Ant colony system solving capacitated location-allocation problems on a line

A. Vlachos*

Department of Informatics

University of Piraeus

80, Karaoli & Dimitriou Str.

Piraeus 185 34

Greece

Abstract

Ant Colony System (ACS) Algorithm is biologically inspired from the behaviour of colonies of real ants. In this project we apply a modification of ACS, especially on update of pheromone, for the solution of the problem: capacitated location – allocation problem on a line (CLAAL). Computational results demonstrate that ACS is a more effective meta-heuristic for the CLAAL.

Keywords : *Location-allocation, ant colony system, meta-heuristic.*

1. Introduction

Location-allocation problems along a line are important to be studied because of the following two reasons: On one hand for economic and technological reasons distribution networks are constructed in hierarchies (high-level distribution channels from which low-level channels branch). On the other hand branching facilities too, may cause high costs and so their number locations along the lines and allocation of destinations to them are important components in the design of a system. For these reasons it is important to solve location-allocation problems on a line.

Distribution networks often have a hierarchical structure, similar to the nerve system. Large distribution channels branch into smaller

**E-mails*: `avlachos@unipi.gr, aris532003@yahoo.gr`

Journal of Information & Optimization Sciences
Vol. 27 (2006), No. 1, pp. 81–96
 0252-2667/06 $2.00 + 0.25

channels and these in turn branch again. Obviously, the primary channels' features (size, speed, conductivity, etc.), result to high costs. For that reason, in order to diminish capital expenditure these channels are constructed as short as possible in straight lines. Given a line we can define on it where we must locate the sources and positive weights of a finite set of n destinations. Then determine the number of facilities and their locations along the line and also the allocation of destinations to them. The aim of the article is double, set-up plus operation costs are to be minimized [1].

The location theory study the Francis, Ginis and White [2]. Capacitated models are considered in Mirchandani et al [3], including a discussion of application-examples.

In this paper we will develop a modified form of Ant Colony System (ACS) [4] algorithm, especially on update of pheromone for the optimization of capacitated Location – Allocation problems on a Line (CLAAL).

The remainder of this paper is organized as follows. In section 2 we present the basic idea of the ant colony system. In section 3 we present the implementation of ACS to the CLAAP. Section 4 presents the case study and in Section 5 we conclude.

2. Ant colony system

The Ant Colony System (ACS) algorithm, proposed by Dorigo and Cambardelila [5], is one of the most recent and meta-heuristics for combinatorial optimization problems. The ACS is inspired by the research on the behavior of real ants. Real ants are capable of finding the shortest path from a food source to their nest without using visual cue [6].

They communicate information concerning food sources via a chemical substance, called pheromone [7]. An ant leaves some quantities of pheromone on the ground and marks the path by a trail of this substance. The pheromone quantity depends on the length of the path and quality of the discovered food source. An ant chooses a specific path in correlation with the intensity of the pheromone. Since ants passing through a food source by a shorter path will come back to the nest sooner than ants via longer paths, the shorter path will have a higher traffic density than the longer one. The pheromone trail evaporates over time if no more pheromone is laid down.

In the ACS used artificial ants that following assumptions are made:

(1) Each artificial ant has some memory.
(2) Artificial ants are not completely blind.
(3) Artificial ants live an environment, which is discrete in time.

The general principles for the ACS simulation of real ant behavior are as follows:

1. *Initialization*. The initialization of the ACS includes two parts: the problem graph representation and the initial ant distribution. First the underlying problem should be represented in terms of a graph, $G = (N, E)$, where N denotes the set of nodes, and E the set of edges. The graph is connected, but not necessarily complete, such that the feasible solutions to the original problem correspond to paths on the graph which satisfy problem-domain constraints. Second, a number of ants are arbitrarily placed on the nodes chosen which are chosen randomly.
2. *Transition rule*. According to the problem-domain constraints are set, some nodes could be marked as inaccessible for an ant. In the traveling salesman problem (TSP) [8], the aim of a single ant is to j find a salesman tour in the graph, whose nodes are the cities, and the edges connecting the cities have been initialized with some amount of pheromone, τ_0. The transition rule is probabilistic. For an ant k on node i, the selection of the next node j given by transition probability,

$$P_{ij}^k = \begin{cases} \dfrac{(\tau_{ij})(n_{ij})^\beta}{\sum\limits_{h \notin tabn_k} (\tau_{ih})(n_{ih})^\beta}\,,, & \text{if } j \notin tabu_k \\ 0\,, & \text{otherwise.} \end{cases} \tag{1}$$

Where,

τ_{ij} is the pheromone trail of edge (ij)
n_{ij} is the value of visibility of edge (i, j)
β is a control parameter
$tabu_k$ means the set of currently inaccessible nodes for the ant k according to the problem-domain constraints.

3. *Global update rule*. After all ants have completed their routes, a global update of pheromone trail takes place.

In order to make the search more directed, global updating is intended to provide a greater amount of pheromone to shorter routs and reinforce them. Therefore, only the globally best ant that found the best solution up to the current iteration of the algorithm is permitted to deposit pheromone. The level of pheromone is then changed as follows:

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \rho\cdot\Delta_{ij}(t) \tag{2}$$

where,

$$\Delta\tau_{ij}(t) = \begin{cases} 1/L^{gb} & \text{if } (i,j) \in \text{ global best tour} \\ 0 & \text{otherwise.} \end{cases}$$

The parameter ρ $(0 \leq \rho \leq 1)$ represents the pheromone evaporation and L^{gb} is the length of the globally best tour found up to the current iteration.

4. *Local update rule*. After having found a tour, an ant deposits pheromone information on the edges through which it went. It constitutes a local update of the pheromone trail according to the formula:

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \rho\cdot\tau_0 \tag{3}$$

where, $1-\rho$, $\rho \in [0,1]$ is the persistence rate of previous trails and τ_0 is the initial amount of pheromone on edge (i,j).

5. *Candidate list*. It is time-consuming to check all the towns, in the problem with big number of towns (especially concerning ATSP problems). For this reason, this list has a number (let c) of the most 'desirable', namely the nearest towns which rank from the nearest to the farthest. Only when all the towns which are on the list of candidates towns, also belong to the tabu list its possible the ants to roam to other towns.

3. Implementation of ant colony system algorithm for solving capacitated location-allocation problems on a line

The capacitated location-allocation problem on a line is modelled as follows, [1]:

$$\min Z = \sum_{j=1}^{n}\left[Fy_j + \sum_{i=1}^{n} w_i d(x_j,(a_i,b_i))z_{i,j}\right] \tag{4}$$

$$\text{subject to } \sum_{j=1}^{n} z_{i,j} = 1\,, \quad i = 1,\ldots,n \tag{5}$$

$$\sum_{i=1}^{n} z_{i,j} \leq Ky_j\,, \quad j = 1,\ldots,n \tag{6}$$

$$y_j = 0/1\,, \quad j = 1, \ldots, n \tag{7}$$

$$z_{i,j} = 0/1\,, \quad i = 1, \ldots, n,\ j = 1, \ldots, n. \tag{8}$$

Where

F denotes the set-up cost
K denotes the capacity of a facility

and

y_j indicates where the j th facility is set
$z_{i,j}$ indicates to which facility the i th destination-customer is allocated
w_i cost per unit distance for i th destination-customer

The function $d(x_j, (a_i, b_i))$ measures the distance between the j th facility located at x_j and the i th destination at (a_i, b_i).

For the problem stated above the *heuristic algorithm* that is based on ant colonies is used. Given that the algorithms for ant colonies are used mostly for discrete problems, and the problem stated above has continuous and discrete parameters some adjustments were made in order to use ant colonies.

Some alterations were made. The first is that in every position there is only a station. Station is the source of the food. The analysis of the algorithm is given below. A straight line with n stations is defined that are in an equal distance. Even though a straight line is mentioned, actually is for a straight line part which has a defined start and a defined finish line. The end from geometric view has no meaning to have an abscissa greater than the maximum abscissa of stations-customers. So, the limits to move about are known.

Transition Rule 1

Suppose that the ant k is on location i. Chose among the customer that has not been served (that are not in Tabu list so far (transition rule)). This is done using a roulette-like mechanism. Use a uniform random number generator, which produces numbers in the interval $[0,1]$. Then select customer j to be allocated

$$j = \begin{cases} \arg\max\{[\tau(i,u)] \cdot [n(i,u)]\}^{\beta}, & \text{if } q \leq q_0 \\ J\,, & \text{otherwise} \end{cases} \tag{9}$$

where

- $\tau(i,u)$ is the pheromone trail on edge (i,u), $n(i,u)$ is value of visibility of edge (i,u)
- q_0 is a parameter that it is selected by the user in the interval $[0,1]$
- q is a random parameter distributed randomly in the interval $[0,1]$
- J is a customer that has not been allocated to a source yet which is selected with probability

$$p_{iJ}^k(t) = \frac{\tau_{ij}(t)\cdot[\eta_{ij}(t)]^\beta}{\sum\limits_{l\notin J_1^k}\tau_{il}(t)\cdot[\eta_{il}(t)]^\beta}, \quad \text{if } j\notin J_1^k \tag{10}$$

among the other candidates (it is another roulette procedure, similar to the one used in genetic algorithms for the selection).

- J_1^k denotes the Tabu list which contains the information about which customers have been served (and by whom)
- $\eta_{ij}(t)$ denotes visibility and is the reverse of the 'stable-distance' between *station-customer* $\eta_{ij}(t) = 1/(w_j d_{ij})$
- $\tau_{ij}(t)$ denotes the pheromone quantity that is on the edge connecting a *station* i with a destination-customer j. During time $t = 0$ (first iteration), the pheromone is set to a starting value τ_0 which is very small.

Transition Rule 2

Suppose that the ant k is on location i which can serve no more customers. Then the ant has to 'jump' onto another location (not used before meaning that it is not in Tabu list $T_2^k(t)$ so far) on the line ant set a new source. This is done using again a roulette-like mechanism. Use a uniform random number generator, which produces numbers (r) in the interval $[0,1]$. Then select location p on the line to set up the new source

$$p = \begin{cases} \max\limits_{u\notin J_2^k}\{T_u(t)\}, & \text{if } r\le r_0 \\ 0, & \text{if } r > r_0 \end{cases} \tag{11}$$

where,

- $T_u(t)$ is the quantity of pheromone that exists in every station
- r_0 is a parameter that it is selected by the user in the interval $[0,1]$
- r is a random parameter evenly distributed in the interval $[0,1]$

- P is a location that has not been used yet which is selected with probability customer that has not been allocated to a source yet which is selected with probability

$$\phi_P^k(t) = \frac{T_l(t)}{\sum\limits_{l \notin J_2^k} T_l(t)}\,, \text{ if } P \notin J_2^k \tag{12}$$

- J_2^k are the location on the line already in use
- $T_1(t)$ pheromone associated to the location of the source

Pheromone

In this problem we have two kinds of pheromone. The pheromone on the edge between a source i and a customer j denoted by $\tau_{ij}(t)$, and a pheromone associated to the location of the source denoted by $T_l(t)$. Both of them change after the completion of a search of all ants ant at the beginning of the algorithm $\tau_{ij}(t)$ pheromone trails are all set equal to a very small value τ_0, while $T_l(t)$ are all set equal to a very small value T_0.

$\eta_{ij}(t)$-visibility

Visibility is a quantity, which in our problem (a static problem) doesn't change over iterations. It the reciprocal of the weighted distance between source i and customer j, $\eta_{ij} = 1/(w_j d_{ij})$.

Pheromone update rules

After all ants have "constructed" a set of "new" solutions (it is not guaranteed that after all the iterations of the algorithm new solutions will be found) the cost for each one of them is computed. If a solution is found that outperforms the global best, then it becomes the new global best. We denote that $\text{Cost}_{\text{best}}(t)$ and $T_{1\,\text{best}}(t)$, $T_{2\,\text{best}}(t)$ ($\text{Cost}_{\text{best}}(t)$ mean that the best solution till iteration t is not the best solution found during iteration t – the same for $T_{1\,\text{best}}(t)$, $T_{2\,\text{best}}(t)$.

The global best solution is the only one that adds pheromone to the existing trails. All the other trails are loosing some of their pheromone. This is done according to the following rules:

Rule 1. $\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}(t)$

Rule 2. $T_i(t+1) = (1-\rho)T_i(t) + \rho\Delta T_i(t)$

and

$$\Delta\tau_{ij}(t) = \begin{cases} 1/\mathrm{Cost}_{\mathrm{best}}(t), & \text{if } (i,j) \in T_{\mathrm{best}}(t) \\ 0, & \text{if } (i,j) \notin T_{1\,\mathrm{best}}(t)\,. \end{cases} \tag{13}$$

Where:

- ρ with $0 \leq \rho \leq 1$ is the evaporation coefficient. P does not have to be the same for both rules.

The algorithm which was used for this problem is described with the 6 steps that follow.

Step 1. Set the pheromone with a starting value for all paths that connect the stations with the customers. Set the second value of the pheromone that is respectively equal for every station with the beginning value. Calculate the total of the balanced distance between the stations and the customers (the distances are multiplied with the coefficients w that are related for every customer). An ant is set in every station.

Step 2. For each ant free customers are set until maximum capacity is reached or until all customers are served. If all customers are served then move to *Step* 4, or otherwise move to *Step* 3.

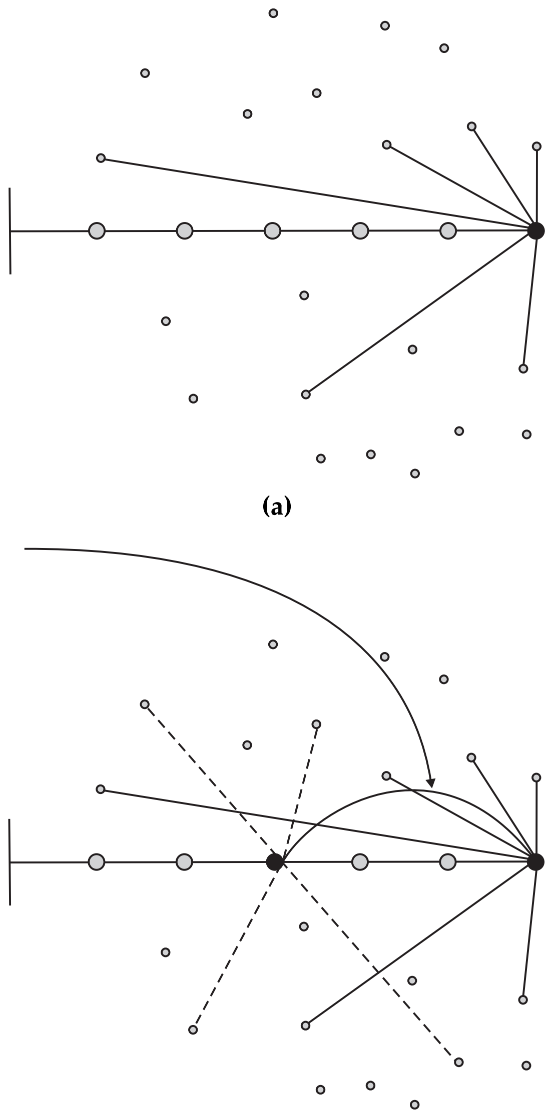
Step 3. Select the station that will be activated based on the second transition rule. Move back to *Step* 2 (set customers in the new station).

Step 4. Calculate the ant cost and save the best.

Step 5. Renew the pheromone quantity using the rule mention above.

Step 6. Repeat procedure from *Step* 2 until a particular number of iterations is completed or until some criteria are satisfied.

A very important element that must be mentioned is that except the pheromone that an ant puts on every edge that connects the station with the destination, there is also the pheromone that is related exclusively with the station. In this simple model of the algorithm, the line on which the station base are located is discrete and no new station is created unless the quantity of the previous is satisfied. The following shape explains better the algorithm used for solving the problem.



(a)

(b)

Figure 1(a, b)

(a) We allocate customers to a source till we reach source's capacity
(b) When the source cannot serve any more customers we "jump" to another location, we establish a new source and we start again allocating customers. We keep doing that till all the customers have been allocated to a source

It is obvious that we need 2 transition rules in this problem. One transition rule to use it for the allocation of customers to a source and a second transition rule to decide where a new source should be placed after the previous has reached its limits.

Again we need to keep track of all the locations where the ant has put a source $T_2^k(t)$ (this is a vector) along with the vector or a matrix it depends on the implementation. A straightforward implementation is to consider a matrix $n \times m$ (number of customers). In that case a line full of zeros indicates that in the corresponding location on the line there is no source.

Customer

	1	2						j							m
1		**1**													**1**
2															
Source															
i								1							
n															

Figure 2

The matrix below need to keep track of all the location where the ant has located a source

$n =$ number of potential source-locations on the line

$m =$ number of customers

According to the above table-matrix (tabu-list), the source located at place 1 (the start of the line) serves 2 customers, customer-2 and customer-m, while the source located i at place 1 serves the customer-j.

The algorithm flowchart follows.

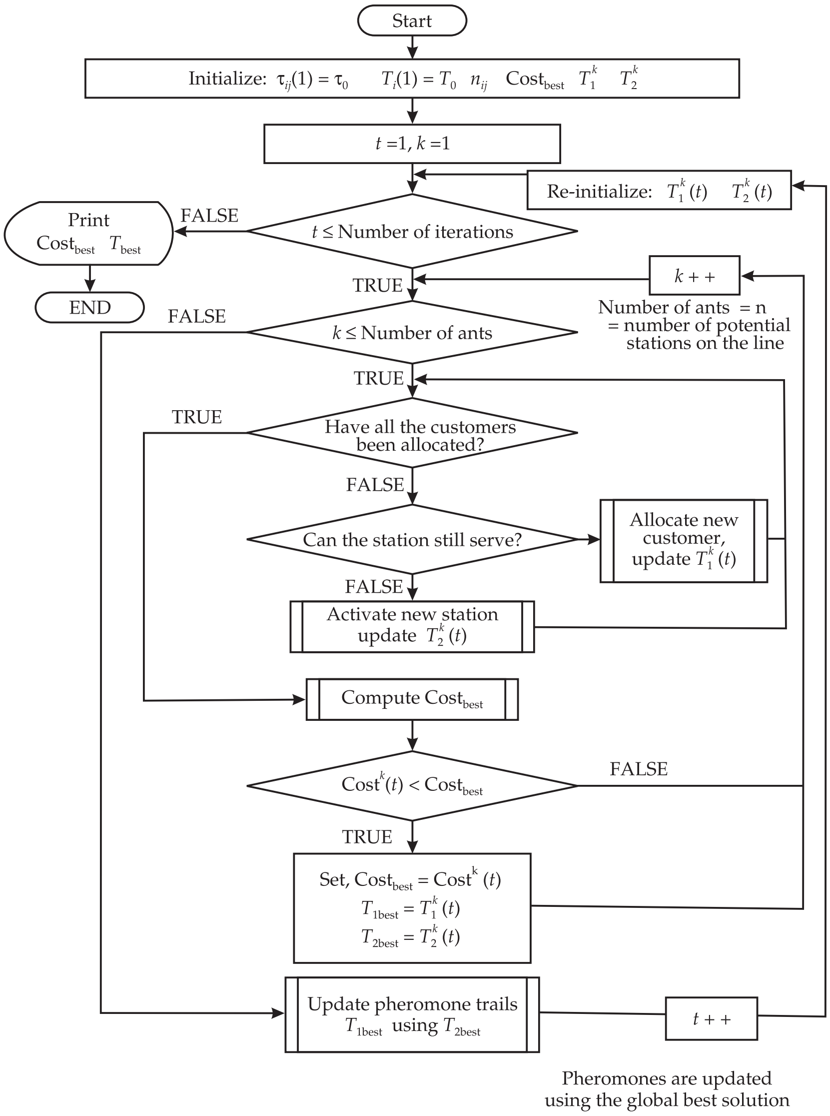


Figure 3
Flowchart of our algorithm

4. Case study

The implementation of the algorithm had as a result the minimization of the cost function, finding in that way the best combination between destinations-customers and stations on the line. Some stations cannot respond anymore to customers because they have reached their capacity, that every destination-customers connects with only one station and all the stations must be served.

First, we set that the stations are three and the destinations-customers six. The Table 1 presents the data for the flow chart of best cost with the price of parameter β.

Table 1
Data for the flow chart in Figure 4

r_0	W_i	Evaporation (ρ)	β	Station Cost	Iteration	Best Cost
0,2	10,...,60	0,1	0,1	200	200	100
0,3	10,...,60	0,1	0,5	200	200	100
0,4	10,...,60	0,1	0,8	200	200	100
0,5	10,...,60	0,1	1,3	200	200	100
0,6	10,...,60	0,1	1,8	200	200	100
0,7	10,...,60	0,1	2	200	200	100
0,8	10,...,60	0,1	2,9	200	200	100
0,9	10,...,60	0,1	3,7	200	200	100

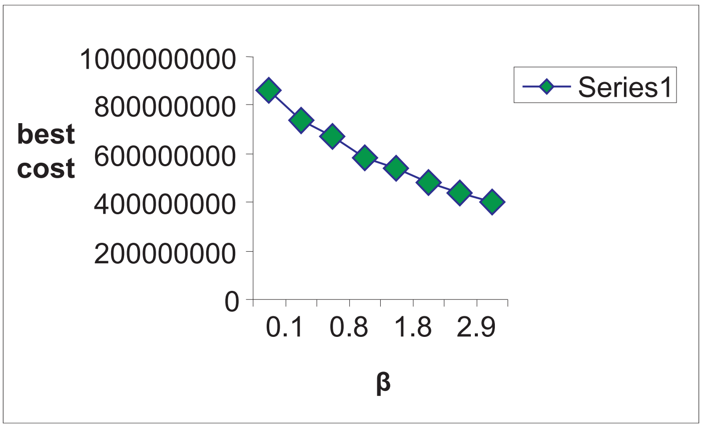


Figure 4
Relation between parameter β and best cost

When the numbers of stations and the customers-destinations is low the cost functions will be small as well. In Figure 4 we show that when parameter β increases then costs will decrease. In Table 2 we show the data that is used in Figure 4 which compare the cost functions when the stations are three with six customers-destinations and when the stations are two with tree customers-destinations.

Table 2
Data for the flow chart in Figure 5

β	Best Cost-customer = 6, station = 3	Best Cost-customer = 3, station = 2
0.1	857800000	555700000
0.5	734800000	455700000
0.8	674800000	415700000
1.3	584800000	355700000
1.8	539800000	325700000
2.	479800000	285700000
2.9	434800000	255700000
3.7	404800000	235700000

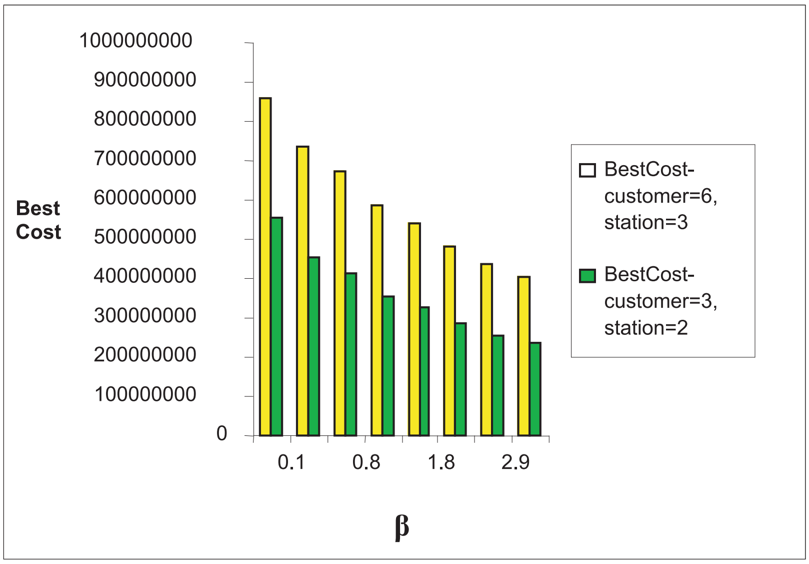


Figure 5
Comparisons of best cost

5. Conclusion

This study examines the improvement of customers and stations problem. The problem was approached according to the heuristic algorithm of Ant Colony System, with modifications which has to do with the update of pheromone. Distribution networks can be constructed as straight lines with stations on them. The stations are placed on a straight line will have a specific capacity and number of destinations which they can serve on the line. The results have shown that it is possible to find a better point of service between stations and customers-destinations that will result to low cost.

Appendix

The pseudo code is presented bellow:

Start
//Initialize
for i: 1 to n (set n stations)
    set $T_i(t) = T_0$
    for j: 1 to m (and set m customers)
    set $\tau_{ij}(t) = \tau_0$
    end for
end for

for i: 1 to n
    for j: 1 to m
        $\eta_{ij}(t) = 1/(w_j d_{ij})$
    end for
end for

for k: 1 to n
    set every ant to one station (initialization of Tabu list for every ant – Two lists, the first is used for the writing the customers that each station served, and the second list is used for writing the stations that were activated) T_1^k, T_2^k)
end for
Initialize minimum Cost, $\text{Cost}_{\text{best}}$ (a very large value)

//Main Loop

for t: 1 to maximum iterations

for $k = 1$ to n

reinitialize Tabu lists (erase the lists that has been created during the previous iteration)

activate the first station

while not all customers have been served

if station can still serve customers

allocate new customer to station using transition rule 1

update Tabu list $T_1^k(t)$

else

activate another station employing transition rule 2

update Tabu list $T_2^k(t)$

endif

endwhile

Compute Cost, $\text{Cost}^k(t)$

If $\text{Cost}^k(t) < \text{Cost}_{\text{best}}$ set $\text{Cost}_{\text{best}} = \text{Cost}^k(T)$ and keep edges (and stations-locations) of best location-allocation: $T_{1\,\text{best}} = T_1^k(t)$, $T_{2\,\text{best}} = T_2^k(t)$

End for

for i: 1 to n

for j: 1 to m

Update pheromone trails (connecting stations to customers by applying the pheromone update rule 1

End for

end for

For i, to n

Update pheromone trails (for station) by applying the pheromone update rule 2

End for

End for

Print $\text{Cost}_{\text{best}} T_{1\,\text{best}}$ (the information of $T_{2\,\text{best}}$ include on $T_{1\,\text{best}}$)

Stop

References

[1] E. C. Moshe and G. M. Abrahamm, Capacitated location-allocation problems on a line, *Computers & Operations Research*, Vol. 29 (2002), pp. 459–470

[2] R. L. Francis, L. F. McGinis and J. A. White, Locational analysis, *European Journal of Operations Research*, Vol. 12 (1993), pp. 220–52.

[3] P. Mirchandani, R. Kohli and A. Tamir, Capacitated location problems on a line, *Transportation Science*, Vol. 30 (1) (1996), pp. 75–80.

[4] E. Bonabean, M. Dorigo and G. Theranlaz (eds.), *S Warm Intelligence, from Natural to Artificial Systems*, New York, Oxford University Press, 1999.

[5] M. Dorigo and L. M. Gambardella, Ant colony system: a cooperative learning approach to the travelling salesman problem, *IEEE Transactions on Frolutionary Computation*, Vol. 1 (1997), pp. 53–66.

[6] R. Beckers, D. L. Deneubourg and S. Gross, Trails and U-turns in the selection of the shortest path by the ant Lasius Niger, *Journal of Theoretical Biology*, Vol. 159 (1992), pp. 397–415.

[7] D. Come, M. Dorigo and F. Glover (eds.), *New Ideas in Optimization*, London, UK, Mc Graw-Hill, 1999.

[8] M. Dorigo and L. M. Gambardella, Ant colonies for the travelling salesman problem, *Bio System*, Vol. 48 (1997), pp. 73–81.

Received March, 2005