Applied Soft Computing 105 (2021) 107298

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Discrete Grey Wolf Optimizer for symmetric travelling salesman problem

Karuna Panwar, Kusum Deep *

Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee, India

ARTICLE INFO

Article history:
Received 1 June 2020
Received in revised form 18 February 2021
Accepted 28 February 2021
Available online 17 March 2021

Keywords:
Travelling salesman problem
2-opt
Metaheuristics
Grey Wolf Optimizer

ABSTRACT

Grey Wolf Optimizer (GWO) is a recently developed population-based metaheuristic algorithm which imitates the behaviour of grey wolves for survival. Initially, GWO was proposed to solve continuous optimization problems where it performed well. In recent years, numerous versions of GWO are available in the literature and GWO has been widely used to solve engineering problems. This paper presents a novel discrete GWO algorithm (D-GWO) to solve complex discrete travelling salesman problem (TSP). The 2-opt algorithm is incorporated into it to improve the performance of the proposed algorithm. To inspect the performance of D-GWO, the results of the proposed algorithm are compared with well-known metaheuristic algorithms such as Bat Algorithm(BA), Discrete Firefly Algorithm, Imperialist Competitive Algorithm and some other classical algorithms over several known TSP instances. In order to obtain unbiased and rigorous comparison, descriptive statistics such as mean and standard deviation are used as well as statistical tests such as Friedman test and Holm's test are also conducted. The D-GWO is implemented in MATLAB environment. The computational result carried out in this study has shown that D-GWO outperforms significantly over other alternative algorithms.

1. Introduction

The combinatorial optimization problems (COPs) are prevalent in the field of optimization, artificial intelligence and in many other applications. There are numerous NP-hard problems in defence, agriculture, industry and other areas which can be directly converted into combinatorial optimization problems and can be solved efficiently using available techniques. To address these problems, several exact and non-exact methods are available in the literature. Due to real life applicability of COPs in various disciplines like power systems, routing, engineering and manufacturing, several algorithms have been developed to find an efficient solution.

Travelling Salesman Problem [1] is one of the most appealed combinatorial optimization problems, also used as a benchmark for algorithms testing too. In last decade, TSP generate a great social, scientific and industrial interest. The aim of salesman in TSP is to travel each city exactly once and form a closed tour by returning to its first city with minimum cost. In TSP, decision variables are discrete. In literature, several exact methods such as cutting plane [2], branch and bound [3], Lagrangian dual [4] etc. are available to solve TSP. Although these traditional methods are most suitable to solve TSP but become unsuitable in some cases such as, the size of problem is too large, lack in the precision of data collection, objectives of problem conflicts etc. Also, due to the complexity of the problem, the computational time for solving TSP using traditional techniques is very high. To overcome this kind of difficulties, the researchers used heuristics and metaheuristics to solve TSP efficiently.

Metaheuristics were proposed many years ago and still remain interesting due to their ability to find near optimal of optimization problems. Metaheuristic methods provide an advantage over traditional methods; these methods help to find a satisfactory solution when traditional methods become unsuitable or fail. The search of an optimal solution may be entirely inappropriate in certain practical applications because of several factors: the size of problem, the difficulty in formulating some constraints or the presence of conflicting objectives, the lack of precision in data collection, the dynamic of work environment etc. In such cases, exact methods either fail to solve the problem or does not lead to acceptable solution. The metaheuristics usually provide acceptable solution for such cases in reasonable computation time. However, compared to classical optimization algorithms and iterative methods, metaheuristics do not guarantee for a globally optimal solution. TSP is one of the NP-hard combinatorial optimization problem (COP). In the field of routing problem, metaheuristics play a vital role and become a hot topic for the scientific community. In order to find optimal solution of TSP, it is difficult to check all the possible feasible solution special when the search space

* Corresponding author.
E-mail addresses: kpanwar@ma.iitr.ac.in (K. Panwar), kusum.deep@ma.iitr.ac.in (K. Deep).

https://doi.org/10.1016/j.asoc.2021.107298
1568-4946/

is large. Therefore, to address these issues, several metaheuristics based on nature, swarm intelligence, biological system, annealing process, physical and chemical process etc. have been proposed in the last decade. PSO algorithm equipped with swap operator and swap sequence mutation methods for BURMA14 problem used by Wang et al. [5]. Akhand et al. [6] proposed another variant of PSO to solve TSP efficiently. Karaboga et al. [7] developed the Combinatorial ABC (CABC) algorithm to solve combinatorial optimization problems. A hybrid approach named as GA-PSO-ACO is presented by Deng et al. [8] where GA and PSO work on exploration phase and ACO is used for exploitation. Chen and Chien [9] proposed another hybrid method to solve TSPs, named as genetic simulated annealing ant colony system with particle swarm optimization (GSA-ACS-PSOT). Khan et al. [10] used 2-Opt and 3-Opt with an artificial bee colony (ABC) algorithm and proposed a swap sequence based artificial bee colony for TSP. Discrete spider monkey algorithm [11] is one of the recently developed algorithm to solve TSP efficiently. Among all these, especially the nature-inspired algorithms such as ACO [12], PSO [13], ABC [10], FA [14] etc. have been applied successfully by the researchers to find optimal solutions of routing problems in recent years. Also, these algorithms hybridized with other algorithms to achieve better results.

This study is focused on one of the recently proposed nature-inspired algorithm known as Grey Wolf Optimizer (GWO) to solve TSP. GWO is population based nature-inspired algorithm proposed by Mirjalili in 2014 [15]. It simulates the social behaviour of the grey wolf for hunting prey. This is a swarm intelligence algorithm which is based on leadership hierarchy. Initially, this was implemented for continuous optimization problem and real world problems in complex environment. GWO outperforms over many other algorithms when tested on complex mathematical benchmark problems. Therefore, the potential of GWO finding global optimal attracts researchers for further investigation. GWO gained wide recognition in solving continuous problems in a very short time of period, but for discrete problems such as assignment and routing problem, this algorithms has not gained much attention from researchers.

Despite large number of research studies, the GWO has rarely been applied to any routing problem. The key reason that inspired the realization of this work was this lack of study. GWO has not been explored significantly yet for routing problems specially TSP. This is one of the first time when GWO addresses the famous routing problem TSP. This study aims to attract the reader towards this popular metaheuristics for solving TSP and the other routing problems. The results obtained by the GWO for TSP also emphasize the future applicability of GWO in efficiently solving the complex discrete optimization problems. The results are compared with other state of art algorithms such as genetic algorithm (GA), evolutionary simulated annealing (ESA) [16], the island based distributed genetic algorithm (IDGA) [17], and some other recently proposed algorithm which are bat algorithm (BA) [18], firefly algorithm (FA) [14] and a discrete imperialist competitive algorithm (DICA) [19]. These algorithms have been selected for the comparison of results with the proposed D-GWO over TSP instances because of some similarities in their implementation. For rigorous and fair comparison, two statistical tests: Friedman test and Holm's test are performed in addition to conventional test based on best result, average, standard deviation etc.

The main contributions of this study to the literature are listed below:

- D-GWO is a novel discrete and permutation coded optimization algorithm for combinatorial optimization problem.
- Experimental findings show that D-GWO provides adequate and comparable solutions for symmetric TSPs.
- This study incorporated 2-opt algorithm with GWO to solve TSP for the first time in order to get better solution.

The rest of the article is arranged as follow: Section 2 presents the work done by the researchers related to GWO in detail. In Section 3 a short idea about the TSP problem and in Section 4 a brief description of original GWO is made. The proposed discrete GWO for solving TSP is presented in Section 5. Section 6 is dedicated to results obtained by D-GWO. Finally, Section 7 concludes the article by providing its future scope.

2. Related work

Several applications of GWO are available in literature for engineering purpose such as image processing, control engineering problems, robotics, power dispatch problems, electric and power engine problems, clustering etc. Also, GWO algorithms have been modified many times by researchers with the intention of addressing real world problems. Komaki et al. [20] developed a discrete version of GWO for two-stage assembly flow shop scheduling problem. In Zhang et al. [21], GWO is used for safe path planning of unmanned combat aerial vehicle path planning with costing minimum fuel. Another application of GWO is addressed by Jayabharathi et al. [22], they solved the economic dispatch problem using GWO, which is a non-convex and discontinuous problem in nature. Li et al. [23] presented a modified discrete grey wolf optimizer (MDGWO) for multiple image thresholding, which improves on the optimal solution updating mechanism of the grey wolves by the weights. Recently, GWO is used to solve a widely studied problem hybrid flow shop scheduling problem by Lu et al. [24]. They proposed multi-objective cellular grey wolf optimizer (MOCGWO) to solve this problem and also used variable neighbourhood search strategy. Abdel-Basset et al. [25] utilized GWO to solve the feature selection for classification problems based on the wrapper methods, where they integrate GWO with two phase mutation. Lu et al. [26] proposed a hybrid multi objective GWO for dynamic scheduling in a real-world welding industry. In their study, a modified social hierarchy is designed to improve its exploitation and exploration abilities of GWO.

Besides of applications, numerous version of GWO have been proposed by researchers are available in literature which based on different updating mechanism, adding new operators, encoding scheme, structure of population, and some of them are problem specific. One example is RW-GWO proposed by Gupta et al. [27], in which a random walk is incorporated with GWO to improve the search ability of grey wolves. Another example is multi objective grey wolf optimizer (MOGWO) presented by Mirjalili et al. [28] to solve the problem having multiple objectives. Additionally, GWO has been hybridized with other metaheuristics such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Whale optimizer (WOA), Biogeography-Based Optimization (BBO) etc. [29–32].

On the other hand, TSP is one of the complex combinatorial optimization problem which is studied extensively by the researchers. Also frequently used to validate the performance of new discrete optimization algorithms. Several state of the art algorithms have been applied to TSP such as GAs [33], SA [34], TS [35] etc. Swarm intelligence based algorithms such as ACO, PSO and ABC also have been widely used to solve TSP [5,7,36]. Furthermore, some hybrid approaches also developed to solve TSP efficiently. The work presented in [37] is a hybrid method based on ACO, PSO and the 3-opt algorithm.

Despite of great amount of research work within a very short time period, GWO has been rarely applied to any routing problem, especially travelling salesman problem (TSP). This lack of studies is the main motivation behind this study. This study covers this

lack of research by introducing a novel discrete version of GWO to solve TSP using hamming distance concept for the first time with GWO. Here, hamming distance is the distance between two grey wolves in the pack of wolves. Each wolf in pack represent the solution of TSP. The concept of hamming distance is used earlier in other metaheuristics when applied to TSP [38]. Osaba et al. [18] proposed an improved bat algorithm using hamming distance technique and 2-opt algorithm for local search; this algorithm is tested on both symmetric and asymmetric travelling salesman problem.

3. The travelling salesman problem (TSP)

The travelling salesman problem (TSP) continuously attracting the research community in past decades. Several approaches proposed in literature are currently in use to solve TSP and its variants such as ATSP (asymmetric travelling salesman problem), MTSP (multiple travelling salesman problem). However, till now, there is no exact method available in literature which guarantees to obtain its optimal solution. This is the reason why heuristic and other approximation methods are introduced to solve it. TSP and its variants are used in numerous research work as well in several application areas too such as interview scheduling, vehicle routing, mission planning etc.

This study is focused on solving symmetric travelling salesman problem (STSP). The STSP can be mathematically modelled over a complete undirected graph $G = (N, \Sigma)$, where $N = \{1, 2, 3....n\}$ is the set of nodes which represent the cities and Σ is the number of edges. Here, each edge of graph represent the path between two cities and length of edge represent the distance between the associated cities. The salesman travel n cities, starting and ending at same city and each city is visited by salesman exactly once. Let y_{lk} represent the distance between city l and k. A cost matrix $Y = (y_{lk})_{n\times n}$ is defined over Σ. The TSP can be mathematically formulated as follows:

$$\begin{aligned} &Determine \quad z_{lk},\ \forall l, k \in N \ and \ l \neq k \\ &Minimize\ f = \sum_{l=1}^{n}\sum_{k=1}^{n} z_{lk}y_{lk} \\ subject\ to \quad &\sum_{l=1}^{n} z_{lk} = 1, \forall k \in N \\ &\sum_{k=1}^{n} z_{lk} = 1, \forall l \in N \end{aligned} \tag{1}$$

where z_{lk} is 1 if salesman visit to city k from city l, otherwise 0.

The objective of salesman is to find a complete tour $(z_1, z_2, \ldots, z_n)$ to

$$Minimize\ f = \sum_{l=1}^{n-1} y_{z_l z_{l+1}} + y_{z_n z_1} \tag{2}$$

4. Grey wolf optimizer (GWO)

GWO is a relatively new population based algorithm inspired by the hunting strategy and leadership hierarchy of grey wolves. Grey wolves usually live in a strict and organized pack. On average, the size of the pack is 5–12. The impressive feature of grey wolves is their strict social dominant hierarchy, which differs it from other swarm intelligence algorithms. The leaders can be either a male or female. Each wolf of pack represent a solution. In GWO, the population of grey wolves is divided into four groups: alpha, beta, delta and omega according to the social hierarchy of wolves. Alpha is assigned to the most powerful wolf i.e. the wolf best in hunting. In mathematical way, the best solution so far is called alpha (α). In the similar way, the second and third best represents the beta (β) and delta (δ) respectively. While the rest of the candidate solutions are considered as omega (ω).

Grey wolves are counted as one of the great predators in nature. Their hunting behaviour can be classified in following three steps:

- Tracking and approaching the prey
- Encircling and harassing the prey
- Attacking on prey

The encircling pattern of wolves around the prey can be mathematically modelled as follows:

$$c = 2 \cdot Rand2 \tag{3}$$

$$\gamma = 2 \cdot r \cdot Rand1 - r \tag{4}$$

where Rand1 and Rand2 are uniformly distributed random number lying between 0 and 1 and r decreased linearly from 2 to 0 over iterations, expressed as

$$r = 2 - t * \frac{2}{iterations} \tag{5}$$

And

$$d = |c \cdot X_P(t) - X(t)| \tag{6}$$

$$X(t+1) = |X_P(t) - \gamma \cdot d| \tag{7}$$

where $X(t+1)$ represents the position of wolf at $(t+1)$th iterations, obtained by using position of prey $X_P(t)$ at tth iterations and difference vector d.

Each wolf use the potential of alpha, beta and delta wolves in hunting strategy as they are best in pack. The wolves update their position with the help of alpha, beta and delta as follows:

$$X_1 = |X_\alpha - \gamma_\alpha \cdot d_\alpha|,\ d_\alpha = |c \cdot X_\alpha - X| \tag{8}$$

$$X_2 = |X_\beta - \gamma_\beta \cdot d_\beta|,\ d_\beta = |c \cdot X_\beta - X| \tag{9}$$

$$X_3 = |X_\delta - \gamma_\delta \cdot d_\delta|,\ d_\delta = |c \cdot X_\delta - X| \tag{10}$$

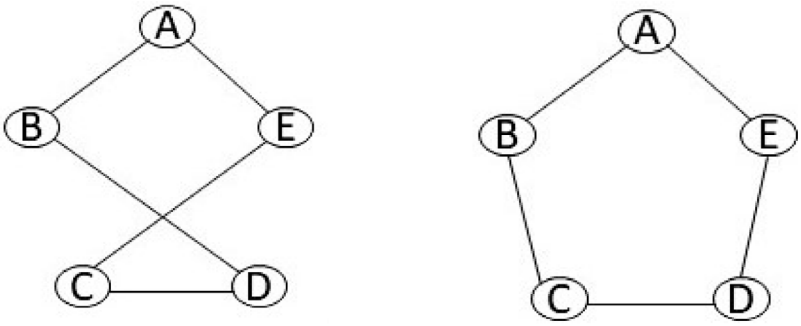
$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \tag{11}$$

where X_α, X_β and X_δ are the approximated positions of alpha, beta and delta wolves. Eq. (11) represents the updated position of wolf.

In GWO, exploration and exploitation are balanced by the parameter c and γ.

Algorithm 1 The Classical GWO

```
Initialize population of wolves x_i(i = 1, 2, ....l)
Initialize the parameters r, γ and c
calculate the fitness of each wolf
Choose
        α= best fit wolf of pack
        β = second best wolf
        δ = third best wolf
Initialize iteration t=0
while t < max. number of iterations do
  for each wolf do
    update the position of current wolf using Eq. (11)
  end for
  update the values of r, γ and c
  update best wolves α, β and δ
  t = t + 1
end while
return the best wolf α
```



(a) Original route (b) New route

Fig. 1. The routes for salesman before and after 2-opt process.

5. The proposed discrete grey wolf optimizer (D-GWO)

In this section, we discussed the proposed D-GWO for solving the travelling salesman problem. Before that, 2-opt algorithm is explained, which plays a vital role in D-GWO.

5.1. 2-opt

The 2-opt algorithm was proposed by Croes [39], and this belongs to the family of local search algorithms. This search techniques help in speeding up the convergence and getting accurate solutions. In the last decades, the 2-opt algorithm has been widely used by the researchers to solve various routing problem [40,41]. The idea behind 2-opt is first to select two edges randomly and then eliminate them from the original route, then reconnect these edges in order to get a new route for salesman with shorter travelling cost. As shown in Fig. 1, the original route was $A-B-D-C-E-A$, and after reconnecting edges $B-D$ and $C-E$ the route transformed into is $A-B-C-D-E-A$. Thus a shorter route is obtained using the 2-opt. The algorithm continues to build on the improved paths by repeating the steps.

Algorithm 2 Pseudo code 2-opt

```
Input define a route R =< A_1, A_2, ....A_k >
Output a new shorter route R =< B_1, B_2, ....B_k >
for i= 1 to n − 1 do
  for j= i + 1 to n do
    dist1= total length of two edges
    dist2= total length of edges after re-connection
    if dist1 > dist2 then
      swap indices of edges
    end if
  end for
end for
```

5.2. Discrete GWO for TSP

As briefly discussed in previous sections, GWO was originally proposed for continuous optimization problems. However, TSP is a combinatorial optimization problem, so GWO cannot be used directly to solve TSP; therefore some necessary modification has been done in original GWO to address TSP. In the proposed D-GWO, 2-opt approach and hamming distance concept are used in order to modify original GWO for TSP. Each grey wolf in the swarm represents a feasible solution for TSP. Additionally, finding a path with minimal cost for the salesman is considered as an objective for this algorithm.

To reduce the complexity of the algorithm and for ease in implementation, some of the parameters of original GWO such as r, γ and c have not been incorporated in the discrete version of GWO. As the parameters of classic GWO cannot be used in the same way in discrete version, so the difference vectors d_α, d_β and d_δ; and the position updating equation are modified. However, the original GWO has adapted accurately, as much as possible.

In the proposed algorithm, the difference vectors are calculated in the following way:

$$\begin{aligned} d_\alpha &= random[1, hd(X_i, X_\alpha)] \\ d_\beta &= random[1, hd(X_i, X_\beta)] \\ d_\delta &= random[1, hd(X_i, X_\delta)] \end{aligned} \tag{12}$$

Here, Firstly the difference between the current wolf and the α wolf is calculated, which is the hamming distance(hd) between the wolf, and then the difference vector d_α is obtained which is a random number between 1 and the calculated hamming distance. In the similar way, the difference vector d_β and d_β are calculated.

Similar to original GWO the position of wolves get updated using d_α, d_β and d_δ. In the proposed algorithm, the concept of 2-opt is also introduced. During each iteration, the wolf i performs a d_α number of 2-opt execution, and similarly d_β and d_δ. Therefore by following this approximation, each wolf up-date their positions as follows:

$$\begin{aligned} X_{i1} &\leftarrow 2-opt(X_i, d_\alpha) \\ X_{i2} &\leftarrow 2-opt(X_i, d_\beta) \\ X_{i3} &\leftarrow 2-opt(X_i, d_\delta) \end{aligned} \tag{13}$$

Furthermore, regarding the generation of new solution from Eq. (13), the original Eq. (11) is slightly changed, here best of X_{i1}, X_{i2}, X_{i3} is taken as the new position of wolf. The flow chart of the proposed algorithm D-GWO is presented in Fig. 2.

Algorithm 3 Pseudo code of proposed D-GWO

```
Initialize population of wolves x_i(i = 1, 2, ....l)
calculate the fitness of each wolf
Choose
        α= best fit wolf of pack
        β = second best wolf
        δ = third best wolf
Initialize iteration t=0
while t < max. number of iterations do
  for each wolf do
    update the position of current wolf using Eq. (12) and
    Eq. (13)
  end for
  update best wolves α, β and δ
  t = t + 1
end while
return the best wolf α
```

Table 1
Experimental parameter configuration of ESA, GA and IDGA.

ESA		GA		IDGA	
Population size	50	Population size	50	Population size	50
Successor functions	2-opt & insertion	Crossover function	OX	Crossover function	OB & OBX
Temperature	$-sup\Delta f/ln(p)$	Mutation functions	Insertion &3-opt	Mutation functions	Insertion & 3-opt
Cooling constant	0.95	Cross. prob.	0.95	Cross. prob.	[0.95,0.9,0.8,0.75]
		Mut. prob	0.25	Mut. prob	[0.05,0.1,0.2,0.25]
		Selection function	Binary tournament	Selection function	Binary tournament
		Survior function	Binary tournament	Survior function	Binary tournament
				Migration strat.	Best–Replace–Worst

Table 2
Comparison of best-so-far solutions found by the proposed D-GWO algorithm with optimal from TSPLIB and other algorithms.

S.N.	Instance	BK	D-GWO				IBA				DFA			
			Best	Avg	Diff	pdBest (%)	Best	Avg	Diff	pdBest (%)	Best	Avg	Diff	pdBest (%)
1	dantzig42	699	**679**	680	**−20**	−2.86	NA	NA	NA	NA	NA	NA	NA	NA
2	pr107	44 303	**44 301**	44 685	**−2**	−0.004	44 303	44 793.8	0	0	44 303	44 790	0	0
3	pr144	58 537	**58 535**	58 601	**−2**	−0.003	58 537	58 876.2	0	0	58 546	58 993	9	0.015

Here, BK represents the best known so far taken from TSPLIB, Best stands for the best result of algorithms out of 20 runs, Avg means the average of 20 runs, Diff is the difference between Best and BK and pdBest (%) is the percentage deviation from the BK.

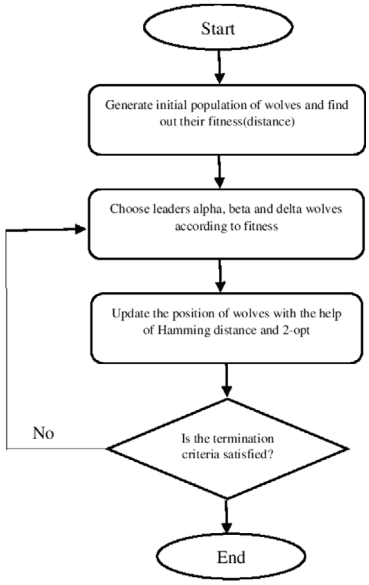


Fig. 2. Flow chart of D-GWO.

6. Simulation and result

In this section, the experimentation carried out in this study are discussed in detail. The TSP data used for experimentation is obtained from MP-TESTDATA (http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html), which consist of: (1) The symmetric travelling salesman problem TSP instances and (2) their best known solutions.

First of all, in Section 6.1 the experimental configuration considered for experiments is detailed. The experiments are conducted in two sets, firstly performance of proposed algorithm D-GWO is compared with IBA and DFA over 3(three) TSP instances. These are the instances for which D-GWO perform exceptionally well. The second experiment is carried out to compare the quality of solution of D-GWO with IBA, DFA, DICA and three classical algorithms GA, ESA and IDGA over 10 TSP instances. Both the experiments are discussed in detail, in Section 6.2. Finally, statistical analysis is carried out in Section 6.3.

6.1. Experimental configuration

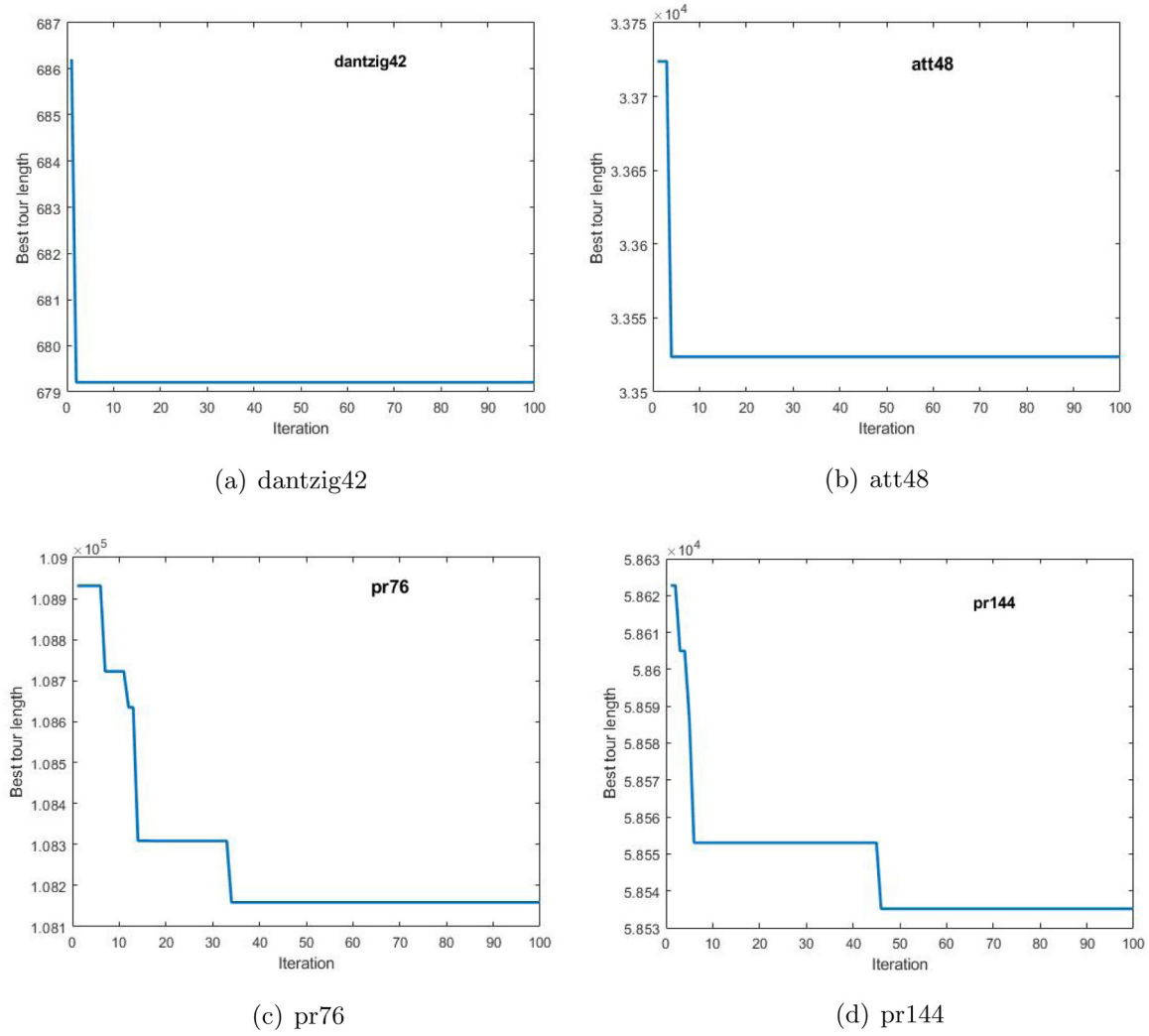
The proposed D-GWO has been tested over 17 TSP datasets. These TSP instances ranging cities from 42 to 1002. For determining the optimum parameters for DGWO, the number of Wolves, Hamming distance concept, movement function (2-opt), number of iterations are analysed. The initial population of wolves is generated randomly. After the repetition of many experiments and conducting a sensitivity analysis, these parameters have been set empirically. The experiments for D-GWO are conducted on a 2.30 GHz CPU Desktop with 4 GB RAM, while implemented on software MATLAB R2019a. It is important to mention that results obtained by all the comparison techniques (IBA, DFA, DICA, GA, ESA, and IDGA) have been taken from recently published work [18]. Also, it is worth mentioning that results depend on random numbers so they can vary slightly. For statistical reliability, 20 independent runs are carried for each algorithm over each problem instance. Moreover, the number of runs is same for all the algorithms used in experiment.

For all the heuristics, the initial population has been generated with random permutation. Also, for the fair comparison, similar parameters have been used in all the algorithms including TSP benchmarks. Among the 20 runs, each run ends when there are $N + \sum_{l=1}^{N}$ generations without improvement in the best solution obtained by the algorithm, where N is the number of nodes in TSP instance. This condition is used as stopping criteria. The same stopping criteria has been used in other algorithm considered for experimentation in this study.

The parameter setup for classical approaches GA, ESA and IDGA summarized in Table 1. It can be seen that the ESA algorithm proceeds with two successor functions 2-opt and insertion; each individual chooses one of them randomly during iterations. In a similar way, IDGA also possesses two crossover functions. While for the DFA, DICA and IBA, the parameter configuration given in [14,18,42] is followed. The concept of Hamming distance has been used for movement in all approaches considered for experimentation.

In the proposed algorithm D-GWO, the initial population size of grey wolves is 50, and the movement function is 2-opt. However, some of the parameters of original GWO are neglected to reduce the complexity of the algorithm. And similar to other algorithms, Hamming distance is used for improvement in solution.

Along with the best solution and average, to check the closeness of the obtained solution with best known (BK) solution, the



(a) dantzig42 (b) att48

(c) pr76 (d) pr144

Fig. 3. Convergence curves of D-GWO.

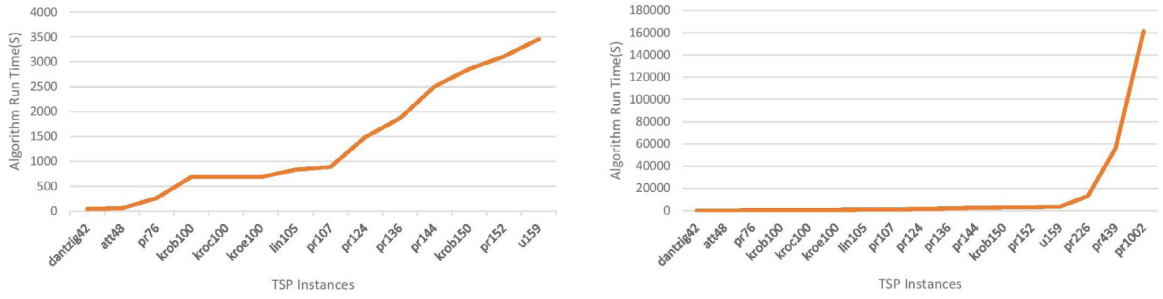


Fig. 4. Run time of proposed algorithm for different TSP instances.

percentage deviation of best solution (pdBest) is calculated as follows:

$$pdBest = \frac{(Best - BK)}{BK} \times 100 \tag{14}$$

The percentage deviation of the average solution of D-GWO is also calculated to compare the performance with other algorithms.

$$pdAvg = \frac{(Avg - BK)}{BK} \times 100 \tag{15}$$

where Best represents the best solution for each algorithm out of 20 runs under each TSP instance, Avg represents average length value for each algorithm over 20 runs under each TSP instance.

Here, run time is not considered as a parameter for comparison as different computers have been used by the researchers to run these algorithms. While Table 3 shows the time taken by the proposed algorithm over various instances (for 20 runs), it is also shown in Fig. 4.

6.2. Discussion of numerical results

Before describing the results of different algorithms and their comparison with D-GWO, a short summary of the basic principles of these algorithms is provided below.

Table 3
Results of proposed D-GWO for TSP instances.

S.N.	Instances	Optimal	Best	Average	Diff	pdBest (%)	pdAvg (%)	Run time (s)
1	dantzig42	699	679	680	−20	−2.86	−2.71	40.69
2	att48	33 522	33 523	33 600	1	0.002	0.23	60.07
3	pr76	108 159	108 159	108 900	0	0	0.68	264.23
4	krob100	22 140	22 159	22 444.6	19	0.085	1.37	689.34
5	kroc100	20 749	20 749	21 078	0	0	1.58	688.43
6	kroe100	22 068	22 131	22 410	63	0.28	1.54	685.46
7	lin105	14 379	14 382	14 520	3	0.02	0.98	685.33
8	pr107	44 303	44 301	44 685.1	−2	−0.004	0.86	830.54
9	pr124	59 030	59 030	59 390.9	0	0	0.61	888.3
10	pr136	96 772	97 826	99 310.5	1054	1.08	2.62	1486.25
11	pr144	58 537	58 535	58 600.5	−2	−0.003	0.1	1866
12	krob150	26 130	26 320	26 756.2	190	0.72	2.39	2503.2
13	pr152	73 682	73 690	74 230	8	0.01	0.74	2856.22
14	u159	42 080	42 142	42 563.3	62	0.14	1.14	3108.6
15	pr226	80 369	80 648	81 135.7	279	0.34	0.95	12 971.29
16	pr439	107 217	110 415	112 850.3	3198	2.98	5.25	56 225.8
17	pr1002	259 047	264 922	267 713.2	5875	2.26	3.34	161 240.9

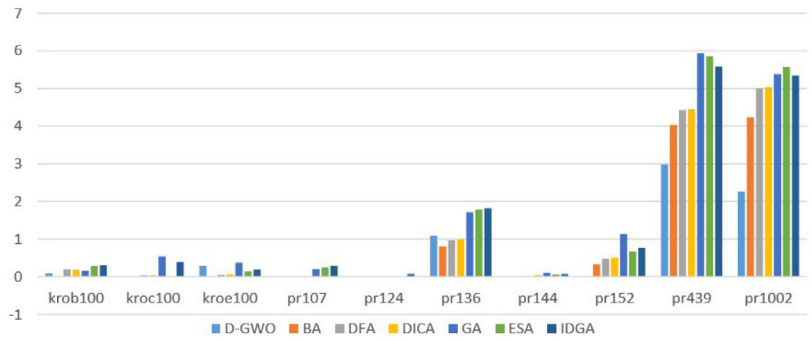


Fig. 5. Percentage deviation of best solution (pdBest) to TSP instances for different algorithms.

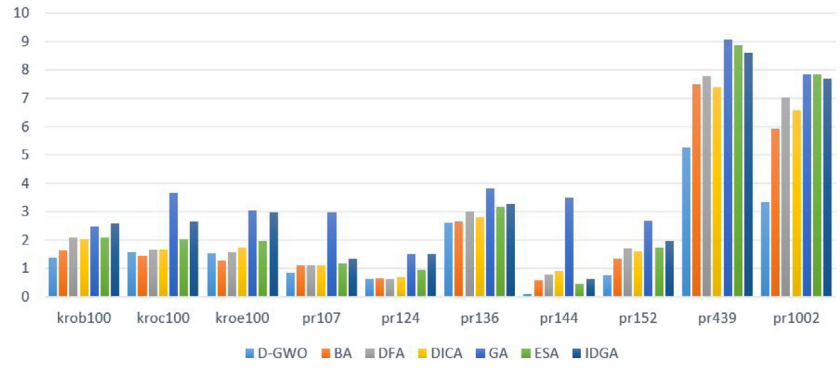


Fig. 6. Percentage deviation of average solution (pdAvg.) to TSP instances for different algorithms.

Genetic Algorithms (GAs): These are one of the most successful metaheuristics used for solving various complex optimization problems. It is easy to apply GAs over the problems and, GAs are very good performer. GAs have been applied to solve several combinatorial optimization problems and other real life application by the researchers in last decades. GA emulates the process of natural evolution and survival of fittest [43]. To overcome some of the drawbacks of original GA such as premature convergence, and also to balance exploration and exploitation, PGA (parallel GAs) were proposed. Island model also belongs to the category of PGA, which consist of multiple populations referred as IDGA [17]. In IDGA, population evolve independently and individuals get exchange occasionally. This approach is taken into account for experimentation.

Evolutionary Simulated Annealing (ESA): SA is a popular technique which has been used widely for local search. This algorithm imitates the behaviour of metal when cooled down. For the fair and rigorous comparison of results, an evolutionary version of SA is used in this study as it is a population based method roughly [16].

Bat Algorithm (BA): BA is a population based metaheuristic introduced by Xin She Yang [44], mimics the echolocation behaviour of microbats. These bats can find the prey even in the complete darkness by varying the pulse rate of emission and loudness. This behaviour has been mathematically modelled in BA. In this study, a discrete version of BA is considered for experimentation [18].

Firefly Algorithm (FA): This optimizer was proposed in [45] inspired by the flashing behaviour of fireflies. These fireflies attract

Table 4
The comparison of the experimental results of D-GWO with classical search techniques GA, ESA and IDGA.

Instance		D-GWO				GA				ESA				IDGA			
Name	Optimal	Best	Avg.	pdBest (%)	pdAvg. (%)	Best	Avg.	pdBest (%)	pdAvg. (%)	Best	Avg.	pdBest (%)	pdAvg. (%)	Best	Avg.	pdBest (%)	pdAvg. (%)
krob100	22 140	**22 159**	**22 445**	0.085	1.375	22 176	22 687.4	0.162	2.472	22 202	22 602	0.28	2.086	22 208	22 712.6	0.307	2.586
kroc100	20 749	**20 749**	**21 078**	0	1.585	20 861	21 510.4	0.539	3.669	20 749	21 170.4	0	2.03	20 830	21 298.7	0.39	2.649
kroe100	22 068	22 131	**22 410**	0.285	1.549	22 150	22 741.3	0.371	3.051	**22 099**	22 499.7	0.14	1.956	22 110	22 721.9	0.19	2.963
pr107	44 303	**44 301**	44 685.1	−0.004	0.862	44 392	**45 620**	0.2	2.971	44 413	44 821.5	0.248	1.17	44 428	44 902.5	0.282	1.353
pr124	59 030	**59 030**	**59 391**	0	0.611	**59 030**	59 910	0	1.49	**59 030**	59 593.6	0	0.954	59 072	59 912.8	0.071	1.495
pr136	96 772	**97 826**	**99 311**	1.089	2.623	98 432	100 472.4	1.715	3.823	98 499	99 858.3	1.784	3.189	98 532	99 932.7	1.818	3.266
pr144	58 537	**58 535**	**58 601**	−0.003	0.108	58 599	60 591.4	0.105	3.509	58 574	58 807.3	0.063	0.461	58 581	58 893	0.075	0.608
pr152	73 682	**73 690**	**74 230**	0.01	0.743	74 520	75 658.3	1.137	2.682	74 172	74 969.5	0.665	1.747	74 249	75 126.7	0.7699	1.96
pr439	107 217	**110 415**	**112 850.3**	2.982	5.254	113 576	116 943.4	5.93	9.071	113 497	116 706.9	5.857	8.851	113 207	116 436.1	5.586	8.598
pr1002	259 047	**264 922**	**267 713.2**	2.267	3.345	273 001	279 384.7	5.386	7.85	273 496	279 419.7	5.577	7.864	272 893	278 951.4	5.344	7.683

Table 5
The comparison of the experimental results of D-GWO with BA, DFA and DICA.

Instance		D-GWO				BA				DFA				DICA			
Name	Optimal	Best	Avg.	pdBest (%)	pdAvg. (%)	Best	Avg.	pdBest (%)	pdAvg. (%)	Best	Avg.	pdBest (%)	pdAvg. (%)	Best	Avg.	pdBest (%)	pdAvg (%)
krob100	22 140	22 159	**22 444.6**	0.085	1.375	**22 140**	22 506.4	0	1.645	22 183	22 604	0.194	2.095	22 180	22 599.7	0.18	2.034
kroc100	20 749	**20 749**	21 078	0	1.585	**20 749**	**21 050**	0	1.45	20 756	21 096.3	0.033	1.673	20 756	21 103.9	0.033	1.681
kroe100	22 068	22 131	22 410	0.285	1.549	**22 068**	**22 349.6**	0	1.27	22 079	22 413	0.0498	1.563	22 083	22 456.3	0.067	1.729
pr107	44 303	**44 301**	**44 685.1**	−0.004	0.862	44 303	44 793.8	0	1.107	44 303	44 790.4	0	1.1	44 303	44 803.3	0	1.116
pr124	59 030	**59 030**	**59 390.9**	0	0.611	**59 030**	59 412.1	0	0.647	**59 030**	59 404.3	0	0.634	**59 030**	59 436.9	0	0.684
pr136	96 772	97 826	**99 310.5**	1.089	2.623	**97 547**	99 351.2	0.8	2.665	97 716	99 683.7	0.975	3.008	97 736	99 583.7	0.996	2.823
pr144	58 537	**58 535**	**58 600.5**	−0.003	0.108	58 537	58 876.2	0	0.579	58 546	58 993.3	0.015	0.779	58 563	59 070.9	0.044	0.903
pr152	73 682	**73 690**	**74 230**	0.01	0.743	73 921	74 676.9	0.324	1.35	74 033	74 934.3	0.476	1.699	74 052	74 886.7	0.502	1.608
pr439	107 217	**110 415**	**112 850.3**	2.982	5.254	111 538	115 256.4	4.03	7.498	111 967	115 558.2	4.43	7.779	111 983	115 763.1	4.445	7.382
pr1002	259 047	**264 922**	**267 713.2**	2.267	3.345	270 016	274 419.7	4.234	5.934	272 003	277 344.7	5.001	7.036	272 080	277 308.1	5.031	6.585

each other, and this attractiveness is proportional to the brightness of flies. The less bright flies get attracted towards brighter. Since its proposal, it has been applied to a variety of optimization proposal. Due to some similar characteristics, a discrete version of FA is used for comparison of results with D-GWO [14].
Imperialist Competitive Algorithm (ICA): This algorithm is based on the concept of imperialism [19]. Like other evolutionary algorithms, ICA is also a gradient free method. ICA considered as the counterpart of GA by the researchers. The ICA divides the whole population into empires and these empires evolve separately. Individual of the population called countries. These countries compete with each other and powerless empires disappear.

In last decades, the TSP is considered as the standard benchmark problem set to measure the performance of discrete optimization algorithms. Here also, TSP instances are used for performance analysis of D-GWO. The main motive of this study is to show that the GWO can be applied successfully over real life routing problems, also it can be implemented easily. With this aim, the performance of D-GWO algorithm is compared with well known algorithms of literature such as GA, SA and other recently proposed metaheuristics. The results clearly show that the proposed algorithm outperforms over these algorithms.

The proposed algorithm D-GWO is tested over 17 TSP instance which has nodes 42 to 1002. The results are shown in Table 3 in the form of best solution (out of 20 runs), average solution in 20 runs, difference with optimal and percentage deviation with optimal known so far for each TSP problem, and compared with known optimal of problems so far.

The extraordinary performance of D-GWO over some TSP instance can be seen in Table 2, when compared with other metaheuristic algorithms which have some common characteristics with D-GWO. From Table 2, it can be concluded that D-GWO not only outperform over other algorithms, also the percentage deviation (pdbest) is less than 0 when compared with best known (BK) so far.

The performance of D-GWO is compared with three classical algorithm GA, ESA and IDGA over 10 TSP instances, and three recently proposed algorithm having some common characteristics with D-GWO. For fair comparison, as much as possible, similar parameters are taken into account for the implementation of all the algorithms. Table 4 shows the optimization results obtained by GA, ESA, IDGA and their comparison with D-GWO in terms of best solution, average, percentage deviation from the best and the average. The best result across all the approaches highlighted in bold. It can be extracted from Table 4 that the D-GWO clearly dominates the GA, ESA and IDGA in terms of best solution as well as in case of average solution for most of the TSP instances. Except for the instance kroe100, D-GWO provides the finest result compare to GA, ESA and IDGA in best results of algorithms. While in case of average solution, D-GWO outperforms over other algorithms for all the instances considered for experimentation except for the instance pr107. This dominant behaviour of D-GWO is verified by the statistical test in the next section.

On the other hand, the result obtained by BA, DFA and DICA are compared with the results of the D-GWO in Table 5. As mentioned previously, similar parameters are used in these algorithms too for an unbiased conclusion. The reason behind consideration of these algorithms is that they are recently proposed and performed very well for TSP. From Table 5, it can be concluded that D-GWO outperforms over other algorithms in case of best results for all the instances except for instances krob100, kroe100 and pr136. While at the same time for average results D-GWO performs worse only for kroc100 and kroe100, outperformed for rest instances.

In Figs. 5 and 6, percentage deviation of best solution and the average solution to the best known so far of TSP instance for different algorithms are compared. It can be concluded from figures that the proposed D-GWO algorithm obtain smaller deviations compare to other algorithms for most of the instances used in experimentation. Fig. 3, demonstrate the simulation tests of a number of TSP problem instances of different scales which were benchmarked to test the effectiveness of the D-GWO algorithm. In these curve for clarity only 100 iterations have been shown, also there was no significant change after 100 iterations. we are only able to show convergence graphs for the proposed algorithm because the other compared algorithms' data were taken from literature and this information was unavailable.

Table 6
Average ranking by Friedman's non-parametric test.

TSP	
Algorithms	Ranking
D-GWO	1.200
BA	2.200
DFA	3.600
DICA	3.600
GA	6.700
ESA	4.900
IDGA	5.800

Table 7
P-value (adjusted and unadjusted) obtained by Holm's post hoc procedure for TSP.

Algorithm	Adjusted p-value	Unadjusted P-value
GA	0.00014	0.000023
IDGA	0.002434	0.000486
ESA	0.015755	0.003938
DICA	0.014971	0.00499
DFA	0.0196	0.0098
BA	0.042028	0.042028

6.3. Statistical analysis

To obtain fair and rigorous conclusion, two different tests have been performed over the obtained results by following the guidelines of [46]. Firstly, Friedman's non-parametric test for multiple comparisons has been conducted to verify whether there is any significant difference between the approaches considered for experimentation or not. The test has been conducted at 95% level of confidence and the result obtained by the test in the form of mean ranking are shown in Table 6. Here, the lower the mean rank, the better the performance. The calculated Friedman statistics with 6 degrees of freedom (distributed according to χ^2) is 49.73. While, with the 95% confidence interval, the critical value (considering χ^2 distribution) is 12.59. Clearly $49.73 < 12.59$ and the observed p-value is 0.0001 (< 0.05), both supports the fact that there is significant difference between the results of different approaches. Thus D-GWO can be considered best among all these approaches, as having the lowest rank.

An additional test is conducted to check the statistical significance, which is Holm's test, considering D-GWO as control algorithm. The adjusted and unadjusted p-values obtained by Holm's post hoc analysis are shown in Table 7. It can be seen that all the p-values are less than 0.05, so we can conclude that D-GWO is significantly better for TSP at 95% confidence level.

7. Conclusion and future direction

In this study, a novel discrete version of GWO has been proposed as a new approach to solve the Travelling Salesman Problem. The GWO algorithm which is inspired by the hunting behaviour of grey wolves, was initially proposed to solve continuous

optimization problems. The original GWO is slightly modified in order to use for discrete problems. To prove that D-GWO is a promising approximation method to solve TSP, 17 TSP instances are tested by this algorithm and its performance is compared with other state of the art algorithms along with 10 TSP instances. The simulation results support the fact that D-GWO performed well for TSP and can be applied to other complex discrete problems. The D-GWO can be used to solve other large scale routing problems. The computational efficiency of the D-GWO algorithm decrease slightly in terms of runtime as the complexity of problem increases, which can be improved by adding some other parameters in D-GWO in future. Also, some of the parameter of classical GWO has not been incorporated, so in future work it is also possible to further improve the algorithm with these parameters. As future work, the author intends for further improvisation in algorithm and use it to solve real life complex problems.

CRediT authorship contribution statement

Karuna Panwar: Conceptualization, Programming, Validation, Writing- original draft. **Kusum Deep:** Supervision, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] E.L. Lawler, The traveling salesman problem: a guided tour of combinatorial optimization, in: Wiley-Interscience Series in Discrete Mathematics, 1985.

[2] G. Laporte, Y. Nobert, A cutting planes algorithm for the m-salesmen problem, J. Oper. Res. Soc. 31 (11) (1980) 1017–1023.

[3] M. Padberg, G. Rinaldi, Optimization of a 532-city symmetric traveling salesman problem by branch and cut, Oper. Res. Lett. 6 (1) (1987) 1–7.

[4] G. Laporte, The traveling salesman problem: An overview of exact and approximate algorithms, European J. Oper. Res. 59 (2) (1992) 231–247.

[5] K.-P. Wang, L. Huang, C.-G. Zhou, W. Pang, Particle swarm optimization for traveling salesman problem, in: Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 03ex693), 3, IEEE, 2003, pp. 1583–1585.

[6] M. Akhand, S. Akter, M. Rashid, Velocity tentative particle swarm optimization to solve TSP, in: 2013 International Conference on Electrical Information and Communication Technology (EICT), IEEE, 2014, pp. 1–6.

[7] D. Karaboga, B. Gorkemli, A combinatorial artificial bee colony algorithm for traveling salesman problem, in: 2011 International Symposium on Innovations in Intelligent Systems and Applications, IEEE, 2011, pp. 50–53.

[8] W. Deng, R. Chen, B. He, Y. Liu, L. Yin, J. Guo, A novel two-stage hybrid swarm intelligence optimization algorithm and application, Soft Comput. 16 (10) (2012) 1707–1722.

[9] S.-M. Chen, C.-Y. Chien, Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques, Expert Syst. Appl. 38 (12) (2011) 14439–14450.

[10] I. Khan, M.K. Maiti, A swap sequence based artificial bee colony algorithm for traveling salesman problem, Swarm Evol. Comput. 44 (2019) 428–438.

[11] M. Akhand, S.I. Ayon, S. Shahriyar, N. Siddique, H. Adeli, Discrete spider monkey optimization for travelling salesman problem, Appl. Soft Comput. 86 (2020) 105887.

[12] M. Dorigo, L.M. Gambardella, Ant colonies for the travelling salesman problem, Biosystems 43 (2) (1997) 73–81.

[13] M. Clerc, Discrete particle swarm optimization, illustrated by the traveling salesman problem, in: New Optimization Techniques in Engineering, Springer, 2004, pp. 219–239.

[14] M. Li, J. Ma, Y. Zhang, H. Zhou, J. Liu, Firefly algorithm solving multiple traveling salesman problem, J. Comput. Theor. Nanosci. 12 (7) (2015) 1277–1281.

[15] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.

[16] P.P. Yip, Y.-H. Pao, Combinatorial optimization with use of guided evolutionary simulated annealing, IEEE Trans. Neural Netw. 6 (2) (1995) 290–295.

[17] E. Alba, J.M. Troya, A survey of parallel distributed genetic algorithms, Complexity 4 (4) (1999) 31–52.

[18] E. Osaba, X.-S. Yang, F. Diaz, P. Lopez-Garcia, R. Carballedo, An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems, Eng. Appl. Artif. Intell. 48 (2016) 59–71.

[19] E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, in: 2007 IEEE Congress on Evolutionary Computation, IEEE, 2007, pp. 4661–4667.

[20] G. Komaki, V. Kayvanfar, Grey wolf optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time, J. Comput. Sci. 8 (2015) 109–120.

[21] S. Zhang, Y. Zhou, Z. Li, W. Pan, Grey wolf optimizer for unmanned combat aerial vehicle path planning, Adv. Eng. Softw. 99 (2016) 121–136.

[22] T. Jayabarathi, T. Raghunathan, B. Adarsh, P.N. Suganthan, Economic dispatch using hybrid grey wolf optimizer, Energy 111 (2016) 630–641.

[23] L. Li, L. Sun, J. Guo, J. Qi, B. Xu, S. Li, Modified discrete grey wolf optimizer algorithm for multilevel image thresholding, Comput. Intell. Neurosci. 2017 (2017).

[24] C. Lu, L. Gao, Q. Pan, X. Li, J. Zheng, A multi-objective cellular grey wolf optimizer for hybrid flowshop scheduling problem considering noise pollution, Appl. Soft Comput. 75 (2019) 728–749.

[25] M. Abdel-Basset, D. El-Shahat, I. El-henawy, V.H.C. de Albuquerque, S. Mirjalili, A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection, Expert Syst. Appl. 139 (2020) 112824.

[26] C. Lu, L. Gao, X. Li, S. Xiao, A hybrid multi-objective grey wolf optimizer for dynamic scheduling in a real-world welding industry, Eng. Appl. Artif. Intell. 57 (2017) 61–79.

[27] S. Gupta, K. Deep, A novel random walk grey wolf optimizer, Swarm and Evolutionary computation 44 (2019) 101–112.

[28] S. Mirjalili, S. Saremi, S.M. Mirjalili, L.d.S. Coelho, Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization, Expert Syst. Appl. 47 (2016) 106–119.

[29] N. Singh, S. Singh, Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance, J. Appl. Math. 2017 (2017).

[30] A.N. Jadhav, N. Gomathi, WGC: hybridization of exponential grey wolf optimizer with whale optimization for data clustering, Alex. Eng. J. 57 (3) (2018) 1569–1584.

[31] X. Zhang, Q. Kang, J. Cheng, X. Wang, A novel hybrid algorithm based on biogeography-based optimization and grey wolf optimizer, Appl. Soft Comput. 67 (2018) 197–214.

[32] M.A. Tawhid, A.F. Ali, A hybrid grey wolf optimizer and genetic algorithm for minimizing potential energy function, Memetic Comput. 9 (4) (2017) 347–359.

[33] H. Braun, On solving travelling salesman problems by genetic algorithms, in: International Conference on Parallel Problem Solving from Nature, Springer, 1990, pp. 129–133.

[34] M. Malek, M. Guruswamy, M. Pandya, H. Owens, Serial and parallel simulated annealing and tabu search algorithms for the traveling salesman problem, Ann. Oper. Res. 21 (1) (1989) 59–84.

[35] J. Knox, Tabu search performance on the symmetric traveling salesman problem, Comput. Oper. Res. 21 (8) (1994) 867–876.

[36] Ş. Gülcü, M. Mahi, Ö.K. Baykan, H. Kodaz, A parallel cooperative hybrid method based on ant colony optimization and 3-opt algorithm for solving traveling salesman problem, Soft Comput. 22 (5) (2018) 1669–1685.

[37] M. Mahi, Ö.K. Baykan, H. Kodaz, A new hybrid method based on particle swarm optimization, ant colony optimization and 3-opt algorithms for traveling salesman problem, Appl. Soft Comput. 30 (2015) 484–490.

[38] L. Zhou, L. Ding, X. Qiang, A multi-population discrete firefly algorithm to solve TSP, in: L. Pan, G. Păun, M.J. Pérez-Jiménez, T. Song (Eds.), Bio-Inspired Computing - Theories and Applications, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 648–653.

[39] G.A. Croes, A method for solving traveling-salesman problems, Oper. Res. 6 (6) (1958) 791–812.

[40] X. Chen, Y. Zhou, Z. Tang, Q. Luo, A hybrid algorithm combining glowworm swarm optimization and complete 2-opt algorithm for spherical travelling salesman problems, Appl. Soft Comput. 58 (2017) 104–114.

[41] K. Karagul, E. Aydemir, S. Tokat, Using 2-opt based evolution strategy for travelling salesman problem, An Int. J. Optim. Control: Theor. Appl. (IJOCTA) 6 (2) (2016) 103–113.

[42] M. Yousefikhoshbakht, M. Sedighpour, New imperialist competitive algorithm to solve the travelling salesman problem, Int. J. Comput. Math. 90 (7) (2013) 1495–1505.

[43] J. Holland, An introductory analysis with applications to biology, control, and artificial intelligence, in: Adaptation in Natural and Artificial Systems, First edn., The University of Michigan, USA, 1975.
[44] X.-S. Yang, A new metaheuristic bat-inspired algorithm, in: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Springer, 2010, pp. 65–74.
[45] X.-S. Yang, Firefly algorithms for multimodal optimization, in: International Symposium on Stochastic Algorithms, Springer, 2009, pp. 169–178.
[46] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.