U-Turning Ant Colony Algorithm for Solving Symmetric Traveling Salesman Problem

Saman M. Almufti[1], Awaz A. Shaban[2]
[1]Department of Computer Science, Nawroz University, Duhok, Kurdistan Region – Iraq
[2]Gara Institute,Duhok, Kurdistan Region - Iraq

ABSTRACT
This paper provides a new Ant based algorithms called U-Turning Ant colony optimization (U-TACO) for solving a well-known NP-Hard problem, which is widely used in computer science field called Traveling Salesman Problem (TSP). Generally U-Turning Ant colony Optimization Algorithm makes a partial tour as an initial state for the basic conventional Ant Colony algorithm. This paper provides tables and charts for the results obtained by U-Turning Ant colony Optimization for various TSP problems from the TSPLIB95.

Keywords: Traveling Salesman Problem (TSP), Ant System (AS), Swarm Intelligence, U-Turning Ant Colony Optimization Algorithm (U-TACO), Symmetric Traveling Salesman Problem (STSP).

1. Introduction

In Computer science field, Swarm Intelligence (SI) algorithms are computational intelligence techniques that study the collective behavior in decentralized systems (Almufti, 2017). In the nature real-ants lives in colonies, Ants of a colony are cooperates the process of food searching. Ants are unsystematically travels searching for a food source, ones an ant reaches a food source it returns to the colony by using a chemical substance called pheromone trail that ant deposits it in the way to the food and can be smelled by other ant. U-Turning Ant Colony Optimization algorithm is a new swarm intelligence algorithm based on the Ant colony algorithm in which ants randomly travel searching for the source of Food (Almufti, 2015). In this paper U-Turning Ant Colony Optimization algorithm is used to solve Symmetric Traveling Salesman Problem in which a salesman want to visit all cities in the graph and return to the start city with minimum time and cost (Almufti, 2015; Asaad, R., Abdulnabi, N. 2018).

2. Symmetric Traveling Salesman Problem (STSP)

TSP is one of widely used NP-hard problem in combinatorial optimization (Asaad, R., Abdulnabi, N. 2018). In a given graph a number of cities in which every city must be visited once and return to the starting city for completing a tour such that the length of the tour is the shortest among all possible tours (Almufti, 2015; Andrej Kazakov, 2009). Symmetric TSP (STSP) is a type of TSP in which the distance between city A and B is equal to the distance between city B and A.

In a graph G(C,A) the distance d(Ci,Cj) = d(Cj, Ci) and the number of tours in the Symmetric TSP (STSP) is (n-1)!/2 for n cities. Consequently the optimal (minimum length) tour to the STSP can be obtained by finding the summation of the length between cities of a permutation list as shown in equation (1) (Almufti, 2015; Asaad, R., Abdulnabi, N. 2018).

$$Btour = \left(\sum_{i=1}^{n-1} d_{\pi(Pi)\,\pi(Pi+1)}\right) + d_{\pi(Pn)\,\pi(P1)} \quad (1)$$

Where p is a probability list of cities with minimum distance between cities (pi and pi+1) (Federico Greco 2008; Asaad, R., Abdulnabi, N. 2018).

3. U-Turning Ant Colony Optimization (U-TACO)

U-Turning Ant Colony Optimization (U-TACO) is metaheuristic algorithm designed by Saman M. Almufti in his master thesis in (2015), it generally based on the same principles of common Ant System (AS) which was first represented in 1992 by Marco Dorigo in his PhD thesis as a nature-inspired metaheuristics for solving hard combinatorial optimization (CO) problems (Dorigo, 1992; Almufti, 2015; Almufti, 2017).

All algorithms that inspired the behavior of real Ant in searching for food source, basically depends on pheromone trail updating and the following of other ants to the pheromone smell: ants deposit pheromones in their way to the food source which takes attention of other swarm ants to the food source and the way that

Academic Journal of Nawroz University
(AJNU) Volume 7, No 4 (2018).
Regular research paper : Published 21 December 2018
Corresponding author's e-mail : saman.almofty@gmail.com

doi : 10.25007/ajnu.v7n4a270

they should take to the food, gradually the way that have more pheromone is the shortest way to the food (Dorigo, 1992; Almufti, 2015) as shown in figure (1).

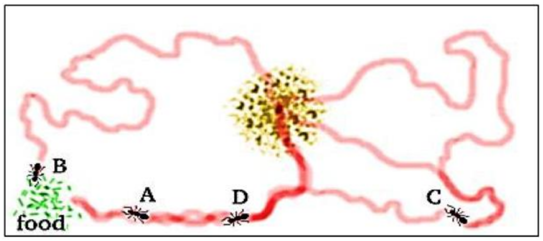

Figure 1: Behavior of real ants (Mute-net.sourceforge.net, 2015)

U-Turning Ant Colony Optimization (U-TACO) involve two phases:

Firstly: initializing pheromone matrix by UAnt to be used in second phase. UAnts makes a partial tour for the given graph and returns feedback in the same path as shown in figure (2), those partial tours leads to update the pheromone trail in the ways that seems to be useful in finding optimal tour by Ant (Almufti, 2015).

Secondly: finding the optimal or near-optimal tour for the problem by Ants, which make complete tours and goes back to the starting position as shown in figure (2).

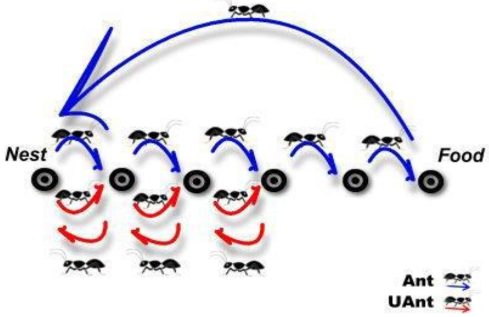

Figure 2: Ant and UAnt tour (Almufti, 2015)

Uants are not responsible of finding an optimal solution for a given problem, where UAnt mission is only initializing the pheromone trail matrix, which will be use by ant to construct an optimal solution (Almufti, 2015).

4. U-TACO and STSP

The traveling salesman problem (TSP) is an important factor in testing and improving the Ant System (AS) algorithms and algorithms that based on the behavior of Ant (Almufti, 2015). The first appearance of Ant System by Marco Dorigo in 1992 was tested on the travelling salesman problem (Dorigo, 1992). As all Ant based algorithms U-Turning Ant Colony Optimization (U-TACO) was first tested in TSP.

U-Turning Ant Colony Optimization (U-TACO) use Ant and UAnt in the process of finding the best tour for the Salesman, number of Ant is equal to the number of cities that the salesman want to visit them, where as the number of UAnt can take any value between 1 and number of cities. Both UAnts and Ants are following the same procedure in starting, updating pheromone trail and pheromone evaporation process, the only deferens occurs in the length and the way that it take back to the start city (Almufti, 2015).

The algorithm follows two phases; the first phase is making partial tours by UAnt which consist of the following steps.

1.Identify the length l that the UAnt should visit in the partial tour (1<= l <= number of cities in graph).

2.Randomly allocate UAnts in cities.

Every UAnt chose the next city upon a probability-based function depend on pheromone trail and heuristic value, called random proportional rule (2), shows the probability of UAnt (k) that locate at the city (i) to visit city (j) (Almufti, 2015).

$$\boldsymbol{P_{ij}^{k}} = \frac{[\boldsymbol{\tau_{ij}}]^{\boldsymbol{\alpha}}.[\boldsymbol{\eta_{ij}}]^{\boldsymbol{\beta}}}{\sum_{\mathbf{1}\in\mathcal{N}_i^k}[\boldsymbol{\tau_{ij}}]^{\boldsymbol{\alpha}}.[\boldsymbol{\eta_{ij}}]^{\boldsymbol{\beta}}} \qquad (2)$$

Where (η_ij) is a priori available heuristic value, (dij) is the distance between cities (i and j), (τ_ij) is the pheromone trail matrix, α is a parameter to regulate the influence of τ_ij and β is a parameter to regulate the influence of η_ij, and ($\mathcal{N}_i^k$) is the set of the neighborhood that ant (k) haven't visit them yet (Almufti, 2015).

3.UAnt visit l cities.

4.Update pheromone trail matrix. Each UAnt after adding new city to its tour makes a local pheromone update as shown in (3) (Almufti, 2015).

$\tau_{ij}new=\tau_{ij}old+\Delta\tau_{ij}$ (3)

5.Gradually the pheromone evaporates in the path (Almufti, 2015).

$\tau_{ij}\boldsymbol{new} = (\mathbf{1} - \boldsymbol{\psi}).\,\tau_{ij}old$ (4)

Whereas (0<ψ<1) is a constant value used for reducing pheromone trail.

After all UAnts complete their partial tour and updating and evaporating the pheromone trail, the first phase complete and the algorithm goes in to the second phase: a Complete tour by Ant.

In the second phase the Ant follow the same steps (initialize, allocate, pheromone trail updates and evaporations) in the first phase but here the ant makes a complete tour and goes to the start city as shown in figure 3.

doi : 10.25007/ajnu.v7n4a270

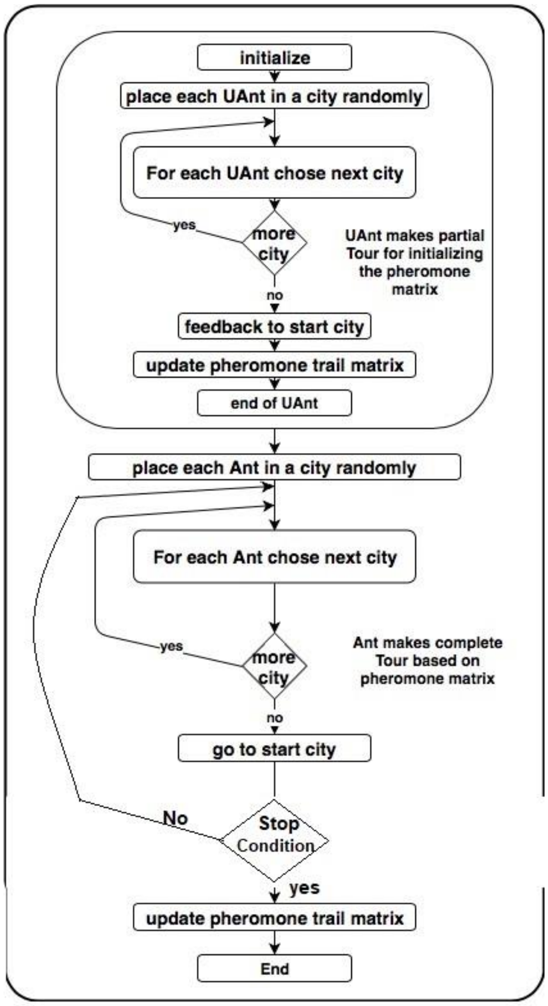
initialize
place each UAnt in a city randomly
For each UAnt chose next city
yes
more city
no
UAnt makes partial Tour for initializing the pheromone matrix
feedback to start city
update pheromone trail matrix
end of UAnt
place each Ant in a city randomly
For each Ant chose next city
yes
more city
no
Ant makes complete Tour based on pheromone matrix
go to start city
No
Stop Condition
yes
update pheromone trail matrix
End

Figure 3: U-Turning Ant Colony flowchart

5. Experimental Results:

This section presents the performance of using U-Turning Ant Colony Optimization (U-TACO) to solve STSP. The algorithms tested in some instances from the TSPLIB library (Karaboga D, Basturk B. 2007). The experiments have been performed on a PC with 2 GHz processor and 8 GB RAM memory running MATLAB R2015b in windows 10. Each instance is run for 100 times.

UAnt can makes partial tour of different length between 1 and total number of city, here we tried three different value for tour length 1/4, 1/2, and 3/4 of city numbers for different TSP problem. The results obtained in 50 iterations for UAnt to initialize pheromone trail matrix and 20 iterations for Ant to construct a complete tour. It was found from the results of (Lin105, Pr107, Eil51) STSP problem that U-TACO have the best results when UAnt make tour with length equal to 1/2 of number of cities in the problem as shown in figure 4, figure 5, and figure 6.

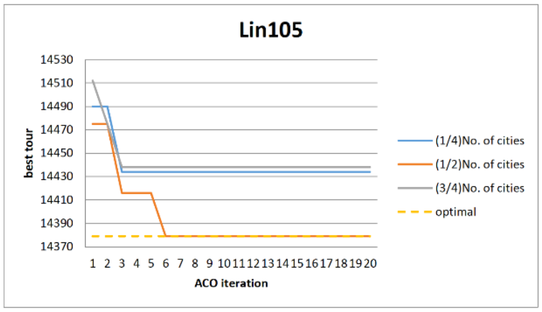

Figure 4: U-TACO for Lin105 over length 1/4, 1/2, and 3/4 of city numbers

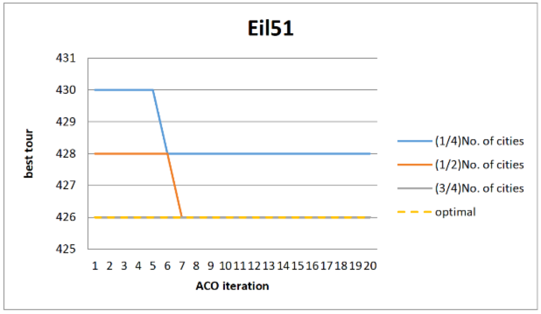

Figure 5: U-TACO for Eil51 over length 1/4, 1/2, and 3/4 of city numbers

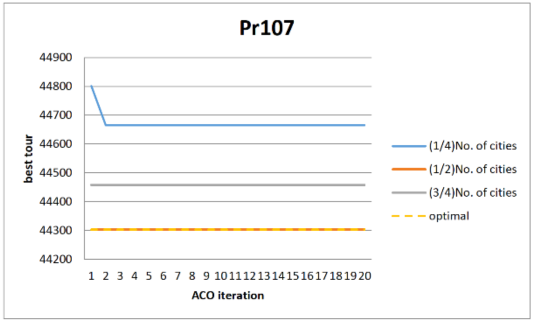

Figure 6: U-TACO for Pr107 over length 1/4, 1/2, and 3/4of city numbers

By considering figures (4, 5, and 6), in this paper simulate U-TACO when UAnt had make a partial tour of length 1/2 * number of cities for the collection of STSP.

Table 1: U-TACO results over 20 iteration

STSP problem	Optimal Tour	U-TACO	Time	# of function Evaluation	Optimal foundation
att48	10628	10628	72.608541	210960	Iteration=2 Time=56.832731 Evaluation= 21096
att532	27686	28218	43566.032999S	2338140	Not
berlin52	7542	7542	92.900329s	228540	Iteration=3 Time=76.081437s Evaluation=34281
Lin105	14379	14379	467.574097s	461475	Iteration=1 Time=358.475811s Evaluation=23073
Ch150	6528	6547	1368.303153s	659250	Not
Eil51	426	427	75.529904s	224145	Not
Eil76	538	538	349.082318s	334020	Iteration=7 Time=313.814483s Evaluation=116907
Eil101	629	629	439.032526s	443895	Iteration=8 Time=367.860212s Evaluation=177558
fl417	11861	12045	20596.700821s	1832715	Not
kroA100	21282	21282	415.284487s	439500	Iteration=6 Time=344.404586s Evaluation=131850
kroA150	26524	26618	1243.611930s	659250	Not
kroA200	29368	29472	2808.580653s	879000	Not
kroB100	22141	22200	439.673347s	439500	Not
kroB150	26130	26242	1333.134116s	659250	Not
kroC100	20749	20798	446.163972s	439500	Not
kroD100	21294	21455	434.844920s	439500	Not
Lin318	42029	42749	14079.427485s	1397610	Not
Pr76	108159	108487	202.229656s	334020	Not
Pr107	44303	44303	524.576606s	470265	Iteration=1 Time=392.417070s Evaluation=23513
Pr124	59030	59030	680.470220s	544980	Iteration=5 Time=545.294855s Evaluation=136245
Pr136	96772	100702	884.805854s	597720	Not
Pr144	58537	58669	1155.302216s	632880	Not
Pr152	73682	73867	1550.821031s	668040	Not
Rat99	1211	1213	981.875703s	435105	Not
Rd100	7910	7935	479.834143s	439500	Not
St70	675	679	212.419589s	307650	Not
Tsp225	3916	3928	3400.762717s	88500	Not

Table (1) presents the simulation results (tour length, time elapsed and number of function evaluation) for solving 27 STSP problems from the TSPLIB95 library by using U-TACO Algorithm. All problems were solved with 20 U-TACO iterations and the parameter that determine the influence of pheromone trail α=1 and heuristic information β=2, for each TSP problem number of ants equal to number of cities over partial tour of length 1/2 * number of cities for the collection of STSP, most of the simulation results were near-optimal and the optimal value for 6 STSP (att48, berlin52, Lin105, Eil76, Eil101, kroA100, Pr107, and Pr124) problems are founded.

6. Conclusion

U-Turning Ant Colony Optimization U-TACO) is used to solve symmetric TSP problems over 50 iterations for UAnts and 20 iterations for Ants, the results were very good (optimal / near optimal) with a small difference between the optimal tour. But generally using U-TACO takes a long time in solving the problem and finding the number of function evaluations.

References

Almufti, S. (2015). U-Turning Ant Colony Algorithm powered by Great Deluge Algorithm for the solution of TSP Problem. [online] Hdl.handle.net. Available at: http://hdl.handle.net/11129/1734 [Accessed 5 Aug. 2018].

Almufti, S. (2017). Using Swarm Intelligence for solving NPHard Problems. Academic Journal of Nawroz University, 6(3), pp. 46-50. doi:10.25007/ajnu.v6n3a78

Andrej Kazakov, (2009), Travelling Salesman Problem: Local Search and Divide and Conquer working together

Asaad, R., Abdulnabi, N. (2018). Using Local Searches Algorithms with Ant Colony Optimization for the Solution of TSP Problems. Academic Journal Of Nawroz University, 7(3), 1-6. doi:10.25007/ajnu.v7n3a193

Dorigo, M. (1992). Optimization, Learning and Natural Algorithms, PhD thesis, Politecnico di Milano, Italy.

Federico Greco, (2008), Travelling Salesman Problem .

Mute-net.sourceforge.net, (2015). MUTE: Simple, Anonymous File Sharing. [online] Available at: http://mute-net.sourceforge.net/howAnts.shtml [Accessed 20 July. 2018] .