INTRODUCTION

Today during rapid development of computers technologies various scientific works in many spheres become the reliable base at their use in the field of decision-making. Some works, like far at first sight from the given area, further become an irreplaceable element in questions of reception effective a way of the decision of those or other problems which searches of the decision considers within the framework of application of computer technologies.

Such work for today is also the method of ant colonies, bases which have been incorporated in biology. Efficiency of this method is caused to feature of behaviour and ability to live of the object, description of the given method of ants.

Where the given method is applicable?

The method of ant colonies has found wide application in the field of the problems connected with find of the shortest way for any purposes, as ants, as biological organisms, have feature the common efforts in the best way to find sources of food, as the main component of maintenance of ability to live of any alive organism.

In the given academic year project the method of ant colonies is used for the decision problems of the direct-sales representative, in other words, for definition of the shortest way, which need to receive, passing, for example, each city only once and to return into start.

On what basis this method is chosen from set of other methods, used at the decision of a similar sort of tasks?

The decision of a task of the direct-sales representative can be made also on the basis of the following methods: *a method of optimization, genetic algorithms and method of ant colonies.*

THE ANT COLONY SYSTEM

It has been observed that ants, while almost blind, manage to establish shortest route paths from their colony to feeding sources and back, only by communicating information by laying on the ground a substance called pheromone . Since this paper is not concerned with the behavior of real ants, the reader is referred for a more complete discussion on real ant colonies and their emergence behavior.

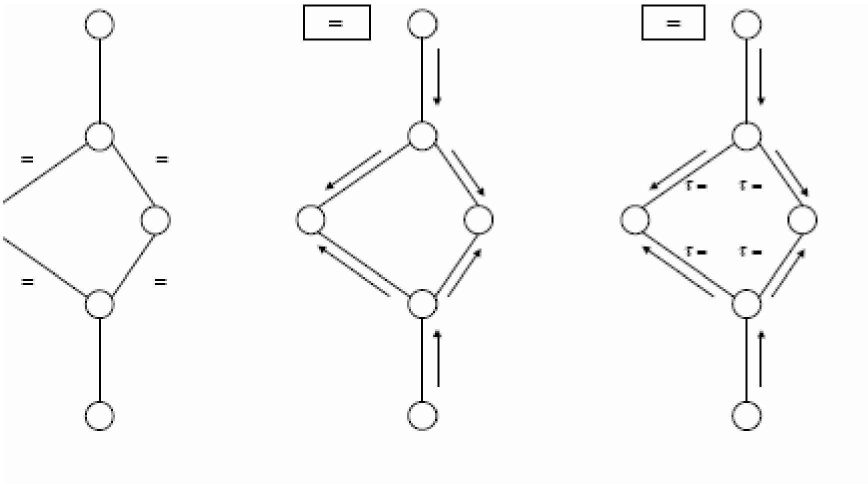

Figure 1

This behavior has inspired what is known as Ant Colony System (ACS) algorithms that use colonies of artificial ants. To illustrate the behavior of the ACS we use the following example. Let as suppose that we have a graph as in Figure 1a) and an ACS as follows:

i. 30 new ants come from A to B and from E to D, at each time unit.

ii. Each ant walks at a speed of 1 per time unit and while walking, it lays down a pheromone trail of intensity 1.

At $t = 0$, there is no pheromone trail and the ants that are in B and D will choose their direction completely at random. Let us suppose then that 15 ants from each node will chose to go toward H and 15 toward C, as in Figure 1b).

At $t = 1$, as in Figure 1c), 30 new ants come to B and D. The ants that have chosen the direction of node C from B and D, have now reached D and B respectively. This means that on the path BCD, a pheromone trail of intensity 30 has been laid. On the other path BHD, the 15 ants that were at B and the 15 ants that were at D have only succeeded to reach H. This means that, on this path, the pheromone trail has intensity 15. The 30 new ants that are now on B will chose their direction based on the intensity of the pheromone trails, in such a way that the expected number of ants that will go to C will be the double of those that will follow the direction of H. The same is also true for the 30 new ants that are now on D. The idea of the ACS is that, if this process continues for a long time, almost all of the new coming ants will follow the shortest path.

This idea can then be applied with impressive results to the Traveling Salesman Problem and to other combinatorial optimization problems (Quadratic assignment, Job-shop scheduling, Vehicle routing, Graph coloring etc). For its application to the TSP, an intuitive explanation of its behavior, is given by M. Dorigo in: “Once all the ants have generated a tour, the best ant deposits (at the end of iterationt) its pheromone, defining in this way a ‘preferred tour’ for earch in the following algorithm iteration t+1. In fact, duringiteration t+1 ants will see edges belonging to the best tour ashighly desirable and will choose them with high probability.Still, guided exploration together with the fact that local updating ‘eats’ pheromone away (i.e., it diminishes the amount of pheromone on visited edges, making them less desirable for future ants) allowing for the search of new, possibly better tours in the neighborhood of the previous best tour. So ACS can be seen as a sort of guided parallel stochastic search in the neighborhood of the best tour.”

In the following paragraph we explain more thoroughly how the ACS is applied to the TSP.

THE ACS ALGORITHM

Given is a set of n nodes in an m - dimensional Euclidean space. We denote by dij the length of the direct path between nodes i and j . In the present case, if the position of the node vi is defined by

$$v_i := \{x_1^i, x_2^i, \ldots, x_m^i\}$$

where, d ij equals the Euclidean distance between nodes i and j :

$$d_{ij} := \sqrt{\sum_{k=1}^{m}\left(x_k^i - x_k^j\right)^2} .$$

Let k the number of ants that are used by ACS_TSP and that all edges have a pheromone intensity of value t_0 . At the beginning of the first iteration, each ant is positioned randomly at a starting node. Each ant moves then to a still not visited node according to the transition probability defined by

$$p_{sj}(t) := \frac{\left[\tau_{sj}(t)\right]\cdot\left[\eta(s,j)\right]^{\beta}}{\sum_{j\in J(s)}\left[\tau_{sj}(t)\right]\cdot\left[\eta(s,j)\right]^{\beta}}$$

where, $t_{sj}(t)$ the intensity of the pheromone trail of the edge (s, j) at $h(s,j):=1/d_{sj}$ is the inverse of the distance between nodes s and j , and b > 0 is a parameter which determines the relative importance of pheromone versus distance. This rule applies of course differently to each ant, supposes that the given ant is at node s, and that the subset of the nodes still not visited by it is given by J(s) . In this way the ACS favors the choice of edges which are shorter and which have a greater amount of pheromone. After each ant has chosen the next node of its path, a local pheromone updating rule is applied. If an ant was at node r and has chosen, by applying the state transition rule described above, to go to node s , then the pheromone level of the (r,s) edge is updated by

$$\tau(r,s) := (1-\rho)\tau(r,s) + \rho\tau_0 ,$$

where ρ and t_0 are parameters. After all ants have completed their tours, they remain to their current position for the following iteration and a global pheromone updating rule is applied for each edge (r,s) in the graph:

$$\tau(r,s) := (1-\alpha)\cdot\tau(r,s) + \alpha\cdot\Delta\tau(r,s) \text{, with}$$

$$\Delta\tau(r,s) := \begin{cases}\left(L_{gb}\right)^{-1}, & \text{if } (r,s) \text{ belongs to the global best tour}\\ 0 & \text{otherwise}\end{cases}$$

0<a<1 is in this case the pheromone decay parameter, and L_{gb} is the length of the globally best tour found up to that point. The ACS proceeds then in this way until the end conditions are reached. Usually, the end conditions are the following:

i. A (user defined) maximum number of iterations is performed, or

ii. A tour of a given length (target length) is found.

For our implementation we have chosen to only apply the first end condition, and the target length is set to zero. As you have noticed, this algorithm makes use of a number of parameters. Dorigo and Gambardella have experimented extensively trying to identify their optimal values, and have reported in [4] the following:

- ¾ · Experimental observation shows that ACS works well when the number of ants used is equal to 10. ·
- ¾ The optimal values of the parameters are largely independent of the problem and experimental observation leads to the following values: $\beta = 2$, α=ρ=0.1 and

$t_0 = (m * Lnn)^{-1}$, where *Lnn* is the tour length produced by applying the Nearest Neighbor heuristic and n the total number of nodes. Note here that t_0 is also used as the initial pheromone intensity value for all edges in the graph

VERIFICATION.

For the purposes of the course ‘Optimization in Logistics’, the ACS algorithm was implemented using the programming language C++ (see Appendix). The main functions used by this implementation have as follows:

- **GetDistanceMatrix** reads from an input file (**TSP.in**) an integer N indicating the number of nodes in the input graph, an integer M indicating the dimension of the Euclidean space of the given TSP instance, and a matrix in $N M$ ´ ¡ containing the exact positions of the N nodes in M ¡
- **NearestNeighborSolve** constructs a solution of the given TSP instance, using the Nearest Neighbor heuristic (starting by a randomly chosen node) and returns the solution (TSP tour) and its length.
- **ACSTSP** solves the given TSP instance, using the ACS algorithm. It requires as input the number of iterations that the algorithm must perform and returns the solution (TSP tour) and its length

For testing our program we use Berlin 52 cities test. Algorithm successful founded optimal length 7861.52, after 37761 iterations. Represent this optimal way on the map Figure 4. Result of iterations per length of tour represented on Figure 4

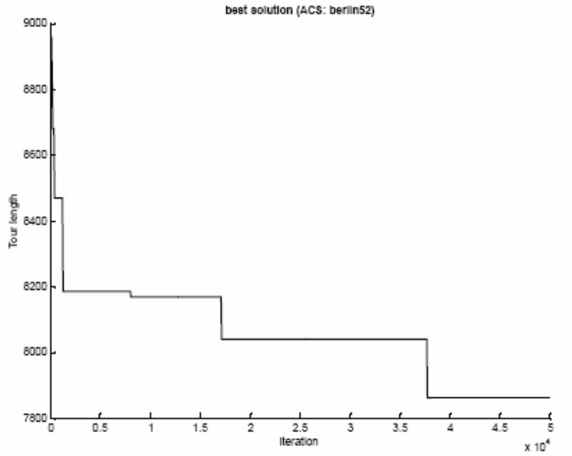

Figure 4

COMPARE WITH OTHER METHOS

In the present paragraph, the performance of the algorithm is illustrated using as a test case the berlin52 TSP instance (taken from TSPLIB [7]), that consists of 52 nodes in 2 ¡ . The optimal tour for this instance has length 7452 and has as in Figure 2

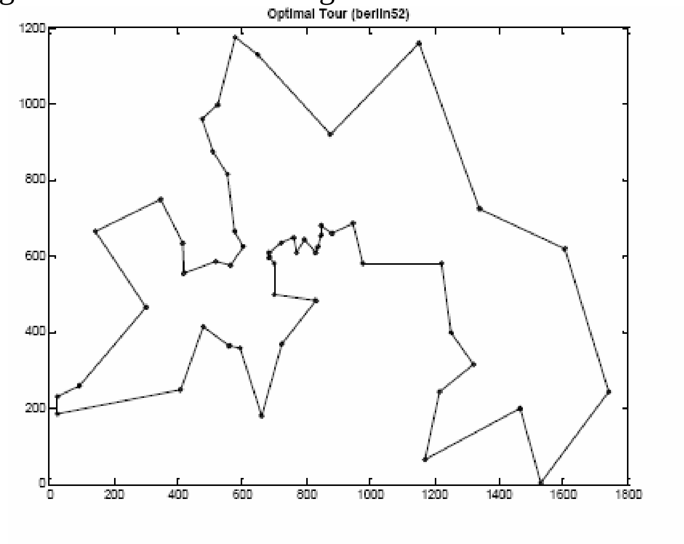

Figure 2

The ACS algorithm succeeded in finding a solution with length 7861.52, after 37761 iterations (note that other runs might perform better or worse). The ACS solution is presented in Figure 3.

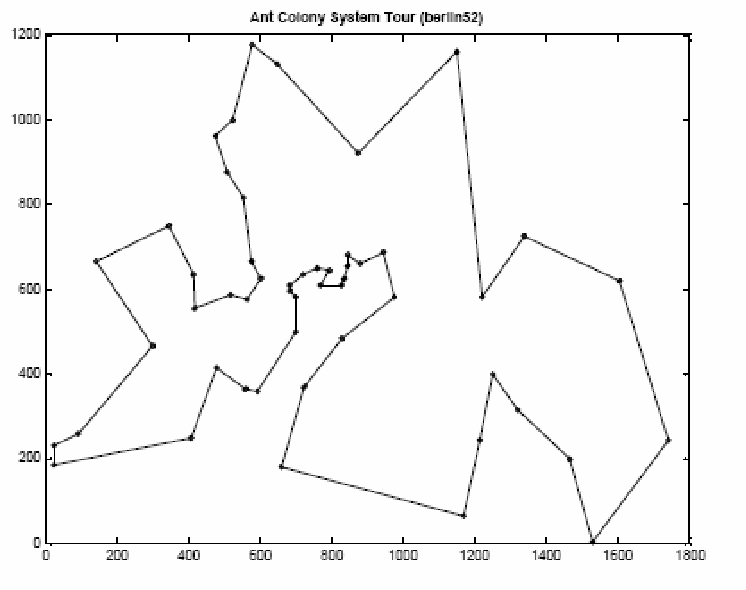

Figure 3

The following figures illustrate the performance of the ACS algorithm for the given instance (berlin52) during a random run. Figure 4 presents the best solution that the ACS has found up to each iteration.

TESTING FOR BELARUS CITIES

For testing our ACO TSP algorithm we use Belarus 42 cities. This cities represented on Figure 5.

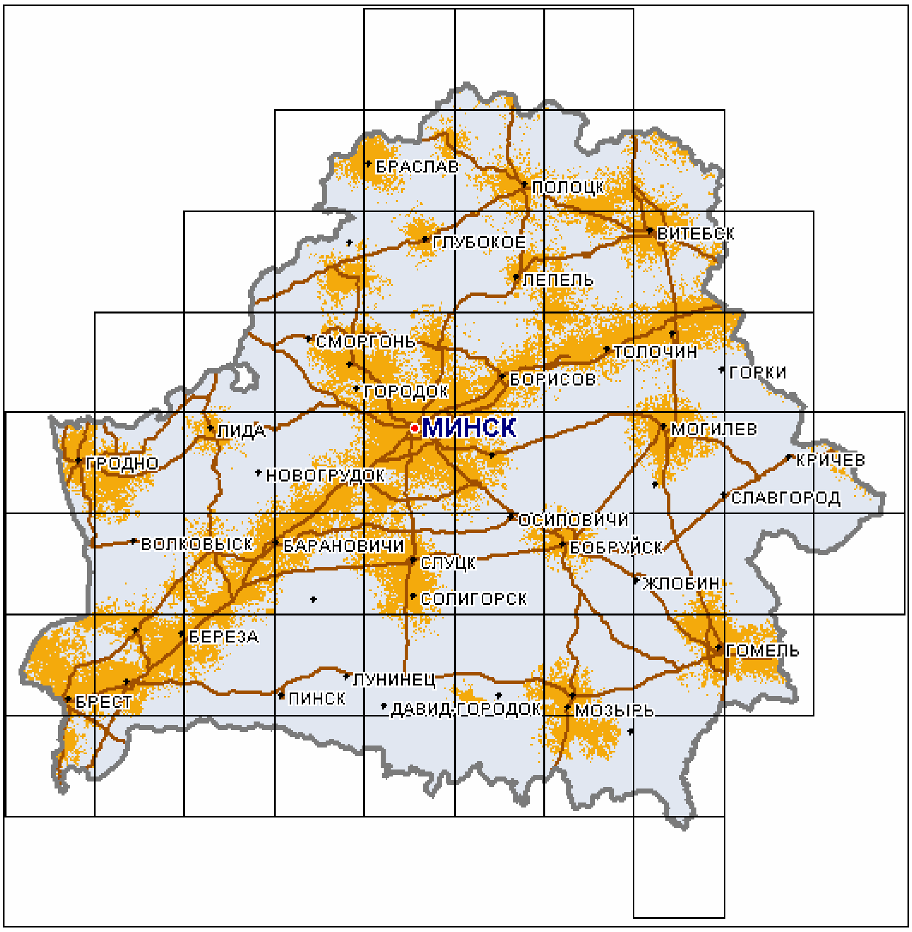

Figure 5

The scale for this map is: 1 centimeter on the map is approximately equal 49 kilometers on the reality. We have coordinates of most poplar cities of Belarus (Table 1).

Table 1

X	Y	City
1,40	2,60	Brest
3,90	4,10	Beresa
2,85	6,10	Volcovisk
1,60	7,90	Grodno
4,60	8,65	Lida
6,10	2,70	Pinsk
5,65	7,65	Novogrudok
6,00	6,10	Baranovichi
7,60	3,10	Luninec
8,40	2,50	David-Gorodok
6,70	10,60	Smorgon
7,80	9,50	Gorordok
8,15	15,30	Braslav
9,10	4,90	Sluck
9,07	5,70	Soligorsk
9,10	8,65	Minsk

9,13	12,90	Glubokoe
11,40	12,00	Lepel
11,05	9,80	Borisov
11,25	6,70	Osipovichi
11,55	14,10	Polack
12,40	6,10	Bobruisk
12,60	2,70	Mozir
13,40	10,43	Tolochin
14,10	5,25	Ghlobin
14,35	13,05	Vitebsk
14,60	8,70	Mogilev
15,90	3,80	Gomel
15,95	7,65	Slavgrad
16,00	10,40	Gorki
17,40	8,00	Krichev

For testing algorithm we use this 42 coordinates. On 37761 iterations algorithm was returned optimal way (Figure 6). Optimal length of tour is 81.1734. In kilometers this value consist 81.1734 * 49 = 3977,4966 kilometers. On this map present graph of optimal tour.

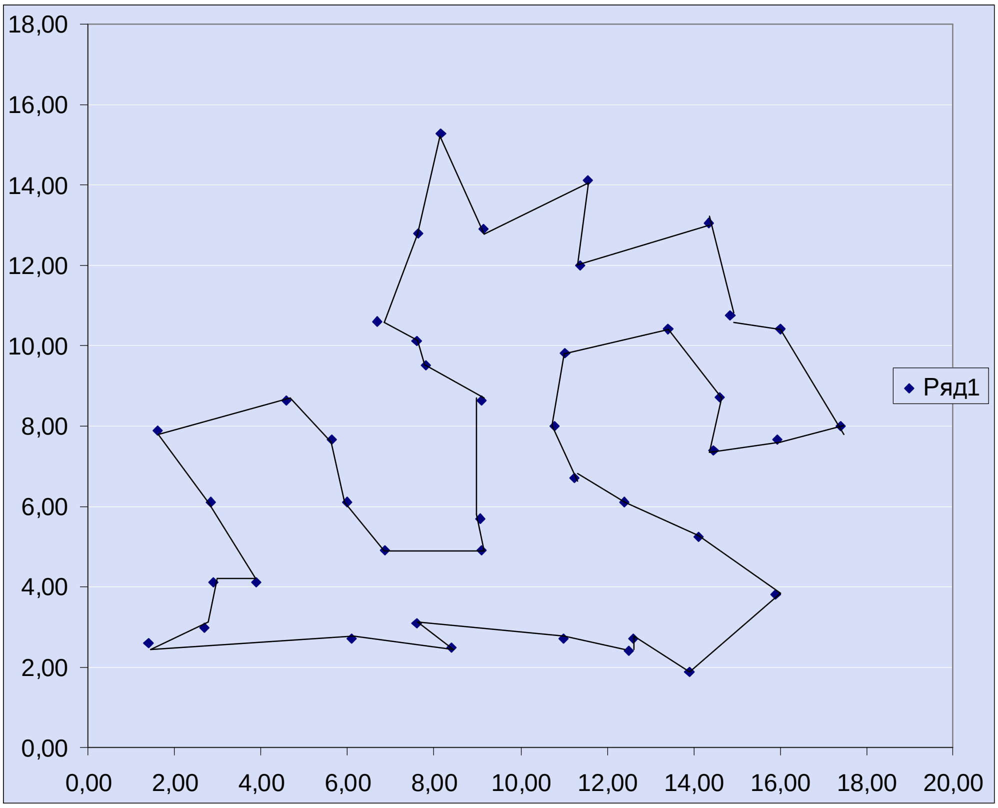

Figure 6

The following figures illustrate the performance of the our ACS algorithm for the given instance (belarus42) during a random run. Figure 7 presents the best solution that the ACS has found up to each iteration.

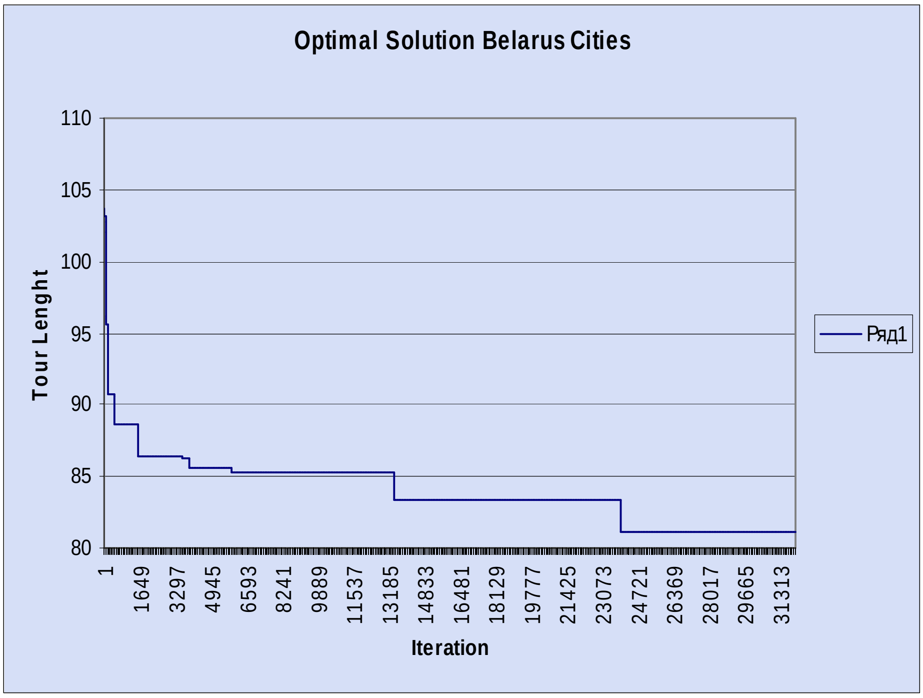
Optimal Solution Belarus Cities
Tour Lenght
110
105
100
95
90
85
80
1
1649
3297
4945
6593
8241
9889
11537
13185
14833
16481
18129
19777
21425
23073
24721
26369
28017
29665
31313
Iteration
Ряд1

Figure 7

THE CONCLUSION

As a result of performance of the given academic year project was develop a program maintenance for the decision of a task of the direct-sales representative a method of ant colonies.

Direct result of realization of the given program is reception of a set of coordinates of the popular cities of Belarus, which defines (i.e. a set of coordinates) the minimal route of detour of all these cities. The initial data

There were also coordinates of these cities

For program realization of the given project the algorithm of the decision has been used solution task residences of the direct-sales representative a method of ant colonies.

As means of realization of the given program project the environment of development has served Microsoft Visual C++ version 6.0.

As the result should be noted, that the received results confirm efficiency of use of a method of ant colonies for application in the decision of the tasks connected to definition of the minimal route and a similar sort of tasks.

Summing up all aforesaid is possible to draw a conclusion,

That the given academic year project completely is executed.

REFERENCES

[1] B. Korte and J. Vygen. “Combinatorial Optimization. Theory and Algorithms.” Germany: Springer, 2002.
[2] H.N. Post. “Transport, Routing- en Schedulingproblemen”Delft: Delft University of Technology, 2004.
[3] M. Dorigo, V. Maniezzo and A. Colorni. “ The AntSystem: Optimization by a colony of cooperating agents” IEEE Transactions on Systems, Man, and Cybernetics – Part B, Vol. 26, No. 1 1996: 1-13.
[4] M. Dorigo, L.M. Gambardella. “Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem” IEEE Transactions on Evolutionary Computation, Vol. 1, No. 1 1997.
[5] M. Dorigo, L.M. Gambardella. “Ant Algorithms for Discrete Optimization” Artificial Life, Vol. 5, No. 3 1999: 137-172.
[6] E. Bonabeau, M. Dorigo and G. Theraulaz. “Swarm Intelligence: From Natural to Artificial Systems (Santa Fe Institute Studies in the Sciences of Complexity Proceedings)” Oxford University Press, 1999.
[7] TSPLIB: http://elib.zib.de/pub/Packages/mp-testdata/tsp/tsplib/tsplib.html
[8] M. Dorigo and L.M. Gambardella. “Ant colonies for the traveling salesman problem” Biosystems, 43 1997: 73-81.
[9] M. Dorigo and T. Sttzle. “Ant Colony Optimization (Bradford Books)” The MIT Press, 2004.
[10] H.M. Botee and E. Bonabeau. “Evolving Ant Colony Optimization” Advances in Complex Systems Vol. 1 1998:149-159 (http://www.tbi.univie.ac.at/~studla/ACS/Contents.html)