An Improved Ant Colony System Based on Dynamic Candidate Set and Entropy for Traveling Salesman Problem

Zar Chi Su Su Hlaing, May Aye Khine
University of Computer Studies, Yangon
zarchisusuhlaing@gmail.com

Abstract

The Ant Colony Optimization (ACO) is a metaheuristic algorithm used for combinatorial optimization problems. It is a good choice for many hard combinatorial problems because it is more efficient and produces better solutions than greedy algorithms. However, ACO is computationally expensive and it can still trap in local optima, take a long time to compute a solution on large problem sets and premature convergence problem. The main idea of the modification is to limit the number of elements choices to a sensible subset, or candidate list, which can limit the selection scope of ants at each step and thus substantially reduce the size of search space and to measure the uncertainty of the path selection and evolution by using the information entropy self-adaptively. Simulation study and performance comparison on Traveling Salesman Problem show that the improved algorithm can converge at global optimum with a high probability. It also shows a faster convergence to the solutions than the standard algorithm.

1. Introduction

In recent years, many research works have been devoted to ant colony optimization (ACO) techniques in different areas. ACO [1, 2, 19, 20] is a recently proposed metaheuristic approach and has been successfully used for solving hard combinatorial optimization problems that are being increasingly applied to real world problems in areas such as communications and transportation.

It is a heuristic algorithm to model the behavior of real ant colonies in establishing the shortest path between food sources and nests. ACO inspired by the foraging behavior of real ant was first introduced by Dorigo and his colleagues [1, 2, 3] and has become one of the most efficient algorithms for TSP. The inspiring source of ACO is the pheromone trail laying and following behavior of real ants which use pheromones as a communication medium. In analogy to the biological example, ACO is based on the indirect communication of a colony of simple agents, called (artificial) ants, mediated by (artificial) pheromone trails. The pheromone trails in ACO serve as distributed numerical information which the ants use to probabilistically construct solutions to the problem being solved and which the ants adapt during the algorithm's execution to reflect their search experience. Artificial ants imitate the behavior of real ants how they forage the food, but can solve much more complicated problem than real ants can.

From the early nineties, when the first ant colony optimization algorithm was proposed, ACO attracted the attention of increasing numbers of researchers and many successful applications are now available. ACO has been widely applied to solving various combinatorial optimization problems such as Traveling Salesman Problem (TSP), Job-shop Scheduling Problem (JSP), Vehicle Routing Problem (VRP), Quadratic Assignment Problem (QAP), Weapon-Target Assignment problems (WTA), etc.

Although ACO has a powerful capacity to find out solutions to combinational optimization problems, it has the problems of stagnation and premature convergence and the convergence speed of ACO is always slow. Those problems will be more obvious when the problem size increases. Therefore, several extensions and improvements versions of the original ACO algorithm were introduced over the years.

To break through this limitation, an improved ant colony algorithm based on candidate set strategy and the average information entropy is proposed here. This modification reduces the size of the search space for the ant colony algorithm. The information entropy is used to judge the stability of the subspace of solutions represented at the given stage of algorithm's evolution and then it is applied to control the parameter of the algorithm. In some degrees, this work can solve the premature convergence of the basic ACO. Various adaptations: an algorithm based on the basis of the ant evolution rules [3], dynamic control of solution construction and mergence of local search ([4], [5], [6]), new pheromone updating strategies [7], max-min ant system [8], a strategy is to partition artificial ants into two groups: scout ants and common ants [9], using candidate lists strategies ([10], [11]), dynamic ant colony system with three level updates ([12, [13]), a new probability selection mechanism by using Held-Karp lower bound to determine the trade-off between the influence of the heuristic information and the pheromone trail [14], and using the path selection controlled by information entropy [15,16] and hybrid ant colony system approach [17] are studied to improve the quality of the final solution and lead to speedup of the algorithm. All these studies have contributed to the improvement of the ACO to some extent, but they have little obvious effect on increasing the convergence speed and obtaining the global optimal solution.

In this paper, a modified ant colony system for solving TSP using candidate set strategy and dynamic updating of heuristic parameter is developed. This algorithm is used to produce near-optimal solutions to the TSP. The paper is organized as follows: Section 2 describes traveling salesman problem. Section 3 and 4 illustrates the algorithm of ant colony system. Section 5 presents candidate list strategy approach to ACO and the other is the improved strategy of entropy. In Section 6, the proposed method is employed into several TSP problems and the results of our approach and of traditional ACO are reported. Finally, Section 7 makes the conclusion.

2. Traveling Salesman Problem

Traveling salesman problem (TSP) is one of the well-known and extensively studied problems in discrete or combinational optimization and asks for the shortest roundtrip of minimal total cost visiting each given city (node) exactly once. Cost can be distance, time, money, energy, etc. TSP is an NP-hard problem and researchers especially mathematicians and scientists have been studying to develop efficient solving methods since 1950's. Because it is so easy to describe and so difficult to solve. Graph theory defines the problem as finding the Hamiltonian cycle with the least weight for a given complete weighted graph.

The traveling salesman problem is widespread in engineering applications. It has been employed in designing hardware devices and radio electronic devices, in communications, in the architecture of computational networks, etc. In addition, some industrial problems such as machine scheduling, cellular manufacturing and frequency assignment problems can be formulated as a TSP.

A complete weighted graph $G=(N, E)$ can be used to represent a TSP, where N is the set of n cities and E is the set of edges (paths) fully connecting all cities. Each edge $(i,j)\in E$ is assigned a cost d_{ij}, which is the distance between cities i and j. d_{ij} can be defined in the Euclidean space and is given as follows:

$$d_{ij}=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2} \tag{1}$$

3. Background

3.1 Ant Behavior

The Ant Colony Optimization (ACO) techniques has emerged recently as a relatively novel meta-heuristic for hard combinational optimization problems. It is designed to simulate the ability of ant colonies to determine shortest paths to food. Although individual ants posses few capabilities, their operation as a colony is capable of complex behavior.

Real ants can indirectly communicate by pheromone information without using visual cues and are capable of finding the shortest path between food sources and their nests. The ant deposits pheromone on the trail while walking, and the other ants follow the pheromone trails with some probability which are proportioned to the density of the pheromone. The more ants walk on a trail, the more pheromone is deposited on it and more and more ants follow the trail. Through this mechanism, ants will eventually find the shortest path. Artificial ants imitate the behavior of real ants how they forage the food, but can solve much more complicated problems than real ants can. A search algorithm with such concept is called Ant Colony Optimization. Figure 1 shows the behavior of real ants [19].

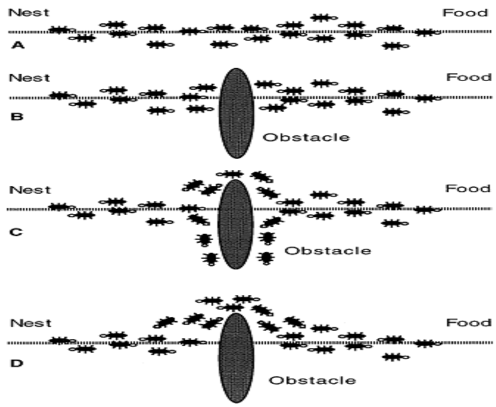

Figure 1. (A) Real ants follow a path between nest and food source. (B) An obstacle appears on the path: ants choose whether to turn left or right with equal probability. (C)Pheromone is deposited more quickly on the shorter path. (D)All ants have chosen the shorter path.

3.2 Ant System (AS)

Ant System was first introduced and applied to TSP by Marco Dorigo et al. [13, 14, and 15]. Initially, each ant is randomly put on a city. During the construction of a feasible solution, ants select the following city to be visited through a probabilistic decision rule. When ant k states in city i and constructs the partial solution, the probability moving to the next city j neighboring on city i is given by

$$p_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum\limits_{u\in J_k(i)} [\tau_{iu}(t)]^\alpha [\eta_{iu}]^\beta} & if \;\; j \in J_k(i) \\ 0 & otherwise \end{cases} \tag{2}$$

where η_{ij} is the heuristic visibility of edge(i,j), generally it is a value of $1/d_{ij}$, where d_{ij} is the distance between city i and city j and τ_{ij} is the pheromone information of edge (i,j). $J_k(i)$ is a set of cities which remain to be visited when the ant is at city i. α and β are are two adjustable positive parameters that control the relative weights of the pheromone trail and of the heuristic visibility. If $\alpha=0$, the closed vertex is more likely to be selected. This is responding to a classical stochastic greedy algorithm. If on the contrary *$\beta=0$*, only pheromone amplification is at work: This method will lead the system to a stagnation situation, i.e. a situation in which all the ants generate a sub-optimal tour. So the trade-off between edge length and pheromone intensity appears to be necessary.

After each ant completes its tour, the pheromone amount on each path will be adjusted according to equation

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

In this equation,

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \tag{4}$$

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \dfrac{Q}{L_k} & \text{, if (i, j)} \in \text{ tour done by ant k} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$(1-\rho)$ is the pheromone decay parameter (*$0<\rho<1$*) where it represents the trail evaporation when the ant chooses a city and decide to move. L_k is the length of the tour performed by ant *k*, Q is an adjustable parameter and *m* is the number of ants.

4. Ant Colony System (ACS)

The ACS is mainly different from the AS in these aspects [19, 20]: The move rules of the ants are different; the global updating rules are different; and local updating rules which adjust the amount of the pheromone on various paths are newly added. Now we will describe the ACS algorithm.

Step 1: Initiation. The amount of the pheromone on each side is initiated into a tiny constant value; allocate *m* ants randomly to *n* cities.

Step 2: Each ant will choose the next city with equation (6)

$$j = \begin{cases} \arg\max\left\{ [\tau_{iu}]^{\alpha}.[\eta_{iu}]^{\beta} \mid u \in allowed_k(i) \right\} & \text{if } q < q_0 \\ J & \text{otherwise} \end{cases} \tag{6}$$

$q_0 \in [0,1]$ is an initially set parameter; q is a random number and $q \in [0,1]$; $allowed_k(i)$ is a set of cities which remain to be visited when the ant *k* is at city *i*; *J* is a random variable determined in accordance with equation (2). This strategy obviously increases the variety of any searching, thus avoiding any premature falling into the local optimal solution and getting bogged down.

Step 3: Local updating of pheromone. After each ant has chosen a city, the amount of pheromone on each side will be updated with equation (7).

$$\tau_{ij}(t+1) = (1-\rho)\,\tau_{ij}(t) + \rho.\tau_0 \tag{7}$$

$$\tau_0 = \frac{1}{n.L_{nn}}$$

where 0< ρ<= 1 is a decay parameter, τ_0 is the initial values of the pheromone trails, where *n* is the number of cities in the TSP and L_{nn} is the cost produced by the nearest neighbor heuristic. Eq. (2) is mainly to avoid very strong pheromone paths to be chosen by other ants and to increase the explorative probability for other paths. Once the edge between city *i* and city *j* has been visited by all ants, the local updating rule makes pheromone level diminish on the edge. So, the effect of the local updating rule is to make an already edge less desirable for a following ant.

Step 4: Computing of the optimal path. After m ants have travelled through all the cities, compute the length of the optimal.

Step 5: Global updating of pheromone. After all the ants have travelled through all the cities, update only the amount of the pheromone on the optimal path according to equation (8).

$$\tau_{ij}(t+1) = (1-\rho)\,\tau_{ij}(t) + \rho.\Delta\tau_{ij}(t) \tag{8}$$

$$\Delta\tau_{ij}(t) = \begin{cases} \frac{1}{L_{gb}} & \text{, if } (i, j) \in \textit{global best tour} \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

L_{gb} is the length of the globally best tour found from the beginning of the trail.

Step 6: If the designated search number is not attained, then repeat the above steps.

5. Proposed Approach

This system presents a powerful approach, called improved ant colony system which is based on dynamic candidate list and improvement strategy based on entropy.

5.1 Dynamic Candidate List

Candidate list is a strategy that tries to improve the performance of an ant algorithm. It was proposed by Gambardella and his colleague [19] to accommodate searching procedure of ACS on larger data. It is used fixed size candidate list. Due to the purpose of improving algorithm performances, the proposed system is also applying candidate list. The proposed candidate list is a dynamic candidate list procedure which captures a suitable number of nodes based on the total number of nodes. It is a static data structure that lists a limited number of preferred closed cities to be visited order by increasing distance. In the ACS algorithm, when the ant chooses the next city, the probability of its transfer from city *i* to city *j* needs to be computed, and then the city whose transfer probability (decision process) is first need to consider those preferred cities listed in the candidate list. Only when an ant cannot find suitable city to choose then the decision to choose a city will consider those which are outside the candidate list. The numbers of closest cities that allowed being included into the candidate list were different from one algorithm to another.

However, it would not allow ants to venture into cities outside the candidate list. The number of cities or the size of the candidate list is also restricted to one fourth of the cities *n*. For example, seven was chosen resulting from the candidate list computation to determine the size of candidate list element for Oliver30 data. The candidate list procedure is as follows:

```
candidate_list=n/4 /*size of candidate list*/
determine cities that not yet visited
do
for i=1 to n
  if city s is not yet visited
        determine distance between city r and city s
          if distance < distance of previous city s
              move city s into node_list
      end for
      candidate_list=node_list
while (until candidate_list is full)
```

5.2 Improved Strategy

In ACO algorithm, α and β are two important adjustable parameters that control the relative weight of trail intensity and desirability. If α is big, the ant will seek for other routes. If β is big, the ant will follow the shortest route found now, which will cause the premature. Usually, α and β are constant, which influence the performance of the algorithm.

The information entropy was introduced to measure the uncertainty of the selection. When ant colony algorithm begins to run, the amount of information on every path equals to each other, information entropy is maximum at this time, but as an enhancement of pheromone on the path, the entropy will be decreased gradually. If the entropy is not controlled currently, the entropy will eventually reduce to 0, that is, the pheromone on only one path is maximum, and the final solution will be mistaken, thus bringing about the premature. In order to overcome the easily-occurred precocious defects for solving complex combinatorial optimization problems with the basic ant colony algorithm, a proposed ant colony algorithm based on information entropy is discussed, using the parameter value selection controlled by information entropy.

The entropy of a random variable is defined as

$$H(X) = -k \sum_{i=1}^{r} P_i \log P_i \tag{10}$$

where p_i represents the probability of the state, and

$p_i \geq 0, \sum_{i=1}^{n} p_i = 1.$ k is a constant weight, and k > 0.

Information entropy has the following characteristics:

(1) Symmetry: When the order of $p_1, p_2, \ldots, p_n$ changes, the entropy does not change. That is $S(p_n, p_2, \ldots, p_1)=S(p_2, p_1, \ldots, p_n)$.

(2) Non-negatice: $S(p_1, p_2, \ldots, p_n)>=0$.

(3) Additive: Relatively independent state, its entropy of addition is equal to the addition of various entropy.

(4) Extremality: When p_i=1/n, the entropy is maximum, its value is lnn.

The information entropy can be obtained to determine the level of certainty for the ants to choose the path. This definition combines the characterstics of ant colony algorithm itself, during algorithm process combining with the information entropy to implement the adaptive adjustment algorithm easily. Therefore, introducing the following here.

$$\alpha'(t) = \frac{E_{\max} - E(t)}{E_{\max}} \tag{11}$$

$$\beta'(t) = 1 - \frac{E_{\max} - E(t)}{2E_{\max}} \tag{12}$$

At the beginning of the algorithm, α (t) is small, and at last, α (t) increases. At the same time, β(t) is biggest at early stage in order to make the algorithm find the optimal route and later it becomes smaller to reinforce the function of random operation and avoiding premature stagnation.

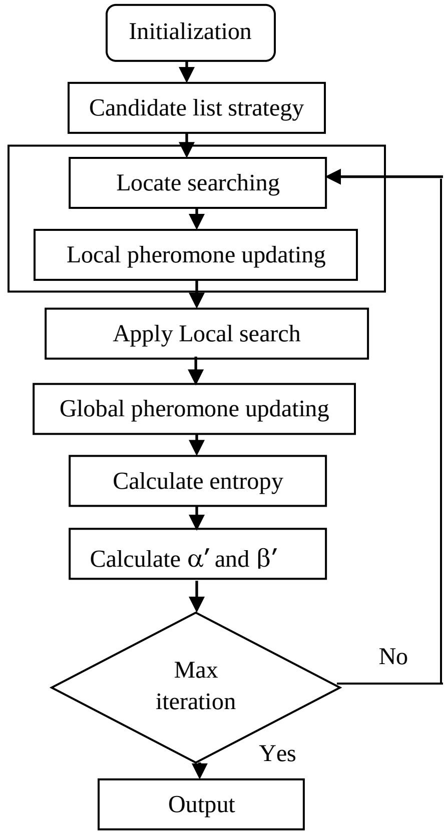

Figure 2. The procedure of proposed algorithm

6. Experimental Results

To demonstrate the proposed method, several TSP problems are considered. They are obtained from the TSPLIB website [21]. In this study, we compared our proposed algorithm results with those of the ACS algorithm in the aspects of algorithm convergence and experiment results.

Table 1 presents the comparison of better results obtained from solving the TSP problems. The parameters are set to the following values: ρ=0.1, q_0 = 0.7, α=1, β=5, the maximum iteration is set 30. In table, it also shows how much deviation from the optimum solution ((best-optimum)/optimum *100). The experiment shows that the proposed algorithm (improved ACO) (IACO) attained better results for TSPs, its efficiency of solution are higher than that of ant colony system.

Table 1: Comparison Results of TSP Problems

	Best Tour Length		
	IACO	ACS+2opt	Chirico's solution [18]
berlin52	7542	7542	7657
eil51	427	428	439

Table 2: Comparison of Tour Length Results of TSP Problems

TSP Problem	Optimum Tour length		Tour Length		
		Algorithm	Best	Average	Deviation
berlin52	7542	ACS	7542	7806.0	0%
		IACS	7542	7641.2	0%
eil51	426	ACS	428	437	0.46%
		IACS	427	431.2	0.23%
st70	675	ACS	679	693.1	0.59%
		IACS	675	686.5	0%
lin105	14379	ACS	14438	14734.3	0.41%
		IACS	14379	14586.2	0%

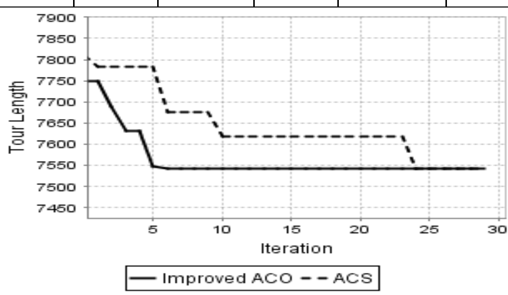

Figure 3. Comparison of convergence speed of berlin52

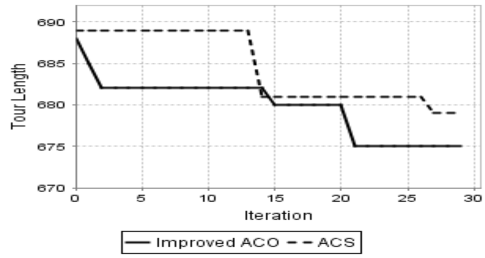

Figure 4. Comparison of convergence speed of eil51

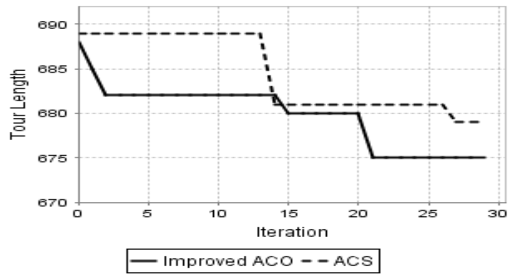

Figure 5. Comparison of convergence speed of st70

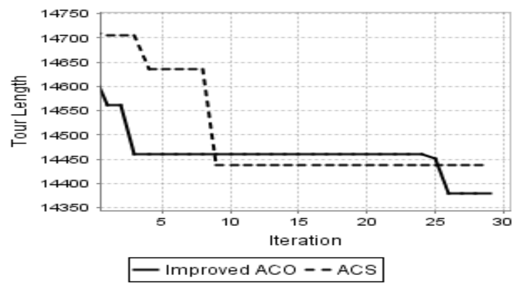

Figure 6. Comparison of convergence speed of lin105

7. Conclusion

This paper presents an approach for solving traveling salesman problem based on improved ant colony algorithm. An improved version of ACO algorithm based on candidate list strategy and a study of the avoidance of stagnation behavior based on entropy is proposed. The solution generated by the ants has been improved by the local search algorithm. From our experimental results, the proposed system is more effective than the ACS algorithm in terms of convergence speed and the ability to finding better solutions.

References

[1] M. Dorigo, V. Maniezzo, and A. Colorni, “The ant system: Optimization by a colony of cooperating agents,” IEEE Transactions on System, Man, and Cybernetics, Part B, Vol.26, pp. 29-41, 1996.

[2] C.Blum, “Ant Colony Optimization: Introduction and recent trends”, ScienceDirect, Physics of Life Reviews 2(2005)353-373.

[3] W. Tsai, F. Tsai, “A New Approach for Solving Large Traveling Salesman Problem Using Evolutionary Ant Rules”, IJCNN 2002, IEEE.

[4] H. Md. Rais, Z. A. Othman, A.R. Hamdan, “Improved Dynamic Ant Colony System(DACS) on symmetric Traveling Salesman Problem (TSP)”, International Conference on Intelligence and Advanced Systems, IEEE, 2007.

[5] J. Han, Y. Tian, “An Improved Ant Colony Optimization Algorithm Based on Dynamic Control of Solution Construction and Mergence of Local Search Solutions”, Fourth International Conference on Natural Computation, IEEE, 2008.

[6] M. Colpan, “Solving Geometric TSP with Ants”, the Pennsylvania State University, 2005

[7] C-Mihaela Pintea, D. Dumitrescu, “Improving Ant System Using a Local Updating Rule”, Proceedings of the Seventh International Symposium and Numeric Algorithms for Scientific Computing (SYNASC’05), IEEE 2005.

[8] T. Stutzle, H.H. Hoos. “MAX-MIN ant system and local search for the traveling salesman problem”, In: IEEE Int’l Conf. on Evolutionary Computation. Indianapolis: IEEE Press, 1997.309~314.

[9] R. Gan, Q. Guo, H. Chang, Y. Yi, “Improved Ant Colony Optimization Algorithm for the Traveling Salesman Problems”, Jouranl of Systems Engineering and Electronics, April 2010, pp 329-333.

[10] C-Xue. Wang, D-Wu. Cui, Y-Kun. Zhang, Z-Rong. Wang, “A Novel Ant Colony System Based on Delauney Triangulation and Self-adaptive Mutation for TSP”, International Joural of Information Technology, Vol.12, No.3, 2006.

[11] Z. A. Othman, H. Md. Rais, A.R. Hamdan,” Strategies DACS3 Increasing its Performances”, European Journal of Scientific Research, 2009.

[12] Z. A. Othman, H. Md. Rais, A.R. Hamdan, “DACS3: Embedding Individual Ant Behavior in Ant Colony System”, World Academy of Science Engineering and Technology 44 2008.

[13] H. Md. Rais, Z. A. Othman, A.R. Hamdan, “Improvement DACS3 Searching Performance using Local Search”, Conference on Data Mining and Optimization, IEEE, 27-28 October 2009.

[14] L. Li, S. Ju, Y. Zhang, “Improved Ant Colony Optimization for the Traveling Salesman Problem”, International Conference on Intelligent Computation Technology and Automation, IEEE, 2008.

[15] X. Wei, “Improved Ant Colony Algorithm Based on Information Entropy”, International Conference on Computational and Information Sciences, IEEE, 2010, 987-0-7695-4270-6.

[16] K. S. Hung, S.F. Su, S. J. Lee, “Improving Ant Colony Optimization for Solving Traveling Salesman Problem”, Journal of Advanced Computational Intelligence and Intelligent Informatics, 2007.

[17] Duan, X. Yu, “Hybrid Ant Colony Optimization Using Memetic Algorithm for Traveling Salesman Problem”, Proceedings of the 2007 IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning (ADPRD 2007).

[18] Ugo Chirico. “A java framework for ant colony systems”, Technical report, Siemens Informatica S.p.A, 2004..

[19] M Dorigo and L.M. Gamgardella, “Ant Colonies for the Traveling Salesman Problem”, Univeristies Libre de Bruxelles, Belgium, Accepted for Publication in BioSystems, 1997.

[20] M. Dorigo, M. Birattari, T. Stuzle, “Ant Colony Optimization, Artificial Ants as a Computational Intelligence Technique”, IEEE Computational Intelligence Magazine, November 2006.

[21] TSPLIB WebPage, http://www.iwr.uniheidelberg.de/groups/comopt/software/TSPLIB95/tsp/