

Genetic algorithm and discrete marine predators hybrid optimization technique for traveling salesman problem

Mukhtar Raji *

Arjan Singh [†]

Satya Bir Singh [§]

Department of Mathematics

Punjabi University Patiala

Patiala 147002

Punjab

India

Abstract

The Traveling Salesman Problem (TSP) is a well-known combinatorial optimization problem, that has gained popularity over the past few years because of its broad applicability in service, logistics, and industrial optimization processes. The search for the optimal path, whose result is minimizing the total cost or distance, has always been a computationally intensive and difficult task.

This paper proposed a new hybrid meta-heuristic algorithm, Genetic Algorithm and Discrete Marine Predators Algorithm (GA-DMPA), for solving the TSP. In the first phase (exploration), the genetic algorithm is used to generate candidate tours per iteration. The second phase (exploitation) is the stage in which the Discrete Marine Predators Algorithm further narrows down on these candidate solutions to improve local search and increase convergence to the global optimum. This hybridization can be used to avoid premature convergence and to improve the quality and strength of the solutions. The practicality of the proposed GA-DMPA is proved by applying it to 20 real Indian cities and a benchmark function. The outcomes of the experiment compared with GA and DMPA algorithms, and the introduced hybrid GA-DMPA algorithm give better results. The results are summarized and present in form of figures and table.

Subject Classification (2020): 90C27, 90C59, 68W20.

Keywords: Traveling salesman problem, Genetic algorithm, Discrete marine predators algorithm, Hybrid optimization, Premature convergence.

* E-mail: rajil23.mukhtar@gmail.com (Corresponding Author)

[†] E-mail: Arjan.pup@gmail.com

[§] E-mail: sbsingh69@pbi.ac.in

1. Introduction

Cook et al. [16] proposed Travelling salesman problem. The Travelling Salesman Problem (TSP) is a standard and problematic computationally inflexible combinatorial optimization problem, and is typically used as a benchmark to evaluate the preference of optimization algorithms. Recently, the TSP has attracted the researchers due to the easy theoretical characteristics and applicability in many practical fields. The most notable uses are car routing [17], manufacturing planning [18], customer-service planning [19], and logistics planning [20]. The TSP is a problem, which asks to determine a minimum-cost circuit in the Hamiltonian (i.e., visiting every city once and once only), with the decision variables being a discrete permutation of the set of cities. the classical formulation has been generalized into two main classes, Symmetric TSP (STSP) and Asymmetric TSP.

In Symmetric Traveling Salesman Problem (STSP), the distance between two cities is the same in both directions hence the graph is undirected, and the number of feasible tours has been cut by a factor of two, as described by Kumar, Panwar, and Deep [15]. The Asymmetric Traveling Salesman Problem (ATSP), in turn, allows directional paths or dissimilar routes between the cities, which is closer to reality (traffic patterns, one-way streets, and different costs of traveling, etc.).

Metaheuristic algorithms represent some of the best methods of addressing complex combinatorial optimization problems, especially when multiple objectives and constraints must be considered, Stefanov [21]. These solutions have been designed to provide quality solutions with realistic run times and minimum memory to keep the solutions small in size. Biologically and socially inspired techniques, such as evolutionary algorithms, genetic algorithms, swarm intelligence, and machine-learning algorithms (e.g., neural networks) have demonstrated strong performance across a wide range of optimization tasks, Abdel-Basset, Abdel-Fatah, and Sangaiah [19]. Furthermore, efficient solution strategies for complex decision-making and multi-objective optimization problems have been investigated in recent mathematical optimization studies, which can be useful for solving complex optimization problems in real-world applications, Chou [22]. In the current paper, a new hybrid metaheuristic GA - DMPA is presented, that will be a combination of a Genetic Algorithm and Discrete Marine Predators Algorithm (DMPA) to tackle the Traveling Salesman Problem (TSP). The hybrid approach is tested empirically using standard benchmark cases and using a real-life dataset that consists of 20 Indian cities. The competitive experiments in comparison with the distinct GA and DMPA show that GA and DMPA are characterized by better quality of solutions and better convergence behavior. The rest of the paper is structured in the following method: Section 2 will contain a review of the related research and available algorithms. In Section 3, the metaheuristic methods used in this work are described. Section 4 describes the implementation process and gives the results

of the experiment. Lastly, Section 5 brings the study to completion and provides recommendations on possible future research.

2. Literature review

Recently, several algorithms were developed to get efficient tours to the Traveling Salesman Problem (TSP). The majority them are metaheuristic methods that can be used to generate high-quality approximate solutions within a reasonable computing time. The close relationship that exists between the vehicle-routing problems and the TSP has the effect of making the numerous schemes that have been developed to deal with vehicle routing flexible enough to be applied to the TSP. As shown in Table 1, recent state-of-the-art metaheuristic algorithms proposed in the past decade have shown remarkable improvement in terms of solution quality and search effectiveness.

Table 1
Recent state-of-the-art metaheuristic algorithms proposed for solving the Traveling Salesman Problem (TSP) during the last decade.

Publication Year	Developed Technique	Core Algorithm	Main Operators and Strategies	Source
2014	Discrete Cuckoo Search (DCS)	Cuckoo Search (CS)	2-opt and double-bridge neighborhood operations	Ouaarab, Ahiod, and Yang [5]
2014	Hybrid Genetic technique (HGA)	Genetic Algorithm	Integration of two local search-based optimization strategies	Wang [14]
2017	Discrete Symbiotic Organisms Search (DSOS)	Symbiotic Organisms Search (SOS)	Swap, inversion, and insertion-based mutation operators	Ezugwu and Adewumi [1]
2018	Discrete Comprehensive Learning Particle Swarm Optimization (D-CLPSO)	Particle Swarm Optimization (PSO)	Use of classical arithmetic operators adapted for the TSP	Zhong et al. [4]
2018	Discrete Water Cycle Algorithm (DWCA)	Water Cycle technique (WCA)	Insertion and 2-opt neighborhood operations	Osaba et al. [3]
2019	Two-phase metaheuristic (IH + VNS + TS)	Time-Dependent TSP (TDTSP)	Insertion Heuristic (IH) + Variable Neighbourhood Search (VNS) + Tabu Search (TS)	Ban [6]

2019	Discrete Symbiotic Organisms Search with Excellence Coefficients and Self-Escape Mechanisms	Symbiotic Organisms Search (SOS)	Enhanced DSOS with excellence coefficients	Wang, Wu, and Xu [12]
2021	Discrete Crow-Inspired Algorithm (DCIA)	Crow Search Algorithm (CSA)	Modular arithmetic and discrete transition operations	Al-Gaphari, Al-Amry, and Al-Nuzaili [13]
2020	Variable Neighborhood-Based Whale Optimization Algorithm	Whale Optimization Algorithm (WOA)	Adaptive weight tuning, Gaussian perturbation, and neighborhood-based search strategies	Zhang, Hong, and Liu [2]
2020	Discrete Spider Monkey Optimization Algorithm	spider Monkey Optimization System	Swap sequence and discrete neighborhood operators	Akhand et al. [11]
2021	Discrete Grey Wolf Optimizer (D-GWO)	Grey Wolf Optimizer	2-opt local search heuristic	Panwar and Deep [10]
2022	Artificial Electric Field Algorithm with Greedy State Transition (GST-AEFA)	Spherical Multiple Traveling Salesman Problem (multiple salesmen on a sphere)	Integer-serialized coding + Greedy State Transition + Artificial Electric Field Algorithm	Bi et al. [7]
2023	Discrete Pathfinder Algorithm (D-PFA)	Discrete Pathfinder Algorithm (D-PFA)	Discretised Pathfinder Algorithm (population-based) with exploration & exploitation modules	Pirozmand et al. [8]
2024	Biased Random-Key Genetic Algorithm with Q-Learning (BRKGA-QL)	Genetic Algorithm (GA) integrated with Q-Learning	2-opt heuristic	Chaves et al. [9]

3. Genetic algorithm

The initial description of the Genetic Algorithm (GA) was by Holland in the early 1970s in the United States at the University of Michigan as he attempted to understand the adaptive processes occurring in natural systems,

Zhong et al. [4]. Whereas the GA is used to binary-encoded representations, there are numerous other encoding schemes that have been implemented in different applications. Overall, a GA is based on the use of the crossover operator, which integrates data of two parent solutions, and the mutation operator, which creates random changes to preserve the diversity of the population.

- **Crossover:** The probability of individuals may be based on their fitness values when deciding whether to use them to be crossover. In the so-called process recombination, the two parent chromosomes will be recombined to form new offspring that will be added to the subsequent generation, Ouaraab, Ahiod, and Yang [5]. As an example, two chromosomes are given:

$$C1=1010$$

$$C2 = 0011$$

In case a random point is chosen to crossover after the second gene, then the chromosomes are partitioned as follow:

$$C1 = 10 | 10$$

$$C2 = 00 | 11$$

After crossover

$$C1' = 1011$$

$$C2' = 0010$$

The two new offspring chromosomes will add to next population and call next generation.

- **Mutation:** After crossover is mutation, in which individuals are chosen randomly, and once an individual is chosen, a mutation point is randomly picked between $[-1,1]$ within this individual. Mutation is important in order to avoid the algorithm reaching a local optimum too quickly. The search may even never get to the global optimum because at the local optimum individuals are fitting to survive. Altering the population at random, mutation assists in keeping a diverse population and gives an opportunity to experiment with new options that would not have been found otherwise.
- **Pseudo-code of simple genetic algorithm:**

```

Generate an initialize population randomly;
Evaluate each individual;
While (termination criterion is not satisfied) do
    Select parents; # the best individuals are selected.
    Crossover;
    Mutation
    Evaluate new individuals;
    Replace worst individuals by the best new ones;
End

```

4. Discrete Marine Predators Algorithm

The standard MPA is formulated to be optimized nonstop optimization and cannot be used directly to solve the combinatorial optimization problem (COPs) such as the Symmetric Traveling Salesman Problem (STSP). In response to this limitation, Kumar, Panwar, and Deep [15] proposed the Discrete Marine Predators Algorithm where memory-saving in addition Food Attraction Dynamics machineries of original MPA are eliminated to make it easier to implement and better adapted to discrete problem domains. In D-MPA, the concepts of the original MPA, which were formulated by Faramarzi et al. [23], are maintained, and the algorithm first produces real-valued candidate solutions. Such solutions are, then optimized by two discrete operators namely permutation crossover and the 2-opt tour refinement method. Once a continuous solution has been obtained, it is mapped into a practical TSP tour with the help of a continuous-to-discrete (CD) mapping process. The mapping parameter of the CD procedure, or the function of converting a real-valued solution into the discrete counterpart, is defined as, Kumar, Panwar, and Deep [15]:

$$\theta(x_1, x_2, \dots, x_D) = (\text{index of LV}, \text{index of } 2^{\text{nd}}\text{LV}, \dots, \text{index of } D^{\text{th}}\text{lv})$$

where LV denotes the lowest values or entries in the solution $(x_1, x_2, \dots, x_D)$.

- **Pseudo-code of simple Discrete Marine Predators algorithm**

```

Input  $Iter_{max}$ : maximum number of iterations to be performed
      n: number of randomly generated feasible solutions
      d: dimension of the initial population (permutation)
calculates the fitness value of each solution.
construct elite matrix  $E$  by replicating optimal solution.
for  $Iter = 1$  to  $Iter_{max}$ 
  if  $Iter < 1/3Iter_{max}$ 
    Phase-1: Update the position of Prey ( $P$ ) by Eq( 4)
  else if  $while\ 1/3Iter_{max} < Iter < 2/3Iter_{max}$ 
    Phase-2: Update the position of first half number of Prey by Eq( 5 )
             Update the position of second half number of Prey by Eq( 6 )
  else if  $Iter > 2/3Iter_{max}$ 
    Phase-3: Update the position of Prey ( $P$ ) by Eq( 7)
  end
  Apply CD function
  Apply permutation crossover
  Apply 2-opt process
end

```

5. Methodology

the first stage (exploration), the genetic algorithm is used to generate candidate tours per iteration. The next step (exploitation) is the step in which Discrete Marine Predators Algorithm further narrows down on these candidate solutions to improve local search and increase convergence to the global optimum.

6. Mathematical formulation

Let N be the number of cities. A tour defines as a permutation $k = (x_1, x_2, \dots, x_n)$ Where each city visiting exactly once.

given a symmetric distance matrix $D = (d_{ij})$,

Minimize $f(k) = \sum_{i=1}^{N-1} d_{x_i x_{i+1}} + d_{x_N x_1}$

Where d_{ij} represents the distance between city i and city j

7. Hybrid Algorithm of GA-DMPA

Exploration (GA)

Initial population p^0 with M random permutation.

$$p^{g+1} = \left\{ \mu \left(c(s(p^g)) \right) \right\}, \quad g = 0, 1, \dots, G - 1$$

S select parents from p^g

C crossover

μ mutation

Exploitation (D-MPA)

$$p_0 = p_{\min} + \text{round}(0,1) \times (p_{\max} - p_{\min})$$

Here, $P_{\max}$ and $P_{\min}$ denote the upper and lower bounds of the problem, respectively, while $\text{Rand}(0,1)$ represents a uniformly distributed random number in the interval $(0,1)$.

$\theta(x_1, x_2, \dots, x_D) = (\text{indices of the lowest value, second} \\ - \text{lowest value, ..., D}^{\text{th}} \text{lowest value})$

mapping parameter of the CD θ mapping of the real-valued solution to its discrete counterpart.

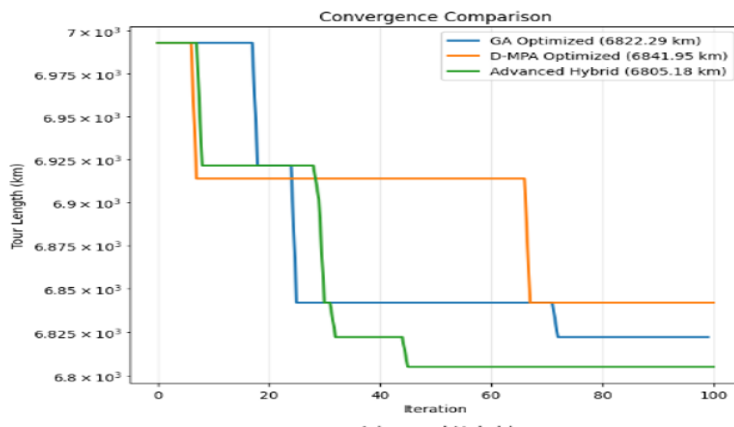


Figure 1

TSP convergence: GA, D-MPA and hybrid (20 Indian cities)

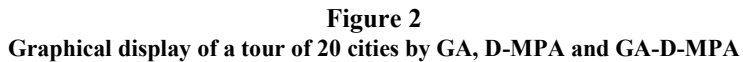


Table 2
Loaded 20 Indian Cities

Algorithm	length of tour (km)	Run time (s)	Iteration
GA	6822.29	2.10	100
D-MPA	6841.95	1.11	100
Hybrid GA-DMPA	6805.18	0.30	100

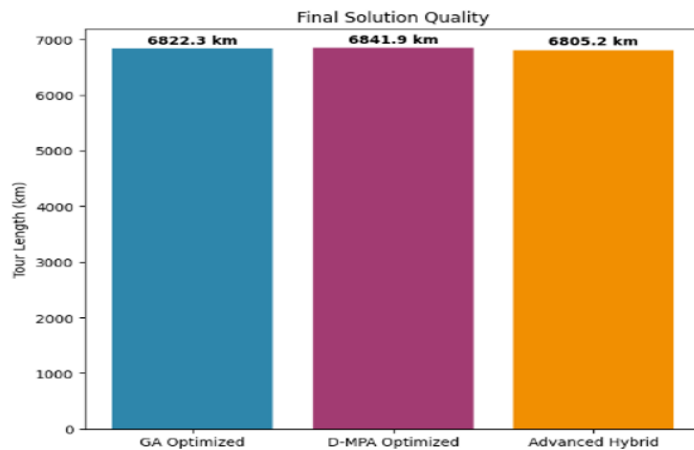


Figure 3
Tour final solution quality generates by GA, D-MPA and GA-D-MPA

8. Results and Discussion

Convergence plot (Figure 1) indicates that D-MPA has quick improvement in the initial stages, but it stagnates too soon at 6841.95 km. GA has more of an exploratory search pattern as it continuously gets better until reaching a standoff at 6822.29 km. Conversely, the Advanced Hybrid algorithm keeps on improving as the 50-iteration run continues, which indicates that there is more balance in exploration and exploitation because the global search of GA and the local refinement mechanism of D-MPA. The tour paths of GA, D-MPA, and Hybrid GA-DMPA shown in Figure 2, where the proposed Hybrid GA-DMPA generates the best tour with least tour length.

The performance analysis of the three optimization techniques, Genetic Algorithm (GA), Discrete Marine Predators Algorithm (D-MPA) and the proposed Hybrid GA-DMPA shows apparent differences in the convergence behaviour and quality of the final solution to the 20 Indian cities for TSP case. As it can be seen in Figure 3, shows the final solution quality, the Advanced Hybrid solution has attained the shortest tour length of 6805.18 km and is superior to the solutions offered by the GA (6822.29 km) and D-MPA (6841.95 km). The three algorithms developed almost identical solutions with a variation of less than 3 percent, however, the convergence pattern of the algorithms varied considerably.

The comparative results in Table 2 also demonstrate the effectiveness of the proposed hybrid approach. Hybrid GA-DMPA not only has the highest solution accuracy in terms of the shortest tour length among the three methods, but also takes the least time (0.30 seconds) to produce a solution compared with 2.10 seconds for GA and 1.11 seconds for D-MPA, and is thus more efficient.

all the algorithms were run for 100 iterations and produced solutions that are very close to each other, but the convergence patterns and run times are different.

Findings have shown that GA demonstrates rapid initial convergence, it may converge prematurely, and D-MPA exhibits stronger exploration but converges to a relatively poorer solution, and the Hybrid GA-DMPA approach presents the most solid and dependable optimization performance. These features render the hybrid approach especially appropriate in situations when routing and transportation planning exercises are complicated and solution quality, convergence stability and time matter.

9. Conclusion

This paper presented a comparative analysis of three optimization strategies Genetic Algorithm (GA), Discrete Marine Predators Algorithm (DMPA), and a hybrid system of Genetic Algorithms and Discrete Marine Predators Algorithms (GA-DMPA) to Traveling Salesman Problem in 20 Indian cities. The experimental findings indicate that all the three algorithms can generate quality, near optimal solutions within 3 percent difference in the end tour lengths. The difference between the methods, however, is significant with regard to convergence dynamics, run time and the features of route generation. In D-MPA, there was fast convergence at the beginning with early stagnation and in GA, there was more coverage of search space at the expense of slower convergence. The hybrid GA-DMPA method exhibited the most equilibrating performance that resulted in constant improvement during the execution, generation of geographically consistent tours and reduction in long inter-city transitions.

The results prove that combining evolutionary and predator prey inspired search strategies can be useful in improving convergence rate and quality of the solutions for TSP. The hybrid GA-DMPA model is a versatile and dependable optimization tool that is applicable in actual routing of the world.

References

- [1] A. E.-S. Ezugwu and A. O. Adewumi, "Discrete symbiotic organisms search algorithm for travelling salesman problem," *Expert Syst Appl*, vol. 87, pp. 70–78 (Nov. 2017), doi: 10.1016/j.eswa.2017.06.007.
- [2] J. Zhang, L. Hong, and Q. Liu, "An Improved Whale Optimization Algorithm for the Traveling Salesman Problem," *Symmetry (Basel)*, vol. 13, no. 1, p. 48 (Dec. 2020), doi: 10.3390/sym13010048.
- [3] E. Osaba, J. Del Ser, A. Sadollah, M. N. Bilbao, and D. Camacho, "A discrete water cycle algorithm for solving the symmetric and asymmetric traveling salesman problem," *Appl Soft Comput*, vol. 71, pp. 277–290 (Oct. 2018), doi: 10.1016/j.asoc.2018.06.047.

- [4] Y. Zhong, J. Lin, L. Wang, and H. Zhang, "Discrete comprehensive learning particle swarm optimization algorithm with Metropolis acceptance criterion for traveling salesman problem," *Swarm Evol Comput*, vol. 42, pp. 77–88 (Oct. 2018), doi: 10.1016/j.swevo.2018.02.017.
- [5] A. Ouaraab, B. Ahiod, and X.-S. Yang, "Discrete cuckoo search algorithm for the travelling salesman problem," *Neural Comput Appl*, vol. 24, no. 7–8, pp. 1659–1669 (Jun. 2014), doi: 10.1007/s00521-013-1402-2.
- [6] H. B. Ban, "An efficient two-phase metaheuristic algorithm for the time dependent traveling Salesman problem," *RAIRO - Operations Research*, vol. 53, no. 3, pp. 917–935 (Jul. 2019), doi: 10.1051/ro/2019006.
- [7] J. Bi, G. Zhou, Y. Zhou, Q. Luo, and W. Deng, "Artificial Electric Field Algorithm with Greedy State Transition Strategy for Spherical Multiple Traveling Salesmen Problem," *International Journal of Computational Intelligence Systems*, vol. 15, no. 1, p. 5 (Dec. 2022), doi: 10.1007/s44196-021-00059-0.
- [8] P. Pirozmand, A. Hosseinabadi, M. J. Chari, F. Pahlavan, S. Mirkamali, G.-W. Weber, S. Nosheen, and A. Abraham, "D-PFA: A Discrete Metaheuristic Method for Solving Traveling Salesman Problem Using Pathfinder Algorithm," *IEEE Access*, vol. 11, pp. 106544–106566 (2023), doi: 10.1109/ACCESS.2023.3320562.
- [9] A. A. Chaves, B. L. Vianna, T. T. da Silva, and C. M. Schenekemberg, "A parallel branch-and-cut and an adaptive metaheuristic to solve the Family Traveling Salesman Problem," *Expert Syst Appl*, vol. 238, p. 121735 (Mar. 2024), doi: 10.1016/j.eswa.2023.121735.
- [10] K. Panwar and K. Deep, "Discrete Grey Wolf Optimizer for symmetric travelling salesman problem," *Appl Soft Comput*, vol. 105, p. 107298 (Jul. 2021), doi: 10.1016/j.asoc.2021.107298.
- [11] M. A. H. Akhand, S. I. Ayon, S. A. Shahriyar, N. Siddique, and H. Adeli, "Discrete Spider Monkey Optimization for Travelling Salesman Problem," *Appl Soft Comput*, vol. 86, p. 105887 (Jan. 2020), doi: 10.1016/j.asoc.2019.105887.
- [12] Y. Wang, Y. W. Wu, and N. Xu, "Discrete symbiotic organism search with excellence coefficients and self-escape for traveling salesman problem," *Comput Ind Eng*, vol. 131, pp. 269–281 (May 2019), doi: 10.1016/j.cie.2019.04.008.
- [13] G. H. Al-Gaphari, R. Al-Amry, and A. S. Al-Nuzaili, "Discrete crow-inspired algorithms for traveling salesman problem," *Eng*

- Appl Artif Intell*, vol. 97, p. 104006 (Jan. 2021), doi: 10.1016/j.engappai.2020.104006.
- [14] Y. Wang, "The hybrid genetic algorithm with two local optimization strategies for traveling salesman problem," *Comput Ind Eng*, vol. 70, pp. 124–133 (Apr. 2014), doi: 10.1016/j.cie.2014.01.015.
 - [15] M. Kumar, K. Panwar, and K. Deep, "Discrete Marine Predators Algorithm for Symmetric Travelling Salesman Problem," *Evol Intell*, vol. 17, no. 5–6, pp. 3833–3848 (Oct. 2024), doi: 10.1007/s12065-024-00960-5.
 - [16] W. J. Cook, D. L. Applegate, R. E. Bixby, and V. Chvátal, *The Traveling Salesman Problem*. Princeton University Press (Sep. 2011), doi: 10.1515/9781400841103.
 - [17] T. Tawanda, P. Nyamugure, S. Kumar, and E. Munapo, "A Labelling Method for the Travelling Salesman Problem," *Applied Sciences*, vol. 13, no. 11, p. 6417 (May 2023), doi: 10.3390/app13116417.
 - [18] R. Jain, M. L. Meena, and K. P. Singh, "Travelling salesman problem optimization using hybrid genetic algorithm," in *Conference Proceedings of ICDLAIR2019*, M. Tripathi and S. Upadhyaya, Eds., Lecture Notes in Networks and Systems, vol. 175. Cham, Switzerland: Springer, 2021, pp. 350–356, doi: 10.1007/978-3-030-67187-7_36.
 - [19] M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, "Metaheuristic Algorithms: A Comprehensive Review," in *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*, Elsevier, pp. 185–231 (2018), doi: 10.1016/B978-0-12-813314-9.00010-4.
 - [20] A. J. Jones, "Genetic algorithms and their applications to the design of neural networks," *Neural Comput Appl*, vol. 1, no. 1, pp. 32–45 (Mar. 1993), doi: 10.1007/BF01411373.
 - [21] S. M. Stefanov, "Multi-objective optimization revisited," *Journal of Information & Optimization Sciences*, vol. 47, no. 3, pp. 833–848 (2026), doi: 10.47974/JIOS-1395.
 - [22] C.-C. Chou, "Touring a Sequence of Spheres in R3," *Journal of Interdisciplinary Mathematics*, vol. 27, no. 7, pp. 1615–1636 (2024), doi: 10.47974/JIM-2002.
 - [23] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm", *Knowledge-based systems*, vol. 191, Art. no. 105190 (2020).