JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125 VOL 7, ISSUE 15, 2020

An Inspired Algorithm for Solving Competitive Travelling Salesmen Problem

Taiser Samer Jasim[1] and Esam Taha Yassen[2]

[1]College of Computer Science and Information Technology, University of Anbar, Iraq. E-mail: Taiser.samer@uoanbar.edu.iq

[2]Computer Centre, University of Anbar, Iraq. E-mail: co.esamtaha@uoanbar.edu.iq

Received: 14 April 2020 Revised and Accepted: 8 August 2020

Abstract Competitive travelling salesman problem (CTSP) is a combinatorial optimization problem in which number of salesmen compete among themselves to fined optimal solution with larger benefit and shortest path. Despite the importance of this problem and its many applications in real life, a few algorithms have been proposed to address this problem. Consequently, the need to either improve the existing algorithms or utilize a new algorithm is still necessary. In the last decades, the nature inspired algorithms, which are seek inspiration from nature and biology phenomena, have been the goal of numerous studies in the most scientific fields, especially in operating research and artificial intelligence. The swarm intelligence describe as an active research area in the developments of new algorithms inspired by nature. One of the recent swarm intelligence algorithms is **salp swarm algorithm (SSA)**. This algorithm is characterized as being simple and flexible, so it motivates scholars to conduct several modifications to improve its performance. But, as any population based metaheuristics, SSA suffers from the slow convergence due to its weak ability to exploit the search space. Thus, this paper proposes enhancing SSA to handle CTSP by utilizing its strong exploring ability and enhancing its exploitation ability. This enhancement achieves via hybridizing the SSA with a single-based meta-heuristics (SBHs) which have strong exploitation ability. In this hybridization, the SSA will be responsible for exploration and the SBH will be responsible for exploitation. The adopted algorithms are applied on CTSP benchmark to test their validity. Results demonstrated that preserving the balance between exploration and exploitation during the search have significant impact on the SSA efficiency. Thus, we concluded that the proposed hybridization managed to improve the effectiveness of SSA in getting good quality solutions.

Keywords: Competitive Travelling Salesman Problem, Salp Swarm Algorithm, Hill Climbing Algorithm, Metaheuristics, Hybrid Metaheuristics

1. Introduction

The travelling salesman problem (TSP) is considered as one of the perplexing optimization problems in transportation and distribution systems. Thus, it has been a fertile area of research which has drawn researching efforts from various communities such as operation research and artificial intelligence. The TSP is generalized version of Hamilton Cycle. A Hamiltonian cycle is a cycle that visits each node only one time in a graph (Shaikh, M., & Panchal, 2012).In TSP, vertices denote to cities; edges denote to the path between cities, and edges cost denote to the distance between cities. The salesmen must start from a city, visit all cities only one time, and return to the first city with minimized cost (Nemati, K., Shamsuddin, M., & Kamarposhti, 2011), (Taba, 2009) . TSP is NP-hard, that's mean there is no known polynomial time that algorithm can find the optimal solution to any given instance (Lawler E, Lenstra J, 1986). There are many applications of TSP some of them are Drilling of printed circuit boards, Overhauling gas turbine engines, X-ray crystallography, Computer wiring, the order-picking problem in warehouses, and the order-picking problem in warehouses, vehicle routing (Davendra, 2010). There are many of variations of TSP. Some of them are Symmetric TSP, Asymmetric TSP, Multiple TSP, Competitive TSP and many others (H., 2013), (Applegate D, Bixby R, 2006) .This study focuses on CTSP which is considered more realistic problem than other variations (Kendall, G. & Li, 2013). In this problem, number of travelling salesmen compete with each other to visit a number of cities. Any salesman will receive a payoff if he is the first one visit a city and he must pay a cost for the distance he traveled(Kendall, G. & Li, 2013). Thus, each salesman goals to visit as many

cities not have been visited before as possible with the minimum distance of travel. The CTSP is an NP-complete problem for the salesmen to design their paths and there is strategic interaction between the salesmen due to their conflict of interest (Kendall, G. & Li, 2013).Due to the importance of CTSP, there are many algorithms have been proposed to solve it. Fekete et.al (2004) propose the competing salesmen problem‖ (CSP), a two-player competitive version of the classical traveling salesman problem. This problem arises when considering two competing salesmen instead of just one. The concern for a shortest tour is replaced by the necessity to reach any of the customers before the opponent does(Fekete, S., Fleischer, F., Fraenkel, A. & Schmitt, 2004). Kendall and Li (2013) have been introduced a new version of the TSP and proposed a "hyper-heuristic methodology" to address it. In a competitive travelling salesmen problem (CTSP), *m* travelling salesmen want to visit *n* cities with non-cooperative relationship among them. Any salesman if he visits a city first one will obtain a benefit and pay a cost for traveling to every city. The aim of each salesman is to visit as many cities not have been visited before as possible, with a minimum distance of travel. Because of the competitive factor, each salesman needs to consider the path of other salesmen when designing his own path. Kendall and Li developed a hyper-heuristic methodology due to that the "equilibrium analysis" is hard in the CTSP. This method assumes that each salesman takes a single heuristic (or number of heuristics) to select his path and each salesman knows that the paths of all other salesmen aren't optimal necessarily. The hyper- heuristic involves of a number of low-level heuristics, each of which can be utilized to initiate a path given the heuristics of the other salesmen, collected with a high-level heuristic that is utilized to choice from the low-level heuristics at each decision point (Kendall, G. & Li, 2013). Mohtadi and Nogondarian (2014) proposed a game theoretic approach for solving the traveling salesman problem in competitive situations. They used to test the problems game theory as a mathematical model. They used tabu search and genetic algorithms. The obtained results show that the computational error is within a reasonable range. Additionally, these results show that tabu search algorithm fined solutions with better quality and less time (Mohtadi, M. & Nogondarian, 2014).Mohannad in (2016) was been used ant colony optimization (ACO) algorithm to solve competitive traveling salesman problem because it is very difficult to compute the Nash equilibrium with large search space. It is inspired from the real ant colony, in which ants leave pheromone trails when looking for source of food in order to guide other ants to the food. In this work he used two different strategies to solve CTSP. In the first strategy, the cities are divided among salesmen equally. To prove the ability of ACO algorithm to solve CTSP, it is compared to (NN) Nearest Neighbors algorithm and (RN) Random Neighbors algorithm which adopted by (Li, J. & Kendall, 2015). In the second strategy, the number of cities is not specific and each salesman will try to visit as much as possible cities according to strategy of the salesman. In this strategy, taken two directions, the first one use for each salesman same plan, while the other uses, for each salesman, every time an update plan. The results that he obtained to it for all experiments clearly indicate that ACO could be a promising method for solving CTSP (Hameed, 2016).
In spite of the importance of competitive salesman problem and its application in our daily life, there is only little research proposed to study it. Consequently, more efficient algorithm that can significantly work well to improve the quality of obtained solution is highly required. Recently, the nature-inspired metaheuristic algorithms can be describe as the most efficient algorithms in tackling and obtaining good quality solutions for complex real-world optimization problems (Esam Taha Yassen, Masri Ayob, 2012). One of these algorithms is Salp Swarm Algorithm (SSA) that has been proposed by (Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, 2017). SSA has been presented as a powerful method to handle many difficult problems and it has not been applied to the CTSP. So, the main objective is to investigate the performance of the SSA in tackling the Competitive Salesman Problem.

2. The Standard SSA

The salp swarm algorithm is one of the efficient recent natures inspired population based meta-heuristic algorithms. The main inspiration of SSA is the swarming behavior of salps when navigating and foraging in oceans (Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, 2017). Based on the fact that the standard SSA is successful in tackling various optimization problems (Laith Abualigah , Mohammad Shehab , Mohammad Al Shinwan, 2019), the study hypothesize that SSA would be successful in tackling CTSP. To solve the CTSP which is highly constrained problem using SSA, the SSA components have to be designed carefully. The main SSA components that are different from one problem to others are ***swarm*** initialization, fitness evaluation and updating the salps' positions steps (Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, 2017). Table 1 illustrates the analogy between CTSP and SSA algorithm terms.

Table 1 the analogy between CTSP and SSA.

CTSP	SSA
Feasible solution	Salp
Agent Route	Decision variable
City	Position
Cities identities	Position range

- **Initialization**

In this step, the salp population (swarm) is randomly initialized using algorithm 1. Firstly, an empty route is created for each salesman. For each route, the following processes are repeated: a seed city is randomly selected and added to the current route. The random un-routed city is inserted as long as the problem constraints are not violated. The insertion process is repeated till all cities have been routed. Then, all the agents' routes are collected in the new salp. Finally, a set of salps (feasible solutions) are added to the swarm. The number of generated salps is equal to N.

```
Algorithm 1 Initialize the salp population (swarm)
Input:-
No. of cities, No of salesmen, size of swarm (N)
 Processing:-
1. Create an empty salp.
2. For each salesman do following step:-
      i.   Create an empty route.
      ii.  Randomly select the seed city and insert it into the route.
      iii. Randomly select the un-visited city and insert it to the last inserted one as long as the
           CTSP constraints are not violated.
      iv.  Repeat the above step (iii) until all cities have been visited.
3. The generated route is inserted into the current salp.
4. Add the current salp to the swarm.
5. The process of creating new salp and inserting salesmen's routes (Steps 1-4) is repeated until the
   swarm has been created.
6. Return swarm.
```

- **Evaluation**

In order to determine the salp with the best fitness (leader), the fitness of each salp is calculated based on the following steps:

1. For each city, the time at which each agent has arrived is calculated.
2. Determining the agent who reached the city before the others.
3. Calculate the payoff for each agent as follows:

   3.1 For each visited city c_i do the following:

   i. **Benefit**. For each visited city c_i, its benefit $\boldsymbol{B_i}$ is given to the current salesman he reached the city before the others. Otherwise, he will receive zero benefit. If two or more salesmen reach at city c_i simultaneously, the $\boldsymbol{B_i}$ is equally divided among them.

$$x_i = \begin{cases} 1 & \textit{if} \text{ agent reached the city before the others} \\ 0 & \text{Otherwise} \end{cases}$$

$$Benefit = \sum_{i=1}^{n} B_i * X_i$$

ii. **Cost**. The current salesman has to pay a cost for their distance of travel c_{ij}, c_{ij} represent the cost for a salesman travelling from city ***i*** to ***j***.

$$Cost = \sum_{i=1}^{n-1} c_{i,i+1} - c_{n,1}$$

3.2. Payoff. for current salesman the payoff is computed by combining the benefit and cost:

$$Payoff = Benefit - Cost$$

4. Calculate the fitness for each salp as follows:

$$fitness_{salp} = \sum_{i=1}^{m} Payoff_i$$

- **Updating the Salps' Positions**

The most important step in SSA is how to update the positions of salps (leader and followers). This step has a prominent role in the performance of SSA as it works to enhance the ability of a SSA to efficiently explore and exploit the search space. In the SSA, in order to update the salp positions, three move operators are utilized; two operators for leader and one for followers:

- **Interchange or Exchange Operator**: Two different positions from each route (each salesman) are randomly selected and exchanged the cities located at these positions. This operator is adopted by leader for making minor changes to its position while the quality of the leader is improved.

- **Two-opt star operator**: This operator is adopted by leader for making big changes to its position while its quality is no longer improved. In this operator, two different segments from different salesmen are randomly selected and replaced.
- **Follower move operator**: the followers, in any situation, they move towards the leader. So, in order to update the follower salps' positions the following algorithm is introduced in this work:

Algorithm 2 Follower move operator

```
Input:-
No. of cities ($N\_cities$), No of salesmen ($NO$), size of swarm ($N$), the swarm
Processing:-
    1. $salp_0$= leader salp.
    2. For each follower salp ($salp_f$) do
    3. For each salesman's route ($R$) do
            i.   $C_{uv}$= set all cities as unvisited cities
            ii.  For each city ($c_f$) in $R$ do
            iii. Calculate the distances between $c_f$ and all unvisited
                 cities ($C_{uv}$).
            iv.  Select a unvisited city ($c_{un}$) that is the nearest to both
                 the $c_f$ and the corresponding city in the $salp_0$.
            v.   // Exchange the $c_f$ and $c_{un}$:
                 Temp = $salp_f.c_f$
                 $salp_f.c_f$ = $c_{un}$
                 $c_{un}$ = Temp
            vi.  Remove $c_{un}$ from $C_{uv}$
            $C_{uv} = C_{uv} - \{c_{uv}\}$
        vii.    The steps (i-v) are repeated for each salesman.
7. **Return** swarm
```

3. The Hybrid SSA

In this paper, the performance of SSA for solving the CTSP is investigated. Due to the fact that the SSA has been effective in exploration but not in exploitation, this section presents the way the SSA exploitation ability is enhanced. Integrating population-based and single-based meta-heuristics (SBH) has produced most efficient metaheuristics to address real word optimization problems (G. V. B. and D. V. O. Vansteenwegen, P., 2009).The SBH possesses an efficient exploitation process which has the ability to compensate for the insufficiency of the population based methods. Consequently, the present study proposes the hybrid SSA to develop its exploitation ability. In this hybridization, the hill climbing algorithm (HC) has been selected according to its strategy to improve the exploitation ability of the SSA. The HC takes place after the leader salp's position is updated. That is, the leader salp is used as an initial solution for the HC, see Figure 1.

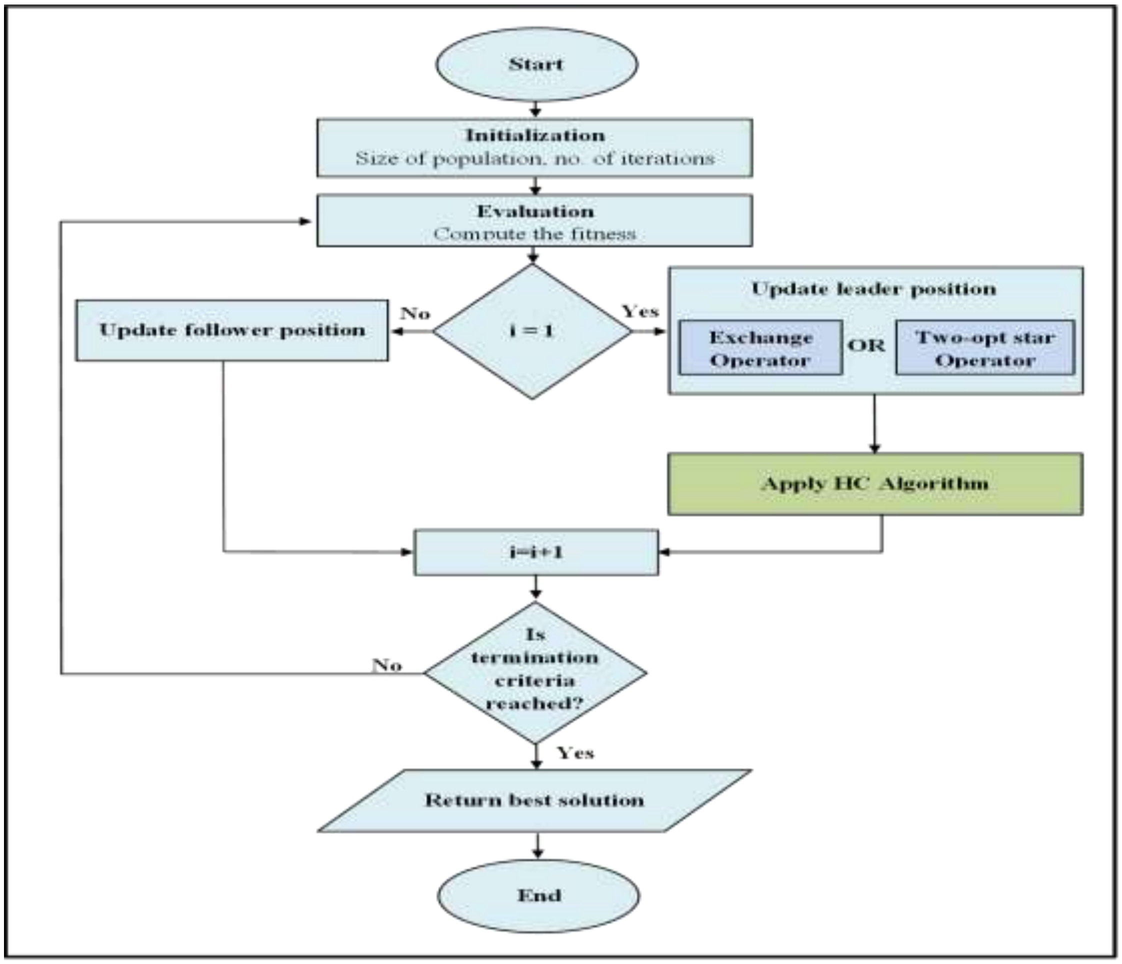

Figure 1 HSSA for CTSP.

So, while the SSA works to explore the search space, the HC works to exploit it. The enhancements proposed in this section will hopefully make SSA more effective in dealing with the CTSP. In this work, the HC works as follows: Given an initial solution S, HC generates a neighbor solution S` based on the **Exchange Operator.** S is replaced with S' if the quality of S' is better than S; otherwise, S' is rejected, and HC begins a new iteration. The search process will be repeated as long as the stopping criterion is not satisfied (Talbi, 2009) , (see Algorithm 3).

Algorithm 3 Hill Climbing Algorithm.

```
s = s_0 ; /∗Generate an initial solution s_0∗/
While not Termination Criterion Do
    Generate (N(s)) ; /∗Generation of candidate neighbors∗/
    If there is no better neighbor Then Stop;
    s = s′; /∗Select a better neighbor s′∈ N(s)∗/
Endwhile
Output Final solution found (local optima).
```

4. Experimental Setup

The proposed algorithms (HC, SSA and HSSA) are programmed in platform: - windows form, IDE: - Visual studio2019, framework:-.Net framework 4.5.2, language visual basic and executed by using PC with windows10, Intel processor core i5 CPU 2.10GHz, RAM 8.00GB. In this work, the appropriate values for the parameters of the proposed algorithms are determined based on the preliminary test or as suggested by previous studies. Finally, the parameter settings adopted in this work are summarized in Table 2.

Table 2 summarizes the parameter setting of standard SSA

Algorithm	Parameters	Value
SSA	Number of iterations	500
	Size of population	10
HC	Number of iterations	500

The CTSP Instances

According to (Kendall, G. & Li, 2013) and Mohannad (Hameed, 2016), two instances are introduced: two-salesmen CTSP on 20 cities and three-salesmen CTSP on 300 cities. In two-salesman CTSP with 20 cities, the cities are randomly distributed in a square space of 100X100. Each city is associated with its visiting payoff (set to be 150). The Euclidean distance between any two cities c_i and c_j is considered as the traveling cost between them. Note that the payoff of visiting a city is bigger than the longest cost between two cities, so the salesmen are motivated to visit all cities. Two salesmen with same traveling speed are present and they are at the first placed in different cities. In terms of CTSP with three salesmen and 300 cities, the 300 cities are randomly distributed in a square area of 100X100. The payoff of visiting every city is fixed to be 150. As in two-salesmen CTSP with 20 cities instance, the cost of travel is equivalent to the distance travelled based on Euclidean distance. The three salesmen with same travelling speed and they are at the first placed in three different cities.

5. Experimental Results

This experiment was designed to investigate the effectiveness of the proposed algorithms in solving the CTSP. For each tested instance, the best (Best) and the average (Avr) are reported. Table 3 illustrates the best results obtained by the standard SSA for CTSP, the first column represents the name of instance, second column shows solutions, third column represent payoff for salesman1 and the last column represent payoff for salesman2. For each instance there are two solutions (sol.1 and sol.2); sol.1 represents the best result based on salesman1 and sol.2 represents the best result based on salesman2.

Table 3 the best results of SSA with two salesmen and three salesmen respectively

Instances	Solutions	Payoff SM1	Payoff SM2	Payoff SM3
20-cities	**Sol.1**	**1082.10**	107.50	-
	Sol.2	102.50	**1139.87**	-
300-cities	**Sol.1**	**3051.54**	-429.99	-2717.75
	Sol.2	-543.96	**1300.24**	-773.77
	Sol.3	-1359.97	-453.67	**1762.24**

The results of standard SSA indicated that it suffers from slow convergence which prevents it from obtaining better results, see figure 2. This figure illustrates that at the beginning periods of the search, the SSA succeeds in tackling the CTSP via enhancing the solution quality. However, the SSA capability of enhancing the solution quality decreases gradually. This is because the SSA is efficient in exploration but not in exploitation.

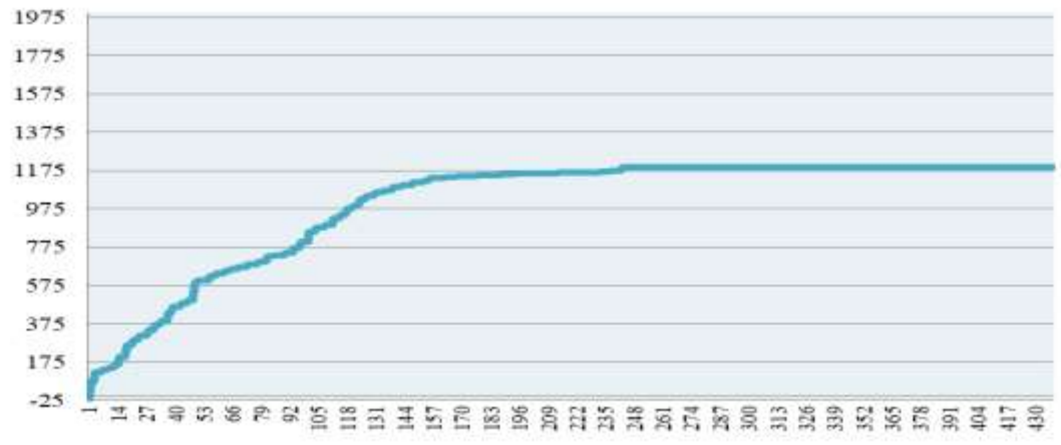

Figure 2 the Behavior of SSA during the Search (20_cities instance).

5.1. The Hybrid SSA Results

In order to compensating the weakness of SSA exploitation, the hybrid SSA (HSSA) emerged. In this hybridization, a single-based meta-heuristic (HC) was integrated with the SSA. This integration made use of SSA to explore the search space and the HC to exploit the search space. The results of SSA and HC are compared with these of the HSSA. This comparison is reported in tables 4 and 5 in terms of Best and Average, respectively.

Table 4 the best results of the standard SSA, HC and HSSA.

Instances	**SSA**		**HC**	**HSSA**
20-cities	Sol1	(1082.10,107.50)	(1238.90,199.34)	(**1633.12,**354.38)
	Sol2	(102.50,1139.87)	(198.16,1690.84)	(290.01,**1915.13)**
300-cities	Sol1	(3051.54,-429.99,-2717.75)	(4615.83,3014.01,4623.29)	(**12820.76**, 8042.83, 8922.50)
	Sol2	(311.37,744.1,-1138.62)	(3622.86,4522.26,4367.59)	(7442.01, **14141.33**, 8035.97)
	Sol3	(-1359.97, -453.67, 1762.24)	(2432.18,4126.27,6335.26)	(8487.22, 6548.03,**14889.72**)

Tables 4 show that HSSA obtain the best result than other algorithms. In this table, each algorithm has two or three solutions, each one represents the best result regarding to a certain salesman. Tables 5 show that HSSA obtain the maximum average than other algorithms.

Table 5 the average results of the standard SSA, HC and HSSA.

Instances	**SSA**	**HC**	**HSSA**
20-cities	(621.73, 619.84)	(580.47 , 1032.14)	(**918.91** , **1082.48**)
300-cities	(-55.99, -95.17,-89.24)	(3772.66,3847.63,5063.85)	(**7174.45, 7504.94, 10892.90**)

Tables 6 and 7 show that any salesman used HSSA to improve their route will obtain better results than others. In these tables, each algorithm has two or three solutions, each one represents the best result regarding to a certain salesman.

Table 6 The result of HSSA compared with SSA on 20- cities with two salesmen.

SM2 / **SM1**	**SSA**		**HSSA**	
SSA	1082.10	107.50	377.88	**1328.68**
	102.50	1139.87	-333.39	**1935.40**
HSSA	**1735.97**	-174.40	**1477.41**	559.64
	1269.51	370.22	290.01	**1773.62**

Table 7 the result of HSSA compared with SSA on 300-cities with three salesmen.

Algorithms	**300 cities**		
SSA, SSA ,SSA	**3051.54**	-429.99	-2717.75
	-543.96	**1300.24**	-773.77
	-1359.97	-453.67	**1762.24**
	12820.75	8042.82	8922.50

Figure 3 shows the behavior of HSSA during the search. This figure illustrates that the HSSA succeeds in tackling the CTSP by enhancing the solution quality during the search. This means that the suggested HSSA is efficient in exploration and exploitation.

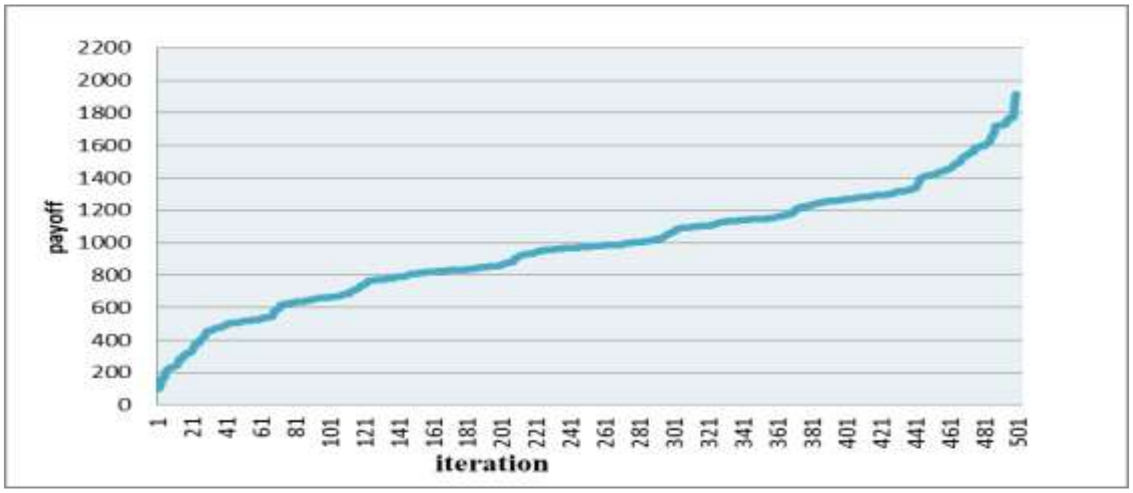

Figure 3 the Behavior of HSSA during the Search (20_cities instance).

5.2. Comparison of HSSA with the State of the Art Methods

To further investigate the effectiveness of the HSSA, the extra experiment is adopted to verify the efficiency of the HSSA in solving the CTSP through comparing the results of hybrid SSA with those obtained by the other algorithms in literature. These results obtained by different methods:

- Hyper_ H: Hyper heuristic proposed by (Kendall, G. & Li, 2013).
- SACO: Static Ant Colony proposed by (Hameed, 2016).
- PACO: Parallel Ant Colony proposed by (Hameed, 2016).
- PACO': Parallel Ant Colony with update plan proposed by (Hameed, 2016).

Table 8 shows these comparison result, first column represent the name of instance, second column represent the results of HSSA, third column represent the results of Hyper-H and the last column represent the SACO, PSACO and PSACO' respectively.

Table 8 the result of standard HSSA comparing with the result of Hyper_H and SACO , PSACO and PSAC' on 20 cities and 300-cities with two and three salesmen.

Instances		HSSA	Hyper_H	ACO	
	Sol.1	(**1633.12**, 354.38)		**SACO**	(1029.47 ,1034.14)

In terms of salesman1 and salesman2, these results show that HSSA in 20-cities instance obtained better results than Hyper_ H, SACO, PACO and PACO'. According to the 300-cities with three salesmen, this table shows that the HSSA obtained better results than others, regarding to salesman3 and competitive results regarding to salesman1 and salesman2.

These results demonstrate that the HSSA work well over 20 cities instance as it obtained best quality solution compared to others. And for the 300 cities instances, even though HSSA did not manage to beat the best known results, the obtained results for these instances are very competitive.

6. Conclusion

The present work contributes to solve the competitive travelling salesman problem (CTSP) aiming to reduce the cost and maximize the benefits via present a more efficient systems that can significantly reduce cost and maximize benefits. This aim was achieved by adopting the standard salp swarm algorithm (SSA) and utilizing its strong abilities to improve the quality of the obtained solutions. As any population based metaheuristics, the SSA suffers from the slow convergence problem due to its weak ability to exploit the search space. Thus, this paper proposes enhancing basic SSA to handle CTSP by utilizing its strong exploring ability and enhancing its exploitation ability. This enhancement achieves via Hybridizing the SSA with hill climbing algorithm (HC) which have strong abilities in exploiting the search space. In this hybridization, the SSA will be responsible for exploration and the HC will be responsible for exploitation. The SSA and hybrid SSA (HSSA) are applied on CTSP benchmark to test their validity. Results demonstrated that achieving balance between the hybrid SSA exploration and exploitation and preserving this balance during the search have significant effect on its efficiency. Thus, we concluded that the proposed hybridization managed to improve the effectiveness of basic SSA in getting good quality solutions for CTSP instances.

References

1. Shaikh, M., & Panchal, M. (2012). Solving Asymmetric Travelling Salesman Problem Using Memetic Algorithm. *International Journal of Emerging Technology and Advanced Engineering 2(11): 634-639.*
2. Nemati, K., Shamsuddin, M., & Kamarposhti, M. (2011). 'Using Imperial Competitive Algorithm for Solving Traveling Salesman Problem and Comparing the Efficiency of the Proposed Algorithm with Methods in Use.' *Australian Journal of Basic and Applied Sciences, 5(10): 540-543.*
3. Taba, M. (2009). 'Solving Traveling Salesman Problem with a NonComplete Graph',. *MSc Thesis, University of Waterloo, Waterloo, Ontario, Canada.*
4. Lawler E, Lenstra J, K. A. and S. D. (1986). The Traveling Salesman Problem: A Guided Tour of

Combinatorial Optimization. *John Wiley & Sons: Hoboken, NJ.*

5. Davendra, D. (2010). Traveling salesman problem theory and applications. *Janeza Trdine 9, 51000 Rijeka, Croatia.*

6. H., D. (2013). Combinatorial Optimization: Solution Methods of Traveling Salesman Problem. *MSc Thesis, Eastern Mediterranean University, Gazimağusa, North Cyprus.*

7. Applegate D, Bixby R, C. V. and C. W. (2006). The Travelling Salesman Problem: A Computational Study. *Princeton University Press: Princeton,.*

8. Kendall, G. & Li, J. (2013). 'Competitive Travelling Salesmen Problem: a Hyper-Heuristic Approach',. *Journal of the Operational Research Society 64(2): 208–216*.

9. Fekete, S., Fleischer, F., Fraenkel, A. & Schmitt, M. (2004). Traveling Salesmen in the Presence of Competition. *Theoretical Computer Science 313: 377-392.*

10. Mohtadi, M. & Nogondarian, K. (2014). Solving the Traveling Salesman Problem in Competitive Situations Using the Game Theory. *Applied Mathematics in Engineering, Management & Technology 2 (3):311-325.*

11. Li, J. & Kendall, G. (2015). A Hyper-Heuristic Methodology to Generate Adaptive Strategies for Games. *IEEE Transactions on Computational Intelligence and AI in Games: 1-10.*

12. Hameed, M. A.-S. (2016). Solving Competitive Traveling Salesmen Problem Using Ant Colony Algorithm'. *MSc Thesis, University of Alanbar.*

13. Esam Taha Yassen, Masri Ayob, M. Z. A. N. R. sabbar. (2012). Multi-parent insertion crossover for vehicle routing problem with time windows. *IEEE Conference on Data Mining and Optimization (DMO).*

14. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, M. S. (2017). Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. *Adv Eng Softw 114:163–191.*

15. Laith Abualigah , Mohammad Shehab , Mohammad Al Shinwan, H. M. A. (2019). 'salp swarm algorithm comprehensive survey. *Neural Computing and Applications*.

16. G. V. B. and D. V. O. Vansteenwegen, P., W. S. (2009). ''Iteratd local search for the team orienteering problem with time windows". *Comput. Oper. Res. 36(12) 3281-3290.,.*

17. Talbi, E. (2009). Metaheuristics from Design to Implementation, John Wiley & Sons, Inc. *New Jersey and Canada.*