A Hybrid Model Of Max-Min Ant System with Genetic Algorithm For Improved To Travelling Salesman Problem

Tuncay Aydoğan and Raed Al-Badri

***Abstract*—Travelling Salesman Problem (TSP) is one of the oldest combinatorial problems that are classified as NP-hard. Hence, solving this type of problems requires a tremendous number of computational complexities by an intensive search process. Though TSP is an old problem, it is still the attractive spot of study for many researchers due to using in its many applications. The TSP problem is tackled by using a new hybrid algorithm approach in this work. This hybrid approach is applied with Genetic Algorithm (GA) that invokes Max-Min Ant System (MMAS) algorithm to minimize the cost, called HGAMMAS. In the experimental results of HGAMMAS reached to BKS(Best Known Solution) values using TSPLIB.**

***Index Terms*—TSP, genetic algorithm, max-min ant system algorithm, hybrid algorithm.**

I. Introduction

The Combinatorial Optimization (CO) research investigates combinatorial and algorithmic approaches to many discrete optimization problems, many of which arise in the context of data science challenges and rapidly changing complex processes. It aims to define a mathematical formulation of some real-world problems that can aid in decision making by applying various optimization methods in order to obtain the best solution among many possible solutions to these problems [1], [2].

TSP is used widespread in engineering applications and some industrial problems such as scheduling, cellular manufacturing and frequency assignment problems which can be formulated as a CO.

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 Equation 1:

$$\mathrm{d_{ij}} = \sqrt{(\mathrm{x}_i - \mathrm{x}_j{}^2) + (\mathrm{y}_i - \mathrm{y}_j{}^2)} \qquad (1)$$

Ant Colony Optimization (ACO) is inspired by the real ant behaviors. There are a lot of ACO algorithms. The original idea is Ant System (AS) that explains and presents the search strategy. Ants deposit pheromone on the path in a quantity proportional to the quality of the solution represented by path. They resolve choices between competing destinations probabilistically. The probabilities are proportional to the pheromone accumulated on previous iterations. Ant Colony System (ACS) and Max-Min Ant System (MMAS) algorithms have been developed according to pheromone update strategy differences [3]-[20].

The MMAS updates the trails depending on the best solution only. Then, it builds the new solutions based on the intensive trails as in the case of the ant system. It generally performs two main processes till meeting the stopping criteria. These processes are generally constructing the new ant solutions and updating the pheromones. Also, there is an optional local search after each solution creation to enhance the solution quality [3]-[5].

Genetic Algorithm (GA) is one of the oldest meta-heuristic approaches that mimic biological operations such as crossover, mutation, selection [6]. GA belongs to Evolutionary Algorithms (EA) that uses the concept of natural selection. In general, GA is used to gain high quality solutions through evolutionary steps. These steps are basically begun with number of random solutions and then evolve them gradually.

In TSP, there has been significant progress in the development of approximate and exact heuristic and meta-heuristic methods. The exact means small problem instances. Due to this fact, the recent researches have been focused on applying artificial intelligence methods for large problem instances, i.e., iterative improvement heuristics and meta-heuristics.

The heuristics for the TSP builds tours from base by adding an unvisited city in each phase relying on the path cost. However, the major issue of deploying local search heuristics is easily fall in local optima of algorithm. Most of the recent research for TSP focus on using advanced meta-heuristics such as Simulated Annealing [7], [8], Tabu Search [9], [10], Genetic Algorithm [11], [12], Ant Colony Optimization (ACO) [13], [14], Particle Swarm Optimization [15], [16], Neural Network [17], [18], Water Flow-Like Algorithm [19].

For a long time, the GA and MMAS have been implemented successfully on various domains and TSP [20], [21]. In specific, these algorithms have been employed for TSP with other methods such as hybrid and other heuristics [22]-[26].

But in this study, it is aimed to improve the MMAS algorithm with GA for TSP different from the literature

This paper dedicated to combinatorial problems which TSP asks for the shortest path of minimal total cost visiting each given city (node) exactly once by the hybrid approach

Manuscript received November 3, 2017; revised January 12, 2018.

Tuncay Aydoğan is with the Department of Software Engineering, Suleyman Demirel University, 32260, Turkey (e-mail: tuncayaydogan@sdu.edu.tr).

Raed Al-Badri was with the Department of Computer Engineering, Suleyman Demirel University, 32260, Turkey (e-mail: raed.albadri@gmail.com).

using MMAS and GA.

II. Material And Methods

This section describes the proposed solution approach for TSP problem. Based on the findings from the previous studies, it is proposed a hybrid model called HGAMMAS that constructs new solution based on two different strategies, i.e., MMAS algorithm process and GA process as shown in Figure 1.

A. MMAS Algorithm Process

In TSP, constructing new ant solutions begin with an empty ant, then each ant will be assigned to one city. Then, at each construction step the solution is extended by adding a feasible solution component from the set of the neighbor solutions. This process is done iteratively until constructing the last ant. Thereafter, the Ant System updates the pheromones by increasing the value of the pheromone that associated with the good solution and decreasing those which related to bad which is known as pheromone evaporation [20].

After all ants visit all cities and complete their tours, they update their pheromone trail. Initially, all traces contain equal amounts of pheromone. After each repetition in the MMAS, only iteration-best or best-so-far ant update the pheromone traces. Pheromone trails are limited to the interval [τ_{min}, τ_{max}].

The pheromone update is performed as in Equations 2, 3, 4 and 5 [20].

$$p^k{}_{ij} = \begin{cases} \dfrac{\tau_{ij}{}^{\alpha}.\,\eta_{ij}{}^{\beta}}{\sum_{c_{ij}\in N(s^p)} \tau_{ij}{}^{\alpha}.\,\eta_{ij}{}^{\beta}} & ,\, cij \in N(s^p) \\ 0 & ,\, \text{otherwise} \end{cases} \tag{2}$$

$p^k{}_{ij}$ illustrates the probability for such ant k located within i city to travel to j city, where τ denotes the pheromone. The parameters α and β control the relative importance of the pheromone τ_{ij} versus the visibility η_{ij} . The visibility represents the heuristic information, which is given by $\eta_{ij} = \frac{1}{\delta_{ij}}$ recognized as the inverse for $\delta(i,j)$ distance between city i and city j. $N(s^p)$ is the set of feasible components.

MMAS differs from the AS in two main aspects: only the best ant is allowed to update the pheromone trails, and the value of pheromone on the paths is bound.

The pheromone update function is implemented as follows:

$$\tau(i,j) \leftarrow \left[(1-\rho).\,\tau_{ij} + \Delta\tau_{ij}^{best}\right]_{\tau_{min}}^{\tau_{max}} \tag{3}$$

where ρ is the evaporation rate, $\Delta\tau_{ij}^{best}$ is the quantity of pheromone laid on path (i, j) by ant best, and τ_{max} and τ_{min} are respectively the upper and lower bounds imposed on the pheromone.

The rule of Eq(3) is seen in Eq(4).

$$[X]_b^a = \begin{cases} a, & \text{if } x > a \\ b, & \text{if } x < b \\ x, & \text{otherwise} \end{cases} \tag{4}$$

$$\Delta\tau_{ij}^{best} = \begin{cases} \dfrac{1}{L_{best}} & ,\text{if } d(i,j) \in G_{best} \\ 0 & ,\text{otherwise} \end{cases} \tag{5}$$

L_{best} is the length of the tour obtained by the best ant.

All possible solutions for MMAS are evaluated in terms of the shortest path (lowest cost or best cost). The best solution of MMAS is determined as LcMMAS.

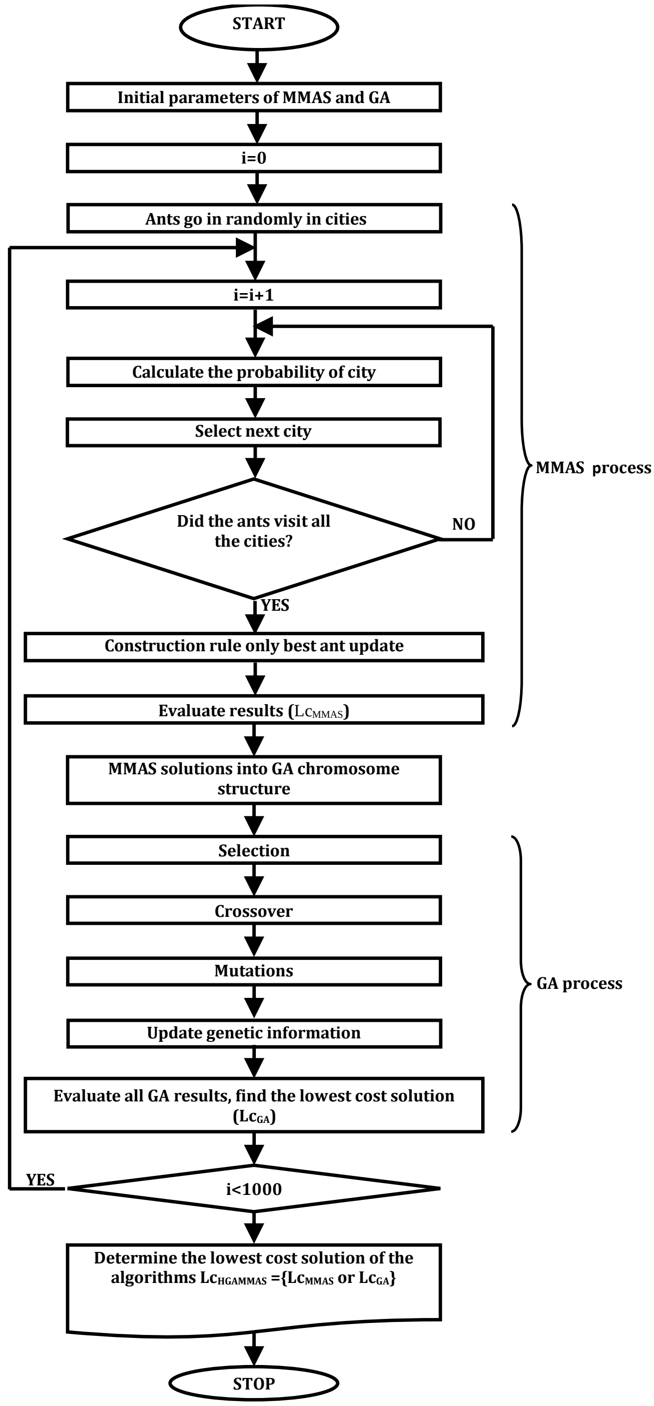

Fig. 1. The proposed hybrid model for TSP problem.

B. GA Process

To introduce the MMAS results into the GA process, the ant-round solutions are transformed into the gene-chromosome form to generate the population for GA as shown in Fig. 2.

MMAS Round Solutions (tours)	GA Population				
2→1→5→3→4 ⇨	2	1	5	3	4

Fig. 2. Demonstration of the Chromosome of the Tour Obtained by Each Ant

Developing the created solutions begin with selecting some solutions from the population by selection operator. Then production operators apply to produce new solutions (offspring) by crossover, mutation. The offspring will take the place of the worst solutions in the population in case of these new solutions are better from the worst.

Practically, the basic GA has numeric parameters, population size, number of generation, crossover rate and mutation rate. These parameters need to be initialized to specific values to control the search process that aims to find the best solution, expected the global best one [11], [12]. These parameters are explained as follows:

Population size parameter represents number of suggested solution that will evolve iteratively via the genetic operators. The typical value of this parameter varies from 10 to 100. In this study it assign fifty individuals as a population size, since it tackles small and medium size of problem instances.

Number of generation's parameter implies the number of performing genetic operators on the population individuals. The value of this parameter has been determined experimentally. Due to the low convergence in proposed method as used two different types of individual productions, it founds one thousands of generations are enough to converge to the best solution.

Crossover rate parameter is the probability of performing crossover operator. Assigning high values for this parameter will lead to premature convergence. Hence, it assigned fifty percent to perform the crossover operation, otherwise the method flow will choose MMAS algorithm to produce the offspring.

Mutation rate parameter determines the probability of modifying the solution locally. In this study, it is assigned one percent due to the variety of the operators that produce the new solutions.

These steps will be repeated until meeting the stopping criterion. And, finally the lowest cost solution of the hybrid algorithms as LcHGAMMAS is the smallest of LcMMAS or LcGA.

III. Results and Discussion

The proposed method has been implemented on TSP. Interface in Fig. 3 developed in java.

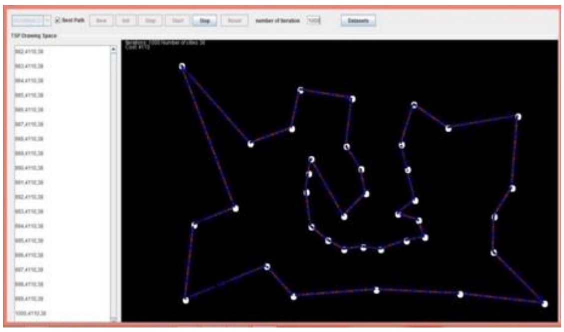

Fig. 3. An Interface of Java Application for TSP

Two experiments were performed to reveal the performance of the developed HGAMMAS algorithm.

In the first experiment, HGAMMAS GA, ACS, AS and MMAS were compared. In this comparison, five random map data sets of 38, 56, 76, 101 and 150 points were used. The experimental results of the different algorithms shown in Table I on the same maps were examined. In this experiment, all algorithms were run ten times for each data set to find the average solution cost (AV).

As a first outcome, ACS, AS and MMAS algorithms from ACO family showed that MMAS produced the solution with the lowest cost. On the contrary, GA calculated with the highest cost.

Table I Results of GA, ACS, AS, MMAS and HGAMMAS Algorithms on Random Maps.

		GA	ACS	AS	MMAS	HGAMMAS
Map-1	AV Cost	4398.75	4360.24	4234.71	4140.41	4117.62
	Start Cost	10460	5739	5739	5739	5739
	Best Cost	4375	4280	4188	4123	4110
Map-2	AV Cost	6270.45	5819.16	5738.70	5486.60	5455.58
	Start Cost	10583	7101	7101	7101	7101
	Best Cost	6122	5794	5659	5448	5398
Map-3	AV Cost	7384.96	6856.84	6526.58	6155.89	5913.44
	Start Cost	14110	7557	7557	7557	7557
	Best Cost	7161	6752	6406	6105	5864
Map-4	AV Cost	11440.38	7630.28	7193.81	6932.73	6565.25
	Start Cost	18300	7934	7934	7934	7934
	Best Cost	9852	7484	7131	6873	6415
Map-5	AV Cost	11953.68	10407.69	10018.98	9174.56	9008.92
	Start Cost	28776	10615	10615	10615	10615
	Best Cost	11583	10272	9957	9089	8855

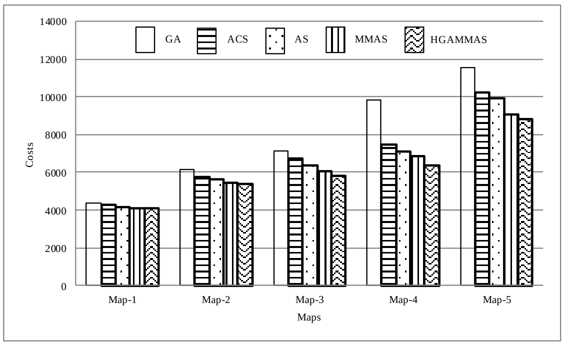

Fig. 4. Cost performance of GA, ACS, AS, MMAS and hgammas algorithms on random maps.

Importantly, at the conclusion of hybridization, both GA and MMAS produced lower cost results. In Figure 4, the cost performance of GA, ACS, AS, MMAS and HGAMMAS algorithms is more clearly seen.

In the second experiment, it is tackled small instances of this problem which are given in TSPLib map [27]. eil51, berlin52, eil76, rad100 and kroa200 data sets have 51, 52, 76, 100 and 200 special universal location points. HGAMMAS has been also compared with GA, MMAS and the other 5 algorithm studies in the literature [17, 19, 28, 29, 30].

The experimental results are shown in Table 2. BKS refers to best known solution cost in literature, and it is regarded as the benchmark. AV is the average cost, and SD is the standard deviation.

Somhom *at al.* proposed a new algorithm, based on a self-organising neural network approach, to solve TSP [28].

Pasti *at al.* and Masutti at al. proposed a new meta-heuristics approach for solving TSP based on a neural network using ideas from the immune system [29], [17]. Chen at al. proposed a new method, called the genetic simulated annealing ant colony system with particle swarm optimization techniques, for solving same problem [30]. Ayman at al. presented a Water Flow-Like algorithm for solving TSP [19]. These studies were taken as reference because they examined the same problem with the same data sets and different approaches.

TABLE II: RESULTS OF HGAMMAS, GA, MMAS AND LITERATURE ALGORITHMS ON TSPLIB MAPS

TSPLIB Library Data Sets		eil51	berlin52	eil76	rd100	kroA200
BKS(Best Known Solution)		426	7542	538	7910	29368
Somhom et al.Algorithm (1997)	AV Cost	440.60	8025	562.30	8239	30416
	SD	3.44	249	5.23	104	133
	Best Cost	433	7715	552	8028	30144
Pasti et al. Algorithm (2006)	AV Cost	438.70	8074	556.10	8253.90	30258
	SD	3.52	270	8.03	149	343
	Best Cost	429	7716	542	7947	29594
Masutti et al.Algorithm (2009)	AV Cost	437.47	7932.50	556.33	8199.77	30190
	SD	4.20	277.30	5.30	80.77	273.40
	Best Cost	427	7542	541	7982	29600
Chen S., et al.Algorithm (2011)	AV Cost	427.27	7542	540.20	7987.60	29738.73
	SD	0.45	0	2.94	62.06	356.07
	Best Cost	427	7542	538	7910	29383
Ayman S., et al.Algorithm (2014)	AV Cost	426.40	7542	538	7942.60	29438.20
	SD	0.52	0	0	15.33	114.84
	Best Cost	426	7542	538	7911	29368
Genetic Algorithm	AV Cost	474	8624.20	638.2	10824.50	57124.50
	SD	13.15	363.12	20.55	523.27	3954.16
	Best Cost	454	7865	588	10181	51389
Max-Min Ant System Algorithm	AV Cost	431.80	7664.20	555.90	8402.90	30972.80
	SD	3.73	154.11	5.46	183.61	528.78
	Best Cost	428	7542	549	8182	30318
HGAMMAS Algorithm	AV Cost	428.80	7542	545.30	8043.40	29984
	SD	2.60	0	4.88	107.38	353.30
	Best Cost	426	7542	538	7910	29427

Table II shows the comparison with 25 experiment's result from the literature. HGAMMAS results are better than 64% (16 exp.) of these experiments, same with 28% (7 exp.) and worse than 8% (2 exp.). The experimental results show that HGAMMAS finds effective solutions in comparison to the related works as shown in Table II.

The same data sets have been tested to see the performance gain of HGAMMAS and its constituent GA and MMAS with 10 experiments's result. HGAMMAS results were better than 90% (9 exp.) of these experiments and gave the same results as 10% (1 exp).

Obviously, in this model yielded a best value for each of the average, and the best, as well as for the standard deviation in most of the instances.

IV. CONCLUSION

This study solved TSP using a hybrid method based on an evolutionary algorithm named GA, and swarm intelligence algorithm known as MMAS. These are two different strategies that search on the best solution based on biological operations in GA and the behavior of the ant foraging in MMAS.

Algorithms combined with the strengths of the hybridization process is designed a new algorithm called HGAMMAS.

HGAMMAS algorithm was tested with GA, MMAS and other 5 algorithms using TSPLib data sets (eil51, berlin52, eil76, rad100 and kroa200).

The new algorithm was found to offer a 3.2% lower cost than the MMAS and a 42.7% lower cost than the GA. These results show that the MMAS algorithm is improved with GA for TSP.

Also, HGAMMAS performance has been shown to achieve the "the Best Known Solution" in literature result on eil51, berlin52, eil76 and rad100 data set.

At the end of the research, it has been shown that the hybridization of high performance algorithms on TSP may give better results. However, the operators of the algorithms to be hybridized must conform to or adapt to each other's mathematical models. The search for adaptive operators of different algorithms that can improve each other is an important research topic.

REFERENCES

[1] C. H. Papadimitriou and K. Steiglitz, "Combinatorial optimization: Algorithms and complexity," *Courier Corporation*, 2013.

[2] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *Journal ACM Computing Surveys (CSUR)*, vol. 35, pp. 268-308, 2003.

[3] C. Blum, "Ant colony optimization: Introduction and recent trends," *Physics of Life Reviews*, vol. 2, pp. 353–373, 2005.

[4] T. Stützle and H. H. Hoos, "Improving the ant system: A detailed report on the max-min ant system," *FG Intellektik, FB Informatik*, TU Darmstadt, Germany, Tech. Rep. AIDA–96–12, 1996.

[5] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," *Book Distributed Optimization by Ant Colonies*, Paris, France, pp. 134 -142, 1991.

[6] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, U Michigan Press, 1975.

[7] T. Peng, W. H. W. Chen, and Z. Dongmo, "Solving the travelling salesman problem by simulated annealing," *Journal of Shanghai Jiaotong University*, 1995.

[8] B. L. Golden and C. C. Skiscim, "Using simulated annealing to solve routing and location problems," *Naval Research Logistics Quarterly*, vol. 33, no. 2, pp. 261-279, 1986.

[9] F. Glover, "Tabu search-part I," *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190 -206, 1989.

[10] F. Huysentruyt and P. D. Causmaecker, *Applying Simulated Annealing and Tabu Search to the Intermittent Travelling Salesman Problem*, 2015.

[11] L. Jiao and L. Wang, "A novel genetic algorithm based on immunity," *IEEE Transactions on Systems, Man, and Cybernetics-part A: Systems and Humans*, vol. 30, no. 5, pp. 552-561, 2000.

[12] J. Yang, C. Wu, H. P. Lee, and Y. Liang, "Solving traveling salesman problems using generalized chromosome genetic algorithm," *Progress in Natural Science*, vol. 18, no. 7, pp. 887-892, 2008.

[13] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53 -66, 1997.

[14] C. B. Cheng and C. P. Mao, "A modified ant colony system for solving the travelling salesman problem with time windows," *Mathematical and Computer Modelling*, vol. 46, no. 9, pp. 1225-1235, 2007.

[15] Y. Marinakis and M. Marinaki, "A hybrid multi-swarm particle swarm optimization algorithm for the probabilistic traveling salesman problem," *Computers and Operations Research*, vol. 37, no. 3, pp. 432 -442, 2010.

[16] X. H. Shi, Y. C. Liang, H. P. Lee, C. Lu, and Q. Wang, "Particle swarm optimization-based algorithms for T SP and generalized TSP," *Information Processing Letters*, vol. 103, no. 5, pp. 169-16, 2007.

[17] T. A. Masutti and C. L. N. De, "A self -organizing neural network using ideas from the immune system to solve the traveling salesman problem," *Information Sciences*, vol. 179, no. 10, pp. 1454 -1468, 2009.

[18] J. C. Créput and A. Koukam, "A memetic neural network for the Euclidean traveling salesman problem," *Neurocomputing*, vol. 72, no. 4, pp. 1250 -1264, 2009.

[19] A. Srour, Z. A. Othman, and A. R. Hamdan, "A water flow-like algorithm for the travelling salesman problem," *Advances in Computer Engineering*, 2014.

[20] P. Guo and Z. Liu, "An ant system based on moderate search for TSP," *ComSIS*, vol. 9, no. 4, pp.1533-1551, 2012.

[21] J. Grefenstette, R. Gopal, B. Rosmaita, and D. V. Gucht, "Genetic algorithms for the traveling salesman problem," in *Proc. the First International Conference on Genetic Algorithms and Their Applications*, pp. 160-168, 1985.

[22] M. K. Rafsanjani, S. Eskandari, A. B. Saeid, "A similarity-based mechanism to control genetic algorithm and local search hybridization to solve traveling salesman problem," *Neural Computing and Applications*, vol. 26, no. 1, pp. 213–222.

[23] P. D. Thanh, H. T. T. Binh, and B. T. Lam, "A survey on hybridizing genetic algorithm with dynamic programming for solving the traveling salesman problem," in *Proc. 2013 International Conference of Soft Computing and Pattern Recognition*, 2015.

[24] L. Rozenberg and R. Rychcicki, "Hybridization of the optimization process as an alternative way to find approximate solutions for NP-hard problems," *Przegląd Elektrotechniczny*, pp. 59-62, 2015.

[25] H. Katagiri, T. Hayashida, I. Nishizaki, and Q. Guo, "A hybrid algorithm based on tabu search and ant colony optimization for k-minimum spanning tree problems," *Expert Systems with Applications*, vol. 39, no. 5, pp. 5681-5686, 2012.

[26] C. F. Tsai and C. Y. Chiang, "Enhancement of traveling salesman problem by hybridization of greedy and 2-opt techniques," in *Proc. 2016 International Conference on Machine Learning and Cybernetics*, 2016.

[27] TSPLib. [Online]. Available: https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/

[28] S. Somhom, A. Modares, and T. Enkawa, "A self-organising model for the travelling salesman problem," *The Journal of the Operational Research Society*, vol. 48, no. 9, pp. 919-928, 1997.

[29] R. Pasti and L. N. Castro, "A neuro-immune network for solving the traveling salesman problem," in Proc. *the 2006 IEEE International Joint Conference on Neural Network Proceedings*, pp. 3760-3766, 2006.

[30] S. M. Chen and C. Y. Chien, "Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques," *Expert Systems with Applications*, vol. 38, pp. 14439–14450, 2011.

Tuncay Aydoğan received the PhD degree in 2005 from Sakarya University. He is currently an associated professor of Suleyman Demirel University. His research interests include smart systems, fuzzy logic, automatic control and ındustrial networks.

Raed Al-Badri received the postgraduate degree in 2017 from Süleyman Demirel University. He is interested in artificial intelligence.