Analysis of a Hybrid Whale Optimization Algorithm for Traveling Salesman Problem

Mehmet Fatih DEMİRAL [iD]

Burdur Mehmet Akif Ersoy University, Faculty of Engineering and Architecture, Department of Industrial Engineering, Burdur, Turkey

ABSTRACT

Whale Optimization Algorithm (WOA) is a fairly new algorithm developed in 2016. WOA was applied to continuous optimization problems and engineering problems in the literature. However, WOA demonstrates lower performance than others in traveling salesman problems. Therefore, in this study, an application of the hybrid algorithm (WOA+NN) has been done in the traveling salesman problem. A set of classical datasets which have cities scale ranged from 51 to 150 was used in the application. The results show that the hybrid algorithm (WOA+NN) outperforms AS (Ant system), WOA, GA, and SA for 50% of all datasets. Ant system (AS) is the second algorithm that is better than other meta-heuristics for 40% of all datasets. In addition, it was given that a detailed analysis presents the number of best, worst, average solutions, standard deviation, and the average CPU time concerning meta-heuristics. The metrics stress that the hybrid algorithm (WOA+NN) demonstrates a performance rate over 50% in finding optimal solutions. AS (Ant system) is better at 40% of all optimal solutions. Finally, the hybrid algorithm solves the discrete problem in reasonable times in comparison to other algorithms for medium-scale datasets.

Keywords: Hybrid algorithm, traveling salesman problem, whale optimization algorithm

Bir Hibrid Balina Optimizasyon Algoritminin Gezgin Satıcı Problemi için Analizi

ÖZ

Balina optimizasyon algoritması (WOA) 2016 yılında geliştirilmiş olan oldukça yeni bir algoritmadır. Balina optimizasyon algoritması literatürde sürekli optimizasyon problemlerine ve mühendislik problemlerine uygulanmıştır. Buna rağmen, WOA gezgin satıcı probleminde diğer algoritmalardan daha düşük performans sergilemektedir. Bu yüzden, bu çalışmada, hibrid algoritmanın (WOA+NN) gezgin satıcı problem üzerinde bir uygulaması yapılmaktadır. Uygulamada 51-150 arasında ölçekli şehirlerden oluşan bir klasik veriseti kullanılmıştır. Sonuçlar, hibrid algoritmanın (WOA+NN), AS (Karınca sistemi), WOA, GA ve SA'dan tüm verisetlerinin %50'sinde üstün olduğunu göstermektedir. Karınca sistemi ise tüm verisetlerinin %40'ında diğer meta-sezgisellerden daha olumlu sonuç veren ikinci algoritmadır. Çalışmada, detaylı bir analiz verilerek meta-sezgisellere göre en iyi, en kötü, ortalama çözümler, standart sapma ve ortalama CPU zamanı sunulmaktadır. Metrikler, hibrid algoritmanın (WOA+NN) optimal çözümleri bulmada %50'nin üzerinde performans sergilediğini göstermektedir. Karınca sistemi (AS) ise, tüm çözümlerin %40'ında daha iyidir. Sonuç olarak, hibrid algoritma orta ölçekli verisetlerinde diğer algoritmalara kıyasla kesikli problemi kabul edilebilir zamanlarda çözmektedir.

Anahtar Kelimeler: Hibrid algoritma, gezgin satıcı problemi, balina optimizasyon algoritması

Mehmet Fatih DEMİRAL, https://orcid.org/0000-0003-0742-0633

INTRODUCTION

Combinatorial optimization is a very popular research area that includes mathematical problems, such as traveling salesman problems, assignment, allocation, scheduling, inventory, routing, timetabling, logistics, etc., also real-world optimization problems. Combinatorial problems are solvable by many techniques, such as exact, heuristic, and meta-heuristics. The problems that have only a small number of data have been solved by exact techniques in short computational times. On the other hand, heuristics and meta-heuristics resolve the relatively large size problems in reasonable times (Mavrovouniotis and Yang, 2011; Parejo et al., 2012; Lin et al., 2016; Cárdenas-Montes et al., 2018). Meta-heuristics originate from nature and bring alternative solutions to mathematical and real-world problems. Meta-heuristics are generally investigated in two logical parts: classical and modern meta-heuristics. Classical meta-heuristics are Simulated annealing (SA) (Kirkpatrick et al., 1983), Tabu search (TS) (Qinghua et al., 2015), Genetic algorithm (GA) (Alp et al., 2003), Particle swarm optimization (PSO) (Pessin et al., 2013), and Ant colony optimization (ACO) (Yin and Wang, 2006). Artificial atom algorithm (Yildirim and Karci, 2018), Bat algorithm (BA) (Yang, 2010), Camel algorithm (CA) (Ibrahim and Ali, 2016), Cuckoo search optimization (CS) (Rajabioun, 2011), Lion optimization algorithm (LOA) (Yazdani and Fariborz, 2016) and Worm optimization algorithm (Arnaout, 2014) are interesting examples of modern meta-heuristics.

Population-based meta-heuristics generally start with a random population or an initial heuristic and improve this population with the algorithm principle. Diversification and intensification are the fundamental features that try to diversify and intensify the algorithm to search for finding alternative solutions in the discrete space. Diversification means that the algorithm would hopefully find new solutions. Intensification expresses that the algorithm escapes from local-optima solutions, and finds near-optimal solutions. Meta-heuristics can be hybridized with other heuristics or be improved in new formats. Nature-inspired algorithms have been applicable in computer engineering, information technology, logistics, combinatorial problems, mathematics, and other fields of science (Kota and Jarmai, 2015; Cherkesly et al., 2016; Long et al., 2019)

The whale optimization algorithm mimics the hunting behavior of humpback whales. In the exploration phase, humpback whales search for prey randomly. In the exploitation phase, humpback whales use the bubble-net behavior of humpback whales. Thus, the bubble-net attacking method is investigated in two distinct mechanisms. They are shrinking encircling mechanism and spiral updating position. WOA is an approach that was proposed for engineering applications in previous researches. It was successfully applied to the 0-1 knapsack problems, job-shop scheduling problems, traveling salesman problem, and discrete optimization problems (Algabalawy et al., 2010; Ahmed and Kahramanlı, 2018; Abdel-Basset et al., 2019; Hussein et al., 2019; Jiang et al., 2019; Luan et al., 2019). Although hybrid whale optimization algorithms have engineering applications in various fields, here includes a new application of WOA (WOA+NN) on TSP in this study. In this study, the WOA+NN is carried out to solve the traveling salesman problem (TSP). The traveling salesman problem is a challenging benchmark problem for evaluating the performance of the optimization algorithms. The discrete problem optimizes the tour length, visiting all cities exactly once and returning to the initial city (Hoffman et al., 2013).

The rest of the paper is organized as follows: In Sect. 2, the discrete problem (TSP) is clearly explained. The HWOA is briefly described in Sect. 3. In Sect. 4, the computational results of the algorithm are given in short, and lastly, Sect. 5 includes conclusion and future work.

TRAVELING SALESMAN PROBLEM

The traveling salesman problem (TSP) is an optimization problem that is widely applicable in engineering, mathematics, and other fields of science. Scientists have been working on the problem to solve it efficiently in competitive times (Elloumi et al., 2014). Various methodologies are proposed for the traveling salesman problem in which a salesman visits all the cities at once and comes back to the home at an optimal distance, optimal time, optimal budget, or other objectives. TSP is generally analyzed in two types that are defined by symmetric and asymmetric distance matrices (Osaba et al., 2016). In symmetric type (s-TSP), the distance equality $\left(d_{ij} = d_{ji}\right)$ is valid for all points. In asymmetric type (a-TSP), the distance inequality $\left(d_{ij} \neq d_{ji}\right)$ is valid for at least one edge. Symmetric TSPs are solved in shorter times than asymmetric TSPs. Besides, there exist many variants of TSP, such as double TSP (d-TSP), multiple TSP (m-TSP), the traveling repairman problem, traveling purchaser problem, and vehicle routing problem. The traveling salesman problem (TSP) is non-deterministic and requires exponential CPU time during the optimization process. If large data size is used, the number of possible solutions increases and it will be impossible to

find optimal results in a polynomial time. In literature, there exist various exact, heuristic, meta-heuristic, improved, and hybrid methods to solve the variants of TSPs in reasonable times (Lin et al., 2003; Wei et al., 2014; Hatamlou, 2018).

To define TSP in a short form, the problem can be described as: *N* is the set of m cities, *E* is the set of the edges, and $D_{ij} = \left(d_{ij}\right)$ is the distance matrix that presents the Euclidean distance between cities *i* and *j*. $V_{ij} = \{n_1, n_2, \ldots, n_m, n_1\}$ is the permutation of the constructed tours. n_1 represents the first node (vertex); n_m represents the last node (vertex) of all the permutations. Then, the mathematical model of the combinatorial problem is briefly given in Eq.1.

$$Min. \sum_{i=1}^{m-1}\left(d_{n_i, n_{i+1}}\right) + d_{n_m, n_1} \tag{1}$$

The Euclidean distance is carried out to calculate the distance between nodes (vertices) using Eq.2.

$$d_{i,j} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \tag{2}$$

HYBRID WHALE OPTIMIZATION ALGORITHM

The whale optimization algorithm (WOA) is one of the interesting algorithms in the literature. It is based on the hunting behavior of humpback whales. In the whale optimization algorithm, the humpback whales (whale population) are searching for the best positions and looking for the candidate preys in different ways (Mirjalili and Lewis, 2016). Therefore, they try to converge the near-optimal positions from random locations in the exploration phase using Eq. 3 and Eq.4.

$$\vec{D} = \left|\vec{C} . \vec{X}^* (t) - \vec{X}(t)\right| \tag{3}$$

$$\vec{X}(t + 1) = \vec{X}^* (t) - \vec{A} . \vec{D} \tag{4}$$

where t declares the iteration number, $\vec{A}$ and $\vec{C}$ are coefficient vectors. The vectors $\vec{A}$ and $\vec{C}$ are defined in the following using Eq. 5 and Eq. 6.

$$\vec{A} = 2 \vec{a} . \vec{r} - \vec{a} \tag{5}$$

$$\vec{C} = 2 . \vec{r} \tag{6}$$

where a is decreased from 2 to 0 depending on the iteration number. $\vec{r}$ is a random vector defined in [0,1]. In the exploitation phase, the bubble-net attacking method is achieved by two mechanisms using the mathematical model described in Eq. 7.

$$\vec{X} = \begin{cases} \vec{X}^* (t) - \vec{A} . \vec{D} & if \ p < 0.5, \left|\vec{A}\right| < 1 \\ \vec{X}_{rand} - \vec{A} . \vec{D} & if \ p < 0.5, \left|\vec{A}\right| \geq 1 \\ \vec{D}' . e^{bl} . \cos(2\pi l) + \vec{X}^* (t) & if \ p \geq 0.5 \end{cases} \tag{7}$$

The vectors $\vec{D}$ and $\vec{D}'$ are defined according to the expressions using Eq. 8.

$$\vec{D}, \vec{D}' = \begin{cases} \left|\vec{C} . \vec{X}^* - \vec{X}(t)\right| & if \ p < 0.5, \left|\vec{A}\right| < 1 \\ \left|\vec{C} . \vec{X}_{rand} - \vec{X}(t)\right| & if \ p < 0.5, \left|\vec{A}\right| \geq 1 \\ \left|\vec{X}^* (t) - \vec{X}(t)\right| & if \ p \geq 0.5 \end{cases} \tag{8}$$

where p is a random number in [0,1].

In discrete optimization as well as traveling salesman problem, the pure whale optimization algorithm does not give well-quality solutions by using Eq.3-8. Therefore, it is a need for improvement with a logical heuristic (Nearest neighbor algorithm-NN). Here, the k=1 is taken for the TSP application (k-NN) because of the most common use and sufficient optimal results for the optimization. In addition, the further analysis will be needed for the optimal "k" in the TSP application for these datasets. The comparison and results are very stunning when all the algorithms are compared with the WOA+NN for medium-scale traveling salesman problems.

In the light of Eq. 3-8, the pseudo-code of the WOA+NN with a dimensional space is shown in Figure 1 (Bozorgi and Yazdani, 2019).

```
1. Initialize the population with NN algorithm, Xᵢ (i = 1, 2, . . ., N)
2. Compute the objective values of candidates in Xᵢ
3. X* = the best candidate solution
4. while (Counter < Maximum Iteration Number)
5. for i = 1 to N do
6. Update a, A, C, l and P;
7. for j = 1 to dim. of solution space do
8. if P < 0.5 then
9. if (|A| < 1)
10. D = | C. X* - Xᵢ |
11. Xᵢ(j) = X*(j) - A.D
12. else if (|A| >= 1)
13. Choose a random individual Xrand
14. D = | C. Xrand - Xᵢ |
15. Xᵢ(j) = Xrand(j) - A.D
16. end if
17. else if P >= 0.5 then
18. D' = |X* - Xi |
19. Xᵢ(j) = D' . eᵇˡ. cos(2pi*l) + X*
20. end if
21. end for j
22. Compute the objective value for Xᵢ
23. end for i
24. end while
```

Figure 1. Pseudo-code of the hybrid algorithm (WOA+NN)

The new solutions are produced by the multiple neighborhoods during the optimization process of the algorithm. The logical operators generate candidate solutions (Szeto et al., 2011; Halim and Ismail, 2019; Demiral and Işık, 2020). In this study, two operators; swap and reverse are relatively intensified operators than insert and swap_reverse operators. Thus, insert and swap_reverse cause diversification in discrete solution space. Each operator is chosen and applied sequentially in each step. As literature declares, the use of multiple structures can be convergent and search feasible regions of the discrete space (Szeto et al., 2011; Halim and Ismail, 2019). To sum up, instead of using a single structure, it is expected to give optimal solutions when multiple structures are used. Then, the use of the best one is defined by the minimal result of the four neighborhoods using Eq. 9.

$$MultiOps(x) = \min\begin{pmatrix} swap(x), \ insert(x), \\ reverse(x), \ swap\_reverse(x) \end{pmatrix} \quad (9)$$

The hybrid algorithm (WOA+NN) converges to the optimal solution in short iteration numbers. That shows the use of combined structures is a robust, clear, and intelligent approach: random swaps (swap), random insertions (insert), reversing a subsequence (reverse), and random swaps of reversed subsequences (swap_reverse) at 200-3000 iterations.

COMPUTATIONAL RESULTS

The ten datasets ranged from 51 to 150 cities were selected from the TSPLIB library in the implementation. In this section, all the experiments were run on Intel® Core™ i7 3520-M CPU 2.9 GHz speed with 8 GB RAM by using Matlab. The algorithms which are WOA+NN, AS, WOA, GA, and SA are compared to demonstrate the performance of the WOA+NN. All the algorithms were run 10 times independently for optimal parameters and 200-3000 iterations for each run. The fundamental parameters are used in the application. In the SA algorithm, initial temperature (T_0 =40000), cooling rate (r=0.80), and the iteration limit for temperature change (L=10-30) are sufficient for optimization. In GA, the crossover rate is 0.80, the mutation rate is 0.02. In WOA, the dimension of space (dim=10), a=2-(2*t/T_{max}) is linearly decreasing function in both exploration and exploitation phases, random number for spiral updating position (l=[-1,1]), logarithmic spiral shape constant (b=0.1), coefficient vectors for updating position of whales; C=$(r+0.9)^2$ and A=a*(C-1) are chosen as optimal parameters. In AS (Ant System), # of ants=20, alfa=1, beta=5, evaporation rate ($\rho = 0.7$), Initial-Feremon=25.The population size is set to 100 for all the population-based meta-heuristics (GA, WOA, and WOA+NN).

Table 1 shows the experimental results and comparison between WOA+NN, AS, WOA, GA, and SA. In this table, the results are given as best, worst, average solution, standard deviation, and CPU Time.

As inferred from Table 1, it can be observed that the quality of the hybrid algorithm (WOA+NN) solutions is better compared to AS, WOA, GA, and SA for 50% of all datasets. Ant system (AS) is the alternative algorithm that outperforms other meta-heuristics for 40% of all datasets. Genetic algorithm (GA) is the optimal algorithm for only one dataset; pr76. Besides, in Table 2, the hybrid algorithm finds 21 optimal, AS finds 16 optimal, GA finds 3 optimal, SA finds never optimal solutions among 40 best results. In summary, Table 2 shows that the hybrid algorithm (WOA+NN) outperforms AS, WOA, GA, and SA for 53% of all optimal solutions.

Table 1. Computational results of algorithms on the medium-scale TSP instances

TSP	Measure	SA	GA	WOA	AS	WOA+NN
eil51	Best	502.67	463.35	526.99	469.12	460.51
(426)	Worst	581.77	504.67	596.95	513.96	492.33
	Avg	537.59	485.75	562.44	487.57	482.91
	Std.	23.09	11.93	25.8	16.03	10.08
	Time	0.28	59.62	16.5	44.33	4.49
	Number	1000	1000	1000	200	200
berlin52	Best	8529.02	8083.51	9136.25	8164.06	7868.66
(7542)	Worst	10472.4	8921.19	11416.3	8171.3	8172.41
	Avg	9636.47	8458.15	10187.6	8168.4	7971.96
	Std.	624.89	254.06	798.26	3.74	94.24
	Time	0.34	59.38	18.03	44.99	4.84
	Number	1000	1000	1000	200	200
st70	Best	934.94	908.07	1029.02	777.83	766.08
(675)	Worst	1106.4	1115.69	1284.7	822.07	803.56
	Avg	1020.14	1019.81	1194.75	799.19	789.59
	Std.	47.94	66.29	69.95	15.3	11.25
	Time	0.26	91.8	18.36	78.8	5.59
	Number	1000	1000	1000	200	200
eil76	Best	660.46	604.61	686.31	587.62	619.9
(538)	Worst	759.58	678.61	825.54	609.25	632.93
	Avg	706.59	650.45	773.01	596.58	627.29
	Std.	32.48	25.24	37.15	6.25	4.59
	Time	0.58	171.38	38.76	143.3	6.77
	Number	2000	2000	2000	300	300
pr76 (108159)	Best	137238	119219	140509	124963	132654
	Worst	153429	131583	200482	132838	137038
	Avg	144165	127009	167946	128793	134651
	Std.	4741.39	4392.13	18472.4	3075.36	1463.27
	Time	0.62	107.9	37.56	139.02	6.57
	Number	2000	2000	2000	300	300
rat99	Best	1880.12	1409.12	2091.99	1382.35	1430.4
(1211)	Worst	2155.03	1602.52	2597.6	1458.73	1485.09
	Avg	1977.09	1488.31	2245.39	1401.78	1462.67
	Std.	109.4	67.6	141.8	31.69	16.96
	Time	0.57	188.42	54.51	306.95	6.4
	Number	2000	2000	2000	300	300
kroa100	Best	30961.3	30118	35199.2	25379.4	24390.8
(21282)	Worst	37964.6	32781.3	43419.1	26843.7	25978.8
	Avg	34614.3	31224	38811.5	26316.5	25306.2
	Std.	2072.4	1036.82	2303.78	398.22	507.05
	Time	1.09	286.78	74.54	453.63	11.3
	Number	3000	3000	3000	500	500
eil101	Best	820.82	778.3	937.84	739.04	739.3
(629)	Worst	1018.93	864.29	1083.41	761.54	766.78
	Avg	921.77	822.29	1014.82	746.15	753.63
	Std.	57.98	33.16	49.92	9.1	9.83
	Time	0.98	288.73	74.12	476.5	11.61
	Number	3000	3000	3000	500	500
bier127	Best	164042	157368	192651	129830	129759
(118282)	Worst	190747	178809	226815	130775	133354
	Avg	180073	166396	212060	130033	131370
	Std.	7342.17	6514.47	10099	264.18	1248.78
	Time	0.99	358.33	79.82	676.03	13.45
	Number	3000	3000	3000	500	500
kroa150	Best	46917.8	47772.8	61472.9	31929.9	31798.3
(26524)	Worst	57732.1	53233.7	70764.4	32291.8	32064.9
	Avg	52002.6	50861.8	65470.4	32028.5	31948.7
	Std.	3694.1	1670.16	2448.44	131.05	86.16
	Time	1.25	460.57	81.99	920.3	14.46
	Number	3000	3000	3000	500	500

Analysis of a Hybrid Whale Optimization Algorithm for Traveling Salesman Problem
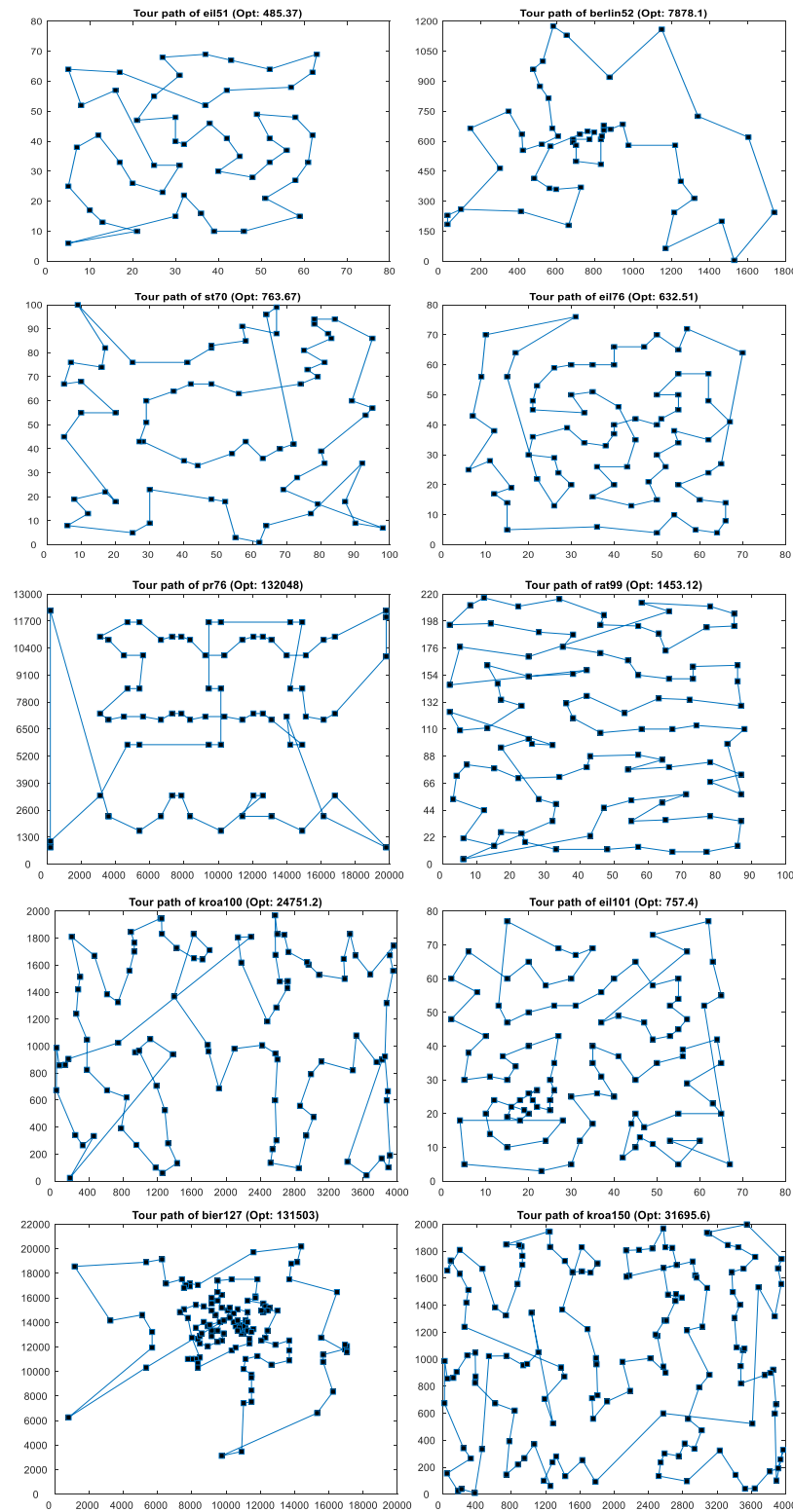
Figure 2. A set of optimal results found by the hybrid algorithm (WOA+NN)

Table 2. The #of optimal solutions and average CPU time

Alg.	Best	Worst	Avg.	Std.	Avg. Time
WOA+NN	6	4	5	6	8.55
AS	3	5	4	4	328.39
WOA	0	0	0	0	49.42
GA	1	1	1	0	207.29
SA	0	0	0	0	0.70

In general, the experimental analysis shows that the hybrid algorithm is a reliable and certain approach for solving the traveling salesman problem. This hybrid meta-heuristic gives better results and reasonable standard deviation as compared to the test meta-heuristics. A low standard deviation specifies that the hybrid algorithm is a more reliable and certain approach to find the optimal results. Lastly, the hybrid algorithm solves the TSP problem in reasonable times in comparison to other algorithms for medium-scale datasets. Figure 2 shows a set of optimal results found by the WOA+NN on the medium-scale datasets.

CONCLUSION

In recent decades, solving discrete problems via modern meta-heuristics is a popular research area. In this paper, the hybrid algorithm is implemented to the symmetric TSP instances. To perform analysis of (WOA+NN), it has been evaluated on ten benchmark test datasets. The experimental results show that the hybrid algorithm can find better solutions compared to the whale optimization algorithm (WOA), genetic algorithm (GA), simulated annealing (SA), and ant system (AS) for 50% of all datasets and 53% of all optimal solutions. Ant system (AS) is the second algorithm that outperforms other meta-heuristics for 40% of all datasets and 40% of all optimal solutions. Genetic algorithm (GA) is the last alternative that produces worse solutions (10% of datasets, and 7% optimality) than the other two alternatives. As CPU time is considered, the hybrid algorithm (WOA+NN) is extremely fast (8.55 secs.) to find optimal solutions. In future works, the hybrid algorithm can be combined with other heuristics and meta-heuristics to increase the performance and evaluated in other combinatorial problems such as scheduling, assignment, timetabling, routing, logistics, and mathematical optimization.

REFERENCES

Abdel-Basset, M., El-Shahat, D., Sangaiah, A.K. (2019). A modified nature inspired meta-heuristic whale optimization algorithm for solving 0–1 knapsack problem. *International Journal of Machine Learning and Cybernetics,* 10: 495–514.

Ahmed, O. M. A., Kahramanlı, H. (2018). Meta-Heuristic Solution Approaches for Traveling Salesperson Problem. *International Journal of Applied Mathematics Electronics and Computers,* 6(3): 21-26.

Algabalawy, M.A., Abdelaziz, A.Y., Mekhamer, S.F., Abdel Aleem, S.H.E. (2010). Considerations on optimal design of hybrid power generation systems using whale and sine cosine optimization algorithms. *Journal of Electrical Systems and Information Technology,* 5(3): 312-325.

Alp, O., Erkut, E., Drezner, Z. (2003). An Efficient Genetic Algorithm for the P-Median Problem. *Annals of Operations Research,* 122(1-4): 21-42.

Arnaout, J.P. (2014). Worm optimization: A novel optimization algorithm inspired by C. Elegans. *Proceedings of the 2014 International Conference on Industrial Engineering and Operations Management*, Bali, Indonesia, 2499-2505.

Bozorgi, S.M., Yazdani, S. (2019). IWOA: An improved whale optimization algorithm for optimization problems. *Journal of Computational Design and Engineering*, 6(3): 243–259.

Cárdenas-Montes, M. (2018). Creating hard-to-solve instances of travelling salesman problem. *Applied Soft Computing,* 71: 268-276.

Cherkesly, M., Desaulniers, G., Irnich, S., Laporte, G. (2016). Branch-price-and cut algorithms for the pickup and delivery problem with time windows and multiple stacks. *European Journal of Operational Research,* 250: 782-793.

Demiral, M.F., Işik, A.H. (2020). Simulated annealing algorithm for a medium-sized tsp data. In: *Artificial Intelligence and Applied Mathematics in Engineering Problems.* Hemanth, D.J., Kose, U. (eds.), Springer, Cham, 457-465.

Elloumi, W., Abed, H. E., Abraham, A., Alimi, A.M. (2014). A comparative study of the improvement of performance using a PSO modified by ACO applied to TSP. *Applied Soft Computing,* 25: 234–241.

Halim, A.H., Ismail, I. (2019). Combinatorial optimization: comparison of heuristic algorithms in travelling salesman problem. *Archives of Computational Methods in Engineering,* 26: 367–380.

Hatamlou, A. (2018). Solving travelling salesman problem using black hole algorithm. *Soft Computing,* 22(24): 8167-8175.

Hoffman, K.L., Padberg, M., Rinaldi, G. (2013). Traveling salesman problem. *Encyclopedia of operations research and management science.* Kluwer Academic Publishers, Springer, Berlin.

Hussein, A.G., Hassanien, A.E., Houssein, E.H., Amin, M., Azar, A.T. (2019). New binary whale optimization algorithm for discrete optimization problems. *Engineering Optimization,* 52(6): 945-959.

Ibrahim, M.K., Ali, R.S. (2016). Novel optimization algorithm inspired by camel traveling behavior. *Iraqi Journal for Electrical and Electronic Engineering,* 12(2): 167-177.

Jiang, T., Zhang, C., Sun, Q-M. (2019). Green job shop scheduling problem with discrete whale optimization algorithm. *IEEE Access*, 7: 43153-43166.

Kirkpatrick, S., Gelatt, C., Vecchi, M. (1983). Optimization by simulated annealing. *Science,* 220(4598): 671–680.

Kota, L., Jarmai, K. (2015). Mathematical modeling of multiple tour multiple traveling salesman problem using evolutionary programming. *Applied Mathematical Modeling,* 39: 3410–3433.

Lin, W.-Y., Lee, W.-Y., Hong, T.-P. (2003). Adapting Crossover and Mutation Rates in Genetic Algorithms. *Journal of Information Science and Engineering,* 19(5): 889-903.

Lin, Y., Bian, Z., Liu, X. (2016). Developing a dynamic neighborhood structure for an adaptive hybrid simulated annealing - tabu search algorithm to solve the symmetrical traveling salesman problem. *Applied Soft Computing,* 49: 937-952.

Long, W., Wu, T., Liang, X., Xu, S. (2019). Solving high-dimensional global optimization problems using an improved sine cosine algorithm. *Expert Systems with Applications,* 123: 108-126.

Luan, F., Cai, Z., Wu, S., Liu, S.Q., He, Y. (2019). Optimizing the low-carbon flexible job shop scheduling problem with discrete whale optimization algorithm. *Mathematics*, 7(8), 688; DOI: 10.3390/math7080688

Mavrovouniotis, M., Yang, S. (2011). A memetic ant colony optimization algorithm for the dynamic travelling salesman problem. *Soft Computing,* 15(7): 1405-1425.

Mirjalili, S., Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software,* 95: 51-67.

Osaba, E., Yang, X.-S., Diaz, F., Lopez-Garcia, P., Carbelledo, R. (2016). An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems. *Engineering Applications of Artificial Intelligence,* 48: 59-71.

Parejo, J.A., Ruiz-Cortés, A., Lozano, S., and Fernandez, P. (2012). Metaheuristic optimization frameworks: a survey and benchmarking. *Soft Computing,* 16: 527–561.

Pessin, G., Sales, D.O., Dias, M.A., Klaser, R.L., Wolf, D.F., Ueyama, J., Osório, F.S., Vargas, P.A. (2013). Swarm intelligence and the quest to solve a garbage and recycling collection problem. *Soft Computing,* 17: 2311–2325.

Rajabioun, R. (2011) Cuckoo optimization algorithm. *Applied Soft Computing,* 11(8):5508–5518

Qinghua, W., Yang, W., Zhipeng, L. (2015). A tabu search based hybrid evolutionary algorithm for the max-cut problem. *Applied Soft Computing*, 34: 827-837.

Szeto, W.Y., Yongzhong, W. and Ho, S.C., (2011). An artificial bee colony algorithm for the capacitated vehicle routing problem. *European Journal of Operational Research,* 215(1): 126-135.

Wei, X. (2014). Parameters Analysis for Basic Ant Colony Optimization Algorithm in TSP. *International Journal of u- and e-Service, Science and Technology (IJUNESST)*, 7(4): 159-170.

Yang, X-S. (2010). Nature-inspired metaheuristic algorithms, 2nd edn. Luniver Press, Frome

Yazdani, M., Fariborz, J. (2016). Lion Optimization Algorithm (LOA): A nature-inspired metaheuristic algorithm. *Journal of Computational Design and Engineering,* 3(1): 24-36.

Yildirim, A.E., Karci, A. (2018). Applications of artificial atom algorithm to small-scale traveling salesman problems. *Soft Computing,* 22(22): 7619-7631.

Yin, P.Y., Wang, J.Y. (2006). Ant colony optimization for the nonlinear resource allocation problem. *Applied Mathematics and Computation,* 174(2): 1438-1453.