Optimize TSP using Ant Colony System using Java

Shatha Habeeb[1] and Zainab Sadiq[2]

[1]University of Technology, Computer Sciences, Baghdad, Iraq.
[2]Al-Mustansiriya University, Baghdad, Iraq

Abstract

The travelling salesman problem (TSP) probably is the most prominent problem in combinatorial optimization. It is simple definition along with its notorious difficulty has stimulated many efforts to find an efficient algorithm. In this research use Ant Colony System to solving TSP and generating good solutions to both. The work has been extended to calculate the correlation coefficient between the number of nodes and number of iterations.

Keywords: TSP, Ant Colony System.

تحسين مشكلة البائع المتجول باستخدام نظام مستعمرة النمل باستخدام جافا

شذى حبيب[1] وزينب صادق[2]

[1]علوم حاسبات، الجامعة التكنولوجية، بغداد، العراق
[2]الجامعة المستنصرية، بغداد، العراق

خلاصة

مشكلة البائع المتجول (TSP) هي على الأرجح المشكلة الأبرز في التحسين الاندماجي. ان التعريف البسيط لهذه المشكلة مع الصعوبة الكبيرة قد تم تمثيلها (ومازالت تمثل) حيث الكثير من الجهود بذلت لإيجاد خوارزمية فعالة. في هذا البحث تم استخدام مستعمرة النمل في حل مشكلة البائع المتجول (TSP) وتوليد الحلول الجيدة. وقد تم عمل حساب معامل الارتباط بين عدد العقد وعدد مرات التكرار .

كلمات مفتاحية: مشكلة البائع المتجول، التحسين الأندماجي.

1. Introduction

1.1. Travelling Salesman Problem (TSP)

The travelling salesman problem (TSP) is an NP-hard problem in combinatorial optimization studied in operations research and theoretical computer science. The task is to find the shortest possible route that visits each city exactly once and returns to the origin city . The problem was first formulated as a mathematical problem in

1930 and is one of the most intensively studied problems in optimization. It is used as a benchmark for many optimization methods. Even though the problem is computationally difficult, a large number of heuristics and exact methods are known, so that some instances with tens of thousands of cities can be solved. A salesman must visit ***n*** cities, passing through each city only once, beginning from one of them which is considered as his base, and returning to it. The cost of the transportation among the cities (whichever combination possible) is given. The program of the journey is requested, that is the order of visiting the cities in such a way that the cost is the minimum[1].

As in figure (1), let's number the cities from 1 to *n* ,and let city 1 be the city-base of the salesman. Also let's assume that c(i,j) is the visiting cost from i to j. There can be c(i,j)<>c(j,i).Apparently all the possible solutions are *(n-1)!*.Someone could probably determine them systematically, find the cost for each and everyone of these solutions and finally keep the one with the minimum cost. These requires at least *(n-1)!* steps.

If for example there were 21 cities the steps required are *(n-1)!=(21-1)!=20!* steps. If every step required a *m.sec* we would need about 770 centuries of calculations. Apparently, the exhausting examination of all possible solutions is out of the question. Since we are not aware of any other quick algorithm that finds a best solution we will use a heuristic algorithm. According to this algorithm whenever the salesman is in town ***i*** he chooses as his next city, the city ***j*** for which the ***c(i,j)*** cost, is the minimum among all ***c(i,k)*** costs, where ***k*** are the pointers of the city the salesman has not visited yet. There is also a simple rule just in case more than one cities give the minimum cost, for example in such a case the city with the smaller ***k*** will be chosen. This is a greedy algorithm which

selects in every step the cheapest visit and does not care whether this will lead to a wrong result or not [2].

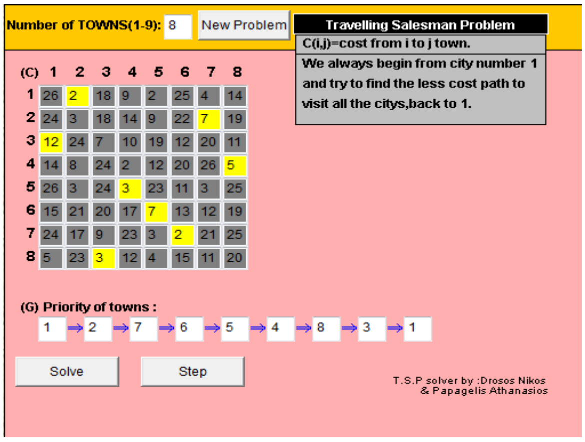


Figure (1) Traditional TSP Solving.

1.2. Ant colony

An initial look at ants in nature does not give an impression of an animal with a high IQ, but a closer look reveals that they are highly efficient in at least one task. Finding the shortest route between two points by starting from the hive they are prone to walk randomly around until they find a point of interest, e.g. a food source. When traveling back to the hive, they will deposit a chemical substance called pheromone as they go, which will help them find their way back to where they came from. When other ants encounter the path of pheromone, they will follow it, becoming less random in their movement. These will then also deposit pheromone, strengthening the already existing path. Because pheromone is a volatile substance, a constant stream of ants is required to keep up the strength of the trail. This means that if a shorter trail exists, the power of this trail's pheromone will be stronger, as the ants will traverse the trail in a shorter amount of time, while the pheromone still evaporates at the same speed. After a (relatively) short time span, the majority of the ants will therefore be following the shortest path, as this path has the strongest pheromone[3].

2. Design of proposal

In this paper problem has been resolved in a manner seller mobile ant as it reaches the best way to travel from one town to another in order to visit all cities once and using Java to solve this problem and examples of movement from Basra to Mosul and all the cities of Iraq. When traveling back to the hive, they will deposit a chemical substance called *pheromone* as they go, which will help them find their way back to where they came from. When other ants encounter the path of pheromone, they will follow it, becoming less random in their movement. These will then also deposit pheromone, strengthening the already existing path. Because pheromone is a volatile substance, a constant stream of ants is required to keep up the strength of the trail. This means that if a shorter trail exists, the power of this trail's pheromone will be stronger, as the ants will traverse the trail in a shorter amount of time, while the pheromone still evaporates at the same speed. After a (relatively) short time span, the majority of the ants will therefore be following the shortest path, as this path has the strongest pheromone[4,5,6].

$$P_{i,J}^{k}(t) = \frac{[\tau_{i,J}(t)]^{\alpha} \cdot [\eta_{i,J}]^{\beta}}{\sum u \in J_i^k [\tau_{i,u}(t)]^{\alpha} \cdot [\eta_{i,u}]^{\beta}} \qquad \text{if } J \in J_i^k$$

τ_{ij} is the amount of pheromone on arc i,j α is a parameter to control the influence of τ_{ij} η_{ij} is the desirability of arc i,j (a priori knowledge, typically $1/d_{i,j}$) β is a parameter to control the influence of η_{ij} . We take the 12 cities will be an Iraqi, he moves salesman through these cities from the city to another using ACO method.

```
Loop
   Randomly position m artificial ants on n cities
   For city=1 to n
      For ant=1 to m
         {Each ant builds a solution by adding one city after
         the other}
         Select probabilistically the next city according to
                     exploration and exploitation mechanism
         Apply the local trail updating rule
      End for
   End for
   Apply the global trail updating rule using the best ant
Until End_condition
```

Description of the Proposal TSP algorithm

Example

From ACO and TSP Let $P = 0.1, \alpha = 0.1, \beta = 2, Q = 0.9$

Random number = R = 0.39 , 0.16, 0.51, 0.01, 0.03, 0.04

Nineveh =A , Salahuddin = B, Diyala =C , Baghdad =D , Babylon= E, Wasit =F, Qadisiyah = G , Dhi Qar = H Muthanna = I , Anbar = J, Maysan =K, Basra = L

Solve the problem includes 5 steps, each step determines the ants move from one city to another, where is calculated according to the equation mentioned above. Set a limit to the distance between the two cities based on the pheromone left by ants during the movement and the way that has the highest pheromone is based, has been in this example to determine distances between cities as well as alpha and beta and the value of pheromone**.**

Step 1

$$\sum_{A}[A]^{\alpha}.[J]^{\beta}+[A]^{\alpha}.[B]^{\beta}$$

$$\sum_{A}[1]^{0.1}.[1/25]^{2}+[1]^{0.1}.[1/15]^{2}$$

$$= 0.0016 + 0.004$$
$$= 0.0056$$

$$P_{A,J}^{1}(1)=\frac{0.0016}{0.0056}=0.286..$$

$$P_{A.b}^{1}(1)=\frac{0.004}{0.0056}=0.714..$$

$$R_1 \prec A.B$$
$$0.39 \prec 0.714$$
$$\therefore A \xrightarrow{*} B$$

$$\text{Cost} = 15$$

$$\text{Pheromone, } \Delta\tau_{ij}(t) = \frac{Q}{\text{cost}} = \frac{0.9}{15} =$$

Step 2

$$\sum_{B}[B]^{\alpha}.[A]^{\beta}+[B]^{\alpha}.[F]^{\beta}+[B]^{\alpha}.[C]^{\beta}+[B]^{\alpha}.[D]^{\beta}$$

$$\sum_{B}[1]^{0.1}.[1/15]^{2}+[1]^{0.1}.[1/35]^{2}+[1]^{0.1}.[1/33]^{2}+[1]^{0.1}.[1/20]^{2}$$

$$= 0.0044 + 0.0008 + 0.0009 + 0.0025$$
$$= 0.0086$$

$$P_{B.A}^{1}(2)=\frac{0.0044}{0.0086}=0.511..$$

$$P_{B.F}^{1}(2)=\frac{0.0008}{0.0086}=0.093..$$

$$P_{B.C}^{1}(2)=\frac{0.0009}{0.0086}=0.104$$

$$P_{B.D}^{1}(2)=\frac{0.0025}{0.0086}=0.290$$

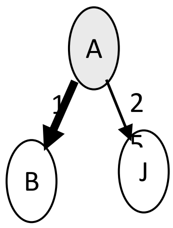
$R_2 \prec B.F$

$0.16 \prec 0.290$

$\therefore B \xrightarrow{*} D$

$\text{Cost} = 35$

$\text{Pheromone, } \Delta\tau_{ij}(t) = \frac{Q}{\text{cost}} = \frac{0.9}{35} =$



Step 3

$$\sum_D [D]^\alpha.[F]^\beta + [D]^\alpha.[E]^\beta + [D]^\alpha.[J]^\beta$$

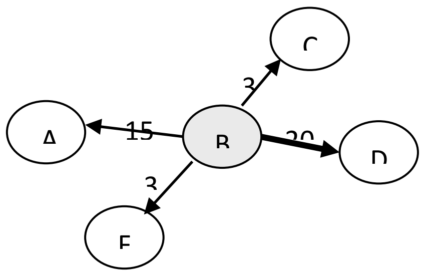
$$\sum_D [1]^{0.1}.[1/45]^2 + [1]^{0.1}.[1/15]^2 + [1]^{0.1}.[1/20]^2$$

$$= 0.0004 + 0.0044 + 0.0025$$
$$= 0.0073$$

$$P^1_{D.F}(3) = \frac{0.0004}{0.0073} = 0.054$$

$$P^1_{D.E}(3) = \frac{0.0044}{0.0073} = 0.602$$

$$P^1_{D.J}(3) = \frac{0.0025}{0.0073} = 0.342$$



$R_3 \prec D.E$

$0.51 \prec 0.602$

$\therefore D \xrightarrow{*} E$

$\text{Cost} = 50$

$\text{Pheromone}, \Delta\tau_{ij}(t) = \frac{Q}{\text{cost}} = \frac{0.9}{50} =$

Step 4

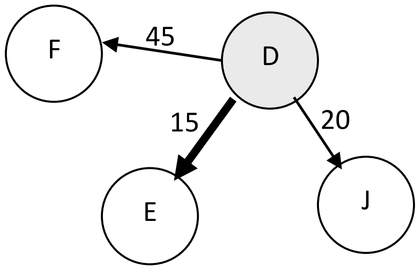
$$\sum_E [E]^\alpha.[G]^\beta + [E]^\alpha.[K]^\beta + [E]^\alpha.[J]^\beta + [E]^\alpha.[F]^\beta + [E]^\alpha.[I]^\beta$$

$$\sum_J [1]^{0.1}.[1/15]^2 + [1]^{0.1}.[1/40]^2 + [1]^{0.1}.[1/40]^2 + [1]^{0.1}.[1/20]^2 + [1]^{0.1}.[1/33]^2$$

$$= 0.0044 + 0.0006 + 0.0006 + 0.0025 + 0.0009$$
$$= 0.009$$

$$P^1_{E.G}(4) = \frac{0.0044}{0.009} = 0.48$$

$$P^{1}_{E.K}(4)=\frac{0.0006}{0.009}=0.006$$



$$P^{1}_{E.J}(4)=\frac{0.0006}{0.009}=0.006$$

$$P^{1}_{E.F}(4)=\frac{0.0025}{0.009}=0.277$$

$$P^{1}_{E.I}(4)=\frac{0.0009}{0.009}=0.1$$

$R_4 \prec E.K$

$0.0.06 \prec 0.06$

$\therefore E \xrightarrow{*} K$

$\text{Cost}=90$

$$\text{Pheromone}\,,\Delta\tau_{\text{ij}}(\text{t})=\frac{\text{Q}}{\text{cost}}=\frac{0.9}{90}=$$

Step 5

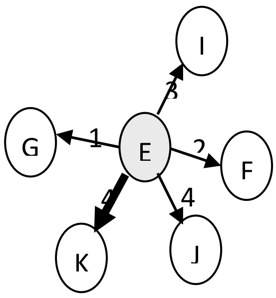
$$\sum_{k}[K]^{\alpha}.[J]^{\beta}+[K]^{\alpha}.[F]^{\beta}+[K]^{\alpha}.[H]^{\beta}+[K]^{\alpha}.[L]^{\beta}+[K]^{\alpha}.[G]^{\beta}$$

$$\sum_{K}[1]^{0.1}.[1/15]^{2}+[1]^{0.1}.[1/40]^{2}+[1]^{0.1}.[1/25]^{2}+[1]^{0.1}.[1/55]^{2}+[1]^{0.1}.[1/150]^{2}$$

$$=\ 0.0044\ +0.0006+0.0016+0.0003+0.00004$$

$$=0.00694$$

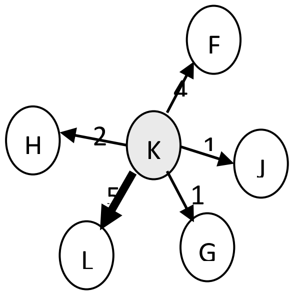
$$P^{1}_{K.J}(5)=\frac{0.0044}{0.00694}=0.634$$



$$P^{1}_{K.F}(5)=\frac{0.0006}{0.00694}=0.086$$

$$P^{1}_{k.H}(5)=\frac{0.0061}{0.00694}=0.250$$

$$P^{1}_{K.L}(5)=\frac{0.0003}{0.00694}=0.0.43$$



$$P^{1}_{K.G}(5)=\frac{0.00004}{0.00694}=0.005$$

$R_5 \prec\!| K.L$

$0.03 \prec 0.043$

$\therefore H \xrightarrow{*} L$

$\text{Cost} = 145$

$$\text{Pheromone}, \Delta\tau_{ij}(t) = \frac{Q}{\text{cost}} = \frac{0.9}{145} =$$

Step 6

$$\sum_{L} [L]^{\alpha}.[I]^{\beta} + [L]^{\alpha}.[H]^{\beta}$$

$$\sum_{L} [1]^{0.1}.[1/200]^{2} + [1]^{0.1}.[1/55]^{2}$$

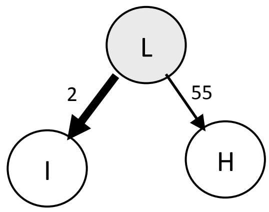
$$= 0.000025 + 0.0003$$
$$= 0.000325$$

$$P^{1}_{L.I}(6) = \frac{0.000025}{0.000325} = 0.07 \qquad P^{1}_{L.H}(6) = \frac{0.0003}{0.000325} = 0.92$$

$R_6 \prec L.I$

$0.0.04 \prec 0.07$

$\therefore L \xrightarrow{*} I$ Gool



$\text{Cost} = 345$

$$\text{Pheromone}, \Delta\tau_{ij}(t) = \frac{Q}{\text{cost}} = \frac{0.9}{345} = 0.0026$$

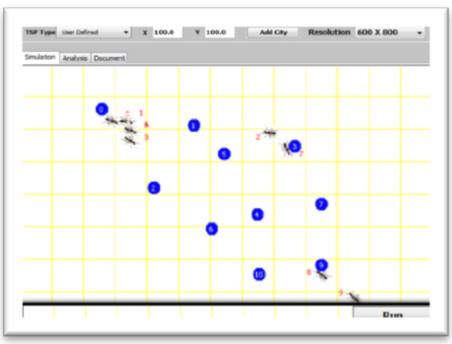


Figure (2) select number of cites and position.

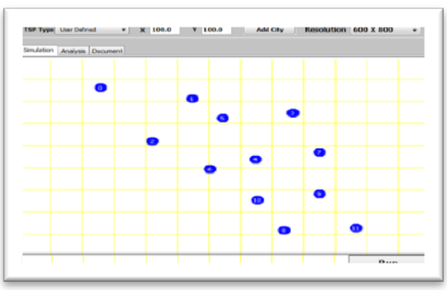


Figure (3) ant chose the cites randomize.

3- The Implementation of the Proposal System

The implementation of the proposal done by using java language the application consists of several interfaces start to select number of cites to visit and location, ant chose the cites randomize , result path to movement ant from cite to anther show in fig.1,2 and 3.

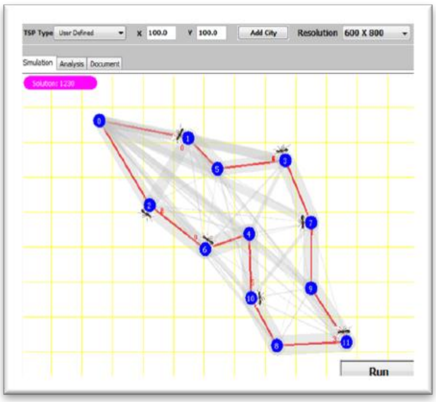


Figure (4): Path to movement ant from cite to anther

4-Conclusions:

1. There are many ways in which TSP can be improved so that the number of tours needed to reach a comparable performance level, making its application to larger problem instances feasible.
2. The elimination of closed-circuit condition that may occur in TSP.
3. Rely on the ants gives the best solutions in the shortest time in the process of movement between cities.

References

1. M´elanie Bourgeois, Gilbert Laporte, and Fr´ed´eric Semet. Heuristics for the black and white traveling salesman problem. Comput. Oper. Res., 30(1):75–85, 2003.
2. Blum C. Beam-ACO—Hybridizing ant colony optimization with beam search: An application to open shop scheduling. Computers & Operations Res 2005;32(6):1565–91.
3. Colorni, M. Dorigo and V. Maniezzo, Distributed Optimization by Ant Colonies, Proceedings of the First European Conference on Artificial Life, MIT Press, Cambridge, MA, 1992.
4. Bianchi, L., L.M. Gambardella, M. Dorigo (2002) An ant colony optimization approach to the probabilistic traveling salesman problem. In *Proceedings of PPSN-VII, Seventh International Conference on Parallel Problem Solving from Nature*, Lecture Notes in Computer Science. Springer Verlag, Berlin, Germany, pp 883-892.
5. Bullnheimer, B., R.F. Hartl, C. Strauss (1999) A new rank-based version of the ant system: a computational study, *Central European Journal of Operations Research* 7(1):25-38.
6. den Besten, M., T. Stützle, M. Dorigo (2000) Ant colony optimization for the total weighted tardiness problem, In *Proceedings Parallel Problem Solving from Nature: 6th international conference*, Lecture Notes in Computer Science. Springer Verlag, Berlin, Germany, pp 611-620.