A DISCRETE SYMBIOTIC ORGANISMS SEARCH BASED 2-OPT ALGORITHM FOR TRAVELING SALESMAN PROBLEM

Vladimir Ilin [a,*], Dragan Simić [a], Nenad Saulić [a]

[a] University of Novi Sad, Faculty of Technical Sciences, Serbia

Abstract: *The traveling salesman problem (TSP) is a well-known combinatorial optimization problem. To address TSP, many exact, heuristic, and metaheuristic algorithms have been developed. In this paper, we have proposed a discrete symbiotic organisms search (DSOS) based 2-OPT algorithm for TSP, named DSOS+2-OPT. The proposed DSOS+2-OPT algorithm is implemented in Matlab environment and tested on symmetric instances from TSPLIB. The overall results demonstrate that the proposed DSOS+2-OPT algorithm offer promising results, with the potential for further improvement, particularly in local search domain.*

Keywords*: discrete symbiotic organisms search algorithm, 2-OPT algorithm, traveling salesman problem.*

1. INTRODUCTION

The traveling salesman problem (TSP) is a combinatorial optimization problem in the fields of computer sciences, operation research, and logistics and transportation. The problem is to find the shortest tour that passes through a set of *n* vertices so that each vertex is visited exactly once. In logistics and transportation, the vertices are represented as cities (Ilin et al., 2020). TSP belongs to the class of NP-hard problems, in which optimal solution to the problem cannot be obtained within a reasonable computational time for large size problems. To address TSP, many exact, heuristic, and metaheuristic algorithms have been developed. In this paper, we have explored the use of a recently proposed metaheuristic algorithm for TSP, named symbiotic organisms search (SOS) algorithm.

The SOS algorithm simulates the interactive behavior noticed between organisms in nature. Symbiosis is derived from the Greek word for ‘‘living together’’. De Bary first used the term in 1878 to describe the cohabitation behavior of different organisms (Sapp, 1994). The most common symbiotic relationships found in nature are mutualism, commensalism, and parasitism. Mutualism denotes a symbiotic relationship between two different species in which both benefit. Commensalism is a symbiotic relationship between two different species in which one benefits and the other is unaffected or neutral.

* v.ilin@uns.ac.rs

Parasitism is a symbiotic relationship between two different species in which one benefits and the other is actively harmed (Cheng and Prayogo, 2014). Initially, the SOS algorithm has been developed to solve numerical optimization over a continuous search space. However, the SOS algorithm can be adapted for discrete optimization problems as well. Several recent papers have developed different discrete symbiotic organisms search (DSOS) algorithms for solving the TSP (Ezugwu and Adewumi, 2017; Wang et al., 2019). In this paper, we have developed and implemented basic version of DSOS algorithm in Matlab environment. In order to improve the search capabilities, the 2-OPT algorithm is used each time an improved organism is found during the search.

The rest of the paper is organized in the following way. Section 2 explains the SOS algorithm in detail and Section 3 explains the main steps in the DSOS+2-OPT algorithm. Experimental results and discussion are presented in Section 4, and finally, Section 5 provides concluding remarks.

2. A SYMBIOTIC ORGANISMS SEARCH ALGORITHM

The SOS algorithm is a new metaheuristic algorithm that was inspired by the symbiosis commonly found between organisms in nature (Figure 1). Symbiotic relationships may help organisms to adapt to changes in their environment. The adaptation is reflected through increased fitness and probability of survival. Therefore, organisms should benefit from symbiosis in the long run.

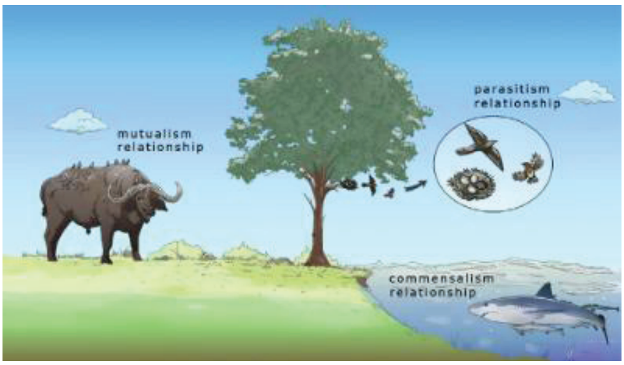

Figure 1. An example of symbiotic relationships in nature (Cheng and Prayogo, 2014)

The SOS algorithm belongs to the group of population-based metaheuristic algorithms. The algorithm has the following characteristics: (1) the SOS uses a population of organisms, which consist of many candidate solutions, examined by the algorithm in a step by step sequence of solutions vector with the hope to approach the global solution over the problem search space; (2) the SOS is also equipped with some kind of special operators that uses the candidate solutions to guide the search process; (3) a selection mechanism is adopted by the algorithm to preserve improved solutions in each iteration; (4) the performance of the algorithm is somewhat dependents on the proper setting of the algorithm's control parameters such as the organisms population size and maximum number of evaluations (Cheng et al., 2015).

The basic steps of SOS algorithm are presented in Figure 2. The main steps are: (1) initialization, (2) symbiotic phase (mutualism, commensalism, and parasitism), and (3) identification of the best solution found. The main parameters in the SOS are termination criteria, named *MaxIter* and number of organisms, named *eco_size*. The computational results are highly dependent on those two parameters.

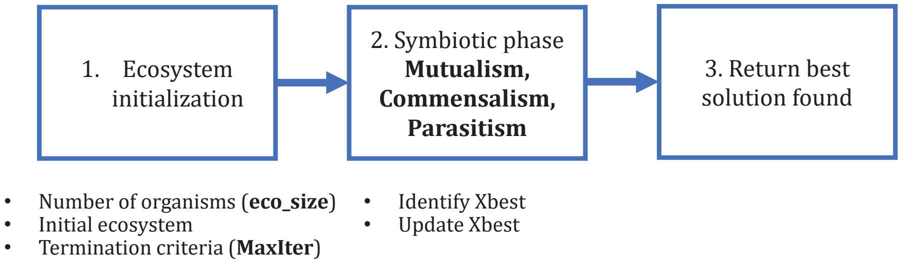

Figure 2. Basic steps of SOS algorithm

In all population-based metaheuristic algorithms specific operations are performed with the objective to generate new organisms in each population. For example, a genetic algorithm has two operations, named crossover and mutation. Harmony search algorithm proposes three rules to improvise a new harmony, named memory considering, pitch adjusting, and random choosing. Artificial bee colony algorithm also proposes three phases, named employed bee, onlooker bee, and scout bee phases. In SOS, these are mutualism phase, commensalism phase, and parasitism phase.

3. A DISCRETE SYMBIOTIC ORGANISMS SEARCH BASED 2-OPT ALGORITHM

The main steps of the DSOS algorithm include mutualism phase, commensalism phase, and parasitism phase. The initial ecosystem is produced randomly with the condition that all solutions need to be feasible.

3.1 Mutualism phase

In mutualism phase, X_i is an organism matched to the i^{th} member of the ecosystem. Another organism X_j is then selected randomly from the ecosystem to interact with X_i. Both organisms engage in a mutualistic relationship with the goal of increasing mutual survival advantage in the ecosystem. New candidate solutions for X_i and X_j are calculated based on the mutualistic symbiosis between organisms X_i and X_j, which is modeled in Eqs. (1) and (2).

$$X_{inew} = X_i + rand(0,1) * (X_{best} - Mutual\_Vector * BF_1) \quad (1)$$

$$X_{jnew} = X_j + rand(0,1) * (X_{best} - Mutual\_Vector * BF_2) \quad (2)$$

where $Mutual\_Vector = \frac{X_i + X_j}{2}$, $rand(0,1)$ generates a value between 0 and 1, and BF_1 and BF_2 are determined randomly using the expression $BF_1 = BF_2 = 1 + round[rand(0,1)]$.

The new candidate solutions X_{inew} and X_{jnew} are accepted only if they provide better fitness values than X_i and X_j, respectively. If an improved solution is found, the 2-OPT algorithm is applied on that solution before it is inserted back into ecosystem.

3.2 Commensalism phase

In commensalism phase, once again, an organism X_j is selected randomly from the ecosystem to interact with X_i. In this interaction, an organism X_i aims to benefit. The new candidate solution of X_i is calculated based on the commensal symbiosis between organisms X_i and X_j, which is modeled in Eq. (3).

$$X_{inew} = X_i + rand(-1,1) * (X_{best} - X_j) \tag{3}$$

where $(X_{best} - X_j)$ represents a benefit provided by X_j to help X_i increase a likelihood of survival. The new candidate solution X_{inew} is accepted only if it provides better fitness value than X_i. If the new and improved solution is found, the 2-OPT algorithm is applied on that solution before it is inserted back into ecosystem.

3.3 Parasitism phase

In parasitism phase, an organism X_i is mutated and the parasite organism, labelled as X_{pv}, is created. The organism X_j is selected randomly from the ecosystem for comparison. In this interaction, if the fitness value of the X_{pv} is better than the fitness value of the X_j, then the X_j will be replaced with the X_{pv}. In addition, the 2-OPT algorithm is applied on the X_{pv}. In opposite, an organism X_j will develop immunity from the parasite organism X_{pv} and the parasite organism will be removed.

The operators, named insertion, inversion, and swap are used to generate the parasite organism. These operators are proposed by Wang, Lin, Zhong, and Zhang (2015). Only one of the three operators is used for computation of the parasite organism based on a random value.

3.4 A 2-OPT algorithm for TSP

The 2-OPT algorithm for TSP was proposed by Lin (1965). In this algorithm, a path is constructed as follows:

Step 1. Find an initial tour randomly or by applying some other algorithm.

Step 2. Try to improve the tour using the two-branch exchange method.

Step 3. Continue Step 2 for all combinations and return the improved tour or the tour that is already 2-optimal.

4. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we have presented experimental results to analyze the performance of the DSOS+2-OPT algorithm for TSP. The parameters used for the proposed DSOS+2-OPT are selected to be to be consistent with the parameters of DSOS (Ezugwu and Adewumi, 2017) in order to make an appropriate comparison. Therefore, MaxIter is 1000, eco_size is 50, and each experiment was executed 10 times independently.

The algorithm is implemented in Matlab and the experiments are run on a desktop computer with an Intel Core i5-2400, 3.1 GHz processor. The authors conducted the experiments on 14 symmetric benchmark instances from TSPLIB (Reinelt, 2022). The obtained experimental results from the proposed DSOS+2-OPT algorithm are compared with the results from the first proposed version of the DSOS (Ezugwu and Adewumi, 2017).

Table 1. Comparison results between the proposed DSOS+2-OPT algorithm and the DSOS algorithm

	DSOS (Ezugwu and Adewumi, 2017)				**DSOS+2-OPT**			
Instance Opt. val.	**Best**	**Mean**	**PDav**	**Time (s)**	**Best**	**Mean**	**PDav**	**Time (s)**
Eil51 426	**426**	427.90	0.45	62.5	428	431.3	1.24	16.28
St70 675	**675**	679.20	0.62	82.58	676	681.7	0.99	16.98
Eil76 538	542	547.40	1.75	93.86	548	554.80	3.12	17.22
KroD100 21294	**21294**	21493.10	0.94	139.70	21464	21686.80	1.84	18.25
Eil101 629	640	650.60	3.43	171.75	649	655.10	4.15	18.16
Pr124 59030	**59030**	59429.10	0.68	235.57	59076	59497.20	0.79	19.20
Pr136 96772	97437	97673.20	0.93	448.08	97755	99290.90	2.60	19.54
Pr152 73682	74013	74785.70	1.50	516.68	74020	74575.70	1.21	20.45
Pr264 49135	50424	52798.90	7.46	622.90	50855	51778.90	5.38	28.25
Pr299 48191	49162	50335.20	4.45	705.27	50109	51132.90	6.10	31.65
Lin318 42029	42201	42972.42	2.24	925.47	43955	44213	5.20	34.06
Rat575 6773	7030	7117.32	5.08	973.86	7312	7382.60	9.00	88.52
Rat783 8806	9045	9102.67	3.37	1043.61	9556	9685.40	9.99	212.89
Pr1002 259045	271381	278381.51	7.46	1843.34	278425	279903.30	8.05	518.06
Average			2.88	561.80			4.49	75.68

In Table 1, if we compare the computed average percentage of the deviation (PDav), the proposed DSOS+2-OPT algorithm is performing slightly worse than the DSOS (Ezugwu and Adewumi, 2017) for 12 out of 14 tested instances. However, if we compare CPU time, the proposed DSOS+2-OPT algorithm is performing significantly better on average basis, than the DSOS (Ezugwu and Adewumi, 2017) for all tested instances. In terms of the quality of the best solution found the DSOS (Ezugwu and Adewumi, 2017) outperforms the proposed DSOS+2-OPT for all tested instances.

Based on the relatively poor quality of the average solution found and high computational time of both compared algorithms we can conclude that both algorithms have relatively poor local search mechanisms. The DSOS (Ezugwu and Adewumi, 2017) algorithm fills that gap with the extensive search which reflects the long execution time. The proposed DSOS+2-OPT can be further improved with some other local search approaches.

5. FINAL REMARKS AND CONCLUSION

In this study, we have presented the basic version of the DSOS algorithm which is additionally improved with the 2-OPT algorithm. The proposed DSOS+2-OPT algorithm is easy to implement, with acceptable execution efficiency, that is, satisfactory CPU times for all tested instances. The algorithm has good global search, but relatively poor local search.

The future work could focus on extending the research on different strategies of how to prevent premature convergence of the proposed DSOS+2-OPT algorithm. The use of some other tour construction and tour improvement algorithms can be investigated. Also, the hybridization with some other metaheuristic algorithms should be explored.

ACKNOWLEDGMENT

This research has been supported by the project "Development of innovative solutions with the aim of improving the field of traffic and transport".

REFERENCES

[1] Cheng, M. Y., Prayogo, D. (2014). Symbiotic organisms search: A new metaheuristic optimization algorithm. Computers & Structures, 139(15), 98–112.

[2] Cheng, M. Y., Prayogo, D., Tran, D.-H. (2015). Optimizing Multiple-Resources Leveling in Multiple Projects Using Discrete Symbiotic Organisms Search. Journal of Computing in Civil Engineering, 30(3), 04015036.

[3] Ezugwu, A. E. S., Adewumi, A. O. (2017). Discrete symbiotic organisms search algorithm for travelling salesman problem. Expert Systems with Applications, 87(30), 70–78.

[4] Ilin, V., Simić, D., Simić, S.D., Simić, S. (2021). Hybrid Genetic Algorithms and Tour Construction and Improvement Algorithms Used for Optimizing the Traveling Salesman Problem. In: Herrero, Á., Cambra, C., Urda, D., Sedano, J., Quintián, H., Corchado, E. (eds) 15th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2020). SOCO 2020. Advances in Intelligent Systems and Computing, vol 1268. Springer, Cham. https://doi.org/10.1007/978-3-030-57802-2_51

[5] Lin, S. (1965). Computer Solutions of the Traveling Salesman Problem. The Bell System Technical Journal, 44, 2245–2269.

[6] Reinelt, G.: TSPLIB, http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/, last accessed 2022/04/07.

[7] Sapp J. (1994). Evolution by association: a history of symbiosis: a history of symbiosis. New York: Oxford University Press.

[8] Wang, Y., Wu, Y.W., Xu, N. (2019). Discrete symbiotic organism search with excellence coefficients and self-escape for traveling salesman problem. Computers & Industrial Engineering, 131, 269–281.