A Survey on Dragonfly Algorithm and its Applications in Engineering

Chnoor M. Rahman [1] and Tarik A. Rashid [2,2]

[1]Affiliation not available
[2]University of Kurdistan Hewler

November 8, 2023

Abstract

Dragonfly algorithm developed in 2016. It is one of the algorithms used by the researchers to optimize an extensive series of uses and applications in various areas. At times, it offers superior performance compared to the most well-known optimization techniques. However, this algorithm faces several difficulties when it is utilized to enhance complex optimization problems. This work addressed the robustness of the method to solve real-world optimization issues, and its deficiency to improve complex optimization problems. This review paper shows a comprehensive investigation of the dragonfly algorithm in the engineering area. First, an overview of the algorithm is discussed. Besides, we also examine the modifications of the algorithm. The merged forms of this algorithm with different techniques and the modifications that have been done to make the algorithm perform better are addressed. Additionally, a survey on applications in the engineering area that used the dragonfly algorithm is offered. A comparison is made between the algorithm and other metaheuristic techniques to show its ability to enhance various problems. The outcomes of the algorithm from the works that utilized the dragonfly algorithm previously and the outcomes of the benchmark test functions proved that in comparison with some techniques, the dragonfly algorithm owns an excellent performance, especially for small to intermediate applications. Moreover, the congestion facts of the technique and some future works are presented. The authors conducted this research to help other researchers who want to study the algorithm and utilize it to optimize engineering problems.

A Survey on Dragonfly Algorithm and its Applications in Engineering

Chnoor M. Rahman[1], and Tarik A. Rashid[2]
[1]Applied Computer Department, College of Health and Applied Sciences, Charmo University, Sulaimany, Iraq.
[2] Science and Engineering Department, School of Science and Engineering, University of Kurdistan Hewler, Erbil, Iraq.

ABSTRACT Dragonfly algorithm developed in 2016. It is one of the algorithms used by the researchers to optimize an extensive series of uses and applications in various areas. At times, it offers superior performance compared to the most well-known optimization techniques. However, this algorithm faces several difficulties when it is utilized to enhance complex optimization problems. This work addressed the robustness of the method to solve real-world optimization issues, and its deficiency to improve complex optimization problems. This review paper shows a comprehensive investigation of the dragonfly algorithm in the engineering area. First, an overview of the algorithm is discussed. Besides, we also examine the modifications of the algorithm. The merged forms of this algorithm with different techniques and the modifications that have been done to make the algorithm perform better are addressed. Additionally, a survey on applications in the engineering area that used the dragonfly algorithm is offered. A comparison is made between the algorithm and other metaheuristic techniques to show its ability to enhance various problems. The outcomes of the algorithm from the works that utilized the dragonfly algorithm previously and the outcomes of the benchmark test functions proved that in comparison with some techniques, the dragonfly algorithm owns an excellent performance, especially for small to intermediate applications. Moreover, the congestion facts of the technique and some future works are presented. The authors conducted this research to help other researchers who want to study the algorithm and utilize it to optimize engineering problems.

KEYWORDS Dragonfly Algorithm, Swarm Intelligence, Metaheuristic Algorithm, Optimization Algorithm, Single and Multi-objective optimization, DA.

I. INTRODUCTION

Many researchers in various areas use swarm intelligence (SI). The ability of natural swarm systems amazed natural scientists and biologists to study the behaviors of swarms and creatures. Swarm-based algorithms are part of the nature-inspired population-based algorithm's family. Regarding complex real-world problems, this group of algorithms produces a good result in terms of cost, speed, and robustness [1]. Bonabeau mentioned SI as the evolving of the combined intellect of sets of modest representatives [2]. Swarm intelligence systems consist of several agents that form a population. Swarm intelligence consists of a collection of intellectual performance of systems that are self-organized and decentralized. Collective clustering and sorting, building nests, and foraging groups of social insects are examples of SI [3]. As discussed in [2], labor division and self-organization are two basic concepts of SI. Self-organization here means the capability of having procedures for developing its innards otherwise agents into fitters procedure with no support from external sources. On the other hand, the labor division indicates the implementation of numerous feasible with meek jobs by people, simultaneously. In SI, agents follow simple rules. No centralized control structure exists to control the behaviors of individuals. In reality, the individual's behaviors are local and random to an extent. Agents, however, interact with each other, which produces intelligent and new actions [4]. SI recently has been applied to different problems in continuous and combinatorial optimization, robotics, telecommunications, etc.; often-magnificent results were produced [5]. Lately, the researchers have proposed some new techniques. Particle swarm optimization or (PSO) that was suggested by Kennedy and Eberhart [6]. PSO is one of the first-born algorithms in the swarm intelligence field. PSO emulates the purposes of a collection of fish or birds. Each particle is a particular agent with a location in the exploration space. S. He et al. proposed Group Search Optimizer (GSO) [7]. GSO mimics the searching behavior of animals. Cuckoo Search (CS) algorithm imitates the process of reproduction in the cuckoo family [8]. Later in 2014, Mirjalili et al. developed Grey Wolf Optimizer [9]. It imitates the hunting behavior of wolfs. Later, in reference [10], Mirjalili proposed a dragonfly optimization algorithm (DA). DA mainly mimics the behaviors of hunting and migration of dragonflies. Harmony Search (HA) algorithm proposed in [11]. It mimics the process of improving music by the musician. The musician tries to provide better harmony depending on his/her experiences and searching for better harmonies. Donkey and Smuggler Optimization algorithm or so-called (DSO) suggested in [12]. DSO imitates the attitudes of donkeys to select and search routes. Yazdani et al. developed another example of nature-inspired algorithms, which is called the Lion Optimization Algorithm (LOA) [13]. The LOA mimics the lion's cooperation behavior and their unique lifestyle. Based on a social organization, the lions divide into residents and nomads. The residents consist of several lions that live together, and they are called pride. Nomads, on the other hand, are mostly seen in pairs and sometimes singularly. Lions may change their lifestyle from nomads to residents or vice versa.

Different researchers have used DA in numerous diverse applications and it gave satisfactory results. Until the end of working on this review paper (March 2019), almost 300 different works cited the dragonfly algorithm in different areas. It produced satisfying results in almost all applications. Additionally, the authors of this review paper published another review paper on the DA and its applications in applied science [14]. In that review paper, the authors cantered their review on the applied science area (such as image processing, machine learning, wireless, and networking). Dragonfly algorithm used for optimizing a huge number of problems in various disciplines. One review paper cannot cover all the articles that used the DA. Thus, in this paper, DA and its engineering applications are focused and reviewed.

This work first shows a short overview of the dragonfly algorithm in section two. Next, we discuss the variants of the algorithm in section three. Afterward, in section four, the authors address some of the hybridization versions related to the DA algorithm with other algorithms. In section five, the applications that were solved by the DA in the field of engineering are presented. Additionally, in section six, the DA is compared with other metaheuristics. The algorithm is then evaluated using the traditional benchmark functions and the CEC-2019 (The 100-Digit Challenge) benchmark functions in section seven. The evaluations are then compared with the FA and PSO. The Wilcoxon rank-sum is utilized for testing the meaning of the outcomes statistically. Furthermore, in section eight, a discussion and some problems that encounter the DA's operators are dealt with in conjunction with giving explanations and prospect works for enhancing the capability of DA. Lastly, the key points of this research work are established in section nine.

II. DRAGONFLY ALGORITHM

In the last few decades, the natural behavior of creatures has widely motivated metaheuristic optimization algorithms. Swarm intelligence is the main inspiration for the metaheuristics [6, 15]. DA is a metaheuristic optimization method. It imitates the swarming attitudes of dragonflies [10].

Dragonflies are little predators. They hunt insects in nature. The main reason for the dragonflies swarming is hunting and migration; in other words, these are two phases; static and dynamic swarms, respectively. In the first phase, which is the static swarming, a set of dragonfly generate sub-swarms and search through different small areas. On the other hand, the second phase, which is dynamic swarming, a set of dragonflies can fly in a much bigger swarm. They fly in one direction towards the most promising global optimum location [10].

In dynamic swarming, dragonflies maintain a reasonable separation and cohesion (intensification or exploitation). In static swarming, conversely, alignment can be too minor besides; cohesion is big for attacking prey (diversification or exploration). Therefore, little cohesion and great alignment weights will be assigned to individuals once exploring the search space. However, they will be assigned to high cohesion and low alignment weights while exploiting the search space. The neighborhood radii proportionally enflamed to the iteration number for changeover between intensification and diversification. Another way for balancing intensification and diversification is tuning the swarming weights adaptively during the process of optimization. The swarming weights are; attraction motion towards food (*f*), separation (*s*), inertia weight (*w*), cohesion (*c*), alignment (*a*), and distraction outwards predators (*e*). The Following are the equations for the swarming weights:

Reynolds in [16] mentioned that Equation (1) can be used for computing separation:

$$S_i = -\sum_{j=1}^{N} \mathrm{X} - \mathrm{X}_j \tag{1}$$

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.

The alignment can be calculated by Equation (2) [10].

$$A_i = \frac{\sum_{j=1}^{N} V_j}{N} \tag{2}$$

A_i specifies the motion of alignment for i^{th} dragonfly.
V specifies a j^{th} dragonfly's velocity in the neighborhood.

Equation (3) is for calculating cohesion:

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

C_i specifies the i^{th} dragonfly's cohesion.
N specifies the neighborhood size.
X_j specifies the j^{th} dragonfly's position in the neighborhood.
X specifies the present individual.

Equation (4) is for calculating attraction motion towards food:

$$F_i = X^+ - X \tag{4}$$

F_i specifies the attraction of food of the i^{th} individual.
X^+ specifies the food source's position.
X specifies the current individual's position.

Equation (5) is for calculating distraction outwards predator:

$$E_i = X^- + X \quad (5)$$

E_i specifies the distraction motion of the enemy for the i^{th} dragonfly.
X^- specifies the position of the enemy.
X specifies the dragonfly's current position.

Individuals' positions of the artificial dragonfly are updated in the exploration space utilizing vectors, namely; ΔX, which is called step vector and X, which is called position vector. ΔX in the dragonfly algorithm is equivalent to the velocity in particle swarm optimization. Updating the position of individuals in DA mainly depends on the PSO algorithm. Whereas X specifies the movement direction in dragonfly individuals. X can be computed as follows [10]:

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + w\Delta X_t \quad (6)$$

s specifies the weight of separation.
S_i specifies the i^{th} individual separation.
specifies the weight of alignment.
A_i specifies i^{th} dragonfly's alignment.
c specifies the weight of cohesion.
C_i specifies i^{th} dragonfly's cohesion.
f specifies the weight of food attraction.
F_i specifies the i^{th} individual food source.
e specifies the weight of enemies' distraction.
E_i specifies the i^{th} dragonfly's enemy position.
w specifies the weight of inertia.
t indicates the counter of iteration.

Once calculating the ΔX, the calculation for the X starts in this manner:

$$X_{t+1} = X_t + \Delta X_{t+1} \quad (7)$$

t specifies the existing iteration.

We should add a random move to the searching technique to upsurge the exploration likelihood of the entire choice space through an optimization technique. When neighboring solutions do not exist to flyover throughout the exploration space, the dragonflies would use a method of random walk or so-called Lévy flight. Here, the dragonfly's position is modified as follows:

$$X_{t+1} = X_t + Lévy(d) \times X_t \quad (8)$$

As mentioned t indicates the present iteration and (d) specifies the position vector's dimension.

Reference [17] stated that although using the Lévy flight improves the performance of DA, however, it might cause very long steps. In the mentioned reference, to avoid this drawback, Brownian motion was used in place of Lévy flight. The motion of Brownian is another mechanism of random motion. The free liquid or gas molecules movement has inspired this. The modified DA complexity was O; the size of the population multiplied by the iteration number. The calculated complexity proved that using Brownian motion did not have an impact on the complexity time of the original DA. By using the Brownian motion, the massive jumps caused by the Lévy flight were corrected. However, occasionally sudden moves may still be required to avoid trapping into local optima. For objectives with local minima, the Brownian motion produced better solutions in a shorter time.

For the changeover between exploitation and exploration, dragonfly individuals change their weights adaptively. To adjust the flying path during the process of optimization, the neighborhood area should be enlarged, hence before the optimization process ends; the whole swarm becomes one group for converging to the global optimum.

III. VARIANTS OF DRAGONFLY ALGORITHM

Dragonfly algorithm has three different versions:

A. SINGLE OBJECTIVE PROBLEMS WITH DA

Like most Si-based optimization algorithms, DA initially creates a solution set randomly for the optimization problem in hand. At first, the position and step vectors of individuals assigned to arbitrary values between both upper and lower variables' bounds. Positions and step vectors are updated for all dragonflies per iteration. For updating the vectors; position and step, the dragonfly's region is selected through the Euclidean distance calculation between all the individuals. Iteratively, the individual's position updating continues until the end criterion is met.

B. BINARY DRAGONFLY ALGORITHM

Since in binary search space only 0 or 1 can be assigned to the position vector, adding step vectors to position vector cannot update the position of search agents. The transfer function produces a binary technique from a continuous SI technique. The velocity (step) values work as an input to the transfer function, and then the transfer function yields a number between (0 and 1) as result, which states the likelihood of moving the individuals and updating their position. Alike to continuous optimization, transfer function reproduces unexpected variations in particles by significant velocity. Equation (9) is for computing the probability of changing the positions of all dragonflies [18].

$$T(\Delta X) = \left|\frac{\Delta X}{\sqrt{\Delta X^2 + 1}}\right| \tag{9}$$

In binary search spaces and after utilizing Equation (9), Equation (10) updates the location of the search agent.

$$X_{t+1} = \begin{cases} \neg X_t & r < T(\Delta X_{t+1}) \\ X_t & r \geq T(\Delta X_{t+1}) \end{cases} \tag{10}$$

r ranges between 0 and 1.

BDA assumes that all of the individuals are in one swarm. Hence, it adaptively tunes the swarming factors such *s*, *f*, *c*, *a*, *e* and *w* to simulate intensification and diversification.

Reference [19] used BDA for feature selection. This work proved the importance of the role of the transfer function for producing the discrete space from the continuous one and an enhanced balance concerning the phases of exploitation and exploration. The proposed work stated that Equation (9) does not provide the right balance concerning the phases of exploitation and exploration, wherein the start of the optimization, the exploration rate should be higher than exploitation. Hence, on the way to rise the BDA's performance and circumvent falling into local optima, time-dependent transfer function (TF) was used.

The value of time-dependent TF linearly gets bigger as the step vector of the search agents gets more significant. Consequently, in the early steps of the algorithm, higher exploration is provided. However, as time passes the probability of exploitation increases, and the probability of exploration decreases.

As proved in this work, the examined TF enhanced the performance of the BDA. The main reason for this was providing the correct balance concerning the phases of exploitation and exploration of the BDA.

The computational complexity of the BDA using the time-dependent TF is the same as the original BDA, and it is $O(ISD)$.

Where *I* specifies the iteration number, *S* specifies the solution number, and *D* specifies the dimension number.

C. MULTI-OBJECTIVE PROBLEMS WITH DA

These problems contain more than one objective. The answer to this type of difficulties can be often known as a set of Pareto Optimal. The finest compromise among the existing objectives present in this set [20]. The Pareto optimal dominance compares two solutions in multi-objective search space [21]. DA is first provided with an archive or record for saving and getting superior solutions of Pareto optimal throughout the process. The food source would come from the archive update the position, and the rest of the procedure is similar to that of the DA.

Likewise, the multi-objective particle swarm optimization or MOPSO algorithm [22], for finding the best Pareto optimal front, the food source can be chosen from the minimum populous zone of the present Pareto optimal front. For including the entire solutions usually, a hypersphere can be defined. In each iteration, equal sub-hyper-spheres are produced by dividing the hyper-spheres. Whenever the segments are produced, for each segment a roulette-wheel technique with a specific probability is utilized for the process of selection [23].

Multi-objective problems with DA or MODA has a better likelihood of selecting the food source from a smaller amount of populous segments. Contrarily, to choose hunters from the record or archive, the equation chooses the worst or most populous hyper-sphere, so that the dragonfly individuals are discouraged to hunt about unpromising zones.
In each iteration, the archive updates regularly, and it may become full during the process of optimization. Hence, to prevent that situation there should be a technique. If as a minimum one of the habitations govern the solution, then it should not go into the

records. On the other hand, if the solution controls some of the solutions of Pareto optimal, then the solution will be added to the record, and all the Pareto optimal solutions will be deleted. If the archive or record becomes full, some solutions from the most populated segments will be deleted [23]. MODA has two extra parameters, which do not exist in the DA: one of the parameters is for describing the max number of hyper-spheres, and the second one is for defining the size of the archive.

In reference [24], the authors modified the multi-objective DA. In this work, the crowded distance selection mechanism from NSGA-III was used instead of the roulette wheel mechanism for the multi-objective DA. NSGA-III developed in [25].

IV. HYBRIDIZED VERSIONS OF DRAGONFLY ALGORITHM

One of the most popular techniques to enhance the ability of metaheuristic algorithms is merging the strong properties of different algorithms. As a result, a novel algorithm will be produced based on the features of the amalgamated algorithms [26]. Some of the hybridized DA versions in other areas are discussed in [14]. The rest of the hybridized versions of the DA can be discussed in this section.

A huge number of social interactions in DA causes trapping into local optima, solving problems with less accuracy, and with an improper balance of exploitation and exploration. In reference [27], to overcome these deficiencies, DA combined by a better type of the Nelder-Mead algorithm or so-called INMDA for making the ability of local exploration more powerful, and avoid falling into local optima, the INMDA can be divided into two steps, the first step, DA utilized to explore the solution space. It provided the necessary variety to the artificial dragonflies to find the global optimum. The second step is the enhanced type of the Nelder-Mead (INM) simplex technique used for finding the worst point and the best point and calculating the population centroid. The key feature of the INM was that the centroid of the population utilized for updating the position. Hence, the chance of trapping into local optima reduced.

For high-dimensional problems, the produced results proved that the examined work performed better comparing to DA and MHDA and that they are not a good choice for solving problems of high dimensional as they rapidly run into a dimensional curse. The high performance of the proposed work came from the improved ability of both exploitation and exploration of reverse learning techniques.

In reference [28], the DA's strength combined with ABC. The aim of hybridizing these two metaheuristics was to eliminate the convergence speed problem and falling into local optima by providing a better steadiness concerning local and global search constituents of the contributed techniques.

Reference [29] proposed an adaptive DA to optimize frame structures. In this article, Coulomb force search strategy or CFSS combined with DA. The exploratory constant parameter (k) is one of the essential parameters in the Coulomb force search strategy. This work examined the utilization of adapted value in the course of the searching procedure of the dragonfly algorithm. The dragonflies encouraged for searching in the search space with giant steps at the beginning of the process and small steps at the end of the process. The above-mentioned adaptive strategy improved the convergence of the algorithm. Hence, it produced an optimal result in a short time evaluated against the standard algorithm, then the compared results to the DA and BDA from the proposed technique proved this. The proposed algorithm used to optimize the front axle of an automobile. In the examined problem, the front axle beam selected. The outcomes substantiated that the convergence speed of the CFSS-DA compared with the BDA and DA.

V. APPLICATIONS OF DA IN ENGINEERING

The ability of DA encourages numerous academics to apply it to optimize different applications in various areas. In the following subsections, we discuss applications of the dragonfly algorithm in Engineering and Physics.

A. MECHANICAL ENGINEERING

Network configuration is the practice of altering the position of open or close switches to make changes in the distribution network's topological structure. In [30], a new reconfiguration schema developed to reduce the net deviation among the nominal voltage value, and the node voltages using a dragonfly optimization algorithm. Dragonfly optimization algorithm based reconfiguration method or DORM enhanced the voltage profile or VP by the net voltage deviation minimization or NVD. The proposed technique examined without making any thermal violations. It also kept the radial structure. In this study, the results obtained using DORM compared to some other nature-inspired algorithms for solving configuration problems, such as PSO [31], GA [32], and BBO [33]. According to the study, the obtained results proved that the DORM provided better configuration that minimized NVD and provided good VP. To share the loads with the conventional power plant, distribute generation units utilized. The mentioned units also used to give the power to the loads individually. Wind turbine (WT), Photovoltaic (PV), gas turbine (GT), and micro-turbine (MT) and storage battery (SB) are the most typical distributed generation units in this type of application.

In reference [34], a novel optimal scheme of a different hybrid power generation system generated, which is called (HPGS). The introduced design consisted of a combination of PV, WT, GT, and SB. Natural gas distribution networks utilized to fuel the GT of the system. To find the optimal design of the proposed work, two metaheuristic techniques; DA [10] and GWO [9] examined. The system considered different weather conditions. Both metaheuristic algorithms in this work used for minimizing the annual

cost and entire emission functions for the system. It concluded that the DA produced better results in respect of the total yearly cost comparing to GWO. In contrast, regarding the system pollution, GWO technique produced better results than the DA technique.

Perforated plates are part of many industrial applications in recent years. Perforated plate cutouts mostly used to decrease the structure weight or to build a point of exit and entry. Cutouts in the plates can change the geometry of the plate, which leads to severe local stresses or called stress concentration throughout the cutouts. This can cause a reduction in strength and premature failure in structures. Therefore, knowing useful parameters to reduce stress concentration in various structures is crucial. In reference [35], DA used to optimize the involved parameters in analyzing the stress of the perforated orthotropic plates. The aim was to achieve the minimum stress value nearby the quasi-triangular cutout positioned in a boundless orthotropic license plate. Dissemination of stress computed employing the suggested technique established on the analytical solution of Lekhnitskii. The variables that designed using the proposed technique included load angle, and the material properties, bluntness, fiber angle, and cut-out orientation angle, the outcomes evaluated against PSO and GA. The factors of the stated algorithms shown in Table 1. The outcomes evidenced that the regular of best values of stress produced via the DA was smaller compared to other algorithms. It concluded that the values of both average and standard deviation for the DA were smaller than the GA and PSO [36]. The comparison of these techniques proved that DA showed excellent performance to solve the problem mentioned above, and it operated more steadily. It was also determined that the high exploration and exploitation rates in the DA were reasons that made the algorithm to perform better. Moreover, DA converged much earlier (18^{th} iteration), whereas PSO and GA converged in the iterations 95^{th} and 146^{th}, respectively. Additionally, depending on the results, it was observed that the most significant levels of stress in all cutout bluntness or *w* happened on 45°-load angle.

The robust non-linear link concerning the array factor and the array's elements and marks the concentric circular antenna array problem or so-called CCAA synthesis challenging. A high maximum sidelobe level or (MSL) is a problem of CCAAs. Reference [37] used DA to design CCAA in a way that was able to get low side lobes. A sub-structured neural network (SSANN) was used instead of a single artificial neural network or ANN, which improved the forecast accuracy of the effectiveness of requalification sub-ANNs and the engine working process. The proposed work aimed at observing and exploring the effectiveness of the DA technique. Moreover, in this work, four different CCAA design cases used to study DA efficiency. Then, the results evaluated against the current approaches like BBO [38], SOS [39], SQP [38], CSO [40], OGSA [41], EP [42], and FA [43]. The proposed work utilized two three-ring designs; CCAA with 4-, 6-, 8- plus 8-, 10-, 12-, besides two cases well-thought-out for each model: CCAA without, and with the center component. For each scheme test, the space between neighboring elements in every ring was fixed to 0.55, 0.606, and 0.75 from the center to the outermost ring. The outcomes of the DA evaluated against the techniques in the literature of the work using the uniform array. The outcomes showed that the DA had better performance for the mentioned problem, and it was competitive with other methods for decreasing MSL.

Table 1: *Parameter Settings of The Algorithms [35]*

DA	GA	PSO
Population size = 30	Population size = 30	Population size = 30
Max. No. Of iterations = 200	Max. No. Of Iterations = 200	Max. No. Of Iterations = 200
Random values = $r_1 = r_2 = [0, 1]$	Probability of crossover (Pc) = 0.8	Cognitive component = $c_1 = 2$
Separation weight (s′) = 0.1	Probability of Mutation (Pm) = 0.03	Social component = $c_2 = 2$
Alignment weight (a′) = 0.1	ncrossover = 2*round(npop *Pc/2)	$\omega = \frac{0.1}{\left\lvert 1\frac{c}{2}\frac{\sqrt{\lvert c^2-4c\rvert}}{2}\right\rvert}$, $c = c_1 + c_2$
Cohesion weight (c′) = 0.7	nmut = npop*Pm	
Food factor (f′) = 1		
Enemy factor (e′) = 1		
Inertia factor (δ+ = 0.9-0.2		
Constant (ζ) = 1.5		

In reference [44], automatic generation control of an interconnected two-area multi-source hydrothermal power system considered. The performance of the scrutinized system was evaluated and planned with proportional-integral (PI), proportional integral derivative (PID), and 2 degrees of freedom PID or so-called 2DOF PID. The DA was used to optimize the controller gains. It concluded that the DA provided superior results compared to classical methods. Furthermore, the 2DOF PID controller optimized by DA produced less overshoot (OS), settling time (ST), undershoot (US). Moreover, smaller values of the objective function provided compared to the 2DOF PID controller optimized by DE.

Optimization can significantly affect the process of grinding by improving the quality of products and reduce operational costs and time of production. Optimizing the grinding process is a challenging process in the engineering field because of the complexity and nonlinearity of the process. In reference [45], multi-objective DA used for obtaining solutions of non-dominated Pareto optimal. In this work, an experimental example in [46] was used. Then, the outcomes were evaluated against the outcomes of an experimental model using NSGA-II in [46]. The solutions of Pareto optimal produced via MODA conquered the attained solutions via the NSGA-II. The outcomes showed that MODA accomplished better compared to the NSGA-II in resolving the multi-objective mathematical model of the grinding process, the reason for this superiority was due to the MODA's efficient operators evaluated against the simple operators of NSGA-II (crossover and mutation). The solutions produced by MODA improved surface roughness significantly and reduced the costs and the total grinding time. The results proved that all the objectives were optimized

by MODA simultaneously using the algorithm's efficient operators. MODA used 30 individuals and 1000 iterations to examine the mathematical model of tri-objective of the grinding process. On the other hand, NSGA-II utilized 100 chromosomes and 1000 iterations that caused a 100,000 number of function evaluations. The results proved that the MODA's computational cost was much lower than the NSGA-II's.

Reference [47] used MODA for optimizing the performance of switched reluctance motor or so-called SRM powered by autonomous stacked proton exchange membrane fuel cells (PEMFC). MODA used to produce the best sets of driving circuit's turn-on/off angles. As mentioned, the best sets produced via DA could improve the savings in energy and increase the performing of isolated PEMFC-SRM. Dragonfly's ability in developing the initial stochastic population and the good exploitation and exploration of DA were the reasons aimed at the superiority of the algorithm for solving this problem. Furthermore, DA provides a high uniformly disseminated Pareto optimal set of solutions in problems of multi-objective [48].

B. ELECTRICAL ENGINEERING

A new technique for designing, modeling, and optimizing a uniform serpentine meander based on MEMS switch incorporating beam puncture effect discovered in [49]. A new analytical model was suggested, which aimed at pull-in voltage in this research work. An optimization technique was introduced for finding the best configuration of the switch to accomplish the least possible pull-in voltage. Here, the analytical model was used as an objective function. For this purpose, the author utilized several great evolutionary optimization methods for achieving the best measurements with less cost computationally and more simplicity. The conducted techniques included PSO, DE, a hybrid PSO with differential evolution (DEPSO), DA, WOA, and human behavior based PSO (HBPSO). A comparison among the applied algorithms showed that the DA had the best minimum pull-in voltage with the smallest errors. The parameter settings for DA in the proposed work were: dimension = 8, search agents = 50, alignment weight, separation weight, and cohesion weight were random between -0.2 and 0.2, food attraction weight was a random, and enemy distraction weight was a value between -0.1 and 1. The results showed that the DA performance was the best to minimize pull-in voltage with minimum errors.

In the power transmission system, the stability of voltage is a significant concern due to inconsistency between demand and power generation. Reference [50] utilized the eigenvalue decomposition (EVD) method and DA in partitioned Y-admittance matrix to identify weak buses for implementing the compensators of reactive power. In this work, DA used to enhance the static VAR compensator' size and cost. Regarding the objective function, line flows, voltage deviation, and reactive power limit was examined as the design constraints. The results proved that the proposed technique maximized the cost of static VAR compensator and the cost of installation with the loading condition. Besides, the voltage deviation and the actual power loss in the DA were much lesser compared to the PSO. Moreover, the DA could show its superiority in reducing real power loss for the IEEE 30 bus system compared to the other algorithms, additionally, DA converged earlier.

Atomic generation control (AGC) problem was examined in reference [51] by using DA. In this work, the DA optimized the control parameters, for example, scaling factors of fuzzy logic and PID gains. The criterion of integral of time multiplied absolute error (ITAE) was used to minimize the settling time with a minimized peak overshoot. The ITAE employed for optimizing the scaling factor and PID gains controller. The addressed control strategy examined through two equal non-reheat thermal interrelated power system areas. The work stretched to two hydrothermal power system areas joined via a high voltage direct current or so-called HVDC transmission link and an AC tie line. To deal with non-linearity, the generation rate constraint (GRC) effect counted. The results proved that in terms of lowest damping oscillations, settling time, peak undershoots, and overshoot in the interrelated three-area power system through GRC non-linearity, the proposed metaheuristic algorithm based fuzzy PID controller provided superior results evaluated against further control methods. The results proved that the DA as an optimization technique produced a better optimum solution of AGC for non-linear and linear interconnected power systems' frequency regulation. Furthermore, the combined fuzzy PID controller proposed in this work proved its superiority over the fuzzy logic and optimized PID controller.

C. OPTIMAL PARAMETERS

Reference [52] optimized the factors in the examining stress of perforated orthotropic plates. In this work, the DA utilized to compute the stress distribution based on the analytical solution of Lekhnitskii. Fiber angle, load angle, orientation cutout, bluntness, and material properties. The results obtained from the dragonfly algorithm in this work evaluated against the results of GA [53] and PSO [6]. The results proved that in comparison to the PSO and GA, the DA converged earlier. Besides, avoiding local optimum and producing better results proved the DA's supremacy compared with the other two algorithms. The DA also produced smaller average values of optimum stress compared to the other algorithms. Furthermore, by using the DA, a standard deviation closer to zero was produced, which was smaller to the ones produced using the PSO and GA.

Providing reliable and continuous supply to customers is a critical ambition of utility and meets the expectations of power balance and the loss of transmission when the generators operate within a specified limit. For achieving this purpose, the value of emission and the fuel cost ought to be as insignificant as conceivable. The allowed deviation in feasible tolerance and fuel cost is named as emission constrained economic dispatch or so-called ECED problem. Reference [54] used DA for finding an explanation for the problem of ECED. ECED problem of a multi-objective. In this work, the value of emission and the fuel cost alongside quadratic function was treated as a problem of multi-objective. To convert the problem to a single-objective, the price penalty factor

technique used. The consequences of penalty factors, such as Min-Min, Min-Max, Max-Max, Max-Min emission value of different gas exhalations, and price penalty factors mentioned in this work. As the results in this work showed that using "Min-Max" as the price penalty factor produced less fuel cost compared to the other penalty factors. In "Common", however, increasing ECED fuel cost by 17% could reduce emission by almost 23% in comparison with the price penalty factor of "Min-Max". The author mentioned that nowadays having a small amount of ECED fuel cost to operate thermal power plants with "Min-Max" price penalty creates contamination in the environment and causes premature death in humans leaving near the thermal power plant.

Reference [55] introduced a new technique to participate in online engine calibration and to control increasing the performance of the engine, and decrease gas emission of the greenhouse. For this purpose, the mentioned reference used a robust model centered on a multi-objective genetic algorithm or NSGA-II, multi-objective dragonfly algorithm, fuzzy dependent on inference system, and sub-structural neural network or SSANN. Throttle angle, injection angle, engine rpm, and injection time were used as the inputs for SSANN. The fuel flow (FF), CO, torque, and NOx were used as outputs. Initially, the data from GT-POWER used to train SSANN. Based on various engine speeds, 15 working points were selected randomly to examine the accuracy of SSANN. Linear regression was utilized for assessing the linear relationship between the measured and predicted outputs. For this problem, MODA converged earlier (at the 40^{th} generation), and it had better inverse generation distance (IGD). However, NSGA-II converged after the 80^{th} generation. Besides, it was discovered that increasing the number of iterations MODA showed better convergence. It was because of the use of the food/enemy selection technique in the MODA.

In reference [56], the vibrant strength of the hybrid energy distributed power system (HEDPS) considered. The HEDPS was subject to wind power and load variations. A controller with three degrees of freedom (3-DOF) proportional-integral-derivative (PID) implemented and designed in the HEDPS to balance frequency fluctuations and power after the perturbation. Unlike the single-degree-of-freedom (1-DOF) controller, the 3-DOF controllers own the ability of an outstanding set-point tracking, and it produces superior regulations for the input disturbance. DA used for optimizing the factors of 3-DOF PID controllers. Also, to optimize the 3-DOF controller gains, integral time absolute error (ITAE) used as an objective function. The achieved outcomes evaluated against the outcomes of other popular metaheuristic algorithms, such as Zeigler-Nichols or so-called ZN. The isolated, interconnected modes of hybrid energy and distributed power system implemented for assessing the proposed controller's performance. For qualitative assessment, the convergence of DA evaluated against other algorithms. The outcomes demonstrated that the dragonfly algorithm established the value of global optimum by a quicker rate and that lesser minimum value for the fitness function generated compared to the other participated algorithms. For this work, all the algorithms generated the optimal global point between 60 to 70 generations, which gave the choice of having 100 iterations. Furthermore, the results concluded that the DA outpaced the other stated algorithms with regards to faster convergence and the value of minimum fitness.

D. ECONOMIC LOAD DISPATCH

Wind integrated system with valve-point effect considered in [57]. DA used to overcome the problem of economic load dispatch (ELD) along with valve-point effect. The Weibull distribution function used to model the stochastic nature of wind. Furthermore, a closed integral function was used to analyze the overestimation/underestimation cost. In the proposed work, the optimization technique started by generating a set of random solutions for the assumed problem. The dragonfly's vectors (position and step) randomly initialized within upper and lower bounds of generators. The outcomes exhibited that the DA successfully resolved the power system of the economic dispatch of the wind thermal integrated system. Two cases and the IEEE-30 bus system implemented for calculating the performance. The problem of non-convex economic dispatch solved in the first case. The obtained results from this case compared to a sequential quadratic programming particle swarm optimization (SQP-PSO) technique. ELD with wind power penetration was solved using DA in the second case. Moreover, the performance of the work in case 2 compared to SQP-PSO [58]. In both cases, 1200MW considered a load demand. The results showed that the DA found a global optimum solution and it was remarkably unrestricted from tricking into local optima.

In reference [59], the dragonfly algorithm applied to improve a novel technique to resolve economic dispatch incorporating solar energy. In carrying out the economic dispatch, the mentioned reference considered prohibited operating zone and valve-point loading constraints. Beta distribution function applied for modeling the solar energy system and the objective function. The output predicted to include four diverse periods. Various loading circumstances considered for each. The proposed work addressed that compared to other optimization methods dragonfly algorithm gave a low cost, minimum power loss, and converges in the minimum running time. It is concluded that more power could be generated if the availability of the sun was abundant in the chosen location. Moreover, in the case of using the produced system power correctly, the economy will maximize, and the system loss will minimize.

The proposed work considered three different cases. In case 1, the system used for testing consisted of six generators and 1263 MW. The results from the first case study compared to the most recent optimization methods. The results proved that the DA was the best regarding the convergence time, the smallest objective function, power loss, and evaluations. Similarly, in terms of generations, cost, and transmission loss, DA was the best. Concerning case 2, the number of used generators was 15, and 2630 MW considered. Here, the total cost generation for the DA was minimum. In case 3, the 86 bus test system utilized in south Indian. It consisted of 7 generators, 131 lines, and 86 buses. This case considered ramp rate constraint, transmission loss, down reserve

constraints, and up the reserve. Here, the obtained results proved that the optimal cost of DA was much smaller than the completive algorithms.

E. LOSS REDUCTION

The research work [60] based on the BDA. A new technique for wrapper-selection was proposed in the research work. The suggested technique aimed at diminishing the number of characteristics concerning the standard feature set and obtain better accuracy in classification at the same time. The K-Nearest Neighbourhood or KNN classifier applied to test the selected subset of the feature. The subset of feature selection is a problem of multi-objective. Problems of multi-objective study two diverse goals. The proposed work aimed at maximizing the accuracy of classification, and diminishing the features. Equation (11) shows the objective function. The proposed approach evaluated using 18 UCI datasets. A comparison made between the proposed technique and the similar techniques that used GA, and PSO. The comparison is concerned with the accuracy of classification and number of the carefully chosen attributes. The outcomes proved that BDA had a superior ability in examining the space of features and choosing the features with more information for the task of classification.

$$Fitness = a y_R(D) + \beta \frac{|R|}{|C|} \quad (11)$$

$y_R(D)$ shows the rate of error of the classification used.
|R| represents the selected subset's cardinality.
|C| signifies the whole number of characteristics included with the dataset.
a And β signify factors representing the classification importance and length of the subset, respectively.
$a \in [0, 1]$ and β = (1- a), the author adopted these from [61].

Reference [62] solved a nearly-zero-energy-building design problem. A comparison made in terms of performance among seven multi-objective algorithms. In the utmost of the cases, the attained solutions enhanced by increasing the generation number. Each algorithm ran 20 times with moderately raising the evaluation number. The optimization results in most running cases proved that the results of MODA were uncompetitive. In terms of contribution and running time, MODA was not competitive, and it was slow. According to this work, MODA did not have any outstanding features.

Power loss, electric distribution system's maximum loadability, and voltage stability margin (VSM) are greatly affected by inadequate reactive power generation. To solve these problems, in reference [63], optimal concurrent as well as multiple separate installations of distributed generation (DG), and capacitor were examined. For this work, minimizing the total of reactive power loss (Q_L) counted as the primary objective, and DA used to optimize the problem. Standard 33-bus distribution systems utilized to test the methodology proposed in this work. The proposed work handled different capacitors and DG installation cases. The results of the proposed work compared to weight improved particle swarm optimization or WIPSO technique. The results proved that the primary behavior of DA for updating the individual's position provided an enhanced Q_L reduction compared to the other methods. The results also showed a better convergence rate by producing fitter solutions in 15 to 20 iterations.

VI. A COMPARISON BETWEEN DRAGONFLY ALGORITHM AND OTHER ALGORITHMS

Reference [64] addressed an assignment of court cases that has an impact on enhancing the effectiveness of the jurisdictional structure. The effectiveness of the jurisdictional structure extremely relies on punctuality and operating the court cases efficiently. In the proposed work, mixed-integer linear programming or MILP utilized to solve the problem of assigning cases in the justice court. The objective function of this issue was assigning *N* cases to *M* groups. Each group might cope with the cases altogether. However, because of the requirement of the cases, personal potentiality, and other assigned cases, the necessary time for each group to solve the same case was not the same. To find the best solution for the proposed work, DA and the firefly algorithm (FA) utilized [41]. Two problems were assessed in a uniform distribution. In the form P1: Lower bound (L_l) = (1, 30), Upper bound (U_i) = (1, 90), efficacy rate (μ_i) = (1, 90), and P2: Lower bound (L_l) = (1, 60), Upper bound (U_i) = (1, 90), efficacy rate (μ_i) = (1, 90). The outcomes exhibited that for finding the best solution the DA required less time and an average percentage deviation to maximize efficacy compared to the firefly algorithm. The outcomes proved that in 50 cases and three-justice groups aiming at trial parameters: P1 (50:3, 4, 5) and P2 (50: 3, 4, 5), the DA was greater compared to FA.

In [65], GWO, DA, and MFO algorithms assessed for optimizing the best sitting of the capacitor in several radial distribution systems or RDSs. The loss sensitivity factor examined for discovering the candidate buses. The authors considered 33-, 69-, and 118-bus RDSs to prove the efficiency and effectiveness of the addressed optimization technique. This study aimed at minimizing the total cost with voltage profile improvement and power loss. The outcomes evaluated against the outcomes of the PSO for showing the advantage of the utilized methods. The GWO-, DA-, and MFO-based techniques produced better outcomes about the PSO-based technique concerning several iterations and the convergence speed for the addressed issue. Furthermore, for the 69-bus distribution system case, DA-, GWO-, and MFA-based optimization exhibited an enhanced convergence level. Additionally, GWO, DA, and MFO were assessed using statistical tests. The results showed that GWO, DA, and MFO had an acceptable root-mean-square error (RMSE).

Reference [66] introduced a novel binary multi verse optimization algorithm. In the article, the authors compared the new algorithm to some other binary optimization algorithms, including binary DA. Binary DA ranked as the second-best algorithm, among others, this was because of the excellent stability between DA's exploitation and exploration phases. Furthermore, the sudden changes in the variables cause a quick convergence.

Reference [67] compared DA with the Harris hawks optimization algorithm or so-called HHS. The algorithms utilized to enhance the multi-layer perceptron's performance, which was used to analyze the stability of two-layered soil. The work compared the accuracy and computational time of the algorithms. Mean absolute error (MAE), the area under the receiving operating characteristic curve (AUC), and Mean square error (MSE) utilized for evaluating the predictive models' performance. In general, both algorithms helped to improve the applicability accuracy of the MLP. However, the DA reached the lowest error within 500 iterations, whereas the HHS needed 1000 iterations for the same task. Hence, the DA provided a better convergence comparing to the HHS for the problem mentioned above.

VII. RESULTS AND EVALUATIONS

The results of the applications in the literature showed that the dragonfly algorithm is suitable to optimize various applications in the engineering field. The provided outcomes proved the superiority of the algorithm. Here, to demonstrate the ability of the DA, it is evaluated against the traditional benchmark functions. Moreover, to further evaluate the algorithm, it was examined on the IEEE Congress of Evolutionary Computation Benchmark Test Functions or CEC-2019, also known as "the 100-digit challenge" [68]. The test functions of Wilcoxon rank-sum are utilized for showing the importance of the outcomes statistically.

To examine the ability of the algorithm and its performance, three groups of traditional benchmark functions utilized with various characteristics in the original work. The groups of the traditional test functions consist of three groups, which are unimodal (F1-F7), multi-modal (F8-F13), and composite test functions (F14-F23). The test functions' results are in Tables 2 and 3 are from [14]. However, we assessed the FA on all the test functions (F1-F23).

It can be seen in Table 2, the results of DA on unimodal test functions outperformed the PSO. The results of the unimodal test functions are evident that the DA has outstanding exploitation and speed of convergence compared to the PSO. Alternatively, for the unimodal benchmark functions, FA provided superior results compared to the PSO and DA. Nevertheless, the results from the references mentioned above are another evidence for the speed of convergence of the DA. Reference [41] utilized the DA and FA for optimizing the same problem. The results showed the high convergence of the DA. Furthermore, the *p*-value in Table 3 for this group of benchmark functions is smaller than 0.05, this proved the statistical importance of the outcomes. Moreover, in the references above, DA proved its high convergence speed.

Table 2: Classical Benchmark Results of *DA, PSO, and FA*

F	Measurements.	DA	PSO	FA
F_1	Mean	**2.85E-18**	4.2E-18	1.72e-10
	Std.	**7.16E-18**	1.31E-18	9.43e-10
F_2	Mean	**1.49E-05**	0.003154	6.01e-07
	Std.	**3.76E-05**	0.009811	3.29e-06
F_3	Mean	**1.29E-06**	0.001891	**1.58e-10**
	Std.	**2.1E-06**	0.003311	**8.66e-10**
F_4	Mean	**0.000988**	0.001748	5.913-03
	Std.	**0.002776**	0.002515	0.029813
F_5	Mean	**7.600558**	63.45331	**2.383765**
	Std.	**6.786473**	80.12726	**1.350716**
F_6	Mean	4.17E-16	**4.36E-17**	1.9e-10
	Std.	1.32E-15	**1.38E-16**	1.04e-09
F_7	Mean	0.010293	**0.005973**	**1.57e-04**
	Std.	0.004691	**0.003583**	**1.01e-04**
F_8	Mean	-2857.58	**-7.1E+11**	-3566.452419
	Std.	383.6466	**1.2E+12**	239.113661
F_9	Mean	16.01883	**10.44724**	7.462188
	Std.	9.479113	**7.879807**	4.41686
F_{10}	Mean	**0.23103**	0.280137	8.47e-07
	Std.	**0.487053**	0.601817	4.64e-06
F_{11}	Mean	0.193354	**0.083463**	0.053309

	Std.	0.073495	**0.035067**	0.053615
F_{12}	Mean	0.031101	**8.57E-11**	1.92e-12
	Std.	0.098349	**2.71E-10**	1.05e-11
F_{13}	Mean	0.002197	0.002197	8.21e-12
	Std.	0.004633	0.004633	4.5e-11
F_{14}	Mean	**103.742**	150	**0.99800**
	Std.	**91.24364**	135.4006	**1.700065e-16**
F_{15}	Mean	193.0171	**188.1951**	**3.77e-04**
	Std.	80.6332	**157.2834**	**1.853-04**
F_{16}	Mean	458.2962	**263.0948**	**-1.031628**
	Std.	165.3724	**187.1352**	**1.06e-15**
F_{17}	Mean	596.6629	**466.5429**	3.0
	Std.	171.0631	**180.9493**	6.05e-15
F_{18}	Mean	229.9515	**136.1759**	**-3.862782**
	Std.	184.6095	**160.0187**	**2.79e-15**
F_{19}	Mean	**679.588**	741.6341	-3.259273
	Std.	**199.4014**	206.7296	0.059789
F_{20}	Mean	**-3.32199**	-3.27047	-9.316829
	Std.	**-3.38E-06**	0.059923	2.21393
F_{21}	Mean	-10.1532	**-7.3874**	-10.147907
	Std.	**6.60E-15**	3.11458	1.396876
F_{22}	Mean	-10.4029	**-8.5305**	-9.398946
	Std.	**1.51E-06**	3.038572	1.99413
F_{23}	Mean	-10.5364	**-9.1328**	-10.2809
	Std.	**2.97E-07**	2.640148	1.39948

Table 3: The Wilcoxon Rank-Sum Test Overall Runs For the Classical Benchmark Functions

F	**DA**	**PSO**
F1	N/A	0.045155
F2	N/A	0.121225
F3	N/A	0.003611
F4	N/A	0.307489
F5	N/A	0.10411
F6	0.344704	N/A
F7	0.021134	N/A
F8	0.000183	N/A
F9	0.364166	N/A
F10	N/A	0.472676
F11	0.001008	N/A
F12	0.140465	N/A
F13	N/A	0.79126
F14	N/A	0.909654
F15	0.025748	0.241322
F16	0.01133	N/A
F17	0.088973	N/A
F18	0.273036	0.791337
F19	N/A	0.472676
F20	0.938062	0.938062
F21	N/A	N/A

F22	0.256157	0.256157
F23	0.59754	0.59754

The results of the test functions of the multi-modal showed the great exploration level of the dragonfly algorithm that aids in discovering the exploration space. Generally, particle swarm optimization for the multi-modal exhibited superior results. Furthermore, as shown in Table 3, these results again are statistically noteworthy as the furthermost of the *p*-values are a smaller amount than 0.05.

The FA outperformed the DA and PSO for the composite test function. Similar to the unimodal test functions, the outcomes of the DA and PSO were competitive. However, in some cases, PSO provided better results, which proved that the FA has a superior balance between the phases of exploration and exploitation compared to DA and PSO PSO. The intention of this is that the DA's level of exploitation is smaller than the level of exploration. Moreover, the majority of the statistical results for this group of benchmark functions are significant and less than 0.05, as shown in Table 3.

Furthermore, in the original work, the CEC benchmark functions were not used to evaluate the DA. Hence, in this paper, the test functions of CEC-2019 are used to assess the DA further. This group of test functions utilizes an annual optimization competition. Professor Suganthan and his colleges improved these benchmark functions to optimize single objective problems [82→68]. All the CEC-2019 benchmark functions are scalable. However, only function number 4 to function number 10 are shifted and rotated. Whereas, the functions (1 to 3) are not. The functions (1 to 3) have different dimensions. However, functions (4 to 10) set as minimization problems with 10 dimensions. See Table 4 for more details about the functions.

Table 4: CEC-2019 Benchmark Functions- 100-Digit Challenge [68]

No.	Functions	Dimension	Range	f_{min}
1	STORN'S CHEBYSHEV POLYNOMIAL FITTING PROBLEM	9	[-8192, 8192]	1
2	INVERSE HILBERT MATRIX PROBLEM	16	[-16384, 16384]	1
3	LENNARD-JONES MINIMUM ENERGY CLUSTER	18	[-4, 4]	1
4	RASTRIGIN'S FUNCTION	10	[-100, 100]	1
5	GRIENWANK'S FUNCTION	10	[-100, 100]	1
6	WEIERSRASS FUNCTION	10	[-100, 100]	1
7	MODIFIED SCHWEFEL'S FUNCTION	10	[-100, 100]	1
8	EXPANDED SCHAFFER'S F6 FUNCTION	10	[-100, 100]	1
9	HAPPY CAT FUNCTION	10	[-100, 100]	1
10	ACKLEY FUNCTION	10	[-100, 100]	1

In the original paper, the DA compared to the PSO; hence, for this group of test functions, DA again will be compared to the PSO. The default parameter settings were not changed during the optimization. For this evaluation, the authors used 100 iterations and 30 agents. As shown in Table 5, DA outperformed the PSO in three CEC-2019 benchmark functions (1, 2, and 3), and the results were competitive in two functions (3 and 10).

VIII. DISCUSSION AND FUTURE WORKS

DA is modest and can be easily applied. For exploring the search space, allocate little weight of cohesion and great weight of alignment to individuals. Contrarily, for exploiting the exploration space, assign individuals to high cohesion and low alignment weights. Another way for balancing exploitation and exploration is adaptively adjusting the swarming weights, such as *s, a, e, c, w, and f* throughout the process of optimization. To make a transition concerning the exploration and exploitation, neighborhood

radii enlarged proportionally to the number of iterations could be applied. It usually provides reasonable results for small to medium-scale problems. However, for large-scale optimization problems, more affords are required, and it causes an increase in convergence time and a reduction in performance, which may cause falling into local optima.

Table 5: The IEEE CEC-2019 Benchmark Results for DA, and PSO

Function No.	Measurements	DA	PSO
1	Mean	**46835.63679**	1.47127E+12
	Std.	**8992.755502**	1.32362E+12
2	Mean	**18.31681239**	15183.91348
	Std.	**0.041929318**	3729.553229
3	Mean	12.70240422	12.70240422
	Std.	1.50E-12	9.03E-15
4	Mean	103.3295366	**16.80077558**
	Std.	20.00405422	**8.199076134**
5	Mean	1.177303105	**1.138264955**
	Std.	0.057569859	**0.089389848**
6	Mean	**5.646572343**	9.305312443
	Std.	**4.27E-08**	1.69E+00
7	Mean	898.5188217	**160.6863065**
	Std.	4.023921424	104.2035197
8	Mean	6.210996106	**5.224137165**
	Std.	0.001657324	0.786760649
9	Mean	2.601134198	**2.373279266**
	Std.	0.233292964	**0.018437068**
10	Mean	20.0506995	20.28063455
	Std.	0.070920925	0.128530895

With growing the complexity of optimizing real-world problems, computing demands are hard to be satisfied with the single version of optimization algorithms. One obstacle that may occur during using the DA is that updating position in this algorithm is not so much correlated with the algorithm's population centroid in the preceding generations. Consequently, the produced solutions have low accuracy, and premature convergence to local optima may occur. Additionally, it may cause it difficult to find the global optimal solution. Furthermore, as mentioned earlier, distraction, cohesion, alignment, and separation in the direction of enemy sources with desirability in the direction of the sources of food mainly determine the exploration and exploitation of the DA. This searching technique maximizes solution diversity and makes the capability of exploration of the DA stronger to some extent. Nonetheless, the performance reduces with a large number of exploitation and exploration operators because they enlarge the convergence time, which causes trapping into local optima.

Similar to other metaheuristic algorithms, DA has several strong points, as well as some weak points. It owns powerful optimization capability. The DA has few parameters for adjusting. Most of the time, it can keep a reasonable convergence rate to the global optima. DA is one of the new algorithms. However, as discussed in the literature, it has been utilized for optimizing an enormous number of applications. The straightforwardness of DA is one of the main reasons for its contributions to various applications. Also, choosing the individuals from the record, the worst hypersphere avoids the DA from discovering the non-promised zones. Another advantage is that DA has few parameters for tuning. Similarly, over other optimization algorithms, the algorithm converges earlier, more stable, and more straightforwardly can be hybridized with diverse algorithms.

Alternatively, for complex optimization problems, as examined in [27], one of the restrictions of the DA is that it easily traps into local optima, and it has a slow convergence speed. Internal memory does not exist in the DA, which is a reason for early arrival at local optimums. This overcame in [69] through emerging a new memory-based hybrid dragonfly or so-called MHDA. Additionally, as presented earlier, DA uses Levy flight as a search process when the neighborhood does not exist. Nevertheless, the giant steps of the Levy flight mechanism caused an interruption. The original work used a step control mechanism to prevent overflowing. However, this distorts the characteristics of the swarm, also, it is a reason for falling into local optima. Hence, utilizing other searching techniques instead of the Levy flight and compared the results of the various methods is highly recommended. Moreover, using an adaptive step instead of the original stochastic step will help in harmonizing phases of exploration and exploitation and enhancing the DA performance. The position updating technique is another way to prevent trapping into local optima. Using the population's centroid technique, as discussed in [70] can reduce the probability of locating into local optima.

Furthermore, after assessing the algorithm in the above section, it was noted that the ability of the DA for balancing between the phases of exploration and exploitation is low; this was because the algorithm has a great exploration level. This great level of the search in the initial phases of the course of optimization is decent, though, in the last iterations of the algorithm, it ought to be diminished, and the exploitation level ought to be improved. For binary dragonfly algorithms, for example, using time-dependent transfer function can increase the balance between both phases of the DA; exploitation, and exploration. Hence, at the beginning of the optimization, the exploration level is great. The exploration level gradually decreases during the process and the exploitation level increases. The mentioned technique will provide a better performance and it prevents trapping into local optima. Tuning parameters automatically improves the performance of different algorithms. Moreover, it improves the stability between the two

phases and the variety of the population [71]. On the other hand, the outcomes of the traditional benchmark function of the unimodal, and the produced outcomes of the majority of the literature works displayed that the dragonfly has a good convergence. The greater convergence of the algorithm makes it outperform most of the mentioned algorithms in the previous works in dealing with small to medium problem sizes.

Generally, the results from the previous section and the applications in the literature proved that the DA has a significant level of exploration and exploitation. The reason for this is the DA's static swarming behavior, which enlarges the exploration's level, and increases the probability of trapping into local optima for simple problems. Additionally, enlarging the number of iterations enlarges the exploitation degree, and enhances the accuracy of the global optimum solution.

However, hybridizing the algorithm with other techniques will give power to the algorithm to overcome the bottlenecks. As discussed, some hybrid versions of DA proposed to overcome the weakness of this algorithm. For example, MHDA was examined for overcoming the shortage that may cause premature convergence to local optima. Moreover, reference [72] utilized Gauss chaotic map to adjust variables. The outcomes exhibited that the hybridized algorithm concerning stability quality, the speed of convergence, classification performance, and the number of selected features provided better results. Although the DA and the hybridized DAs provided some good results for several problems of complex optimization, yet, more or less disadvantages found. In DA, high exploration and exploitation acquire through desirability on the way to food and diversion on the road to enemies. The correlation of updating the position in the DA with the population centroid from the preceding generation is not high. Thus, it may solve with that traps into local optima, low accuracy, and struggles to find the global optima. Hence, finding a new technique for updating the position of individuals is highly recommended. Another research area that will improve the algorithm is finding suitable stability concerning phases of exploration and exploitation. Proper stability concerning exploration and exploitation will circumvent DA from falling into the local optima. Besides, merging new searching methods with the DA is highly recommended to researchers. Moreover, tuning parameters dynamically during the practice of optimization will have significant guidance on enhancing the exploitation and exploration balance of the algorithm.

IX. CONCLUSIONS

This paper reviewed one of the new metaheuristic algorithms. The various types of the algorithm, including the merging versions with other techniques, were discussed. Besides, most of the optimization problems in engineering and physics that used DA were discussed. From the reviewed works, the authors discovered that DA is one of the practical techniques in the area. The simplicity of the algorithm was one of the reasons that encouraged the researchers to use the algorithm to optimize the problems in hand. Moreover, the accuracy and convergence speed of the algorithm are other reasons. For instance, in general, for small to medium problems, the algorithm provided good results. However, similar to other algorithms, for some problems (especially complex problems) DA cannot produce reasonable results. The exploration of the algorithm is high, which may cause trapping into local optima, mainly for the complex problems. Moreover, the produced results from the test functions (F20-F23) proved that the results of the DA and PSO are competitive. Besides, the results of the CEC-2019 test function also showed that the DA is comparative with the PSO, and FA showed better results than both DA and PSO. Finally, reviewing the DA and its applications proved that we could utilize the mentioned algorithm successfully to optimize almost all the problems in the real world.

As an extension of this work, the authors are willing to find a technique for providing a decent balance about the exploration and exploitation phases of DA. Likewise, the representation of the DA can be assessed and evaluated against other well-known and competitive algorithms, such as Donkey and Smuggler Optimisation Algorithm [12], WOA-BAT Optimisation Algorithm [73], Fitness Dependent Optimiser [74], Modified Grey Wolf Optimiser [75], etc.

Funding:

This work did not take any grant from funding agencies in the public, commercial, or not-for-profit sectors.

REFERENCES

[1] Dorigo M., (1992) “Optimization, Learning and Natural Algorithms”, PhD thesis [in Italian], Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy.

[2] Bonabeau E, Dorigo M, Theraulaz G. “Swarm Intelligence: From Natural to Artificial Systems”. Journal of Artificial Societies and Social Simulation. 1999;4: 320.

[3] Ab Wahab, M., Nefti-Meziani, S. and Atyabi, A., 2015. A Comprehensive Review of Swarm Optimization Algorithms. *PLOS ONE*, [online] 10(5), p.e0122827. Available at: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0122827> [Accessed 14 March 2020].

[4] Zhang, Y., Wang, S. and Ji, G. (2015). *A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications*. [online] Available at: https://www.hindawi.com/journals/mpe/2015/931256/ [Accessed 4 Feb. 2018].

[5] Ducatelle, F., Di Caro, G. and Gambardella, L., 2010. Principles and applications of swarm intelligence for adaptive routing in telecommunications networks. *Swarm Intelligence*, [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. (1995). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*. [online] Available at: https://ieeexplore.ieee.org/document/488968 [Accessed 7 Feb. 2018].

[7] He, S., Wu, Q. and Saunders, J., 2009. Group Search Optimizer: An Optimization Algorithm Inspired by Animal Searching Behavior. *IEEE Transactions on Evolutionary Computation*, [online] 13(5), pp.973-990. Available at: <https://dl.acm.org/doi/10.1109/TEVC.2009.2011992> [Accessed 19 March 2020].

[8] Gandomi, A., Yang, X. and Alavi, A., 2011. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with Computers*, [online] 29(1), pp.17-35. Available at: <https://link.springer.com/article/10.1007/s00366-011-0241-y> [Accessed 19 March 2020].

[9] Mirjalili, S., Mirjalili, S. and Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, [online] 69, pp.46-61. Available at: https://www.sciencedirect.com/science/article/pii/S0965997813001853 [Accessed 3 Jan. 2018].

[10] Mirjalili, S. (2015). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications*, [online] 27(4), pp.1053-1073. Available at: https://link.springer.com/article/10.1007/s00521-015-1920-1 [Accessed 2 Jan. 2018].

[11] Yang, X., 2009. Harmony Search as a Metaheuristic Algorithm. *Music-Inspired Harmony Search Algorithm*, [online] pp.1-14. Available at: <https://link.springer.com/chapter/10.1007/978-3-642-00185-7_1> [Accessed 10 March 2020].

[12] Shamsaldin, A., Rashid, T., Al-Rashid Agha, R., Al-Salihi, N. and Mohammadi, M. (2019). Donkey and smuggler optimization algorithm: A collaborative working approach to path finding. *Journal of Computational Design and Engineering*, [online] 6(4), pp.562-583. Available at: https://www.sciencedirect.com/science/article/pii/S2288430018303178 [Accessed 1 May 2019].

[13] Yazdani, M. and Jolai, F. (2016). Lion Optimization Algorithm (LOA): A nature-inspired metaheuristic algorithm. *Journal of Computational Design and Engineering*, [online] 3(1), pp.24-36. Available at: https://www.sciencedirect.com/science/article/pii/S2288430015000524 [Accessed 8 Mar. 2019].

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

[15] Yang, X. (2014). *Cuckoo search and firefly algorithm: Theory and Applications*. Cham: Springer.

[16] Reynolds, C. (1987). Flocks, herds and schools: A distributed behavioral model. *Proceedings of the 14th annual conference on Computer graphics and interactive techniques - SIGGRAPH '87*, [online] 21(4), pp.25-34. Available at: https://dl.acm.org/citation.cfm?id=37406 [Accessed 14 Feb. 2018].

[17] Acı, Ç. and Gülcan, H. (2019). A Modified Dragonfly Optimization Algorithm for Single- and Multiobjective Problems Using Brownian Motion. *Computational Intelligence and Neuroscience*, [online] 2019, pp.1-17. Available at: https://www.hindawi.com/journals/cin/2019/6871298/ [Accessed 21 Sep. 2019].

[18] Mirjalili, S. and Lewis, A. (2013). S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization. *Swarm and Evolutionary Computation*, [online] 9, pp.1-14. Available at: https://www.sciencedirect.com/science/article/abs/pii/S2210650212000648 [Accessed 14 Feb. 2018].

[19] Mafarja, M., Aljarah, I., Heidari, A., Faris, H., Fournier-Viger, P., Li, X. and Mirjalili, S. (2018). Binary dragonfly optimization for feature selection using time-varying transfer functions. *Knowledge-Based Systems*, [online] 161, pp.185-204. Available at: https://www.sciencedirect.com/science/article/abs/pii/S095070511830399X?via%3Dihub [Accessed 21 Sep. 2019].

[20] Mirjalili, S. and Lewis, A. (2015). Novel performance metrics for robust multi-objective optimization algorithms. *Swarm and Evolutionary Computation*, [online] 21, pp.1-23. Available at: https://www.sciencedirect.com/science/article/abs/pii/S2210650214000777 [Accessed 22 Feb. 2018].

[21] Coello Coello, C. (2009). Evolutionary multi-objective optimization: some current research trends and topics that remain to be explored. *Frontiers of Computer Science in China*, [online] 3(1), pp.18-30. Available at: https://link.springer.com/article/10.1007/s11704-009-0005-7 [Accessed 22 Feb. 2018].

[22] Coello Coello, C. and Lechuga, M. (2002). MOPSO: a proposal for multiple objective particle swarm optimization. *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*. [online] Available at: https://ieeexplore.ieee.org/document/1004388 [Accessed 22 Feb. 2018].

[23] Coello, C., Pulido, G. and Lechuga, M. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, [online] 8(3), pp.256-279. Available at: https://ieeexplore.ieee.org/document/1304847 [Accessed 9 Mar. 2018].

[24] Li, J., Lu, J., Yao, L., Cheng, L. and Qin, H. (2019). Wind-Solar-Hydro power optimal scheduling model based on multi-objective dragonfly algorithm. *Energy Procedia*, [online] 158, pp.6217-6224. Available at: https://www.sciencedirect.com/science/article/pii/S1876610219304990 [Accessed 21 Sep. 2019].

[25] Deb, K. and Jain, H. (2014). An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation*, [online] 18(4), pp.577-601. Available at: https://ieeexplore.ieee.org/document/6600851 [Accessed 25 Nov. 2019].

[26] K.S., S. and Murugan, S. (2017). Memory based Hybrid Dragonfly Algorithm for numerical optimization problems. *Expert Systems with Applications*, [online] 83, pp.63-78. Available at: https://www.sciencedirect.com/science/article/pii/S0957417417302762 [Accessed 9 Mar. 2018].

[27] Xu, J. and Yan, F. (2018). Hybrid Nelder–Mead Algorithm and Dragonfly Algorithm for Function Optimization and the Training of a Multilayer Perceptron. *Arabian Journal for Science and Engineering*. [online] Available at: https://link.springer.com/article/10.1007%2Fs13369-018-3536-0 [Accessed 2 Mar. 2019].

[28] Ghanem, W. and Jantan, A. (2018). A Cognitively Inspired Hybridization of Artificial Bee Colony and Dragonfly Algorithms for Training Multi-layer Perceptrons. *Cognitive Computation*, [online] 10(6), pp.1096-1134. Available at: https://link.springer.com/article/10.1007/s12559-018-9588-3 [Accessed 12 Mar. 2019].

[29] Yuan, Y., Lv, L., Wang, X. and Song, X. (2019). Optimization of a frame structure using the Coulomb force search strategy-based dragonfly algorithm. *Engineering Optimization*, [online] pp.1-17. Available at: https://www.tandfonline.com/doi/full/10.1080/0305215X.2019.1618290 [Accessed 21 Sep. 2019].

[30] K., T. and Aravindhababu, P. (2017). Dragonfly Optimization based Reconfiguration for Voltage Profile Enhancement in Distribution Systems. *International Journal of Computer Applications*, [online] 158(3), pp.1-4. Available at: https://www.semanticscholar.org/paper/Dragonfly-Optimization-based-Reconfiguration-for-in-Abhiraj/6970179fb3b97a55dc881e8aa1c42e4dac44dc5d [Accessed 11 Mar. 2018].

[31] Andervazh, M., Haghifam, M. and Olamaei, J. (2013). Adaptive multi-objective distribution network reconfiguration using multi-objective discrete particles swarm optimisation algorithm and graph theory. *IET Generation, Transmission & Distribution*, [online] 7(12), pp.1367-1382. Available at: https://ieeexplore.ieee.org/document/6674159 [Accessed 11 Mar. 2018].

[32] Gupta, N., Swarnkar, A. and Niazi, K. (2014). Distribution network reconfiguration for power quality and reliability improvement using Genetic Algorithms. *International Journal of Electrical Power & Energy Systems*, [online] 54, pp.664-671. Available at: https://www.sciencedirect.com/science/article/pii/S0142061513003578 [Accessed 11 Mar. 2018].

[33] Aruul, S. and Santhi, R. (2016). New Reconfiguration Method for Improving Voltage Profile of Distribution Networks. *International Journal of Computer Applications*, [online] 135(7), pp.25-29. Available at: https://www.semanticscholar.org/paper/New-Reconfiguration-Method-for-Improving-Voltage-of-Santhi-Jabr/382de46c554feb5d53a550579f7aa48af9ddd6ce [Accessed 11 Mar. 2018].

[34] Algabalawy, M., Mekhamer, S. and Abdelaziz, A. (2017). Optimal Design of a New Configuration of the Distributed Generation Units using Grey Wolf and Dragonfly Optimizers. *MASK International Journal of Science and Technology*. [online] 2(1). Available at: https://www.academia.edu/31230265/Optimal_Design_of_a_New_Configuration_of_the_Distributed_Generation_Units_using_Grey_Wolf_and_Dragonfly_Optimizers. [Accessed 11 Mar. 2018].

[35] Jafari, M. and Bayati Chaleshtari, M. (2017). Using dragonfly algorithm for optimization of orthotropic infinite plates with a quasi-triangular cut-out. *European Journal of Mechanics - A/Solids*, [online] 66, pp.1-14. Available at: https://www.sciencedirect.com/science/article/pii/S0997753817304370 [Accessed 12 Feb. 2019].

[36] Bloomfield, M., Herencia, J. and Weaver, P. (2010). Analysis and benchmarking of meta-heuristic techniques for lay-up optimization. *Computers & Structures*, [online] 88(5-6), pp.272-282. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0045794909002648 [Accessed 12 Feb. 2019].

[37] Babayigit, B. (2017). Synthesis of concentric circular antenna arrays using dragonfly algorithm. *International Journal of Electronics*, [online] 105(5), pp.784-793. Available at: https://doi.org/10.1080/00207217.2017.1407964 [Accessed 13 Feb. 2019].

[38] Bloomfield, M., Herencia, J. and Weaver, P. (2010). Analysis and benchmarking of meta-heuristic techniques for lay-up optimization. *Computers & Structures*, [online] 88(5-6), pp.272-282. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0045794909002648 [Accessed 12 Feb. 2019].

[39] Dib, N. (2017). Design of planar concentric circular antenna arrays with reduced side lobe level using symbiotic organisms search. *Neural Computing and Applications*, [online] 30(12), pp.3859-3868. Available at: https://link.springer.com/article/10.1007/s00521-017-2971-2 [Accessed 13 Feb. 2019].

[40] Ram, G., Mandal, D., Kar, R. and Ghoshal, S. (2015). Circular and Concentric Circular Antenna Array Synthesis Using Cat Swarm Optimization. *IETE Technical Review*, [online] 32(3), pp.204-217. Available at: https://www.tandfonline.com/doi/abs/10.1080/02564602.2014.1002543?journalCode=titr20 [Accessed 14 Feb. 2019].

[41] Ram, G., Mandal, D., Kar, R. and Prasad Ghoshal, S. (2015). Opposition-based gravitational search algorithm for synthesis circular and concentric circular antenna arrays. *scientia Iranica*, [online] 22(6). Available at: http://scientiairanica.sharif.edu/article_3796_e57ae2fe002d2736cf943f020f5ac2fc.pdf [Accessed 14 Feb. 2019].

[42] Mandal, D., Ghoshal, S. and Bhattacharjee, A. (2010). Design of Concentric Circular Antenna Array with Central Element Feeding Using Particle Swarm Optimization with Constriction Factor and Inertia Weight Approach and Evolutionary Programing Technique. *Journal of Infrared, Millimeter, and Terahertz Waves*, [online] 31(6), pp.667-680. Available at: https://link.springer.com/article/10.1007/s10762-010-9629-9 [Accessed 14 Feb. 2019].

[43] Sharaqa, A. and Dib, N. (2013). Circular antenna array synthesis using firefly algorithm. *International Journal of RF and Microwave Computer-Aided Engineering*, [online] 24(2), pp.139-146. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1002/mmce.20721 [Accessed 15 Feb. 2019].

[44] Simhadri, K., Mohanty, B. and Mohan Rao, U. (2018). Optimized 2DOF PID for AGC of Multi-area Power System Using Dragonfly Algorithm. *Advances in Intelligent Systems and Computing*, [online] pp.11-22. Available at: https://link.springer.com/chapter/10.1007/978-981-13-1819-1_2 [Accessed 20 Feb. 2019].

[45] Khalilpourazari, S. and Khalilpourazary, S. (2018). Optimization of time, cost and surface roughness in grinding process using a robust multi-objective dragonfly algorithm. *Neural Computing and Applications*, [online] pp.1-12. Available at: https://link.springer.com/article/10.1007/s00521-018-3872-8 [Accessed 19 Feb. 2019].

[46] Gholami, M. and Azizi, M. (2014). Constrained grinding optimization for time, cost, and surface roughness using NSGA-II. *The International Journal of Advanced Manufacturing Technology*, [online] 73(5-8), pp.981-988. Available at: https://link.springer.com/article/10.1007/s00170-014-5884-6 [Accessed 20 Feb. 2019].

[47] El-Hay, E., El-Hameed, M. and El-Fergany, A. (2018). Improved performance of PEM fuel cells stack feeding switched reluctance motor using multi-objective dragonfly optimizer. *Neural Computing and Applications*. [online] Available at: https://link.springer.com/article/10.1007%2Fs00521-018-3524-z [Accessed 21 Sep. 2019].

[48] Amroune, M., Bouktir, T. and Musirin, I. (2018). Power System Voltage Stability Assessment Using a Hybrid Approach Combining Dragonfly Optimization Algorithm and Support Vector Regression. *Arabian Journal for Science and Engineering*, [online] 43(6), pp.3023-3036. Available at: https://link.springer.com/article/10.1007%2Fs13369-017-3046-5 [Accessed 21 Sep. 2019].

[49] Guha, K., Laskar, N., Gogoi, H., Borah, A., Baishnab, K. and Baishya, S. (2017). Novel analytical model for optimizing the pull-in voltage in a flexured MEMS switch incorporating beam perforation effect. *Solid-State Electronics*, [online] 137, pp.85-94. Available at: https://www.sciencedirect.com/science/article/pii/S0038110117300229 [Accessed 14 Apr. 2018].

[50] Vanishree, J. and Ramesh, V. (2017). Optimization of size and cost of static var compensator using dragonfly algorithm for voltage profile improvement in power transmission systems. *International Journal of Renewable Energy Research (IJRER)*, [online] 8(1), pp.56-66. Available at: https://ijrer.org/ijrer/index.php/ijrer/article/view/6933 [Accessed 20 Feb. 2019].

[51] Kouba, N., Menaa, M., Hasni, M. and Boudour, M. (2018). A Novel Optimal Combined Fuzzy PID Controller Employing Dragonfly Algorithm for Solving Automatic Generation Control Problem. *Electric Power Components and Systems*, [online] pp.1-17. Available at: https://www.tandfonline.com/doi/abs/10.1080/15325008.2018.1533604?af=R&journalCode=uemp20 [Accessed 23 Feb. 2019].

[52] Jafari, M. and Bayati Chaleshtari, M. (2017). Using dragonfly algorithm for optimization of orthotropic infinite plates with a quasi-triangular cut-out. *European Journal of Mechanics - A/Solids*, [online] 66, pp.1-14. Available at: https://www.sciencedirect.com/science/article/pii/S0997753817304370 [Accessed 2 Mar. 2018].

[53] Mathworks.com. (2018). "Genetic Algorithm". [online] Available at: https://www.mathworks.com/discovery/genetic-algorithm.html [Accessed 27 May 2018].

[54] Bhesdadiya, R., Pandya, M., Trivedi, I., Jangir, N., Jangir, P. and Kumar, A. (2016). Price penalty factors based approach for combined economic emission dispatch problem solution using Dragonfly Algorithm. *2016 International Conference on Energy Efficient Technologies for Sustainability (ICEETS)*. [online] Available at: https://ieeexplore.ieee.org/document/7583794 [Accessed 1 Mar. 2018].

[55] Guo, S., Dooner, M., Wang, J., Xu, H. and Lu, G. (2017). Adaptive engine optimisation using NSGA-II and MODA based on a sub-structured artificial neural network. *2017 23rd International Conference on Automation and Computing (ICAC)*. [online] Available at: https://ieeexplore.ieee.org/document/8082008 [Accessed 15 Feb. 2019].

[56] Guha, D., Roy, P. and Banerjee, S. (2018). Optimal tuning of 3 degree-of-freedom proportional-integral-derivative controller for hybrid distributed power system using dragonfly algorithm. *Computers & Electrical Engineering*, [online] 72, pp.137-153. Available at: https://www.sciencedirect.com/science/article/pii/S0045790618304609 [Accessed 19 Feb. 2019].

[57] Pathania, A., Mehta, S. and Rza, C. (2016). Economic load dispatch of wind thermal integrated system using dragonfly algorithm. *2016 7th India International Conference on Power Electronics (IICPE)*. [online] Available at: https://ieeexplore.ieee.org/document/8079422 [Accessed 14 Apr. 2018].

[58] Zhang, Y., Yao, F., Iu, H., Fernando, T. and Wong, K. (2013). Sequential quadratic programming particle swarm optimization for wind power system operations considering emissions. *Journal of Modern Power Systems and Clean Energy*, [online] 1(3), pp.231-240. Available at: https://link.springer.com/article/10.1007/s40565-013-0030-2 [Accessed 14 Apr. 2018].

[59] Suresh, V. and Sreejith, S. (2016). Generation dispatch of combined solar thermal systems using dragonfly algorithm. *Computing*, [online] 99(1), pp.59-80. Available at: https://link.springer.com/article/10.1007/s00607-016-0514-9 [Accessed 16 May 2018].

[60] Mafarja, M., Eleyan, D., Jaber, I., Hammouri, A. and Mirjalili, S. (2017). Binary Dragonfly Algorithm for Feature Selection. *2017 International Conference on New Trends in Computing Sciences (ICTCS)*. [online] Available at: https://ieeexplore.ieee.org/document/8250257 [Accessed 10 May 2018].

[61] Emary, E., Zawbaa, H. and Hassanien, A. (2016). Binary ant lion approaches for feature selection. *Neurocomputing*, [online] 213, pp.54-65. Available at: https://www.sciencedirect.com/science/article/pii/S0925231216307263 [Accessed 2 Jul. 2018].

[62] Hamdy, M., Nguyen, A. and Hensen, J. (2016). A performance comparison of multi-objective optimization algorithms for solving nearly-zero-energy-building design problems. *Energy and Buildings*, [online] 121, pp.57-71. Available at: https://www.sciencedirect.com/science/article/pii/S0378778816301724 [Accessed 1 Mar. 2018].

[63] Arulraj, R. and Kumarappan, N. (2018). Simultaneous Multiple DG and Capacitor Installation Using Dragonfly Algorithm for Loss Reduction and Loadability Improvement in Distribution System. *2018 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*. [online] Available at: https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8501352&filter=issueId%20EQ%20%228521560%22&pageNumber=2 [Accessed 19 Feb. 2019].

[64] Wongsinlatam, W. and Buchitchon, S. (2018). The Comparison between Dragonflies Algorithm and Fireflies Algorithm for Court Case Administration: A Mixed Integer Linear Programming. *Journal of Physics: Conference Series*, [online] 1061, p.012005. Available at: https://iopscience.iop.org/article/10.1088/1742-6596/1061/1/012005 [Accessed 23 Feb. 2019].

[65] Diab, A. and Rezk, H. (2018). Optimal Sizing and Placement of Capacitors in Radial Distribution Systems Based on Grey Wolf, Dragonfly and Moth–Flame Optimization Algorithms. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, [online] 43(1), pp.77-96. Available at: https://link.springer.com/article/10.1007/s40998-018-0071-7 [Accessed 24 Feb. 2019].

[66] Al-Madi, N., Faris, H. and Mirjalili, S. (2019). Binary multi-verse optimization algorithm for global optimization and discrete problems. *International Journal of Machine Learning and Cybernetics*. [online] Available at: https://www.springerprofessional.de/en/binary-multi-verse-optimization-algorithm-for-global-optimizatio/16442930 [Accessed 21 Sep. 2019].

[67] Moayedi, H., Abdullahi, M., Nguyen, H. and Rashid, A. (2019). Comparison of dragonfly algorithm and Harris hawks optimization evolutionary data mining techniques for the assessment of bearing capacity of footings over two-layer foundation soils. *Engineering with Computers*, [online] pp.1-11. Available at: https://link.springer.com/article/10.1007%2Fs00366-019-00834-w [Accessed 21 Sep. 2019].

[68] K. V. Price, N. H. Awad, M. Z. Ali, P. N. Suganthan. (2018). The 100-Digit Challenge: Problem Definitions and Evaluation Criteria for the 100-Digit Challenge Special Session and Competition on Single Objective Numerical Optimization. *Nanyang Technological University*, Singapore.

[69] Ma, H., Simon, D., Fei, M., Shu, X. and Chen, Z. (2014). Hybrid biogeography-based evolutionary algorithms. *Engineering Applications of Artificial Intelligence*, [online] 30, pp.213-224. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0952197614000189 [Accessed 5 Mar. 2018].

[70] Mirjalili, S. and Lewis, A. (2013). S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization. *Swarm and Evolutionary Computation*, [online] 9, pp.1-14. Available at: https://www.sciencedirect.com/science/article/abs/pii/S2210650212000648 [Accessed 2 Mar. 2019].

[71] Tejani, G., Savsani, V. and Patel, V. (2016). Adaptive symbiotic organisms search (SOS) algorithm for structural design optimization. *Journal of Computational Design and Engineering*, [online] 3(3), pp.226-249. Available at: https://www.scopus.com/record/display.uri?eid=2-s2.0-84991722784&origin=inward&txGid=94c67ebb12be33b0fb72098991bc1ffc [Accessed 24 Nov. 2019].

[72] Sayed, G., Tharwat, A. and Hassanien, A. (2019). Chaotic dragonfly algorithm: an improved metaheuristic algorithm for feature selection. *Applied Intelligence*, [online] 49(1), pp.188-205. Available at: https://link.springer.com/article/10.1007%2Fs10489-018-1261-8.

[73] Mohammed, H., Umar, S. and Rashid, T. (2019). A Systematic and Meta-Analysis Survey of Whale Optimization Algorithm. *Computational Intelligence and Neuroscience*, [online] 2019, pp.1-25. Available at: https://www.hindawi.com/journals/cin/2019/8718571/ [Accessed 14 Dec. 2019].

[74] Abdullah, J. and Ahmed, T. (2019). Fitness Dependent Optimizer: Inspired by the Bee Swarming Reproductive Process. *IEEE Access*, [online] 7, pp.43473-43486. Available at: https://ieeexplore.ieee.org/document/8672851 [Accessed 14 Dec. 2019].

[75] Rashid, T., Abbas, D. and Turel, Y. (2019). A multi hidden recurrent neural network with a modified grey wolf optimizer. *PLOS ONE*, [online] 14(3), p.e0213237. Available at: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0213237 [Accessed 14 Dec. 2019].