) Swarm Intelligence:From Natural to Artificial Systems. J Artif Soc Soc Simul 4:320

3. Wahabm Ab, Nefti-Meziani M. S. and Atyabi, A. A Comprehen-sive Review of Swarm Optimization Algorithms. PLOS ONE,
2015, [online] 10(5), p.e 0122827. Available at: <https://journals.plos.org/plosone/article?id=https://doi.org/10.1371/journal.pone.0122827> [Accessed 14 March 2020].

4. Zhang, Y., Wang, S. and Ji, G. A Comprehensive Survey on Par-ticle Swarm Optimization Algorithm and Its Applications. Math-ematical Problems in Engineering, 2015. [online]
Available at: https://www.hindawi.com /journals/mpe/2015/931256/ [Accessed4 Feb. 2018].
5. Ducatelle, F., Di Caro, G. and Gambardella, L., Principles and applications of swarm intelligence for adaptive routing in telecommunications networks. Swarm Intelligence, 2010. [online] 4, 3,
pp.173–198. Available at: <http://people.idsia.ch/~frederick/sij-submitted.pdf> [Accessed 15 March 2020].
6. Kennedy, J. and Eberhart, R. Particle swarm optimization. Proceedings of ICNN'95 - International Conference on Neural Net-
works. 1995. [online] Available at: https://ieeexplore.ieee.org/ document/488968 [Accessed 7 Feb. 2018].
Evolutionary Intelligence1 3
7. Song X, Zhang Y, Guo Y, Sun X, Wang Y (2020) Variable-size cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data. IEEE Trans Evol
Comput 24(5):882–895
8. Ji, X., Zhang, Y., Gong, D. and Sun, X., 2021. Dual-Surrogate Assisted Cooperative Particle Swarm Optimization for Expensive Multimodal Problems. IEEE Transactions on Evolutionary
Computation, pp.1–1.
9. Chakraborty C (2017) Chronic wound image analysis by particle swarm optimization technique for tele-wound network. Wireless Pers Commun 96(3):3655–3671
10. He, S., Wu, Q. and Saunders, J., .Group Search Optimizer: An Optimization Algorithm Inspired by Animal Searching Behavior. IEEE Transactions on Evolutionary Computation, 2009.
[online] 13(5), pp.973–990. Available at: <https://dl.acm.org/doi/https://doi.org/10.1109/TEVC.2009.2011992> [Accessed 19 March 2020].
11. Gandomi, A., Yang, X. and Alavi, A., Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Engineering with Computers, 2011. [online] 29(1), pp.17–35. Available at: <https://link.springer.com/article/https://doi.org/10.1007/s00366-011-0241-y> [Accessed 19 March 2020].
12. Mirjalili, S., Mirjalili, S. and Lewis, A. Grey Wolf Optimizer. Advances in Engineering Software, 2014. [online] 69,pp.46–61. Available at: https://www.sciencedirect.com/science/article/pii/S0965997813001853 [Accessed 3 Jan. 2018].
13. Mirjalili, S. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, andmulti-objective problems. Neural Computing and Applications,
2015. [online] 27(4), pp.1053–1073. Available at: https://link.springer.com/article/https://doi.org/10.1007/s00521-015-1920-1[Accessed 2 Jan. 2018].
14. Yang, X. Harmony Search as a Metaheuristic Algorithm. Music Inspired Harmony Search Algorithm, 2009. [online] pp.1–14.Available at: <https://link.springer.com/chapter/https://doi.org/10.1007/978-3-642-00185-7_1> [Accessed 10 March 2020].
15. Shamsaldin, A., Rashid, T., Al-Rashid Agha, R., Al-Salihi, N.and Mohammadi, M. Donkey and smuggler optimization algorithm: A collaborative working approach to path finding. Journal of Computational Design and Engineering, 2019. [online] 6(4), pp.562–583. Available at: https://www.sciencedirect.com/science/article/pii/S2288430018303178 [Accessed 1 May 2019].
16. Yazdani, M. and Jolai, F.. Lion Optimization Algorithm (LOA):A nature-inspired metaheuristic algorithm. Journal of Computational Design and Engineering, 2016. [online] 3(1), pp.24–36. Available at: https://www.sciencedirect.com/science/article/pii/S2288430015000524 [Accessed 8 Mar. 2019].

17. Rahman C, Rashid T (2021) A new evolutionary algorithm:learner performance based behavior algorithm. Egypt Inform J 22(2):213–223

18.Dai C, Lei X (2018) A Decomposition-based multiobjective evolutionary algorithm with adaptive weight adjustment. Complexity 2018:1–20

19. Rahman, C. and Rashid, T. Dragonfly Algorithm and Its Applications in Applied Science Survey. Computational Intelligence and
Neuroscience, 2019 [online] 2019, pp.1–21. Available at: <https://www.hindawi.com/journals/cin/2019/9293617/> [Accessed 16March 2020].

20. Ram, G., Mandal, D., Kar, R. and Ghoshal, S. Circular and Concentric Circular Antenna Array Synthesis Using Cat Swarm Optimization. IETE Technical Review, 2015. [online] 32(3), pp.204–217.Available at: https://www.tandfonline.com/doi/abs/https:// doi. org/ 10. 1080/ 02564 602. 2014. 10025 43? journalCode=titr20 [Accessed 14 Feb. 2019]