www.dergipark.gov.tr
ISSN:2148-3736
El-Cezerî Fen ve Mühendislik Dergisi
Cilt: 7, No: 2, 2020 (526-535)

El-Cezerî Journal of Science and Engineering
Vol: 7, No: 2, 2020 (526-535)
DOI :10.31202/ecjse.662864

ECJSE

Research Paper / Makale

A Parameters Analysis of Sine Cosine Algorithm on Travelling Salesman Problem

Mehmet Fatih DEMİRAL

Burdur Mehmet Akif Ersoy University, Faculty of Engineering and Architecture, Department of Industrial Engineering, Istiklal Campus 15030 Burdur, TURKEY. mfdemiral@mehmetakif.edu.tr

Received/Geliş: 21.12.2019 **Accepted/Kabul:** 12.03.2020

Abstract: Sine Cosine Algorithm (SCA) is a fairly new algorithm developed in 2016 by Mirjalili, likewise Black Hole Algorithm (BHA), Whale Optimization Algorithm (WOA), Artificial Atom Algorithm (A^3) and Physarum-Energy Optimization Algorithm (PEO) proposed in 2013, 2016, 2018 and 2019, respectively. Due to new ideas in SCA, a few publications have been published on SCA. SCA was applied to continuous and discrete optimization problems. Besides, there exist remarkable implementations of SCA in the field of engineering, science, and technology. In this work, a parameter analysis of SCA has been done on a classical TSP (Berlin52-CTSP) and randomly generated TSP (RTSP). To do parameter analysis, major parameters have been changed gradually. For classical TSP, symmetric data has been taken from TSPLIB (TSP Library in net). The results are given as best, mean, worst solutions, std. deviation and CPU time for CTSP and RTSP. Besides, figures and tables demonstrate the effect of parameters for solving TSP. After adequate experimentation, based on trial-and-error methodology, optimal parameters and best solutions have been found. As a result, the findings indicate that major parameters of SCA influence the performance of that algorithm significantly.

Keywords: Combinatorial Problems, Meta-heuristics, Sine Cosine Algorithm, Travelling Salesman Problem

Sinüs Kosinüs Algoritmasının Gezgin Satıcı Problemi Üzerinde Parametre Analizi

Öz: Sinüs Kosinüs Algoritması (SCA) 2016 yılında, Mirjalili tarafından geliştirilmiş ve kara delik algoritması (BHA), balina optimizasyon algoritması (WOA), yapay atom algoritması (A^3) ve physarum-enerji optimizasyon algoritması (PEO) gibi sırasıyla 2013, 2016, 2018 ve 2019 yıllarında önerilmiş olan oldukça yeni algoritmalardan biridir. SCA' daki yeni fikirlerle birlikte, SCA üzerine birkaç yayın yayımlanmıştır. SCA sürekli ve kesikli optimizasyon problemleri üzerinde uygulanmıştır. Ek olarak, SCA' nın mühendislik, bilim ve teknoloji alanında dikkate değer uygulamaları mevcuttur. Bu çalışmada, SCA'nın bir klasik gezgin satıcı problemi (Berlin52-CTSP) ve rassal olarak alınmış TSP verisetinde (RTSP) parametre analizi yapılmaktadır. Parametre analizi yapabilmek için, ana parametreler kademeli olarak değiştirilmiştir. Klasik TSP için, simetrik veri net deki TSPLIB' den alınmıştır. Sonuçlar, CTSP ve RTSP için en iyi, ortalama, kötü çözümler, standard sapma ve CPU zamanları olarak verilmektedir. Bunun yanında, şekiller ve tablolar TSP' nin çözümünde parametrelerin etkisini göstermektedir. Yeterli deneme sonucunda, deneme yanılma metodolojisi ile, optimal parametreler ve en iyi çözümler bulunmaktadır. Sonuç olarak, bulgular SCA' nın ana parametrelerinin algoritma performansı üzerinde önemli derecede etki yaptığını göstermektedir.

Anahtar kelimeler: Kombinatöryel Problemler, Metasezgiseller, Sinüs Kosinüs Algoritması, Gezgin Satıcı Problemi

1. Introduction

Optimization denotes finding near - optimal or optimal solutions due to using proper resolution methodology: exact, heuristic or meta-heuristic. Optimization needs a wide perspective because of local optima issues when solving combinatorial optimization problems [1].

How to cite this article
Demiral, M.F., "*A Parameters Analysis of Sine Cosine Algorithm on Travelling Salesman Problem*", El-Cezerî Journal of Science and Engineering, 2020, 7 (2); 526-535.

Bu makaleye atıf yapmak için
Demiral, M.F., *"Sinüs Kosinüs Algoritmasının Gezgin Satıcı Problemi Üzerinde Parametre Analizi", El-*Cezerî Fen ve Mühendislik Dergisi, 2020, 7(2); 526-535.

Although many algorithms have been developed, it is still waiting to be solved. SCA was proposed in 2016 by Mirjalili as an alternative to classical optimization meta-heuristics: Simulated Annealing (SA), Tabu Search (TS), Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) [2,3].

In recent years, there have been many meta-heuristics introduced in the literature. Hence, it becomes very popular to work on new trend meta-heuristics. There is a main classification about meta-heuristics according to its inspiration source: Swarm-intelligence based, evolutionary, physics-based, bio-inspired. Some popularly studied meta-heuristic algorithms in recent years are Cuckoo optimization algorithm (CSO), black hole algorithm (BH), sine-cosine algorithm (SCA) and heart algorithm (HA) [4,5].

Particular to SCA, it has unique parameters and meta-heuristic structure as distinct from other meta-heuristics. Some parameters should be initially defined and the rest has to be stated inside the algorithm. To balance between intensification and diversification phase of the SCA, it has to be done proper parameter tuning. In literature, there has not been adequate research upon parameter analysis of SCA. Some publications have taken parameters as ordinary, so as it is needed to make a deep research on parameters analysis of SCA when solving a particular combinatorial optimization problem. In the work, the subject is the investigation of major parameters of the SCA which was suggested in 2016 based on simulating the mathematical functions. As known, sine, and cosine mathematical functions have a character of wave function so then this kind of optimization with proper parameter tuning would hopefully give better results than other meta-heuristics [6-8].

This paper is organized as follows. Section 2 gives a brief description of the Travelling Salesman Problem. Section 3 describes the Sine Cosine Algorithm (SCA) in detail. Section 4 sufficiently presents Experimental Analysis. The last section serves the results of the study.

2. Travelling Salesman Problem

Travelling Salesman Problem (TSP) is a popular combinatorial optimization problem which needs an adequate solution (near-optimal) being solved by meta-heuristics. The problem can be defined as the minimum tour length in which a salesman travels all cities exactly once and return to home. Since the number of possible solutions for a graph with n nodes is (n-1)!/2, it requires exponentially computational time to solve TSP. The exact algorithms can only solve small and a few numbers of medium-sized problems. The rest of the problems, the larger sets of data could be solved by meta-heuristics and TSP gives near-optimal results [9-11].
As inside this paper, the Euclidean distance is applied to calculate the distance between cities using the below formula [12]:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \qquad (1)$$

As in the below, Z is the set of all nodes in the Euclidean Space. The objective of classical TSP is the sum of distances between all nodes traveled exactly once from home to return point (home). The basic formula for traveling salesman problem is expressed as follows [13]:

$$Min. \quad T_D = \sum_{i=1}^{N-1} d\left(c_{\pi(i,i+1)}\right) + d\left(c_{\pi(N,1)}\right) \qquad \forall i \in Z \qquad (2)$$

Travelling Salesman Problem has several application areas in the field of engineering, science, and technology. TSP and its variations have been applied in the main research areas [14,15]:

- Logistics and Planning of Goods
- Distribution of Resources and Goods
- Manufacturing and Production of Materials and Goods
- Routing of Buses and Vehicles
- Machine Scheduling

3. Sine Cosine Algorithm (SCA)

Sine Cosine Algorithm is a new trend population-based optimization algorithm which requires initial solutions to fluctuate outwards or towards the best solution using sine and cosine mathematical functions. As in previous meta-heuristics, the usual is the analysis of optimization into two stages: exploration versus exploitation. In the exploration phase, a high rate of randomness occurs, while a small number of changes happen in the exploitation phase [16, 17].
The updating equations for SCA are suggested for both stages [18]:

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \times \sin(r_2) \times \left| r_3 P_i^t - X_i^t \right|, & r_4 < 0.5 \\ X_i^t + r_1 \times \cos(r_2) \times \left| r_3 P_i^t - X_{ii}^t \right|, & r_4 \geq 0.5 \end{cases} \tag{3}$$

where r_4 is a random number in [0,1]. As in equation (3), the basic parameters in SCA: $r_1, r_2, r_3, and\ r_4$
The parameter r_1 is defined as follows:

$$r_1 = a - \frac{a*t}{T} \qquad T = \max .\#of\ iterations \quad \wedge \quad a = constant \tag{4}$$

On the other hand, the parameter r_2 is stated as follows:

$$r_2 = 2*\pi*rand \tag{5}$$

The third parameter r_3 is expressed as the below equation:

$$r_3 = b*rand \qquad b = constant \quad \wedge \quad b > 1 \tag{6}$$

As the above formulas denote, there exist four basic parameters in SCA: *r1, r2, r3,* and *r4.* The parameter *r1* indicates the particular region which could be in the space between the solution and objective or outside it. The parameter *r2* clarifies how far the movement should be towards or outwards the objective. The parameter *r3* adds a random weight to the objective to randomly stress (*r3*>1) or unstressed (*r3*<1) the effect of objective on the next solution. Finally, the parameter *r4* equally prefers the sine or cosine function according to Eq. (3). Therefore, the mathematical functions (Eq. 3) realize the intensification and diversification of the search space [19].

In this study, the parameters r_1 and r_3 are investigated so that dependent parameters, $a, b\ and\ T$ are initially defined. During the experimentation, the major parameters, r_1 and r_3 have been changed gradually.

In the TSP application, discrete TSP is discussed. To apply the discrete version of the problem, the objective values are taken as X_i^t solution and the best objective is the P_i^t [13].

In light of the above information, the steps of the Discrete Sine Cosine Algorithm are in Algorithm 1:

Algorithm 1. Discrete Sine Cosine Algorithm

Initialize major parameters a, b and T
Generate Initial Population
For i=1: N
  Calculate r_1, r_2, r_3 *and* R_1=rand
  Identify best solution HC_b
  If $R_1 < \text{Pr}ob$

$$HC_j^{g+1} = HC_j^g + r_1 * \sin(r_2) * \left| r_3 (HC)_i^g - HC_j^g \right|$$

Else

$$HC_j^{g+1} = HC_j^g + r_1 * \cos(r_2) * \left| r_3 (HC)_i^g - HC_j^g \right|$$

  End if
End For

Figure 1. The Steps of the Discrete Sine Cosine Algorithm [13].

4. Experimental Results

The Discrete Sine Cosine Algorithm (DSCA) for the TSP problem was developed using MatlabR2017b. All computations have been implemented in MATLAB and run on an Intel® Core™ i7 3520-M CPU 2.9 GHz speed with 8 GB RAM. The discrete SCA converged to the near-optimal solutions within an acceptable time. The coded algorithm was run 10 times independently for each set of parameters and 500-3000 iterations for each run. For comparison with other algorithms, GA, ACO System, and BH, the same computational environment is valid through implementation. However, the iteration limit is set to be 200 for the ACO system.

4.1. Parameter Setting of SCA

For Experimental Analysis, the Berlin52 city problem is taken from TSPLIB [20]. For all the experiments, the population size is fixed up to *100*. The major parameters a, b *and* T are tuned to observe the effect of them on the TSP problem Berlin52 to find optimal parameters and best-known solutions [1,6,13].

The best solution is observed as test data 1 from Experiment 1 shown in Table 1. The optimal parameters are *a=0.1, b=2 and T=1000.* The best, average and worst solutions are *7755.94, 8357.6 and 8630.71, respectively.* The std. deviations and CPU times are acceptable.

Table 1. The Simulation Results (a,b are varied; T is constant)

a	b	T	Best	Average	Worst	Std. Dev.	CPU Time
0.1	2	1000	7755.94	8357.69	8630.71	237.75	15.61
0.5	3	1000	8421.47	8802.83	9194.13	281.46	13.74
2	5	1000	8864.02	9238.4	9609.71	244.87	14.09
3	7	1000	8363.02	8925.88	9391.02	366.16	14.01
5	9	1000	8449.34	9172.41	9965.27	446.35	13.83
10	10	1000	8571.26	9067.21	9395.07	248.39	13.66

The (%) deviations of best, average and worst results from the best-known value in literature are *2.84, 10.82 and 14.44,* respectively.

$$(\%)\ Deviation = \frac{(7755.94 - 7542)}{7542} * 100 = 2.84.$$

$$(\%)\ Deviation = \frac{(8357.69 - 7542)}{7542} * 100 = 10.82.$$

$$(\%)\ Deviation = \frac{(8630.71 - 7542)}{7542} * 100 = 14.44.$$

It is inferred that when *a and b* parameters are incrementally increased, tour length is reasonably increased in Figure 2. Though the problem has a local optimum at *a=3, b=7,* it has a global optimum at *a=0.1, b=2.*

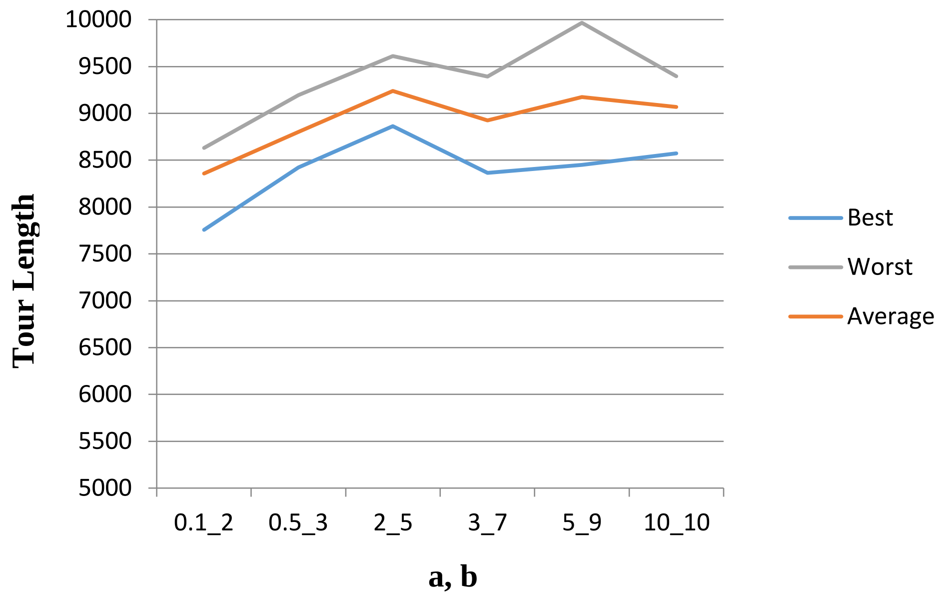

Figure 2. a, b versus Tour Length

The best solution is observed as test data 1 from Experiment 2 shown in Table 2. The optimal parameters are *a=0.1, b=2 and T=1000.* The best, average and worst solutions are *7755.94, 8357.69 and 8630.71, respectively.* The std. deviations and CPU times are acceptable. The deviations (%) from the best-known solution in literature are the same as Experiment 1.

Table 2. Simulation Results (a is varied; b and T are constant)

a	b	T	Best	Average	Worst	Std. Dev.	CPU Time
0.1	2	1000	7755. 94	8357.69	8630.71	237.75	14.26
0.2	2	1000	8333.76	8523.87	8756.52	135.85	14.19
0.3	2	1000	8786.75	8950.98	9319.44	181.62	14.25
0.5	2	1000	8805.35	9141.99	9664.22	274.38	13.71
1	2	1000	8692.22	9227.29	9769.28	399.22	13.82
3	2	1000	8305.83	9041.39	9655.34	443.03	13.76

The global optimum attained at *L=7755.94* while *b*, *T (2, 1000)* are kept constant is presented in Figure 3. When *a* is gradually increased, tour length follows a concave curve. TSP gives local optimum at *a=3, b=2* and *T=1000.*

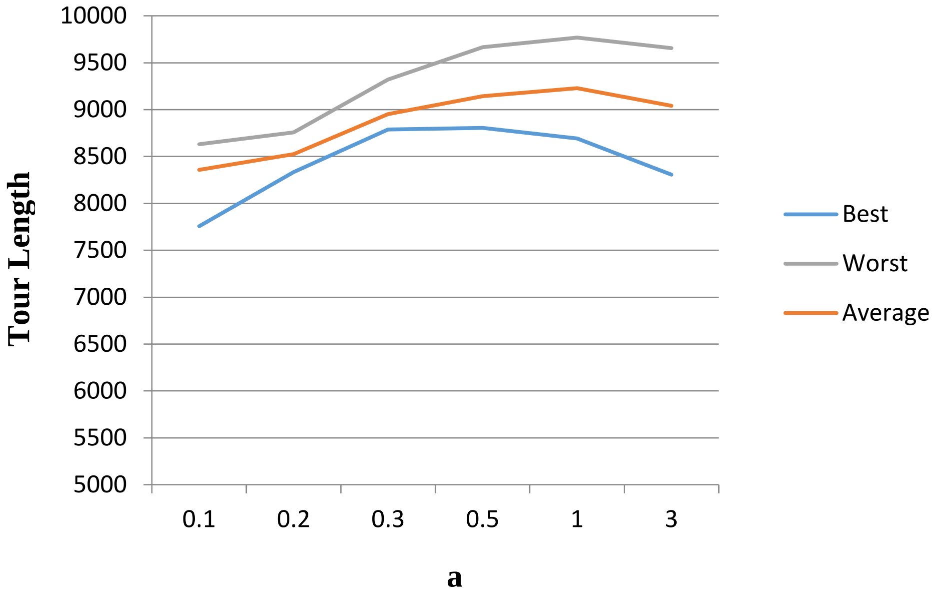

Figure 3. a versus Tour Length

The best solution is observed as test data 6 from Experiment 3 shown in Table 3. The optimal parameters are *a=0.1, b=2 and T=3000.* The best, average and worst solutions are *7549.89, 7683.26 and 7790.39.* The std. deviations and CPU times are also acceptable.

Table 3. Simulation Results (a and b are constant; T is varied)

a	b	T	Best	Average	Worst	Std. Dev.	CPU Time
0.1	2	500	9060.94	9273.1	9493.93	172.08	7.16
0.1	2	750	8424.11	8618.98	8904.02	129.44	10.48
0.1	2	1000	8219	8357.98	8569.04	109.81	13.84
0.1	2	1500	7920.81	8055.24	8212.83	91.66	20.61
0.1	2	2000	7606.54	7830.91	7955.75	123.33	28.66
0.1	2	3000	7549.89	7683.26	7790.39	86.25	41.55

The (%) deviations of best, average and worst results from the best-known value in literature are 0.1, 1.9 and 3.3.

$$(\%)\ Deviation = \frac{(7549.89 - 7542)}{7542} * 100 = 0.1.$$
$$(\%)\ Deviation = \frac{(7683.26 - 7542)}{7542} * 100 = 1.9.$$
$$(\%)\ Deviation = \frac{(7790.39 - 7542)}{7542} * 100 = 3.3.$$

The global optimum (*L=7549*.89) at *T=3000 (a, b are constant)* is presented in Figure 4. When *T* is increased, tour length is highly falling down. Simulated results show that the maximum iteration range would be acceptable between *T*∈ [1000, 3000].

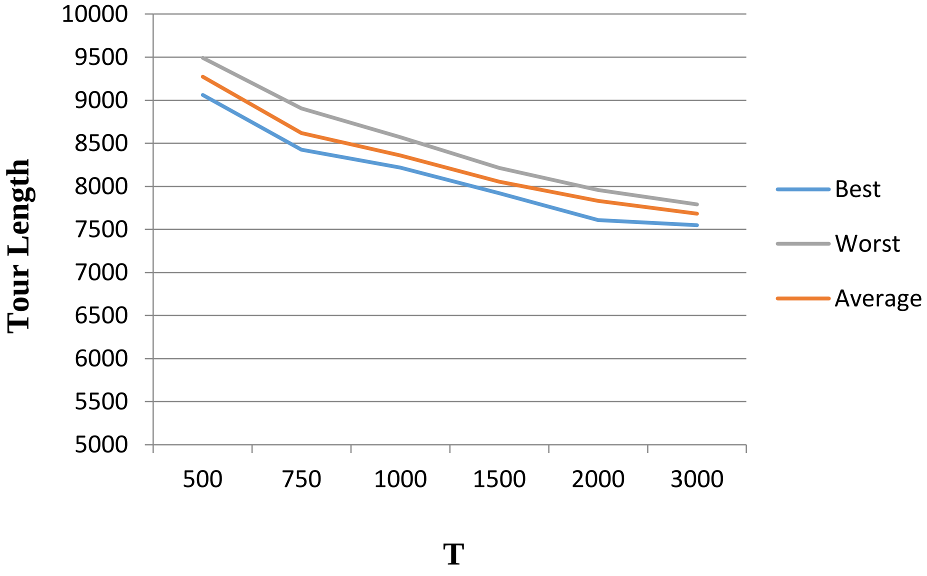

Figure 4. T versus Tour Length

The shortest route found by Sine Cosine Algorithm is demonstrated in Figure 5. The best tour length is found as *7549.89* while optimal parameters are *a=0.1, b=2, T=3000.*

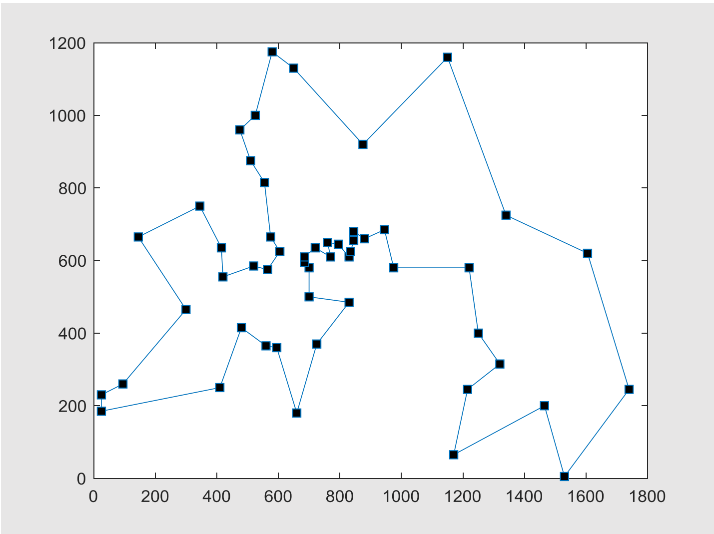

Figure 5. Best Tour Graph (L=7549.89)

4.2. Performance Comparison of Meta-heuristics

The paper has deeply explored and analyzed three important parameters' effect of *"a"* inside of r_1 which shows the movement direction, "b", and "T" inside of r_3 which denote the effect of objective on solution and maximum number of iterations [1,13]. As the experiment result which is demonstrated in Table 3, the sine cosine algorithm for the Berlin52 node problem has the major parameters as *a=0.1, b=2, and T=3000.* The optimal result for those parameters is 7549.89. Combined with simulated experiments concluded the best value range of three parameters: *a*∈ [0.1, 3], *b*∈ [2.0,7.0], *T*∈ [1000,3000].

The performance of SCA has been compared with the performance of other nature-inspired meta-heuristic methods on two randomly generated TSPs *(RTSP40, RTSP45)* and one classical TSP *(Berlin52):* genetic algorithm, ant colony optimization, black hole algorithm, and sine cosine algorithm. The parameters are set for GA, *cr-rate=0.8, mut-rate=0.02;* for ACO System, *# of ants=20, alfa=1, beta=3, evaporation rate* $(\rho = 0.5)$, *Initial-Feremon=25*; for SCA, *a=0.1, b=2* [1,4,21,22,23]. The population size, except the ACO System, is fixed up to *100*. All the algorithms, except ACO System (*200 iterations*), were run at *1000 iterations* and *10 times* independently and following results have obtained in Table 4:

Table 4. Comparison of Meta-heuristics for Solving Different TSP Instances

Problem		GA	ACO	BH	SCA
RTSP40	Best	519.65	506.18	494.83	497.41
	Average	594.29	508.76	502.49	505.34
	Worst	657.3	519.10	510.32	514.99
	Std. Dev.	49.88	5.45	5.29	5.64
	CPU Time	25.23	37.78	13.16	12.02
RTSP45	Best	589.97	562.22	551.85	549.25
	Average	657.28	562.22	568.17	563.45
	Worst	699.95	562.22	574.95	581.66
	Std. Dev.	29.06	0	7.80	11.29
	CPU Time	31.42	55.67	13.56	12.35
Berlin52	Best	8725.31	8016.97	7913.9	7755.94
	Average	9610.47	8150.80	8312.13	8357.69
	Worst	10420.2	8171.3	8717.33	8630.71
	Std. Dev.	535.29	47.12	242.27	237.75
	CPU Time	40.48	63.24	15.04	14.24

As the algorithm performance is shown in Table 4, SCA is superior to other above-mentioned meta-heuristic algorithms. It means that SCA finds better solutions and reasonably fast compared to other meta-heuristics. BH is the latter algorithm in ranking.

GA converges faster than the other two algorithms shown in Figure 6. However, it finds worse solutions than the other two. Besides, BH leads to a better solution quality than GA and converges between the other two. In addition to that, SCA achieves the best solution and converges later than the other two algorithms.

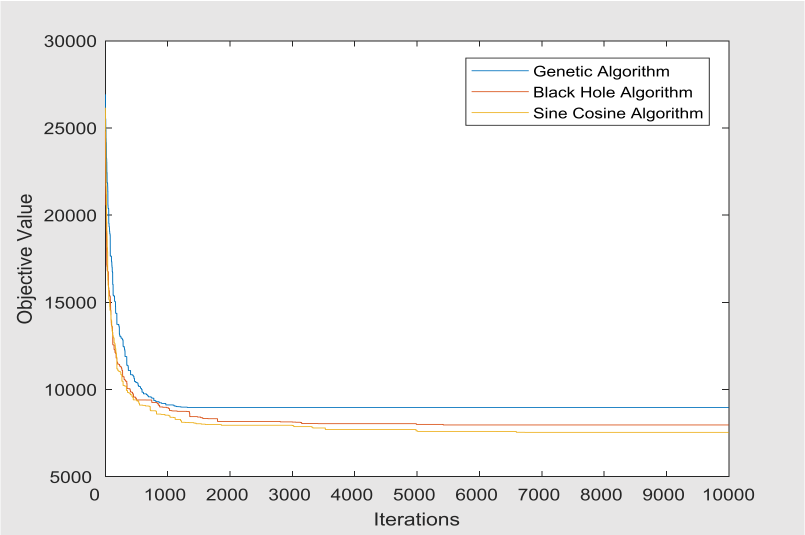

Figure 6. Convergence Curves for GA, BH, and SCA (One run for Berlin52 TSP problem)

In Figure 7, it is concluded that ACO converges at the earliest stage and finds a better solution than GA. However, ACO finds worse solutions than BH and SCA.

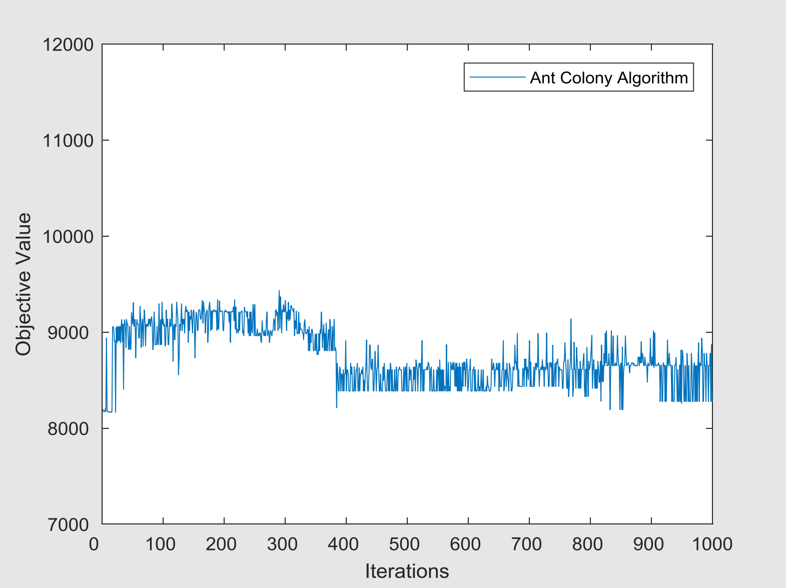

Figure 7. Convergence Curve for Ant Colony Algorithm (One run for Berlin52 TSP problem)

5. Results and Discussion

This work put forths that SCA finds better solutions than the other discussed meta-heuristic algorithms when optimal parameters are used for the combinatorial problem, TSP. However, it converges worse than other algorithms. Additionally, results show that major parameters of SCA (*a, b and T*) influence the performance of SCA significantly. To find better solutions with SCA, the parameters *a and b* should be chosen less and *T* must be high (*a=0.1, b=2 and T=3000*). Also, the performance of algorithms denotes that BH and SCA are dominant to the two classical algorithms.

In future studies, more related works on parameter tuning of SCA and other recent meta-heuristics should be done. When detailed analysis is made, it could be achieved more accurate results. Besides that, investigating more combinatorial problems with new trends and classical meta-heuristics can expose more interesting results. Parameter tuning provides practically optimal solutions and so it helps to realize them on real-world applications.

Acknowledgments

This research received no specific grants from any funding agency in public, commercial or non-profit sectors.

References

[1]. Mirjalili, S., "A Sine Cosine Algorithm for solving optimization problems", Knowledge-Based Systems, 2016, 96, 120-133.

[2]. Osaba, E., Ser, J.D., Sadollah, A., Bilbao, M.N., Camacho, D., "A discrete water cycle algorithm for solving the symmetric and asymmetric traveling salesman problem", Applied Soft Computing, 2018, 71, 277-290.

[3]. Ali, R.S., Alnahwi, F.M., Abdullah, A.S., "A modified camel travelling behavior algorithm for engineering applications", Australian Journal of Electrical and Electronics Engineering, 2019, 16(3), 176-186, https://doi.org/10.1080/1448837X.2019.1640010.

[4]. Hatamlou, A., "Solving travelling salesman problem using black hole algorithm", Soft Computing, 2018, 22 (24), 8167-8175, https://doi.org/10.1007/s00500-017-2760-y.

[5]. Ibrahim, M.K., Ali, R.S., "Novel Optimization Algorithm Inspired by Camel Traveling Behavior", Iraqi Journal for Electrical and Electronic Engineering, 2016, 12(2), 167-177.

[6]. Das, S., Bhattacharya, A., Chakraborty, A.K., “Solution of short-term hydrothermal scheduling using sine cosine algorithm”, Soft Computing, 2018, 22(19), 6409-6427, https://doi.org/10.1007/s00500-017-2695-3.
[7]. Long, W., Wu, T., Liang, X., Xu, S., “Solving high-dimensional global optimization problems using an improved sine cosine algorithm”, Expert Systems with Applications, 2019, 123, 108-126.
[8]. Sindhu, R., Ngadiran, R., Yacob, Y.M., Zahri, N.A.H., Hariharan, M., “Sine-cosine algorithm for feature selection with elitism strategy and new updating mechanism”, Neural Computing and Applications, 2017, 28(10), 2947-2958. https://doi.org/10.1007/s00521-017-2837-7.
[9]. Yildirim, A.E., Karci A., “Applications of artificial atom algorithm to small-scale traveling salesman problems”, Soft Computing, 2018, 22(22), 7619-7631, https://doi.org/10.1007/s00500-017-2735-z.
[10]. Mavrovouniotis, M., Yang, S., “A memetic ant colony optimization algorithm for the dynamic travelling salesman problem”, Soft Computing, 2011, 15(7), 1405-1425, https://doi.org/10.1007/s00500-010-0680-1.
[11]. Xiutang G., Zhihua C., Wei Y., Deqian S., Kai Z., “Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search”, Applied Soft Computing, 2011, 11(4), 3680-3689.
[12]. Jangra, R., Kait, R., “ACO Parameters Analysis of TSP Problem”, International Journal of Computer Science and Mobile Applications, 2017, 5(8), 24-29.
[13]. Tawhid, M.A., Savsani, P., “Discrete Sine-Cosine Algorithm (DSCA) with Local Search for Solving Traveling Salesman Problem”, Arabian Journal for Science and Engineering, 2019, 44(4), 3669-3679.
[14]. Bektas, T., “The multiple traveling salesman problem: an overview of formulations and solution procedures”, Omega, 2006, 34(3), 209-219.
[15]. Zhong, Y., Lin, J., Wang, L., Zhang, H., “Discrete comprehensive learning particle swarm optimization algorithm with Metropolis acceptance criterion for traveling salesman problem”, Swarm and Evolutionary Computation, 2018, 42, 77–88, https://doi.org/10.1016/j.swevo.2018.02.017.
[16]. Kumar S., Datta D., Singh S.K., Black Hole Algorithm and Its Applications. In: Azar A., Vaidyanathan S. (eds) “Computational Intelligence Applications in Modeling and Control”, Studies in Computational Intelligence, vol 575, Springer, Cham, (2015).
[17]. Reddy, K.S., Panwar, L.K., Panigrahi, B. K., Kumar, R., “A New Binary Variant of Sine–Cosine Algorithm: Development and Application to Solve Profit-Based Unit Commitment Problem”, Arabian Journal for Science and Engineering, 2018, 43(8), 4041–4056, https://doi:10.1007/s13369-017-2790-x.
[18]. Qu, C., Zeng, Z., Dai, J., Yi, Z, He, W., “A Modified Sine-Cosine Algorithm Based on Neighborhood Search and Greedy Levy Mutation”, Computational Intelligence and Neuroscience, 2018, vol. 2018, Article ID 4231647, 19 pages, https://doi.org/10.1155/2018/4231647.
[19]. Ekinci, S., “Optimal design of power system stabilizer using sine cosine algorithm”, Journal of the Faculty of Engineering and Architecture of Gazi University, 2019, 34(3), 1329-1350.
[20]. TSP Library (TSPLIB), 2019. http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/ (Accessed on 10.12.2019).
[21]. Wei, X., “Parameters Analysis for Basic Ant Colony Optimization Algorithm in TSP ”, International Journal of u-and e-Service, Science and Technology (IJUNESST), 2014, 7(4), 159-170.
[22]. Shweta, K., Singh, A., “An Effect and Analysis of Parameter on Ant Colony Optimization for Solving Travelling Salesman Problem”, International Journal of Computer Science and Mobile Computing (IJCSMC), 2013, 2(11), 222-229.
[23]. Lin, W.-Y., Lee, W.-Y., Hong, T.-P., “Adapting Crossover and Mutation Rates in Genetic Algorithms”, Journal of Information Science and Engineering, 2003, 19(5), 889-903.