International Journal of Scientific Research in Science and Technology
Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijsrst.com)
doi : https://doi.org/10.32628/IJSRST218446

Improved Dragonfly Algorithm with Neighbourhood Structures

S. Rajalakshmi, S. Kanmani, S. Saraswathi
Department of Information Technology, Pondicherry Engineering College, Puducherry, India

Article Info
Volume 8, Issue 4
Page Number : 303-309

Publication Issue
July-August-2021

Article History
Accepted : 10 July 2021
Published : 20 July 2021

ABSTRACT

Dragonfly algorithm is a recently proposed optimization algorithm inspired on the static and dynamic swarming behaviour of dragonflies. Because of its simplicity and effectiveness, DA has received interest of specialists from various fields. Premature convergence and local optima is an issue in Dragonfly Algorithm. Improved Dragonfly Algorithm with Neighbourhood Structures (IDANS) is proposed to overcome this drawback. Dragonfly Algorithm with Neighborhood structures utilizes candidate solutions in an iterative and intuitive process to discover promising areas in a search space. IDANS is then initialized with best value of dragonfly algorithm to further explore the search space. In order to improve the efficiency of IDANS, Neighbourhood structures such as Euclidean, Manhattan and Chebyshev are chosen to implement these structures on IDANS to obtain best results. The proposed method avoids local optima to achieve global optimal solutions. The Efficiency of the IDANS is validated by testing on benchmark functions and classical engineering problem called Gear train design problem. A comparative performance analysis between IDANS and other powerful optimization algorithms have been carried out and the results shows that IDANS gives better performance than Dragonfly algorithm. Moreover it gives competitive results in terms of convergence and accuracy when compared with other algorithms in the literature.

Keywords - Dragonfly Algorithm, Neighbourhood Structures, Unimodal, Multimodal, Gear Train

I. INTRODUCTION

Nature consists of different life forms expressing variety of social behaviors for migrating and foraging to survive. Nature inspired algorithms have been derived based on the deep analysis of these social behaviors and applied on various real world problems. It is observed that several nature inspired metaheuristic algorithms exist such as Genetic algorithm (GA) [7], Particle Swarm Optimization algorithm (PSO) [8], Cuckoo Search algorithm (CS) [9], Bat Algorithm (BA) [11], Flower Pollination

Algorithm (FPA) [12], States of Matter Search algorithm (SMS) [13] for solving optimization problems.

The most commonly and efficiently used metaheuristic algorithms like Artificial Bee Colony Algorithm (ABC) [3], Ant Colony Optimization algorithm (ACO) [4] and Firefly algorithm (FA) [6] are inspired from the behavior of insects. Some other insect based algorithms are Ant Lion Optimizer algorithm (ALO) [5], Butterfly Optimization algorithm (BOA) [1], Dragonfly Algorithm

(DA) [2], Monarch Butterfly Optimization algorithm (MBO) [10] and Moth Flame Optimization algorithm (MFO) [14].

Artificial Bee Colony algorithm is inspired by the intellectual and food penetrating traits of Honeybee [3]. In ABC, bees are grouped as employee bee, onlooker bee and scout bee according to their functions. ABC has produced optimal solutions for several real world problems.

Ant colony optimization algorithm is inspired by the social information sharing behavior of the ants based on pheromone trails [4]. Strongest pheromone trails form the shortest route for food source. This concept of finding the shortest route is used in the algorithm to troubleshoot various real world issues.

Firefly optimization algorithm uses flashing pattern and flashing intensity behavior of fireflies [6]. The flashing lights used by fireflies are used to search suitable mates. This social behavior of the fireflies has been used in Firefly Algorithm to find optimal solution for the problems.

Ant lion optimization algorithm is inspired by the hunting characteristics of antlions [5]. Ants and Antlions make random walks which promote exploration in the algorithm. Based on antlions fitness value, it digs conical shaped pits and waits for ants to be trapped in it. The shrinking of traps results in exploitation of the search space. Because of these exploration and exploitation properties, ALO is suitable to solve unknown search area problems.

Butterfly optimization algorithm is derived from the food searching characteristics of butterflies [1]. Its smell is used to find food and mates. Based on fragrance values, information is shared between butterflies. This inspired behavior is introduced in BOA to obtain better solutions for the constrained problems.

Dragonfly metaheuristic Algorithm [2] is influenced by the hunting and migrating behaviors of dragonflies. This algorithm is used to solve single objective and Multiobjective optimization problems since it exhibits proper balance of exploitation and exploration.

This paper is an extension of the Dragonfly Algorithm done in Seyedali Mirjalili (2016) [2]. This work is tested on Unimodal and Multimodal benchmark test functions and Gear Train design problem and the results are compared with Improved Dragonfly algorithm and other algorithms in the literature.

II. Basic Dragonfly Algorithm

Dragonfly Algorithm is a nature inspired insect algorithm which is stimulated by the Global and local search behaviour of Dragonflies [2]. In nature dragonfly is a hunting insect which hunt small insects and fishes. Dragonflies are the search agents in Dragonfly algorithm to implement optimization.

In local search, dragonflies form small search area and fly within that area to hunt small preys. In Global search, huge number of dragonflies forms the swarm to travel in one route over long distances. To survive, all the dragonflies attracted in the direction of food and distracted away from the opponents. Based on these two characteristics, dragonflies' position is updated.

The Separation of Dragonflies is computed as:

$$SE_i = -\sum_{j=1}^{No} Y - Y_j \quad (1)$$

Where,

SE_i – separation

Y – individuals current position

Y_j – Neighbouring individual jth position

N –Total neighbouring individuals

Alignment of Dragonflies is computed as:

$$AL_i = \frac{\sum_{j=1}^{No} VE_j}{No} \quad (2)$$

Where,

AL_i – alignment

VE – jth neighbouring individual velocity

The cohesion of Dragonflies is computed as:

$$CO_i = \frac{\sum_{j=1}^{No} Y_j}{No} - Y \quad (3)$$

Where,

CO_i – cohesion

Y_j – Neighbouring individual jth position

Y – individuals current position

Attraction in the direction of food source is computed as:

$$FO_i = Y^+ - Y$$

(4)

Where,

FO_i – food

Y^+ – food position

Y – individuals current position

Enemy Distraction is computed as:

$$EN_i = Y^- + Y \quad (5)$$

EN_i – Enemy

Y^- – Enemies position

Y – individuals current position

Step (ΔY) and Position(Y) vectors are the two important vectors which are used to update dragonflies' position and its movements

$$\Delta Y_{k+1} = (sSE_i + aAL_i + cCO_i + fFO_i + eEN_i) + w\Delta Y_k \quad (6)$$

Where

k – current iteration

s – separation weight

a – alignment weight

c – cohesion weight

f – food element

e – enemy element

Position vectors are computed as follows:

$$\Delta Y_{t+1} = Y_t + \Delta Y_{t+1} \quad (7)$$

If there is no neighbouring results, dragonflies position are computed as:

$$Y_{t+1} = Y_t + \text{Lévy}(f) \times Y_t \quad (8)$$

$$\text{Lévy}(f) = 0.01 \times \frac{ra_1 \times \sigma}{|ra_2|^{\frac{1}{\beta}}} \quad (9)$$

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{1/\beta} \quad (10)$$

Where,

ra1, ra2 – random numbers

β – constant

III. Improved Dragonfly Algorithm with Neighbourhood Structures

Recently proposed nature inspired dragonfly algorithm is based on the static and dynamic intelligent behaviour of dragonflies.

Dragonfly population Initialization Y_i $(i = 1,2 \ldots n)$
step vectors Initialization ΔY_i $(i = 1,2 \ldots n)$
while the end criteria is not fulfilled
objective value calculation of all dragonflies f(x)
update the food position and enemy position
Update the weights of a, w, f, c and e
Calculate A, S, F, C and E using Eqn (1 to5)
Select random index and dimension
Change the neighbourhood
Evaluate y
Update the radius of neighbourhood
if a dragonfly has one or more than one neighbouring dragonflies
Update the Velocity using Eqn (6)
Update the position vector using Eqn (7)
else
Update the Position Vector using Eqn (8)
end if

if f(x)<f(y)

Move to new solution (y ← x)

end if

Based on the boundaries correct the new position

end while

Fig. 1 Pseudo-codes of the Improved Dragonfly Algorithm with Neighbourhood Structures

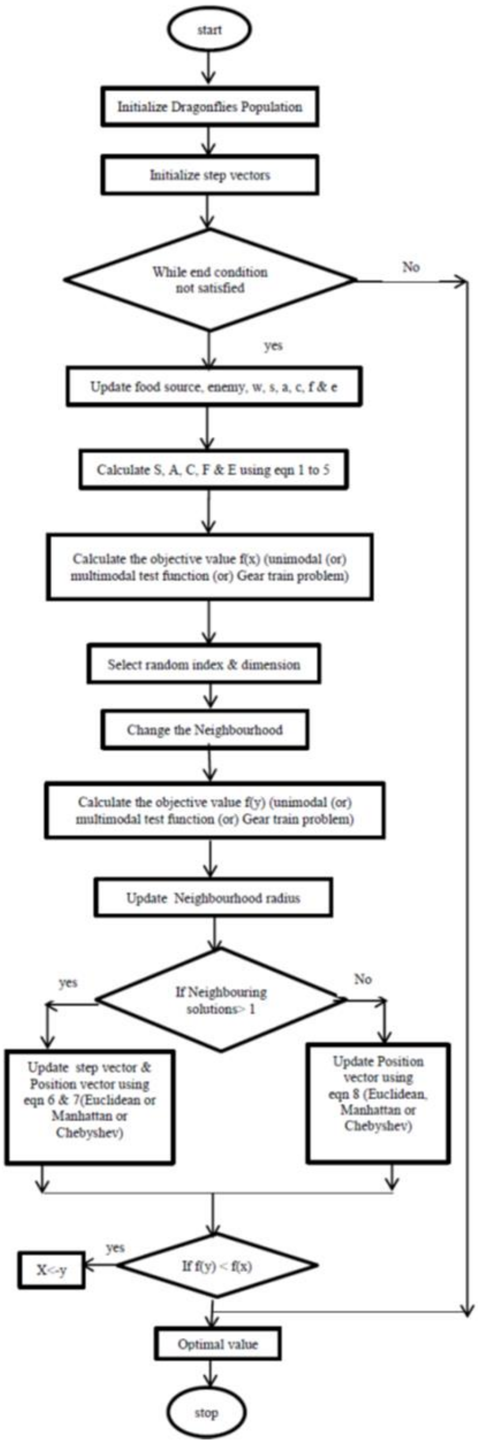


Fig 2 Flowchart of Improved Dragonfly Algorithm with Neighbourhood Structures

Due to its effectiveness and simplicity, many researchers has attracted towards it. But the disadvantage of dragonfly algorithm is premature convergence, local optima and unpredictable results. To overcome the above disadvantage improvement mechanism called neighbourhood search with this algorithm will be investigated. Neighbourhood search [15] is an improvement mechanism which iteratively improves the solution and change the neighbourhood of the dragonfly in the search area until the best solution is found. Improved Dragonfly Algorithm with neighbourhood structures (IDANS) discovers auspicious areas in the search area by using successor solutions in an iterative process. IDANS is then initialized with best value of dragonfly algorithm to further explore the search space. Pseudo-code for dragonfly algorithm with neighbourhood structures and their Flowchart are shown in Fig 1 & 2.

In dragonfly algorithm with neighbourhood structure, each and every individual chooses its own neighbourhood independently. This independence will allows exploration and exploitation of the search space. The solution will be updated only if the newly generated solution will be better than the previous solution, otherwise the solution will remain same. In Neighbourhood search different neighbourhood structures [15] can be used in an efficient way to perform search in a search space. Euclidean, Manhattan, Chebyshev, Mahalanobis and Canberra distances are some of the examples of neighbourhood structures. In our proposed system three neighbourhood structures called Euclidean, Manhattan and Chebyshev distance are chosen to implement on IDANS. The distance of the Neighbourhood are calculated by using any one of the neighbourhood structures such as Euclidean Distance which permits any direction of movement, Manhattan Distance permits four direction of movement and Chebyshev distance allows eight direction of movement between all Dragonflies to find neighbourhood in the search space.

TABLE I
COMPARISON RESULTS OF IDANS ON NEIGHBOURHOOD STRUCTURES

S.no	Function Name	Improved Dragonfly Algorithm with Neighbourhood Structures (IDANS)		
		Euclidean	Manhattan	Chebyshev
1	Sphere	0	0	0
2	Schwefel 2.22	0	0	1.0802e-06
3	Schwefel 2.26	-35420.9356	-34557.3836	-19550.208
4	Rastrigin	0	0.99496	0

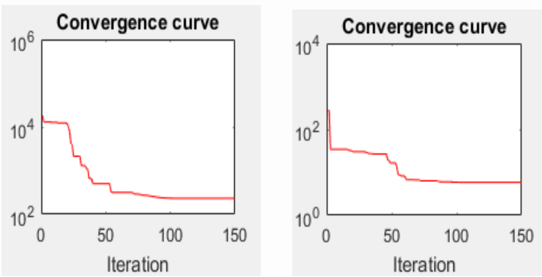
TABLE II RESULT ANALYSES ON THE TEST FUNCTIONS USING IDANS, DA, PSO & GA

S.No	Test Function	Runs	Improved Dragonfly Algorithm with Neighbourhood Structures (IDANS) Euclidean		Dragonfly Algorithm (DA)		Particle Swarm Optimization (PSO)		Genetic Algorithm (GA)	
			Avg	Std	Avg	Std	Avg	Std	Avg	Std
1	Sphere	30	0	0	2.85E-18	7.16E-18	4.2E-18	1.31E-17	748.5972	324.9262
2	Schwefel 2.22	30	0	0	1.49E-05	3.76E-05	0.003154	0.009811	5.971358	1.533102
3	Schwefel 2.26	30	-35420.9356	7.2832e-12	-2857.58	383.6466	-7.1E+11	1.2E+12	-3407.25	164.4776
4	Rastrigin	30	0	0	16.01883	9.479113	10.44724	7.879807	25.51886	6.66936

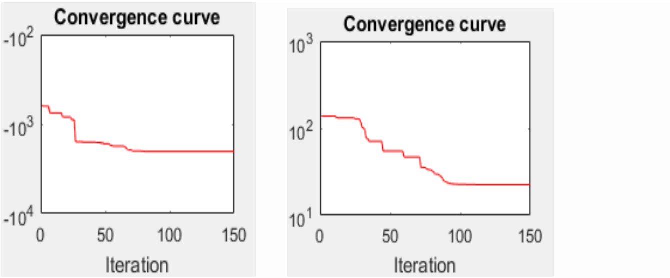
IV. RESULTS AND DISCUSSION

Improved Dragonfly algorithm with neighbourhood structures is applied on Unimodal and multimodal benchmark functions and gear train design problem to verify and analyze the performance of the algorithm. Euclidean, Manhattan and Chebyshev neighbourhood structures are applied on IDANS to examine the performance of algorithm on different neighbourhood structures. It was observed that Euclidean and Manhattan distance provides competitive results as compared to Chebyshev distance for Unimodal and Multimodal Benchmark Functions. Result Analysis on the Test Functions using IDANS (Euclidean, Manhattan and Chebyshev), DA, PSO & GA are described in Table 1.

As per the information provided in the Table 2 it was found that IDANS performs better as compared to other three algorithms and IDANS always provide Best results as compared to DA, PSO and GA. Figure 3 shows the convergence curve of IDANS on Sphere, Schwefel 2.22, Schwefel 2.26 and Rastrigin benchmark functions. This indicates the estimation of the global optimum becomes more as iteration rises. Other fact that can be noted hers is the convergence curves accelerated trend. IDANS converge gradually to obtain global optimum at the final step. As the iteration increases IDANS converge gradually towards global optimum, this emphasis the exploitation of the IDANS.



(a) (b)



(c) (d)

Fig 3 convergence curve of IDANS on Sphere (a), Schwefel 2.22(b), Schwefel 2.26 (c)and Rastrigin (d)

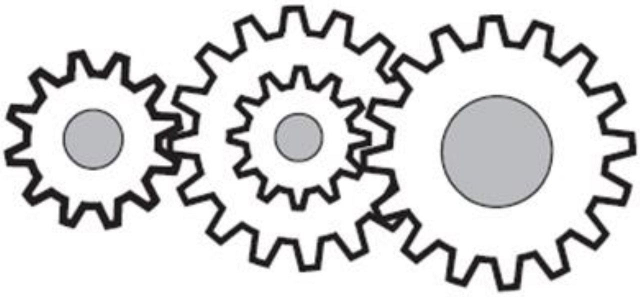
IDANS is applied on Classical engineering problem called Gear Train Design problem [8] which is shown in Figure 4. Gear Train is a discrete optimization problem and it has four integer variables in the range of 12 and 60. The goal of the Gear train problem is to minimize the gear ratio and to find the optimal number of tooth for gear train problem. The Scientific calculation for this problem is

$$f(T_A, T_C, T_B, T_D) = \left(\frac{1}{6.931} - \frac{T_A\ T_C}{T_B\ T_D}\right) \quad (11)$$

TABLE III

Comparative Analysis of Gear Train Problem

Algorithm	Optimal values for variables				f_{min}	Max.eval.
	T_A	T_B	T_C	T_D		
IDANS	22	12	13	52	1.8168e-11	100
ALO	49	19	16	43	2.7009e-012	120
CS	43	16	19	49	2.7009e-012	5000
MBA	43	16	19	49	2.7009e-012	10000
ISA	N/A	N/A	N/A	N/A	2.7009e-012	200
ABC	19	16	44	49	2.78e-11	40000
GA	33	14	17	50	1.362e-09	N/A
ALM	33	15	13	41	2.1469e-08	N/A



The optimal design obtained by IDANS and other algorithms are presented in table 3. The other algorithms designs for Gear train are taken from the paper Seyedali Mirjalili [5]. This table shows that the proposed algorithm IDANS is compared with Ant Lion Optimizer (ALO) [5] algorithm, Cuckoo Search (CS) [9] algorithm, MBA [16], Interior Search (ISA) [17] Algorithm, Genetic Algorithm (GA) [7], Artificial Bee Colony (ABC) [3] algorithm & ALM [18]. IDANS algorithm provides very competitive results as compared to other algorithms in the literature. The result of the Gear Train problem indicates IDANS algorithm is suitable for discrete and constrained problems.

V. CONCLUSION

This paper proposes a new efficient "Improved Dragonfly algorithm with neighbourhood structures" which allows candidate solutions to choose its neighbourhood independently. This allows exploration and exploitation of the search space. Neighbourhood structures like Euclidean, Manhattan and Chebyshev are applied on IDANS to evaluate the performance of the proposed algorithm, out of these three structures Euclidean and a Manhattan structure achieves better than Chebyshev structure. Unimodal and Multimodal benchmark functions and Gear train design problem are chosen to test the performance of the IDANS. From the result it can be find that IDANS gives best results as compared to other algorithms in the literature. In future different classical engineering problems will be implemented on IDANS to investigate the performance of the Improved Dragonfly Algorithm with Neighbourhood Structures.

VI. REFERENCES

[1]. Arora and Singh, "Butterfly optimization algorithm: a novel approach for global optimization", Journal of Soft Computing, pp 1-20,08 (2018).

[2]. S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems", Neural Computing & Applications, pp 1053-1073 (2015).

[3]. Karaboga D," Powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm", Journal of Global Optimization, pp 459–471 (2007).

[4]. Dorigo M, "The ant colony optimization metaheuristic: algorithms, applications, and advances", Handbook of metaheuristics, pp 250–285, (2003).

[5]. S. Mirjalili, "The Ant Lion Optimizer", Advances in Engineering Software, pp 80-98 (2015).

[6]. Arora S , "The firefly optimization algorithm: convergence analysis and parameter selection", International Journal of Computing Applications, pp 48–52 (2013).

[7]. Goldberg DE, "Genetic algorithms and machine Learning", Machine Learning, pp 95–99 (1988).

[8]. Eberhart and Shi, "Particle swarm optimization: developments, applications and resources", Evolutionary computation, pp 81–86 (2001).

[9]. Yang and Gandomi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems", Engineering Computing ,pp 17–35 (2013).

[10]. Wang , Deb and Cui , "Monarch butterfly optimization", Neural Computing and Applications (2015).

[11]. Yang X-S, "A new metaheuristic bat-inspired algorithm", Nature inspired cooperative strategies for optimization, pp 65–74 (2010).

[12]. Yang X-S, "Flower pollination algorithm for global optimization", Unconventional computation and natural computation, pp 240–249 (2012).

[13]. Cuevas, "An optimization algorithm inspired by the States of Matter that improves the balance between exploration and exploitation", Applied Intelligence,pp 256–272 (2014).

[14]. S. Mirjalili, "Moth-flame optimization algorithm: A novel nature inspired heuristic paradigm", Knowledge –Based Systems, pp 228- 249 (2015)

[15]. Rafael Stubs Parpinelli, "Population-based Variable Neighborhood Search Algorithm Applied toUnconstrained Continuous Optimization", International Journal of Bio-Inspired Computation,pp-73-80 (2017).

[16]. Sadollah A, Bahreininejad A, Eskandar H, Hamdi M. Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. Appl Soft Comput 2013;13:2592–612.

[17]. Gandomi AH. Interior search algorithm (ISA): a novel approach for global optimization. ISA Trans 2014.

[18]. Kannan B, Kramer SN. An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. J Mech Des 1994;116:405–11.

Cite this article as :

S. Rajalakshmi, S. Kanmani, S. Saraswathi, "Improved Dragonfly Algorithm with Neighbourhood Structures", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 8 Issue 4, pp. 303-309, July-August 2021. Available at doi : https://doi.org/10.32628/IJSRST218446
Journal URL : https://ijsrst.com/IJSRST218446