*DOI: 10.24996/ijs.2021.62.5.34*

ISSN: 0067-2904

# Single-based and Population-based Metaheuristics for Solving NP-hard Problems

**Saman M. Almufti[1*], Ridwan B. Marqas[2], Pawan Sh. Othman[3], Amira Bibo Sallow[4]**

[1]Department of Computer science, Nawroz University, Duhok, Iraq
[2]Department of Information Technology, Duhok Private Technical Institute, Duhok, Iraq
[3]Department of Computer System, Ararrat Private Technical Institute, Duhok, Iraq
[4] Department of Computer Sciences, Nawroz University, Duhok, Iraq

Received: 26/5/2020 Accepted: 17/11/2020

**Abstract**

Metaheuristic is one of the most well-known fields of research used to find optimum solutions for non-deterministic polynomial hard (NP-hard) problems, for which it is difficult to find an optimal solution in a polynomial time. This paper introduces the metaheuristic-based algorithms and their classifications and non-deterministic polynomial hard problems. It also compares the performance of two metaheuristic-based algorithms (Elephant Herding Optimization algorithm and Tabu Search) to solve the Traveling Salesman Problem (TSP), which is one of the most known non-deterministic polynomial hard problems and widely used in the performance evaluations for different metaheuristics-based optimization algorithms. The experimental results of Elephant Herding Optimization algorithm and Tabu Search for solving ten different problems from the TSPLIB95 library are compared.

**Keywords**: Metaheuristics, Elephant Herding Optimization, Tabu Search, NP-Hard, Traveling Salesman Problem.

# أحادية الأساس والمجاميع للأدلة العليا في حل المشاكل الصعبة غير القطعية أداء خوارزميات

**سامان محمد المفتي[1*], ريدوان بويا مرقس[2], پاوان شڤان عثمان [3], اميرة بيبو سلو[4]**

[1]علوم حاسبات, جامعة نوروز, دهوك , العراق
[2]تكنولوجيا المعلومات, معهد دهوك التقني الاهلي , دهوك , العراق
[3]انظمة الحاسبات , معهد ارارات التقني الاهلي , دهوك , العراق
[4]علوم حاسبات, جامعة نوروز, دهوك , العراق

**الخلاصة**

يعتبر ميتاهيورستك (Metaheuristic) هو واحد من أكثر المجالات المعروفة التي تستخدم لبحث ولإيجاد الحل الأمثل لمشكلات كثيرة الحدود غير الحتمية الصعبة (NP–Hard) ، والتي يصعب إيجاد الحل الأمثل لها في زمن كثير الحدود. يستعرض هذا البحث الخوارزميات المبنية على ميتاهيورستك وتصنيفاتها، والمشكلات الصعبة متعددة الحدود غير الحتمية، و ومقارنة أداء خوارزميتين تعتمدان على ميتاهيورستك (Elephant Herding Optimization algorithm and Tabu Search) لحل مشكلة بائع متجول (TSP) ، وهي واحدة من أكثر المشاكل المعروفة التي تنتمي إلى مشكلة صعبة متعددة الحدود غير حتمية وتستخدم على نطاق واسع في تقييم اداء خوارزميات ايجاد الحل الامثل القائمة على ميتاهيورستك. النتائج

*Email: saman.almofty@gmail.com

التجريبية للبحث تقارن نتائج الخوارزميتين EHO و TS لحل عشر مشاكل مختلفة من مكتبة TSPLIB95والتي تضم أمثلة عن البائع الجوال (والمشكلات ذات الصلة) من مصادر مختلفة وأنواع مختلفة.

**Introduction**

Over the last years, complexity of problems has been increased so that it is very difficult for basic mathematical approaches to obtain an optimum solution in an optimal time [1]. This has led researchers to develop the Metaheuristic-based Optimization Algorithms, which are appropriate techniques widely used in optimizing solutions for complex problems [2, 3]. Typically, metaheuristic algorithms are developed based on heuristics.

Non-deterministic polynomial hard (NP-hard) problems are those problems in which basic algorithms cannot reach their optimal solutions within a polynomial-bounded computation-time [2].

Commonly, two categories of algorithms are used to solve or optimize a solution for NP-hard problems, namely the complete search algorithms and the approximate search algorithms.

In the complete search algorithms, the proposed algorithms have to test all possible solutions for a given NP-hard problem. Thus, they require exponential computing-time to find an optimal solution. On the other hand, the approximate search algorithms, which is subdivided into single-based and population-based search algorithms, attempts to decrease the solution time by testing the solutions that are most probable to obtain a relatively good (near optimal) solution instead of testing all possible solutions for obtaining optimal solutions [2].

Generally, using approximate search algorithms is more preferable for most NP-hard problems, because it obtains a near-optimal solution in a significantly short time. These algorithms are also known as metaheuristic-based algorithms [3].

In the real life, several problems cannot be solved in a polynomial time and belong to NP-hard problems, such as (Traveling Salesman, Maximum Clique, Min-Color, Longest Path, Subset Sum, Vertex Cover, Circuit Satisfiability, Independent Set, Dominating Set, Graph Coloring, Subgraph Isomorphism, Hamiltonian Path, Knapsack,…etc.) [3, 4, 5]

Generally the problems that belong to non-deterministic polynomial-time hard problems cannot be solved in an optimal time range [2]. In the computer science field, a necessary requirement for any efficient algorithm is that it can reach a goal in a polynomial time [6, 7]. Metaheuristic algorithms characterize a master strategy that guides and modifies other heuristics to produce solutions beyond those that are normally generated in a quest for local optimality [10]. These algorithms attempt to discover an optimal solution for the optimization of NP-hard problems in a polynomial time by constructing random modification and local-searches in the problem search space [4]. This study uses Traveling Salesman Problem (TSP) to test the efficiency of metaheuristic-based algorithms for solving NP-hard problems. It compares two metaheuristics-based algorithms to optimize the solution of one of the most well-known NP-Hard problems, i.e., the TSP [4, 5]. The first proposed algorithm in this study is the Elephant herding optimizations (EHO), which is a metaheuristic-based method inspired from the herding behaviours of living elephants in their clans [6, 7]. The second proposed algorithm is the Tabu Search (TS), which is a metaheuristic-based method that depends on two strategies, namely the hill-climbing and scanning, to avoid earlier solutions [6].

**Traveling Salesman Problem**

TSP is a widely used problem for evaluating and improving many optimization algorithms and belongs to the NP-hard problems [3]. TSP is a problem which requires to find a minimum distance to visit each city once only and return to the starting city. TSPs are divided into two categories, which are the symmetric (STSP) and the asymmetric (ATSP) [5, 8]. For STSP, the distance from α to β is the same as the distance from β to α and the number of trips is calculated by (C-1)! /2 for C city [8, 10]. The optimal trip can be calculated by the summation of distances as shown in (1) [4].

$$optimal\_trip = \left(\sum_{i=0}^{C-1} d_{p(i)p(i+1)}\right) + d_{p(C)p(1)} \quad (1)$$

where p represents the probability list of distances between cities (α and β) [1].

TSPLIB is a library that contains different sample instances for both categories of TSP (ATSP and STSP) from various sources of various types with their known optimal solutions [11].

**Metaheuristic Algorithms**

Metaheuristic is a higher-level problem-independent algorithmic framework that affords a number of guidelines or approaches for developing heuristic-based optimization algorithms [1, 10].

The term metaheuristics was invented by Glover (1986), which is the combination of the Greek word "Meta" that means a higher-level and the word "Heuristic" that means discovering a solution or goal by trial and error [1, 6]. Metaheuristics algorithms are divided into two categories, which are the single solution-based algorithms and the population-based algorithms, as shown in Figure-1.

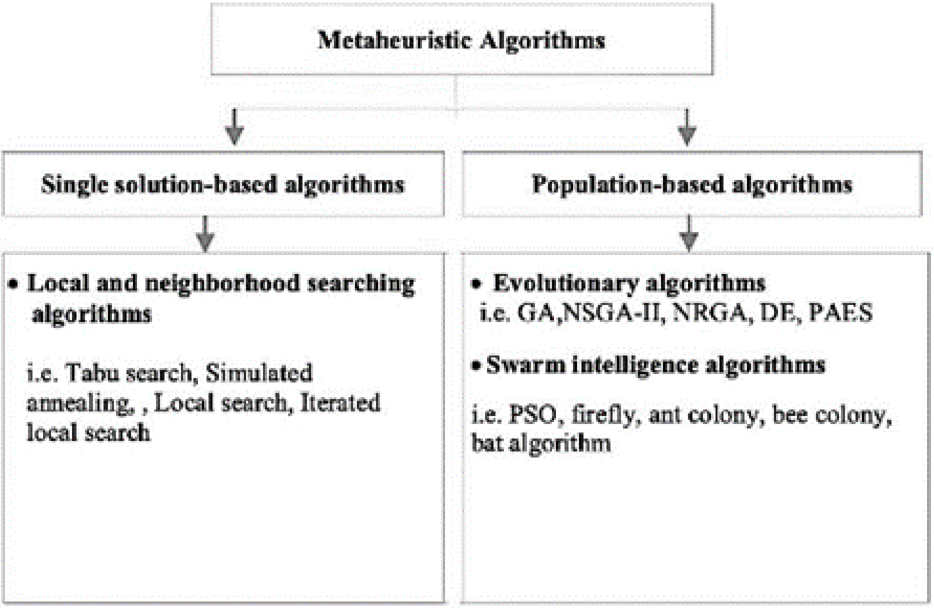


**Figure 1-** Categories of the metaheuristics algorithms

This study considers a comparison between the two algorithms of EHO and TS, which belong to two different branches of metaheuristic Algorithms. TS is a well-known algorithm that belongs to local and neighbourhood searching single-based metaheuristics algorithms, whereas EHO belongs to Swarm intelligence population-based metaheuristic algorithms.

Local and neighbourhood searching algorithms are among the interesting branches of the single solution-based metaheuristics algorithms. Generally, local search metaheuristics discover optimal solutions by iteratively changing procedures from the current single solution. These changes are called "Move" and could be regarded as walks through neighbourhoods or search trajectories of the search space of the problem. Through the search process in each iteration, the current solution is changed by a solution from the neighbourhood set. The technique that is used for selecting a new solution is called the search strategy. There are various search strategies, such as: (1) Steepest-sescent or Steepest-ascent strategy is a widely known search strategy in which the most suitable (best) move from the neighbourhood is selected. The metaheuristics of this strategy is called hill-climbers. (2) Random-improving solution, in which a solution that is better than any other solution in its neighbourhood is called a local optimum. (3) First-improving strategy, which is also called mildest ascent/descent strategy, in which the first move that improves the current solution is selected.

Tabu Search is the most popular local search metaheuristic algorithms. It is a general metaheuristic procedure for guiding search to find a good solution in complex search spaces. TS was proposed in its present form a few decades ago by Glover (1986). A complete neighbourhood checking with TS algorithm provides generally high-quality solutions for various optimization problems to many new fields.

Swarm intelligence (SI) is one of the most interesting branches of population-based metaheuristics algorithms in the artificial intelligence field. It is a collection of intelligent multi-agent techniques that cooperate with each other to accomplish a specific task [3]. Those algorithms are inspired by the behaviours of social communities of living beings in the nature, such as elephants, birds, wolfs, ants, and bees. The most interesting characteristics of swarm systems are self-organization and decentralized-control, which naturally leads to an emergent behaviour in the group of living agents.

The emergent behaviour represents an interactive behaviour which emerges as a local interaction between group agents and is not possible to be achieved alone by any single agent in the group [1].
Generally, these algorithms do not guarantee discovering an optimal solution [1, 10]. Famous instances of metaheuristics algorithms are Particle Swarm Optimization (1995), Artificial Swarm Intelligence (2015), Ant Colony optimization (1992), Elephant Herding optimizations (2015), and Tabu Search (1986) [5, 10, 12].
There are several main characteristics of metaheuristic algorithms [10]. First, they involve a number of approaches that control the search procedure. Second, the idea of their usage is to professionally discover a search space to obtain an optimal solution. Third, they are approximating approaches and generally non-deterministic.

**A. Elephant Herding Optimization**

Elephant herding optimizations is a population-based metaheuristics algorithm that was designed by Wang in 2015 to solve optimization problems [9, 12]. EHO is inspired by herding behaviours of elephants in their clan [13, 14].

- Herding behaviour of elephants can be described in several points [13]. An elephants group consists of a number of sub-groups, called clans, which contain a calve and many female elephant, as shown in Figure 2- (a). A matriarch typically supervises each clan. Male calves leave the clan when they grow up to adulthood, as shown in Figure 2- (b).

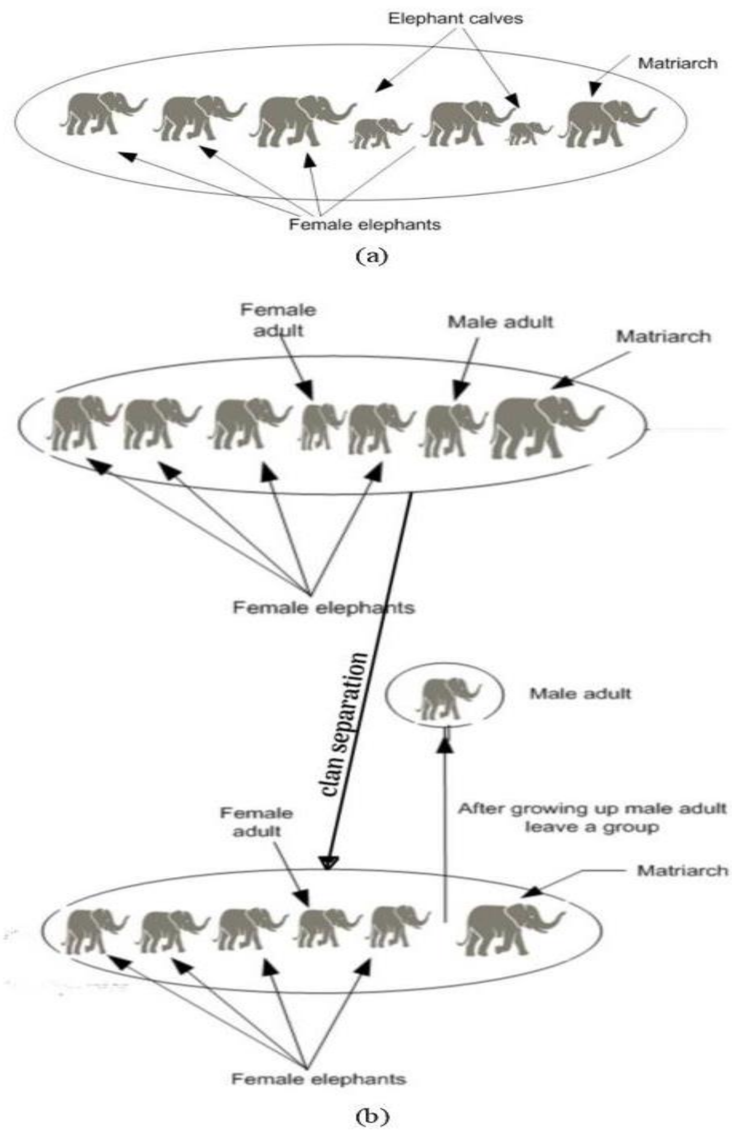


**Figure 2-** Elephants clan characteristics: (a) a clan with calves before (b) adult male elephant separation process.

The EHO algorithm consists of two phases [15], as described below.

**1) Clan updating**

In an elephant clan, a matriarch supervises other female elephants. The location of all elephants in the clan are affected by the matriarch position. The clan updating process is based on the following equation (2) [13].

$$El_{new} = El_{old} + \alpha(El_{best} - El_{old})R \tag{2}$$

where $El_{old}$ characterizes the previous location , $El_{new}$ characterizes the next location for j elephants in $El_{Clan}$ clan, and α characterizes a scale operator ∈ {0, 1} to regulate the effect of matriarch elephant $El_{Clan}$ on $El_{old}$ [13, 15]. $El_{best}$ represents the matriarch elephant in the $El_{Clan}$ [14]. R is a kind of distribution ∈ [0, 1] for enhanced diversity of elephant populations at each iteration [8].

In $El_{Clan}$, the matriarch $El_{best}$ is not affected in (2). EHO uses (3) to update $El_{best}$.

$$El_{new} = \beta El_{center} \tag{3}$$

where $El_{center}$ characterizes the center-point of $El_{Clan}$ based on the information collected by the $El_{Clan}$ elephants, and $\beta$ ∈ {0,1} determines the influences of $El_{center}$ on $El_{new}$. The centre position $El_{center}$ in $El_{Clan}$ can be calculated by E (4).

$$El_{Center} = \frac{1}{n_{clan}} \sum_{j=1}^{n_{clan}} El_{old} \tag{4}$$

where $n_{clan}$ represents the population number of elephants in $El_{Clan}$ [13, 15]. The clan updating phase is shown in Algorithm 1.

Algorithm 1: Clan updating phase [14].

| | |
|---|---|
| 1: | Start |
| 2: | Loop (i=1: Total No. of Clans) |
| 3: | Loop (j=1: $n_{clan}$ ) |
| 4: | Update $El_{old}$ & find $El_{new}$ by (2). |
| 5: | If ( $El_{old}$ == $El_{best}$ ) |
| 6: | Updates $El_{old}$ & find $El_{best}$ by (3) |
| 7: | EndLoop |
| 8: | EndLoop |
| 9: | End |

**2) Clan Separation phase**

In all clans, male elephants leave the group to live alone after reaching the adult age. In optimization problems, this separating process is called separating operator [13, 12]. In EHO method, the adult male with the worst efficiency separates the clan in each generation, as shown by using equation (5) [16].

$$El_{worest} = El_{\min} + (El_{\max} - El_{\min} + 1)R \tag{5}$$

where $El_{worest}$ denotes the worst male elephant in the $El_{Clan}$ [13]. $El_{\min}$ and $El_{\max}$ denote the lower and upper bounds of elephants' positions. R is a type of stochastic and uniform distribution ∈ {0,1} [16]. The separating operation is shown in Algorithm 2 [14].

Algorithm 2: Separating operator [14].

```
Start
Loop (i=1: Total No. of Clans)
update El_worest individual in El_Clan by (5).
EndLoop
End
```

The STSP is a combinatorial-optimization problem in which the solution is denoted by a series that can be shown as a vector [2]. The EHO method was designed to find a solution for continued-optimization problems in which a number represents the solution. Therefore, the EHO algorithm is unusable for solving STSP directly [14]. EHO can be adapted to solve STSP with small changes in its operators by respecting the real behaviours of elephants, as shown below:

- Random initialization for elephant position.
- The position of each elephant denotes a node in STSP.
- The distance between elephants (El1 – El2) represents the set of probabilities of El2 to obtain El1.
- The addition between the set of permutations and a position (El + sp) applies the set of permutations to the El position.

The process of adapting EHO algorithm for solving STSP is shown in Figure 3.

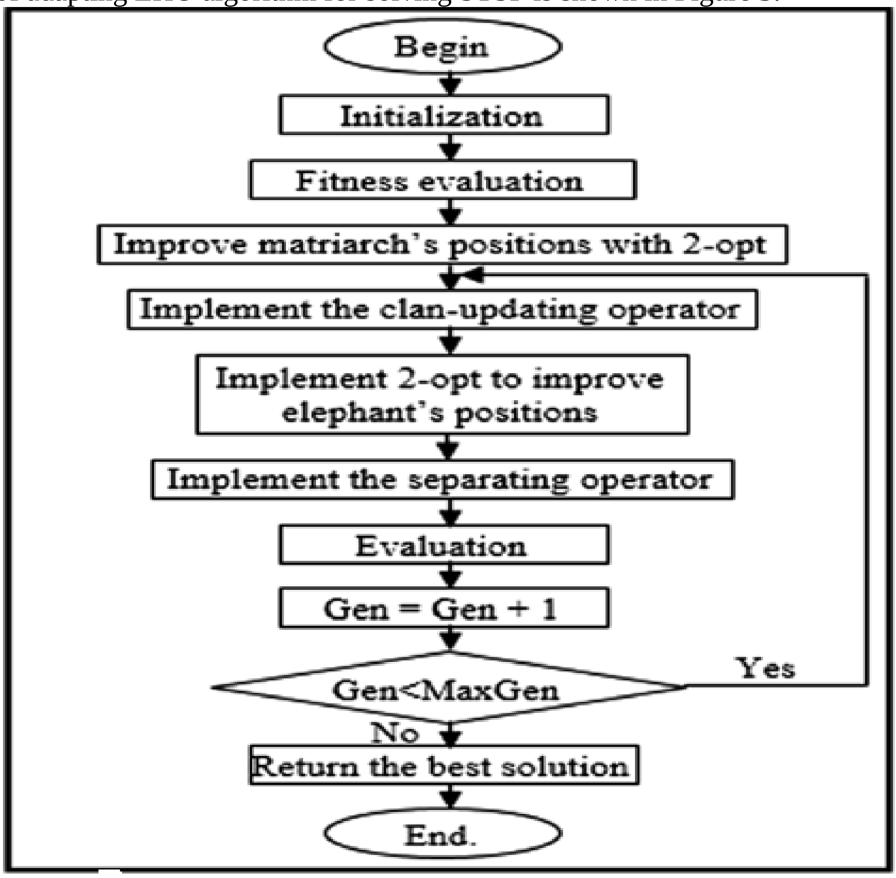


**Figure 3-** EHO algorithm for solving STSP.

**B. Tabu Search**

Tabu search is one of the broadly known single solution-based metaheuristical algorithms designed by Glover (1986) to find good (approximate) solutions for combinatorial optimization problems. Tabu search can be described as the algorithm that starts at a random initial node and then goes to a neighbouring node [17]. A neighbouring node is created by a set of allowable moves. At each iteration, the method moves to the best node in the neighbourhood of the current set until a chosen termination criterion is satisfied [12]. Generally, this algorithm has a better execution time than the other local-search algorithms, because it includes a short-term memory that records the recent history of the search to prevent revisiting the recently visited nodes. It also includes a longer-term frequency memory that reinforces a good-looking node. TS memories are called Tabu lists [17, 18, 19].

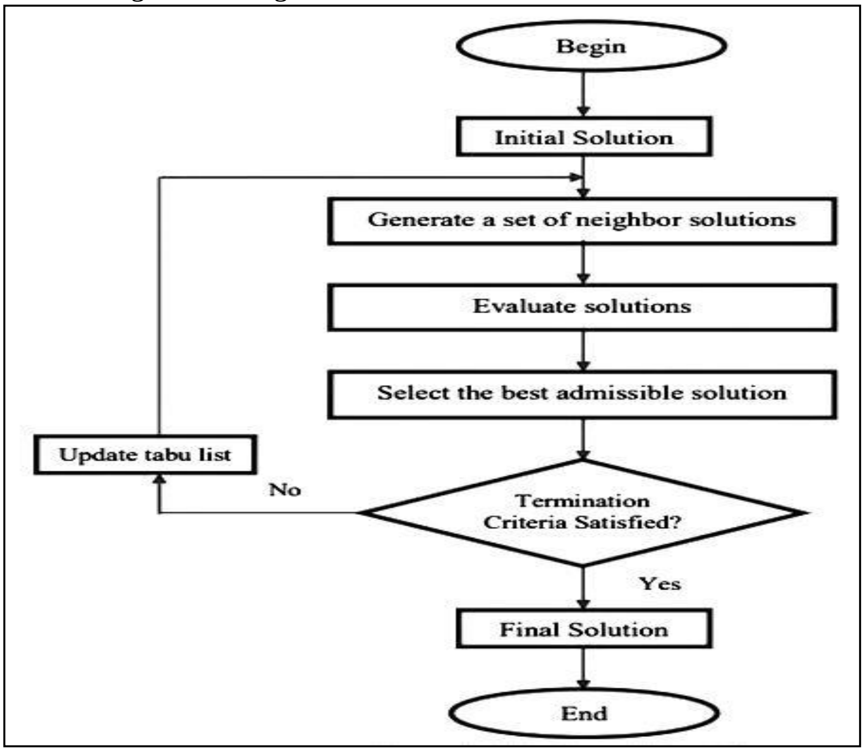


**Figure 4-** Tabu search algorithm procedure

Figure 4 shows the Tabu search algorithm procedure for solving STSP, starting from randomly initializing the problem nodes, then visiting the neighbourhood node, and updating the Tabu list until it reaches the optimal or near-optimal solutions.

**Experimental Results**

This section presents an evaluation of the performance of both elephant herding optimizations and TS in solving the STSP that belongs to NP-hard problems. Both EHO and TS are used to solve 10 different samples of STSP problems selected from the TSPLIB. The size of the selected problems ranged between 52 and 150 cities.

Both algorithms were tested in the same conditions and all problems were solved on Core i7 laptop with 8 GB ram, using MATLAB 2018b.

**Table 1-** EHO and TS Experimental Results in solving TSP

| No | STSP | Opt. sol | EHO | | | TS | | |
|---|---|---|---|---|---|---|---|---|
| | | | worst | best | time | worst | best | time |
| 1 | $berlin52$ | 7542 | 9707 | **7542** | 10.255 | 9513 | 8175 | 8.728 |
| 2 | $kroA100$ | 21282 | 33348 | **21282** | 18.634 | 36567 | 29381 | 16.78 |
| 3 | $kroB100$ | 22141 | 31827 | **22141** | 33.069 | 33181 | 30095 | 16.20 |
| 4 | $kroC100$ | 20749 | 33063 | **20749** | 18.912 | 32511 | 29333 | 17.70 |
| 5 | $kroD100$ | 21294 | 40031 | **21294** | 18.074 | 32114 | 29054 | 17.65 |
| 6 | $Eil101$ | 629 | 1088 | 630 | 13.10 | 831 | 764 | 10.66 |
| 7 | $Lin105$ | 14379 | 16352 | **14379** | 14.714 | 26371 | 20302 | 10.15 |
| 8 | $Ch150$ | 6528 | 8216 | 6550 | 15.67 | 10096 | 9435 | 12.36 |
| 9 | $kroA150$ | 26524 | 29254 | **26524** | 22.28 | 46841 | 40887 | 20.11 |
| 10 | $kroB150$ | 26130 | 31252 | **26130** | 20.16 | 46965 | 39668 | 18.81 |

Table 1 shows the experimental results of both algorithms and the optimal solutions for each problem. The optimal solutions were collected from the TSPLIB. The experimental results show that the EHO reached the optimal solutions for 8 over 10 problems, whereas the TS did not reach any optimal solution for all proceeded problems.

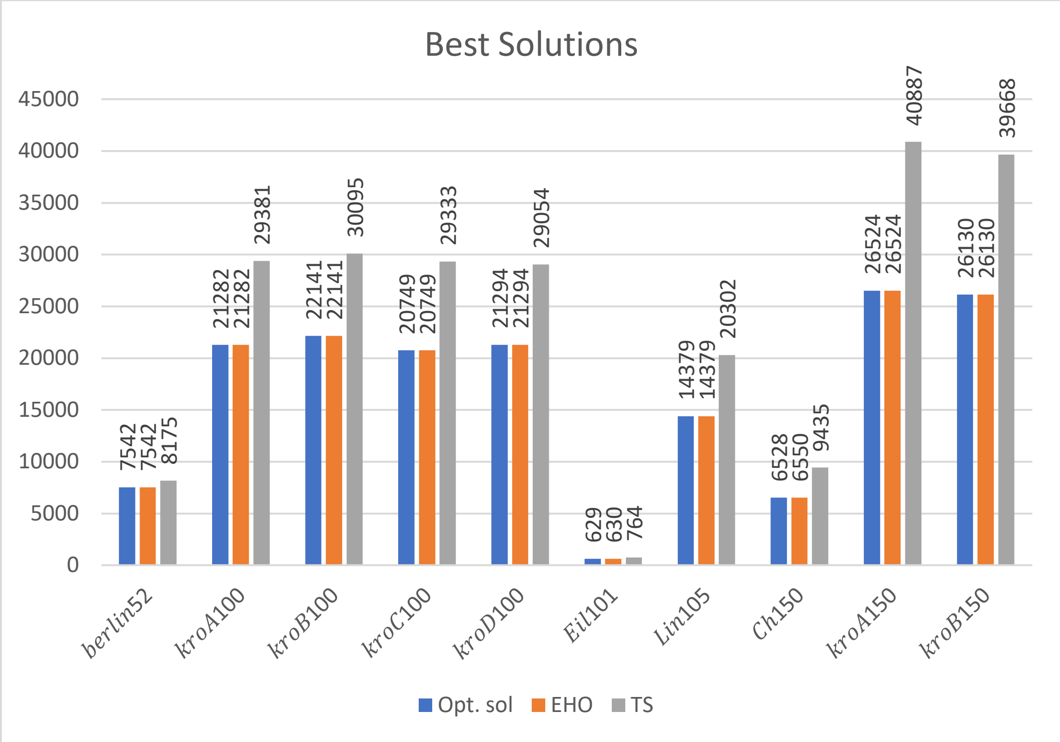


**Figure 5-** Comparison of EHO and TS best and optiimal solutions.

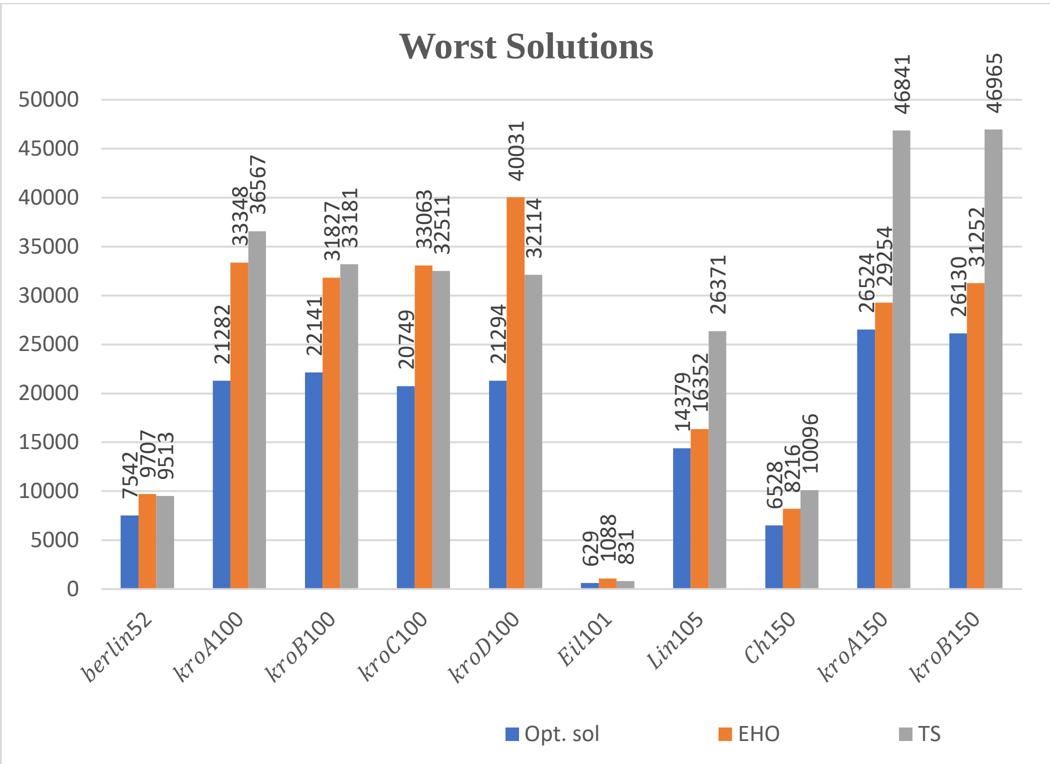


**Figure 6-** Comparison of EHO and TS worst and optimal solutions.

Figures 5 and 6 show a graphical comparison of worst and best solutions found for both EHO and TS. The best solutions of both the proceeded algorithms were compared with the TSPLIB optimal solutions. Generally, the EHO best solutions found for the chosen STSP problems were very close to the optimal solutions, whereas the best solutions of TS were far from the optimal ones. The worst solutions of both EHO and TS were very far from the optimal solutions. The error percentage for the best solution of both algorithms is calculated by equation (6).

$$E = ((Best - Optimal)/Optimal)*100\% \quad (6)$$

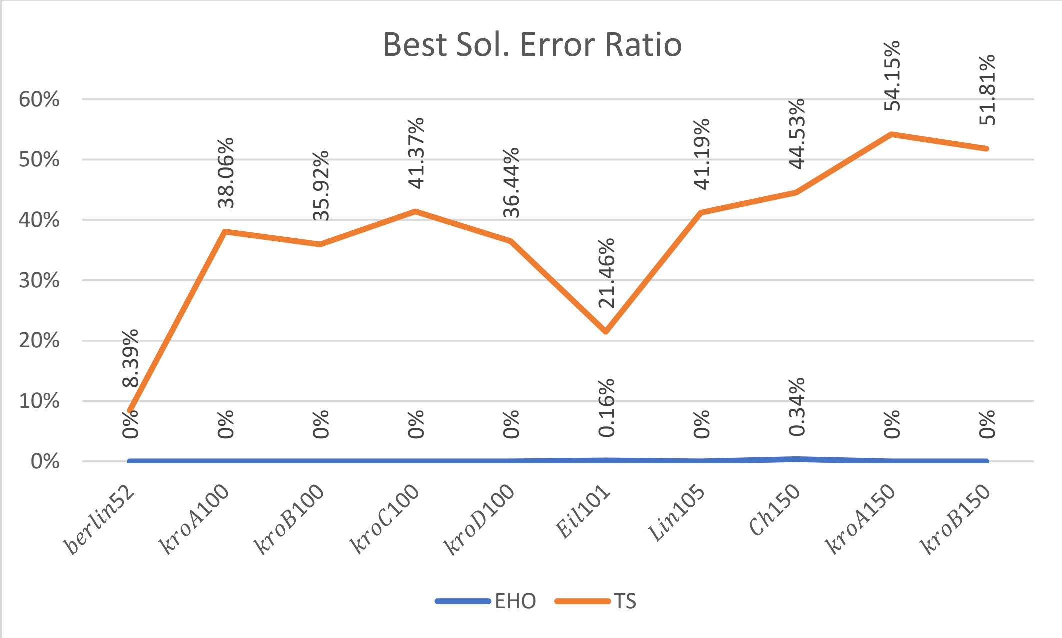


**Figure 7-** Error ratio for EHO and TS best solutions

Figure 7 shows the graphical difference for the error ratio between the EHO and TS best solutions. It can be clearly observed that the EHO performed better than TS in obtaining the best (optimal or near optimal) solutions for STSP (TS).

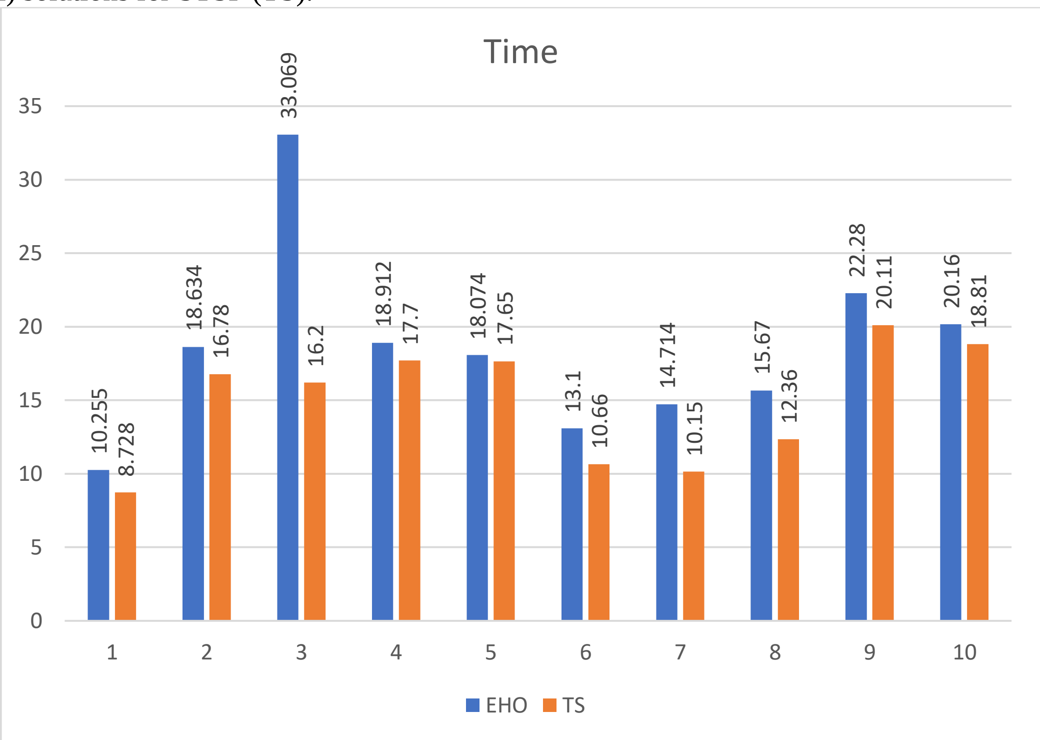


**Figure 8-** Comparing EHO and execution time

Figure 8 shows the graphical difference for the execution time between the EHO and TS algorithms over 10 iterations. It can be clearly indicated that has a shorter execution time than that of the EHO to complete 10 iterations for almost all of the proposed STSP problems. The total time required by each algorithm is compared in Fig.9.

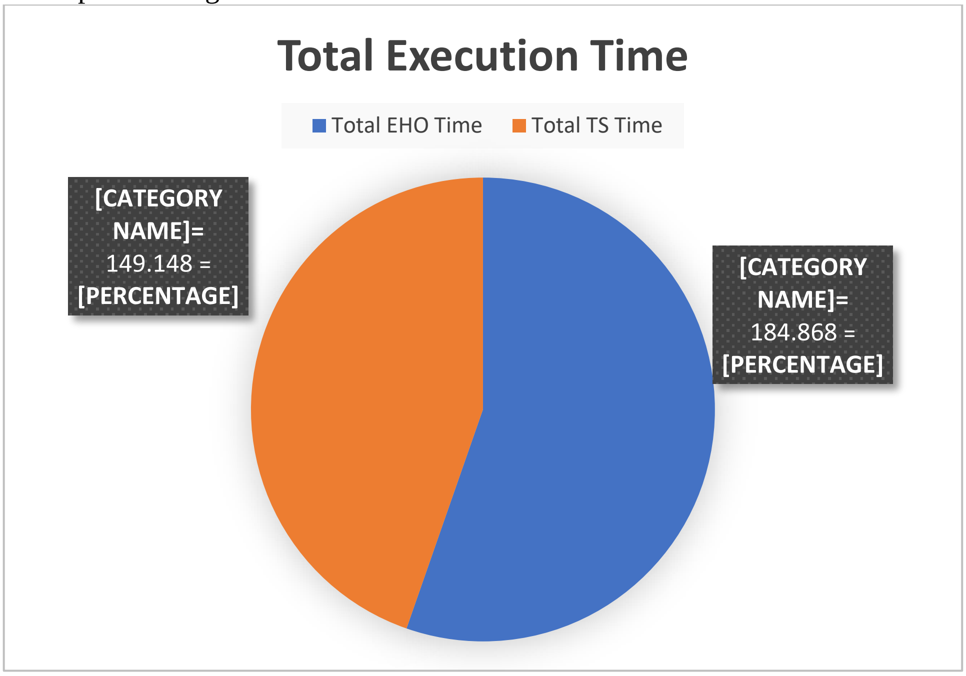


**Figure 9-** EHO and TS algorithms total execution time

**Conclusions**

This paper presents comparisons between two algorithms, namely EHO which belongs to the population-based metaheuristics algorithms and TS which belongs to the single-based metaheuristics algorithms, in solving one of the widely used NP-hard problems, i.e. the STSP. The results of simulations are compared with the optimal solutions from TSPLIB library.

As shown in the experimental results, all algorithms are adapted to solve the same STSP problem in the same experimental conditions. The results for 10 STSP problems demonstrated that EHO obtained the optimal solutions in 8 problems, whereas TS did not achieve any optimal solution.

The error ratio obtained by TS to solve KroA150 was very high, reaching a value of 54.15, which is twice the optimal solution.

A comparison of the overall time needed for the execution of both algorithms in solving ten STSP problems revealed that the overall execution time by using EHO is less than that by using TS.

This paper concludes that EHO performs better that TS in solving STSP, and, based on that, population-based metaheuristics algorithms perform better than single-based metaheuristics algorithms in solving NP-hard problems.

**References**

1. Yang X., 2011, '*Metaheuristic Optimization, Scholarpedia*', 6(8).
2. Almufti S., 2015, '*U-Turning Ant Colony Algorithm powered by Great Deluge Algorithm for the solution of TSP Problem*', Available from: Hdl.handle.net, [last accessed Mar 2020].
3. Almufti S., 2017, '*Using Swarm Intelligence for solving NPHard Problems*', Academic Journal of Nawroz University, vol. 6, no. 3, pp. 46–50.
4. Almufti S., Marqas R. & Asaad R., 2019, '*Comparative study between elephant herding optimization (EHO) and U-turning ant colony optimization (U-TACO) in solving symmetric traveling salesman problem (STSP)*', Journal Of Advanced Computer Science & Technology, 8(2), 32,. https://doi.org/10.14419/jacst.v8i2.29403.

**5.** Marqas, R., Almufti, S., Othman, P. and Abdulrahman, C., 2020. '*Evaluation of EHO, U-TACO and TS Metaheuristics algorithms in Solving TSP'*. JOURNAL OF XI'AN UNIVERSITY OF ARCHITECTURE & TECHNOLOGY, XII(IV).

**6.** Rajpurohit J., Sharma T. & Abraham A., 2017, '*Glossary of Metaheuristic Algorithms*', International Journal of Computer Information Systems and Industrial Management Applications, vol. 9, pp. 181-205.

**7.** Barrington D., & Maciel A., 2010, '*IAS / PCMI Summer Session 2000 Clay Mathematics Undergraduate Program Basic Course on Computational Complexity Lecture 7: 7: NP-Complete Problems*'.

**8.** Erickson J., 2002, '*CS 373: Combinatorial Algorithms, Dcs.fmph.uniba.sk'*,[Online]. Available: http://www.dcs.fmph.uniba.sk/~fduris/VKTI/3color_HamCycle.pdf. [Accessed: 06- Apr- 2020].

**9.** Almufti S., Asaad R. & Salim B.,2019, '*Review on Elephant Herding Optimization Algorithm Performance in Solving Optimization Problems'*. International Journal of Engineering & Technology, [S.l.], v. 7, n. 4, p. 6109-6114. ISSN 2227-524X. Available at: <https://www.sciencepubco.com/index.php/ijet/article/view/28473/15566>.

**10.** April J., Glover F., Kelly J., & Laguna M., 2003, '*Practical introduction to simulation optimization*'. Winter Simulation Conference.

**11.** Reinelt G., 1997, '*TSPLIB', Elib.zib.de*, Available at: http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html. [last accessed: Feb 2020].

**12.** Almufti, S., Marqas, R., Othman, P., & Abdulrahman, C. (2020). '*Evaluation of EHO, U-TACO and TS Metaheuristics algorithms in Solving TSP'*. JOURNAL OF XI'an UNIVERSITY OF ARCHITECTURE & TECHNOLOGY, 12(4). https://doi.org/10.37896/jxat12.04/1062

**13.** Chibani S. & Tari A., 2017, '*Elephant Herding Optimization for Service Selection in QoS-Aware Web*', International Journal of Computer and Information Engineering.

**14.** Wang G., Deb S. & Coelho L., 2015, '*Elephant Herding Optimization'*, 3rd International Symposium on Computational and Business Intelligence (ISCBI).

**15.** Hossam A., Bouzidi A. & Riffi M., 2019, '*Elephants Herding Optimization for Solving the Travelling Salesman Problem*', Advances in Intelligent Systems and Computing, pp. 122-130.

**16.** Wang G., Dos L., Gao X. & Deb S., 2016, '*A new metaheuristic optimisation algorithm motivated by elephant herding behaviour*', International Journal of Bio-Inspired Computation, vol. 8, no. 6, p. 394.

**17.** Stefan E. & Stefan S., 2012, '*Chapter 14 - selective search'*. In Stefan Edelkamp and Stefan Schrödl, editors, Heuristic Search, Morgan Kaufmann, San Francisco, p 633 – 669.

**18.** Gendreau M., Iori M., Laporte G. & Martello S., 2006, '*A Tabu Search Algorithm for a Routing and Container Loading Problem*', Transportation Science, pp. 342-350.

**19.** Almufti S., 2019, '*Historical survey on metaheuristics algorithms'*, International Journal Of Scientific World, 7(1), 1. doi: 10.14419/ijsw.v7i1.29497.