
Improved Ant Colony Optimization Algorithm with Pheromone Correction Strategy for the Traveling Salesman Problem

M. Tuba

R. Jovanovic

Milan Tuba*

Faculty of Computer Science
Megatrend University, Belgrade, Serbia
Bulevar umetnosti 29
11070 N. Belgrade, Serbia
tuba@ieee.org
*Corresponding author

Raka Jovanovic

Texas AM University at Qatar
PO Box 23874, Doha, Qatar
rakabog@yahoo.com

Abstract

A new, improved ant colony optimization algorithm with novel pheromone correction strategy is introduced. It is implemented and tested on the traveling salesman problem. Algorithm modification is based on undesirability of some elements of the current best found solution. The pheromone values for highly undesirable links are significantly lowered by this a posteriori heuristic. This new hybridized algorithm with the strategy for avoiding stagnation by leaving local optima was tested on standard benchmark problems from the TSPLIB library and superiority of our method to the basic ant colony optimization and also to the particle swarm optimization is shown. The best found solutions are improved, as well as the mean values for multiple runs. The computation cost increase for our modification is negligible.

Keywords: ant colony optimization (ACO), traveling salesman problem (TSP), nature inspired algorithms, metaheuristics, swarm intelligence.

1 Introduction

Most real-life problems can be represented as some kind of optimization. Nowadays only hard optimization problems are of research interest. Many discrete (combinatorial) as well as continuous optimization problems are of great practical interest but intractable i.e. cannot be solved within reasonable time by standard, mathematical, deterministic methods. Traveling salesman problem (TSP) is a classic combinatorial example that was researched for the longest period of time and because of that is often used as a benchmark. There are also a number of engineering design problems with mixed continuous and discrete variable and nonlinear objective function and nonlinear constraints that are used as benchmarks for optimization algorithms.

For such intractable problems acceptable suboptimal solutions can usually be found by some metaheuristics, recently very successfully by nature inspired algorithms and as a subclass, swarm intelligence algorithms which include artificial bee colony [1], [2], particle swarm optimization [3], [4], cuckoo search algorithm [5], [6], seeker optimization algorithm [7], [8] etc.

A large number of different algorithms have been developed to find suboptimal solutions for the traveling salesman problem (TSP) in polynomial time. Examples are nearest neighbor, greedy, insertion heuristics, christofides, 2-opt and 3-opt. The non-deterministic metaheuristics like simulated annealing (SA) [9], [10], tabu search [11], genetic algorithms (GA) [12] and particle swarm optimization (PSO) [13] have also been used, giving better results than previously mentioned deterministic algorithms. There are efficient algorithms for the TSP [14] (<http://www.tsp.gatech.edu/concorde.html>) but the research continues on different metaheuristic algorithms [15].

Ant colony optimization (ACO) is another metaheuristic that was successfully used for the TSP [16]. The effectiveness of the ACO has been improved by use of different types of hybridization, like addition of local searchers [17] or combining ACO with GA [18], [19], [20], differential evolution (DE) [21] or kangaroo algorithm [22]. These types of hybridization methods usually have a drawback of complexity of implementation and computation time penalty. It has also been shown that, although adding a local searcher is a good approach in the majority of cases, it may prevent ACO from finding the optimal solution [23]. A very interesting hybridization of ACO is given in article [24] where scout ants, which search the solution space in a more systematic way, are added to the algorithm. Multi-colony systems [25] have been developed, as well as variations of the basic ACO like elitist ant colony, rank based ant colony system and min-max ant system (MMAS) to improve the performance on the TSP [26] in a more natural way, with different pheromone strategies. All these variations have the problem of becoming trapped in local optima due to the fact that they increase the efficiency by making their search more greedy by intensifying the search near the best found solution. Min-max ant system (MMAS) [27] tries to solve this problem by bounding the pheromone values and resetting the pheromone trail if better solutions have not been found in a large number of iterations. Minimum pheromone threshold strategy (MPTS) is another approach to avoid early stagnation applied to the quadratic assignment problems [28]. MPTS adds an additional minimum threshold value and if a pheromone value falls below, it is set to the maximal allowed value of the pheromone trail. Thus the MPTS avoids reinitialization of the pheromone trail like in the MMAS and explores the solution search space more systematically. Another interesting approach is in introducing variable pheromone sensitivity within population of ants [29].

In this paper we propose a new type of hybridization for the ACO and implement it for the TSP. We improve the ACO algorithm with a strategy for avoiding stagnation i.e. leaving local optima if trapped in search for optimal solution. This method is based on the pheromone trail correction. The correction calculation is based on the properties of the best-found solution so far. We have previously used a similar type of improvement on the minimum weight vertex cover problem with great success [30]. We have adopted this approach to TSP and further improved it. The basic idea of this correction is to lower the possibility of edges with high level of undesirability to belong to the optimal solution. We show that our strategy is simple to implement, computationally inexpensive and improves results compared to the pure ACO or PSO [13]. Other relevant research includes [31], [32], [33], [34], [35].

In Section 2 we give an outline of the ACO algorithm. In Section 3 our hybridization used for escaping search stagnation is explained. In Section 4 we analyze and compare experimental results of our algorithm against pure ACO and PSO algorithms for TSP tested on benchmark problems from the TSPLIB [36].

2 Ant Colony Optimization

The ACO algorithm is based on mimicking the behavior of ants colony while gathering food. Each ant starts from the nest and walks towards food until it reaches an intersection, where it has to decide which path to select. In the beginning this choice is random, but after some time the majority of ants will be moving along the optimal path. This happens because of the colony's collective intelligence. Each ant, as he moves, deposits chemical called pheromone thus marking the route taken. Pheromone trail evaporates as time passes. Accordingly, a shorter path will have more pheromone because it will have less time to evaporate before it is deposited again. Each ant chooses paths that have more pheromone so shorter routes will be selected with higher and higher probabilities until practically all ants go along shortest path. But, in case of dynamic change, some new obstacle or new passage, ants will quickly adopt to new situation.

The described behavior can be converted into a computational system as presented by Marco Dorigo and Luca Maria Gambardella [16], with small modifications:

$$s = \begin{cases} \arg \max_{u \notin M_k} \{ \tau_{rs}^\alpha \eta_{rs}^\beta \} & , q \leq q_0 \\ S & , q > q_0 \end{cases} \quad (1)$$

$$p_{rs}^k = \begin{cases} \frac{\tau_{rs}^\alpha \eta_{rs}^\beta}{\sum_{u \notin M_k} \tau_{ru}^\alpha \eta_{ru}^\beta} & , s \notin M_k \\ 0 & , s \in M_k \end{cases} \quad (2)$$

Equations (1) and (2) define the probabilistic decision method that is used by ants during the iterative construction of the path. An artificial ant at step k , currently at node r , after visiting nodes in M_k , uses these probabilities for choosing the next node s . In these expressions q is a random variable chosen uniformly from $[0,1]$ and q_0 is a predefined parameter that gives a balance between exploitation (use of known good paths, $q \leq q_0$) and exploration (search for new paths, $q > q_0$).

In the case of exploitation, the next node is selected by the highest value of S , which gives the value of desirability of an edge depending on the amount of pheromone and its length. In Equations (1) and (2), τ_{rs} is the value corresponding to the amount of pheromone deposited on edge connecting r and s , and η_{rs} is the length of rs which is used as a heuristic. α and β are predefined parameters that specify the influence of the pheromone and the heuristic, respectively.

In the case of exploration the next node is chosen at random with a probability distribution given by Equation (2), where p_{rs} is the probability of choosing edge rs .

The pheromone trail is created using two types of updates. Global update is used to reward good paths by depositing more pheromone on better paths. This is achieved by using the following formula:

$$\tau_{ij} = (1 - \gamma)\tau_{ij} + \gamma\Delta\tau^k, \quad \forall (ij) \in B^k \quad (3)$$

where B^k is the set of all edges in the path ant k used, $\Delta\tau^k$ is the quality of that solution, and γ is a predefined constant from the interval $[0,1]$.

In practical application of ACO, not all the ants deposit pheromone but only the one with the best path. In some cases, only the best path found so far is used to deposit pheromone, to make the search more greedy.

The local updating is used to avoid creation of a very strong edge used by all ants, and it emulates pheromone evaporation. Every time an edge is chosen by an ant it loses some pheromone according to the following formula:

$$\tau_{ij} = (1 - \delta)\tau_{ij} + \delta\tau_0 \quad (4)$$

where δ is a predefined constant from the interval $[0, 1]$, and τ_0 is the quality of the solution created by the greedy algorithm for solving TSP when the heuristic function is the length of an edge.

3 Suspicious elements exclusion pheromone correction strategy (SEE)

The analysis of the ACO algorithm for the TSP when it gets trapped in local optima indicates which corrections should be made, or more precisely what should not appear in the shortest path. There are two simple criteria that can be used on the edges belonging to the best found tour: very long edges and intersecting edges are very unlikely to be a part of the optimal path. The next step was to find a way to, without major corrections to the ACO algorithm, remove suspicious elements from the ants search path. The solution was to significantly lower the amount of pheromone on randomly selected highly suspicious edges belonging to the best path and letting the colony resume its search.

We have divided the correction into two parts: one considering the edge lengths, and the other one that is related to intersecting edges. All intersecting edges are considered highly suspicious and the pheromone trail is corrected on them. Edges for which pheromone trail correction will be applied due to their length are defined in the following way. First we define a heuristic for suspicion $Sus(rs) = length(rs)$. Using this heuristic, we define the probability of edge rs being selected for pheromone correction

$$p_{selected}(rs) = \frac{RK - RankSusp(rs)}{2 * RK} \quad (5)$$

In Equation (5) instead of using the value of Sus for edges, we used $RankSusp$ which represents their rank by suspicion. RK is the maximal number of edges that are considered for correction. The final step is to lower the pheromone trail for the selected edges:

$$(\forall(rs) \in Selected)(\tau_{rs} = \delta\tau_{rs}) \quad (6)$$

where τ_{rs} is the amount of pheromone deposited on edge rs and δ is a fixed small constant that is used to significantly reduce the value of pheromone on edge rs .

The use of suspicion defined as length is not fully effective because the same group of edges would be repetitively selected until a better tour was found. Because of this we introduce an improved suspicion criteria:

$$CorSusp(rs) = Susp(rs) * ExSusp(rs) \quad (7)$$

The improvement consists of tracking which edges have already been selected and preferring the selection of new edges. To do this, a new array $ExSusp$ is introduced with elements initially set to 1. If edge rs is selected, the following correction is done

$$ExSusp(rs) = ExSusp(rs) * \lambda * \frac{len(rs)}{len(MaxEdge)} \quad (8)$$

In Equation (8) $\lambda \in (0, 1)$ is a fixed parameter and $MaxEdge$ is the longest edge in the best solution. The fraction in Equation (8) makes the probability that the edge will be removed proportional to the length of the inspected edge, which makes longer edges selected more often

that short ones, even after a large number of corrections. This is an improvement to our previous implementation of SEE to the minimum weight vertex cover problem where the suspicion was always equally reduced [30]. If a new best set is found, the values of *ExSusp* are reset to 1. A more complex suspicion criteria could have been used that would better analyze the properties of the best found solution, but we wished to show that even with a relatively simple heuristic, improvements can be archived.

For stagnation criterion we used the fact that there was no change to the global best solution for at least n iterations. If this criterion is satisfied, we apply previously defined correction algorithm to the pheromone trail. The pseudo-code for an iteration step for the improved ACO is:

```

Reset Solution for AllAnts
while(! AllAntsFinished) do
  for AllAnts do if(AntNotFinished) do begin
    Add new edge  $rs$  to the solution based on probability
    Local update rule for  $rs$ 
  end for; end while;
Compute Global Update
If (Stagnation) and (UsingSuspisionImprovement)
  Use SuspisionCorrectionMethod

```

The computation cost for this modification is negligible: fraction of one percent of the total computation time. It can be estimated as follows:

The time required to generate one ant solution is of the order n^2 since it includes n steps to create a solution (adding nodes) and for each of these steps on average $n/2$ steps to recalculate probabilities and pheromone corrections.

Computation time for one correction is also of the order n^2 . It consists of three steps where the first one is dominant. The first step is to check all pairs of nodes for intersections and it requires $(n - 1)n/2 \sim n^2$ operations. This step is not even performed if the solution has not been improved since last correction. The second step is to sort links, which requires order of $n * \log(n)$ operations and finally, to select suspicious elements in n operations. The total time is $(n - 1) * n/2 + n * \log(n) + n \sim n^2$, or even less when the first term is missing.

One corrections is computationally equivalent (or less) to one ant solution generation and since the correction is done rarely, for our tests we used 30 iterations as a criterion, and each iteration includes 10 ants, the total correction time is 1/300 of the total time or 0.33%.

The most important difference between our proposed algorithm and all other algorithms used so far is in the way how trapping in local minima is avoided. All other algorithms include fresh elements in the search by rather nondiscriminatory increasing pheromone level for large number of elements currently not in the best solution. Our method, however, lowers the pheromone levels for few undesirable elements in the current best solution letting the ant colony replace them with the most appropriate elements. This proved to be a significant new approach applicable to many situations [30], [37].

4 Tests and Results

In this section we present the results of applying our improved version of the ACO on 11 standard benchmark problems from TSPLIB [36], [38] (acceptable by Concorde software). All of the test have been done on the symmetric version of the TSP. We compared our hybridized ACO algorithm with the PSO algorithm [13] and simple ACO in MMAS variation. For each of

these three methods we compared the best found solution and the average error over all runs. The results for ACO are calculated using our software system [39]. Results are presented in Table 1.

Table 1: Comparison of PSO, simple ACO and ACO combined with SEE

Problem	OPT	PSO		ACO		SEE	
		Best	AvgErr%	Best	AvgErr%	Best	AvgErr%
eil51	426	427	2.58	427	0.52	427	0.23
berlin52	7542	7542	3.85	7542	0.28	7542	0.13
st70	675	675	3.34	676	1.50	675	1.36
eil76	538	546	4.17	538	1.21	538	1.19
pr76	108159	108280	3.82	108359	2.94	108358	2.62
kroa100	21282	-	-	21282	0.77	21282	0.72
lin105	14379	-	-	14379	0.38	14379	0.38
pr124	59030	-	-	59385	0.84	59030	0.72
pr136	96772	-	-	96785	1.09	96781	0.70
u159	42080	-	-	42080	1.27	42080	0.45
kroa200	29368	-	-	29532	1.21	29490	0.93

Columns 3, 5 and 7 (Best) show best results of all runs, columns 4, 6 and 8 (AvgErr) show quality of average solution of all runs: $AvgErr = (Avg - Opt)/Opt * 100\%$.

For the first five problems we used 100 runs to make it comparable to [13]. We also included 6 larger cases and for these cases we used 10 runs since there was no significant difference. For each of the runs we created 200,000 tours. All the colonies had the following parameters: $\alpha = 4$ and $\beta = 1$ which control the influence of pheromone and edge length in the transition rule, $q_0 = 0.9$ specifies the exploitation/exploration rate, $\gamma = 0.1$ and $\delta = 0.1$ specify the global and local update rules, the initial value of the pheromone is given by $\tau_0 = 1/n * L_{nn}$ where L_{nn} is the length of the path calculated by the nearest neighbor heuristic.

For SEE we used the following parameters $RK = ProblemSize/10$, $\lambda = 0.9$, $\delta = 0.1$ and $n = 30$. In our test we used 10 colonies for both, ACO and the improved SEE version. In both cases we used random seeds with values from 0 to 9. This make possible comparison of the effect of SEE. SEE used a separate random number generator for the corrections which provides ants having the same behavior.

The first five problems are the same as used in [13] and the results of our method are better compared to the PSO algorithm. Best solutions were equal in three cases, better in one and worse in one case, but average solutions were significantly better in all five cases. We also compared our hybridized algorithm to simple ACO in MMAS variation. Here we have best solutions equal 6 times and improved 5 times by our algorithm, but again average solutions are improved in 10 cases and equal in one.

5 Conclusion

We have introduced a new type of hybridization for the ACO applied to the TSP. It adds a pheromone trail correction method to ACO metaheuristic which is activated in the situations when the search algorithm has started to stagnate. It is based on the analysis of properties of the best-found tour. SEE adds a new heuristic for determining the undesirability of edges

belonging to the tour and significantly decreases their pheromone values. The calculation time for this heuristic is negligible compared to the other parts of the algorithm. We compared our hybridization to the use of the simple ACO and PSO and the positive effect of this hybridization was evident on standard benchmark problems.

This type of improvement has the potential of application to different problems previously solved using ACO with appropriate heuristics defined for measuring the desirability or suspicion on parts of the solution. In most cases these heuristic functions are easy to calculate and implement. A great advantage of SEE is that it can easily be added to the existing ACO algorithms, with minimal change to the original source code.

References

- [1] Brajevic, I. and Tuba, M. (2012) An upgraded artificial bee colony algorithm (ABC) for constrained optimization problems. *Journal of Intelligent Manufacturing*, **published Online First**, DOI:10.1007/s10845-011-0621-6.
- [2] Bacanin, N. and Tuba, M. (2012) Artificial bee colony (ABC) algorithm for constrained optimization improved with genetic operators. *Studies in Informatics and Control*, **21(2)**, 137–146.
- [3] Secui, D. C., Felea, I., Dzitac, S., and Popper, L. (2010) A swarm intelligence approach to the power dispatch problem. *International Journal of Computers, Communications & Control*, **5(3)**, 375–384.
- [4] Zamfirescu, C.-B. and Filip, F. G. (2010) Swarming models for facilitating collaborative decisions. *International Journal of Computers, Communications & Control*, **5(1)**, 125–137.
- [5] Yang, X. S. and Deb, S. (2010) Engineering optimisation by cuckoo search. *Int. J. of Mathematical Modelling and Numerical Optimisation*, **1(4)**, 330–343.
- [6] Tuba, M., Subotic, M., and Stanarevic, N. (2012) Performance of a modified cuckoo search algorithm for unconstrained optimization problems. *WSEAS Transactions on Systems*, **11(2)**, 62–74.
- [7] Dai, C., Chen, W., Song, Y., and Zhu, Y. (2010) Seeker optimization algorithm: a novel stochastic search algorithm for global numerical optimization. *Journal of Systems Engineering and Electronics*, **21(2)**, 300–311.
- [8] Tuba, M., Brajevic, I., and Jovanovic, R. (2013) Hybrid seeker optimization algorithm for global optimization. *Applied Mathematics and Information Sciences*, **7(3)**, 867–875.
- [9] Wang, Z., Geng, X., and Shao, Z. (2009) An effective simulated annealing algorithm for solving the traveling salesman problem. *Journal of Computational and Theoretical Nanoscience*, **6(7)**, 1680–1686.
- [10] Meer, K. (2007) Simulated annealing versus metropolis for a TSP instance. *Information Processing Letters*, **104(6)**, 216–219.
- [11] Gendreau, M., Laporte, G., and Semet, F. (1998) A tabu search heuristic for the undirected selective travelling salesman problem. *European Journal of Operational Research*, **106(2-3)**, 539–545.

- [12] Liu, F. and Zeng, G. (2009) Study of genetic algorithm with reinforcement learning to solve the TSP. *Expert Systems with Applications*, **36(3)**, 6995–7001.
- [13] Shi, X. H., Liang, Y. C., Lee, H. P., Lu, C., and Wang, Q. X. (2007) Particle swarm optimization-based algorithms for TSP and generalized TSP. *Information Processing Letters*, **103(5)**, 169–176.
- [14] Applegate, D., Bixby, R., Chvatal, V., and Cook, W. (1998) On the solution of the travelling salesman problems. *Documenta Mathematica*, **Extra Volume ICM(III)**, 645–656.
- [15] Rego, C., Gamboa, D., Glover, F., and Osterman, C. (2011) Traveling salesman problem heuristics: Leading methods, implementations and latest advances. *European Journal of Operational Research*, **211(3)**, 427–441.
- [16] Dorigo, M. and Gambardella, L. M. (1997) Ant colonies for the travelling salesman problem. *Biosystems*, **43(2)**, 73–81.
- [17] Chengming, Q. (2008) An ant colony algorithm with stochastic local search for the VRP. *3rd International Conference on Innovative Computing Information and Control*, Los Alamitos, CA, USA, pp. 464–468, IEEE Computer Society.
- [18] Lee, Z.-J., Su, S.-F., Chuang, C.-C., and Liu, K.-H. (2008) Genetic algorithm with ant colony optimization (GA-ACO) for multiple sequence alignment. *Applied Soft Computing*, **8(1)**, 55–78.
- [19] Jun-Qing Li, Q.-K. P. and Xie, S.-X. (2010) A hybrid variable neighborhood search algorithm for solving multi-objective flexible job shop problems. *Computer Science and Information Systems*, **7(4)**, 907–930.
- [20] Negulescu, S., Dzitac, I., and Lascu, A. (2010) Synthetic genes for artificial ants. diversity in ant colony optimization algorithms. *International Journal of Computers, Communications & Control*, **5(2)**, 216–223.
- [21] Zhang, X., Duan, H., and Jin, J. (2008) DEACO: Hybrid ant colony optimization with differential evolution. *IEEE Congress on Evolutionary Computation*, pp. 921–927, IEEE Computer Society.
- [22] Serbencu, A., Minzu, V., and Serbencu, A. (2007) An ant colony system based metaheuristic for solving single machine scheduling problem. *The Annals of Dunarea De Jos University of Galati*, **3**, 19–24.
- [23] Neumann, F., Sudholt, D., and Witt, C. (2008) Rigorous analyses for the combination of ant colony optimization and local search. *Ant Colony Optimization and Swarm Intelligence, LNCS Vol. 5217*, Berlin, Heidelberg, pp. 132–143, Springer-Verlag.
- [24] Gan, R., Guo, Q., Chang, H., and Yi, Y. (2010) Improved ant colony optimization algorithm for the traveling salesman problems. *Journal of Systems Engineering and Electronics*, **21(2)**, 329–333.
- [25] Jovanovic, R., Tuba, M., and Simian, D. (2010) Comparison of different topologies for island-based multi-colony ant algorithms for the minimum weight vertex cover problem. *WSEAS Transactions on Computers*, **9(1)**, 83–92.

-
- [26] Stützle, T. and Dorigo, M. (1999) ACO algorithms for the traveling salesman problem. *Evolutionary Algorithms in Engineering and Computer Science: Recent Advances in Genetic Algorithms, Evolution Strategies, Evolutionary Programming, Genetic Programming and Industrial Applications*, K Miettinen, P Niettaanmaki, M M Makela and J Periaux, editors, p. 500, Willey.
- [27] Stützle, T. and Hoos, H. H. (2000) MAX-MIN ant system. *Future Generation Computer Systems*, **16**(9), 889–914.
- [28] Wong, K. Y. and See, P. C. (2009) A new minimum pheromone threshold strategy (MPTS) for max-min ant system. *Applied Soft Computing*, **9**(3), 882–888.
- [29] Pintea, C.-M., Chira, C., Dumitrescu, D., and Pop, P. C. (2011) Sensitive ants in solving the generalized vehicle routing problem. *International Journal of Computers, Communications & Control*, **6**(4), 228–231.
- [30] Jovanovic, R. and Tuba, M. (2011) An ant colony optimization algorithm with improved pheromone correction strategy for the minimum weight vertex cover problem. *Applied Soft Computing*, **11**(8), 5360–5366.
- [31] Gan, R., Guo, Q., Chang, H., and Yi, Y. (2010) Improved ant colony optimization algorithm for the traveling salesman problems. *Journal of Systems Engineering and Electronics*, **21**(2), 329–333.
- [32] Crişan, G. C. and Nechita, E. (2008) Solving fuzzy TSP with ant algorithms. *International Journal of Computers, Communications & Control*, **III**(S.), 228–231.
- [33] Huang, H., Yang, X., Hao, Z., and Cai, R. (2006) A novel ACO algorithm with adaptive parameter. *Computational Intelligence and Bioinformatics, LNCS 4115*, pp. 12–21, Springer-Verlag Berlin Heidelberg.
- [34] White, C. and Yen, G. (2004) A hybrid evolutionary algorithm for traveling salesman problem. *IEEE Congress on Evolutionary Computation, Vol.2*, pp. 1473–1478, IEEE Computer Society.
- [35] Duan, H. and Yu, X. (2007) Hybrid ant colony optimization using memetic algorithm for traveling salesman problem. *Approximate Dynamic Programming and Reinforcement Learning*, pp. 92–95, IEEE Computer Society.
- [36] Reinelt, G. (1991) TSPLIB - a traveling salesman problem library. *ORSA Journal on Computing*, **3**(4), 376–384.
- [37] Jovanovic, R. and Tuba, M. (2013) Ant colony optimization algorithm with pheromone correction strategy for the minimum connected dominating set problem. *Computer Science and Information Systems (ComSIS)*, **10**(1), 133–149, DOI:10.2298/CSIS110927038J.
- [38] <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/> .
- [39] Jovanovic, R., Tuba, M., and Simian, D. (2008) An object-oriented framework with corresponding graphical user interface for developing ant colony optimization based algorithms. *WSEAS Transactions on Computers*, **7**(12), 1948–1957.

Raka Jovanovic (b. September 10, 1978) received his M.Sc. in Computer Science (2007) and Ph.D. in Computer Science (2012) from University of Belgrade. Now he is in the process of election for Assistant Professor of Computer Science at Faculty of Computer Science, Megatrend University of Belgrade. He was research assistant/associate at the Institute of Physics, University of Belgrade and since 2009 he is a research associate at Texas AM University at Qatar. His current research interests include different aspects of Optimization problems, Data compression, Image processing, Numeric simulation and Fractal imaging. He has (co-)authored more than 20 papers.

Milan Tuba (b. August 27, 1952) received his B. S. in Mathematics, M. S. in Mathematics, M. S. in Computer Science, M. Ph. in Computer Science, Ph. D. in Computer Science from University of Belgrade and New York University. Now he is full professor of Computer Science and Provost for mathematical, natural and technical sciences at Megatrend University of Belgrade. From 1983 to 1994 he was in the U.S.A. first as a graduate student and teaching and research assistant at Vanderbilt University in Nashville and Courant Institute of Mathematical Sciences, New York University and later as Assistant Professor of Electrical Engineering at Cooper Union Graduate School of Engineering, New York. During that time he was the founder and director of Microprocessor Lab and VLSI Lab, leader of scientific projects and supervisor of many theses. From 1994 he was Assistant Professor of Computer Science and Director of Computer Center at University of Belgrade, from 2001 Associate Professor, Faculty of Mathematics, and from 2004 also a Professor of Computer Science and Dean of the College of Computer Science, Megatrend University Belgrade. His research interest includes mathematical, queuing theory and heuristic optimizations applied to computer networks, image processing and combinatorial problems. He is the author or coauthor of more than 130 scientific papers and coeditor or member of the editorial board or scientific committee of number of scientific journals and conferences. Member of the ACM since 1983, IEEE 1984, New York Academy of Sciences 1987, AMS 1995, SIAM, IFNA.