Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2013) 21: 2015 – 2036
© TÜBİTAK
doi:10.3906/elk-1109-44

An efficient solving of the traveling salesman problem: the ant colony system having parameters optimized by the Taguchi method

Musa PEKER,[1,*] **Baha ŞEN,**[2] **Pınar Yıldız KUMRU**[3]

[1]Department of Computer Engineering, Faculty of Engineering, Karabük University, Karabük, Turkey
[2]Department of Computer Engineering, Faculty of Engineering, Yıldırım Beyazıt University, Ulus, Ankara, Turkey
[3]Department of Industrial Engineering, Faculty of Engineering, Kocaeli University, Kocaeli, Turkey

Received: 22.09.2011 • **Accepted:** 19.06.2012 • **Published Online:** 24.10.2013 • **Printed:** 18.11.2013

Abstract: Owing to its complexity, the traveling salesman problem (TSP) is one of the most intensively studied problems in computational mathematics. The TSP is defined as the provision of minimization of total distance, cost, and duration by visiting the n number of points only once in order to arrive at the starting point. Various heuristic algorithms used in many fields have been developed to solve this problem. In this study, a solution was proposed for the TSP using the ant colony system and parameter optimization was taken from the Taguchi method. The implementation was tested by various data sets in the Traveling Salesman Problem Library and a performance analysis was undertaken. In addition to these, a variance analysis was undertaken in order to identify the effect values of the parameters on the system. Implementation software was developed using the MATLAB program, which has a useful interface and simulation support.

Key words: Ant colony system, Taguchi method, route planning, traveling salesman problem

1. Introduction

Route planning is a type of problem that aims to determine the shortest available route from point (x) to point (y) on a map. The most popular application in the route planning problem is the traveling salesman problem (TSP), in which the points of the start and finish are the same [1,2]. The TSP is the type of problem used to determine the shortest (most cost-effective) route to return to the starting point by stopping at each point for each given M point (city) [3]. It is an easily definable but hard to solve NP-hard problem (nondeterministic polynomial-time hard). As the number of points increases, the degree of difficulty increases exponentially. The time complexity of this method is O(n!) [1]. For example, even in instances where 40 cities are used, the number of permutations is so high (40!) that computers cannot solve them in a short time. For this reason, the approximation methods (heuristic and approximation algorithms) that help to reach good solutions in a short time have been used in many fields, in addition to more definite methods [4,5]. This solution to the problem is used in daily life in road and route planning, business planning, and the identification of a sequence of operations during perforation for printed circuit boards.

Scientists have developed successful optimization algorithms by examining animal behaviors. These techniques have been applied to many scientific and engineering problems successfully. Since modeling through

*Correspondence: pekermusa@gmail.com

deterministic methods requires an abundance of mathematical infrastructure and proves inadequate when the dimensions of the problem increase, the attractiveness of the heuristics increases day by day [6,7].

The ant colony algorithm is a population-based heuristic algorithm developed for solving optimization problems. There are many algorithms derived from ant colony metaheuristics that are used in the solution of various problems. These algorithms are different from each other in terms of their formulations; however, they all share the common features of ant colony metaheuristics. The details of the successful ant colony optimization (ACO) variants can be seen in Blum's study [8]. A brief survey of the ACO variants used in the literature and the fields in which they are applied is provided in the Appendix.

Simulation-based software was developed in this study to find the most available route, with the help of the ant colony system (ACS), by visiting the points of settlement whose distances are provided. In the study, the TSP was selected as the route planning method. The study aims to reach the best solution for the problems Berlin52, Eil51, Eil76, Eil101, A280, KroA100, KroC100, Pr76, Lin105, Pr1002, and D1291 in the Traveling Salesman Problem Library (TSPLIB) [9]. The parameter optimization of the Taguchi method was provided in the study in order to increase the performance of the ant colony algorithm. The experimental results for the proposed study were compared with the best known values and studies [10–14] in the literature. Section 2 provides information about these studies.

2. Related work

In recent years, some methods have been presented for solving the TSP. Angeniol et al. [10] presented a method using self-organizing feature maps to solve the TSP. Somhom et al. [11] presented a self-organizing model to solve the TSP. Alaykıran and Engin [12] presented an ant colony algorithm for solving the TSP. Pasti and Castro [13] proposed new metaheuristics for solving TSPs based on a neural network trained using ideas from the immune system. Vallivaara [14] proposed a team ACO (TACO) for solving the TSP. Dorigo and Gambardella [15] presented an ACS to solve the TSP. Ellabib et al. [16] presented a multiple ACS with exchange strategies for solving the TSP. Nguyen et al. [17] presented a genetic algorithm to solve the TSP. Xie and Liu [18] presented a multiagent optimization system for solving the TSP. Yi et al. [19] presented a fast elastic net method for solving the TSP. Saadatmand-Tarzjan et al. [20] presented a novel constructive-optimizer neural network for solving the TSP. Sauer and Coelho [21] presented a discrete differential evolution with a local search method to solve the TSP. Shi et al. [22] presented an ACO method with time windows to solve the prize-collecting TSP. Li et al. [23] presented an improved ACO method for solving the TSP. Liu and Zeng [24] presented a genetic algorithm with reinforcement learning to solve the TSP. Chien and Chen [25] presented a method for solving the TSP based on the parallelized genetic ACS. Cheng and Wang [26] presented a genetic algorithm with a decomposition technique to solve the vehicle routing problem with time windows. Naimi and Taherinejad [27] presented an ant colony algorithm with a new interpretation of a local updating process for solving the TSP. Marinakis and Marinaki [28] presented a hybrid algorithm by combining genetic algorithms and particle swarm optimization algorithms for solving the vehicle routing problem. Karaboğa and Görkemli [29] developed a new combinatorial artificial bee colony algorithm for solving the TSP. You et al. [30] proposed a parallel ACO algorithm based on a quantum dynamic mechanism for the TSP. Masutti and Castro [31] suggested some adjustments on an immune-inspired self-organizing neural network for solving the TSP. Bianchi et al. [32] introduced the ACO for a different version of the TSP, the probabilistic TSP, where each customer has a given probability of requiring a visit. Attempts have been made to improve the performance of ACO algorithms since their introduction.

Tsai et al. [33] presented a new metaheuristic approach called the ACO with multiple ant clans (ACOMAC) algorithm to solve the TSP.

Analyses are undertaken in studies using ant colony algorithms to find out how the parameters are determined. According to the analyses, the optimal parameter series in ant colony algorithms and other metaheuristic methods (genetic algorithms, simulated annealing, particle swarm algorithms, etc.) are generally determined through the use of experiments. These experiments are based on trial and error. In many studies, it can be seen that the authors follow the suggestions obtained from the studies of Dorigo and Gambardella [34]. The Appendix displays the details of the parameter determination methods used in previous studies and the parameter values that they optimized. There are studies in the literature focusing on the importance of parameter values and the need for them to be identified systematically. Baykasoğlu et al. stated that there is no optimal strategy to determine the best parameter setting in the ant colony algorithm and that it is imperative for experts in the field to identify a practical method approved by researchers [35]. Eiben et al. explained the importance of parameter values in metaheuristic methods [36]. Chan and Swarnkar stated that the ACS parameters Q, α, β, and ρ should be chosen carefully, as they might lead to poor performance of the algorithm, and they investigated the parameter behaviors graphically in their study [37]. Simon et al. indicated that it is only through the appropriate selection of parameters that it will be possible for the ACS to show a high level of convergent behavior [38]. Karpenko et al. maintained that the algorithm performance is sensitive to the parameter selection in the ACS [39]. Shi et al. indicated that the α, ρ, Q, and β parameters in the ACS affect the calculation efficiency and the convergence of the algorithm directly or indirectly [40].

In light of this information, the method proposed in this study is thought to provide a good solution in terms of parameter optimization. Although it is possible to obtain optimum values through experiments that use trial-and-error methods, that would require a large number of experiments to achieve the desired results, which would then be problematic in terms of time. On the other hand, the method proposed in this study will provide a new approach to the solution of the problem. It is also thought that the proposed method could be used not only for ant colony algorithms, but also for other heuristic algorithms.

3. Method

3.1. Ant colony system

Ants have the ability to find the shortest route from the food source to their ant hills without using their sense of sight [41]. They also have the ability to adapt to environmental change. For example, let us think about the shortest route between the ant nest and the food that was discovered by the ants. In case this route cannot be used due to external reasons, the ants can discover the shortest route again [42,43].

At the beginning, the ants follow a straight route and they leave pheromones in their path, which helps the other ants to find their way (see Figure 1). When a barrier is placed in front of the ants, they cannot follow the pheromones anymore and they randomly select 1 of the 2 routes they could use. Since the use of the short route will be higher in unit time, the amount of pheromone will also be higher. Accordingly, the number of ants selecting the short route will increase in time. After a specific time, all of the ants will prefer the shortest route.

The following of the densely traced route by controlling the traces of the ants who acted randomly at first is an autocatalytic type of behavior and there is a synergic effect in the interaction of the ants. Since the algorithm is inspired by ant colonies, the system is called the ant system (AS) and the algorithm is called the ant colony algorithm. Ant colonies are used in optimization problems. The ants in the AS are different from natural ants. They have memory, they are not completely blind, and they live in an environment where time is discrete/discontinuous.

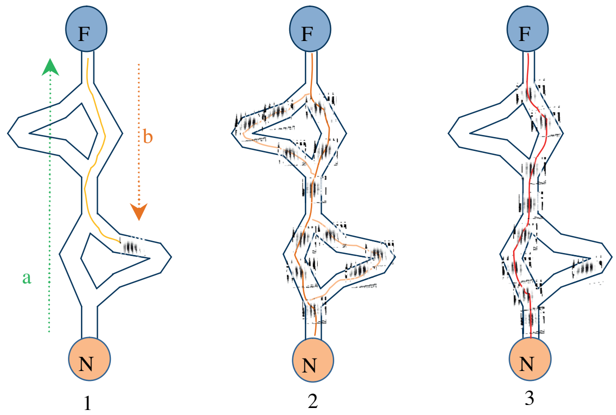

Figure 1. The probability of real ants finding the shortest route [44].

The basic operations in the working of the algorithm are the increase in the amount of pheromones in the routes that the artificial ants pass through at the end of their tours, the vaporization of specific amounts of pheromones, the finding of the best solution and pheromone updating accordingly, and the realization of new tours according to these renewed pheromone amounts.

3.2. Ant colony formulation

The ant colony algorithm can be described briefly as follows. Initially, put m ants into n cities randomly, and each edge has an initial pheromone $\tau ij\,(0)$ between 2 cities. The first element of each ant's tabu list is set to be equal to its starting town [15]. Thereafter, every ant moves from town i to town j. According to the following probability function, ants select the next city [45].

$$p_{ij}^{k}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^{\alpha}\cdot[\eta_{ij}]^{\beta}}{\sum\limits_{l\in allowedk(t)}[\tau_{il}(t)]^{\alpha}\cdot[\eta_{il}]^{\beta}} & if \quad j\in allowedk(t) \\ 0 & otherwise \end{cases} \tag{1}$$

where:

- $\tau_{ij}(t)$: Pheromone trace amount in t time in the (i,j) corners.
- The desirability value (also referred to as the visibility or heuristic information) between a pair of cities is the inverse of their distance $\eta_{ij} = 1\,/\,d_{ij}$, where d_{ij} is the distance between cities i and j. Hence, if the distance on the arc $(i,\,j)$ is long, visiting city j after city i (or vice versa) will be less desirable.
- α (alpha) is the parameter that shows the relative importance of the pheromone trace.
- β (beta) is the parameter that shows the importance of the visibility value.
- $allowed_k(t)$ is the set of cities not visited by ant k at time t.

A data structure, which is also called a *tabu list*, is associated with each ant in order to stop the ants from visiting a city more than once [46]. This list, $tabuk(t)$, sustains a set of visited cities up to time t by the kth ant. Therefore, the set $allowedk(t)$ can be stated as follows: $allowedk(t) = \{j \mid j \notin tabuk(t)\}$. When a tour is completed, the $tabuk(t)$ list $(k = 1, \ldots, m)$ is cleared and every ant is once more free to select an alternative tour for the next cycle. The term tabu list is used here to imply a simple memory consisting of a set of already visited cities, and it bears no relation to tabu search [46].

After visiting all of the points in the problem, a tour or iteration is completed [45]. At this point, the pheromone trace amount is updated according to Eq. (2) [47].

There are 2 basic elements in pheromone updating:

a) Vaporization of the pheromones in all of the routes in specified ratios (rate of vaporization).

b) Increasing of the pheromone amounts in the routes that the ants use by inverse proportioning to the route distance [48].

The evaporation rate decreases the importance of previous solutions. An increase in pheromone inversely proportional with tour length provides an increase in the importance of good solutions [41].

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

where:

- $0 < \rho < 1$ is a value that shows the pheromone trail evaporation. Parameter ρ is used to refrain from unlimited collection of the pheromone trails and it also permits the algorithm to 'forget' previous bad judgments [49]. If an arc is not selected by the ants, its related pheromone strength diminishes exponentially.

- At the beginning of the optimization, the initial pheromone value $\tau_{ij}(0)$ is usually set to a constant value. Dorigo et al. did some experiments on $\tau_{ij}(0) = 0$, $\tau_{ij}(0) = 1/(n\,x\,Lnn)$, where $\tau_{ij}(0)$ is inspired by Q-learning [50]. Lnn is the tour length produced by the nearest heuristic neighbor, where n is the number of decision points [51]. The authors found that the result on $\tau_{ij}(0) = 0$ was worse than the other ones, whereas the latter ones performed rather well. In this paper, we choose to employ $\tau_{ij}(0) = 1/(n\,x\,Lnn)$ for its simplicity and the adequacy of the performance.

- $\Delta\tau_{ij}$ shows the pheromone trace amount owing to the selection of the (i, j) corners in the ant's tour [52]. This amount is calculated according to Eq. (3):

$$\Delta\tau_{ij}(t, t+1) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t, t+1) \tag{3}$$

where

- m is the total number of ants, and

- $\Delta\tau_{ij}^{k}$ is the pheromone trace amount left by ant k in the (i, j) corners.

Eq. (4) shows the contribution of ant k to the pheromone trace amount at any point in the (i, j) corners.

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{Lk}, & if \quad k^{th}\, ant\, uses\, edge\, (i,j)\, in\, its\, tour\, (between\, time\, t\, and\, (t+n)) \\ 0, & otherwise \end{cases}, \tag{4}$$

where

- Q is a constant, representing the amount of pheromone an ant put on the path after an exploitation, and Lk is the tour length of ant k.

4. Application and discussion

In this study, a solution for the TSP is proposed utilizing the ACS and parameter optimization taken from the Taguchi method. The implementation was tested by various data sets in the TSPLIB and a performance analysis was undertaken. In addition, a variance analysis was done in order to identify the effect values of the parameters on the system. Implementation software was developed using the MATLAB program, which employs a useful interface and simulation support. The flow chart for the proposed system is provided in Figure 2.

4.1. Parameter set

The values of the parameters used in the solution of the problem were found with the help of optimization. Values α and β in Eq. (1) are control parameters and they affect the rates of the pheromone trace and contribution to heuristic functions ($\alpha \geq 0, \beta \geq 0$) [48]. If ($\alpha = 0$), the selection will be based only on visibility, i.e. heuristic function, and the algorithm is turned into a stochastic greedy search algorithm. If ($\beta = 0$), only pheromone amplification is at work; this will lead to the rapid emergence of a stagnation situation with the corresponding generation of tours, which, in general, are strongly suboptimal [53]. Thus, they both need to contribute weight to the function as required. However, without pheromone vaporization (ρ) the ACS cannot provide good solutions ($0 < \rho < 1$). The pheromones do not contain any information at this stage since the first inquiries in the search space are completely random [53]. Hence, a vaporization mechanism is needed in order for the other ants to forget these useless values easily and to follow routes through good solutions.

Another important parameter in the ACS is the current number of ants. A high number of ants in the system causes suboptimal results, whereas too few ants remove the collective work feature, as a result of pheromone vaporization [54]. For this reason, Dorigo stated that the number of ants should be equal to the number of cities ($m = n$) and that they should be distributed to cities randomly at first. In this study, we equalized the number of ants with the number of cities according to this information.

Parameter Q is a parameter related to the quantity of routes initiated by the ants and it stops the cumulative pheromone levels from being too high or too low [55]. In many studies, it has been stated that the parameters do not have a very important effect on the solutions provided that they are not too small [12,50,55]. This parameter was added to the parameter optimization in order to identify its effect on the ACO. The values of the Q parameter that was to be optimized were identified as $Q \in$ 1, 10, 100, 1000, 10000}. These values were identified according to the response of the algorithm obtained as a result of intensive experiments prior to optimization and by utilizing other studies [55–59].

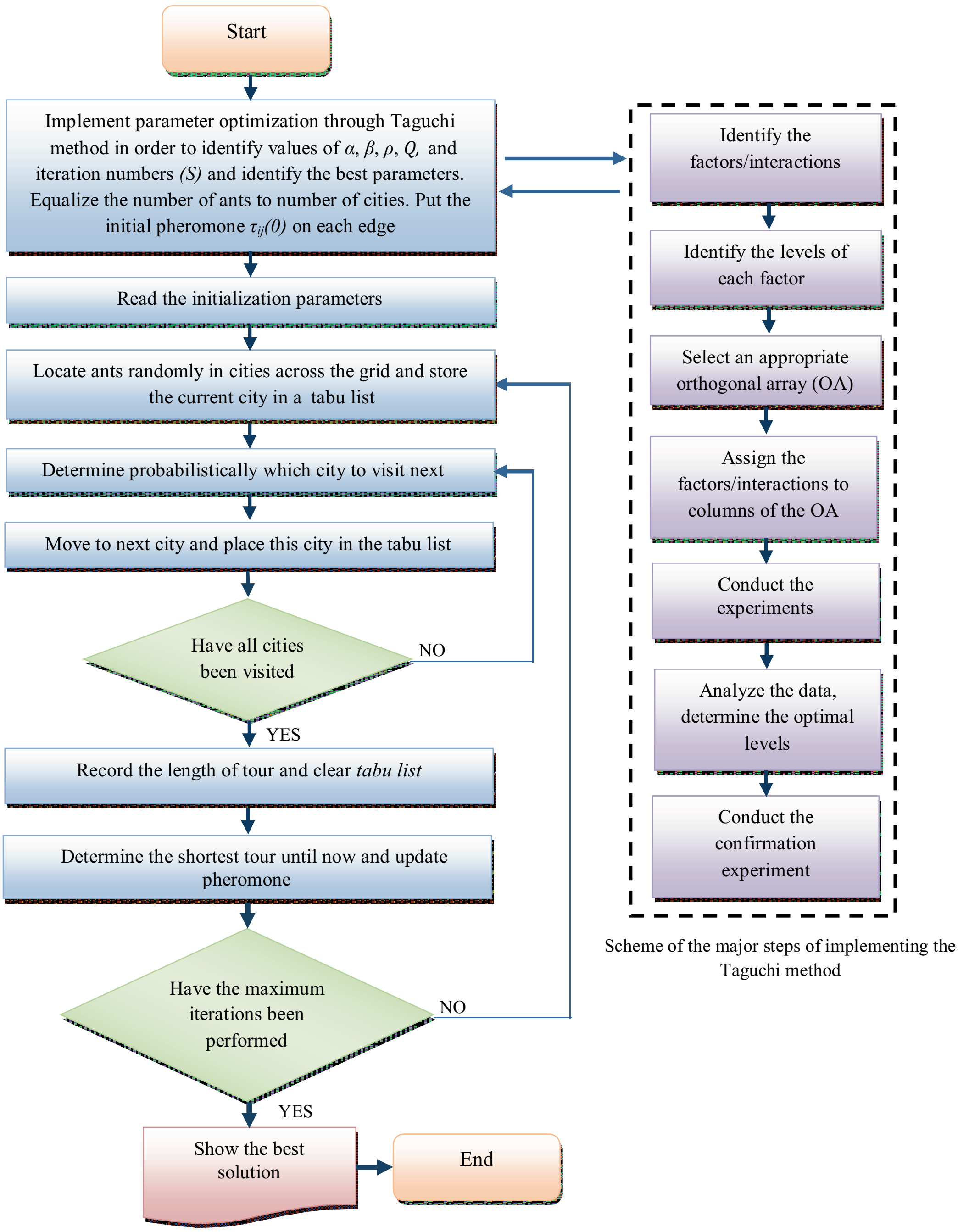

Figure 2. The flow chart of the proposed method.

The values of the iteration number (S) were identified as $S \in$ 1000, 2000, 3000, 4000, 5000$\}$. It can be seen in the literature that these values are used extensively [11,33,41,55,57]. This parameter was included in the optimization in order to learn which iteration value best represented the problem.

4.2. Selection of parameters through the Taguchi method

Many trials were performed on the test problem in order to find the most suitable values for the parameters that are specific to the problem. The 'Lin105.tsp' problem from the TSPLIB, consisting of 105 cities, was used as the test problem [9]. This problem is a hard problem owing to its complex nature.

Taguchi experimental design was used in order to identify the range of effectiveness for the parameters affecting the ACS by utilizing a minimal number of experiments. The Taguchi experimental design is a design method that aims to minimize the variability in a product or operation in line with a specified function (least best, most best, targeted value best) by selecting the most suitable combinations of the controllable factor levels compared to the uncontrollable factors that create variability for a specific product or operation [60–62]. Figure 2 presents the steps of the Taguchi method.

The parameters in Table 1, whose lowest and highest levels have been determined, are classified in levels of 5. Table 2 displays the level values obtained for the factors. In the case of an experiment where classic full factorials are used, $5^5 = 3125$ experiments would be needed for each observation value, whereas it is sufficient to undertake 25 experiments with a selected L25 orthogonal index (5 parameters and 5 levels in each parameter). This method ensures the identification of effective parameters and levels with fewer experiments by providing balance among the orthogonal index, parameters, and levels [63].

Table 1. Range of the experimental parameters.

Factors	Lower limit	Upper limit
α	1	5
β	1	5
ρ	0	1
Q	1	10,000
S	1000	5000

Table 2. Level values obtained for factors.

Factors	Level 1	Level 2	Level 3	Level 4	Level 5
α	1	2	3	4	5
β	1	2	3	4	5
ρ	0.1	0.3	0.5	0.7	0.9
Q	1	10	100	1000	10,000
S	1000	2000	3000	4000	5000

Table 3 presents the results obtained from the experiments undertaken with the help of Minitab [64] in the related orthogonal index, parameter, and value levels. The table lists the values of α, β, ρ, Q, and S, created in the L25 octagonal order. The table also presents the tour values obtained through implementing the parameters to the problem. The parameters were implemented twice in order to increase the reliability of the experiments and the average of 2 different observation values was obtained.

The results obtained with the help of the Taguchi experimental design were evaluated by transforming the results into signal/noise (S/N) ratios (see Table 4). The signal value represents the real value that the system provides and the noise factor represents the unwanted factors in the evaluated value [65,66]. This table shows us the order of importance of the variables and displays the level at which variables should be used in order to achieve the best result. Delta is the difference between the maximum and minimum values of the variable. Rank shows the level according to which variables should be used in order to receive the best solution. The level with the highest S/N value is the most suitable level. The S/N values were calculated using the 'least best' formula, in Eq. (5), since the shortest total route length was desired for the TSP. Here, Y is the performance characteristic value (strain) and n is the number of Y values [67].

$$\frac{S}{N} = -10 \times \log\left(\frac{1}{n} \times \sum_{i=1}^{n} Y_i^2\right) \tag{5}$$

Table 3. Parameter values obtained by the Taguchi method in line with parameter range values and results of the experiments.

Standard order	α	β	ρ	Q	S	Average of the results of the observation
1	1	4	0.7	1000	4000	16,865.4
2	2	3	0.7	10,000	1000	18,697.5
3	1	3	0.5	100	3000	14,385.6
.	.	.	.	.	.	.
.	.	.	.	.	.	.
24	5	4	0.5	10	1000	25,624.1
25	5	5	0.7	100	2000	17,697.3

Table 4. S/N ratios obtained in the Taguchi experimental design.

Level	α	β	ρ	Q	S
1	–84.39	–84.94	–84.96	–85.84	–85.92
2	–85.36	–84.74	–85.10	–85.27	–85.03
3	–85.03	–84.76	–85.52	–84.45	–84.89
4	–85.04	–86.28	–84.68	–85.17	–84.91
5	–85.86	–84.97	–85.43	–84.97	–84.93
Delta	1.47	1.54	0.84	1.40	1.03
Rank	1	2	4	3	3

It is possible to see the effect values of factors graphically according to the S/N values in Figure 3. When Table 4 and Figure 3 were taken into consideration, the best parameter set was identified as $\alpha1{-}\beta2{-}\rho4 - Q3 - S3$ according to the S/N values. The real values of these levels are presented in Table 5.

Table 5. Parameter set obtained through optimization.

Parameter	Value
α	1
β	2
ρ	0.7
Ant number (m)	City number
Iteration number (S)	3000
Q	100

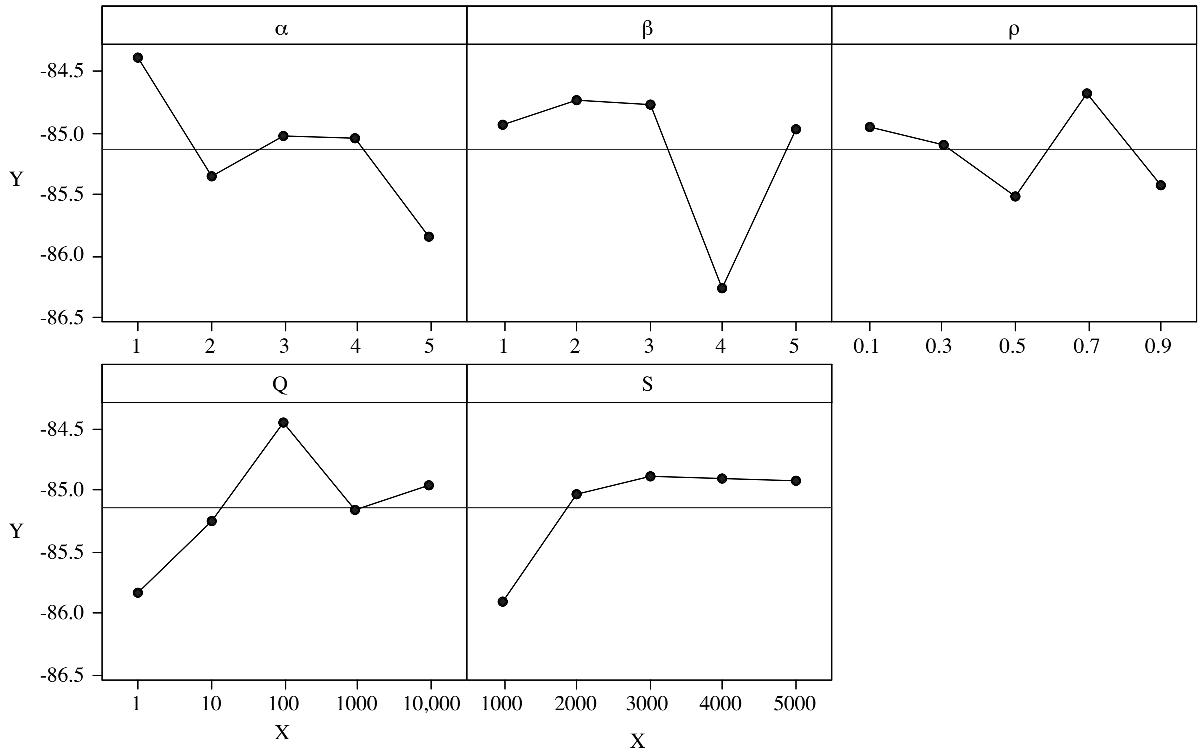

Figure 3. Main effects plot for the S/N ratio.

After identifying the best parameter levels, the percentages of the parameter effect on the system were examined through variance analysis (ANOVA). ANOVA helps to statistically identify which factor affects which operation and allows the combination that creates the best performance to be identified [68]. Moreover, the statistical reliability of the results is tested. The ANOVA table was created in light of the S/N values obtained through experiments, and the important factors were identified.

When Table 6 is examined, it is seen that parameters are statistically significant at P = 0.05 (5%). Five important elements were found in the analysis by alpha (α) (20.10%), beta (β) (29.09%), evaporation rate (ρ) (17.83%), Q (8.35%), and S (13.70%). Figure 4 presents the effect percentages of the parameters graphically. According to this result, the most important parameter value was obtained as beta (β), although all of the parameters are important in the ACS.

Table 6. Results obtained by ANOVA, P < 0.05.

Parameters	Degrees of freedom	Sum of squares	F	Sig (P)	Percentage value
α	4	5.75	1.27	0.001	20.10%
β	4	8.32	2.06	0.000	29.09%
ρ	4	5.10	1.09	0.002	17.83%
Q	4	2.39	0.46	0.010	8.35%
S	4	3.92	0.80	0.007	13.70%
Error	4	3.12			10.90%
Total	24	28.6			100%

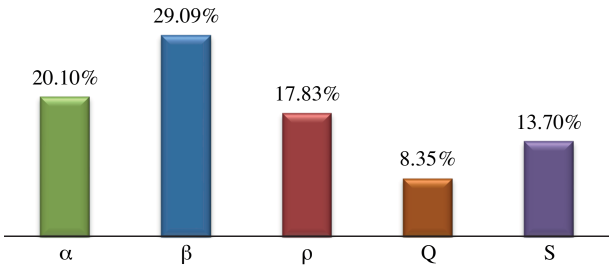

Figure 4. Effect percentages of the parameters on the system.

4.3. Solution of TSP and performance analyses

The TSP was solved with a computer with an Intel Pentium 2.0 GHz microprocessor and 2 GB RAM. The TSP was implemented using the MATLAB software package (MATLAB version R2010a). The problems were solved with the parameter set in Table 5. Figure 5 presents the working of the simulation software and the screen shots of the results.

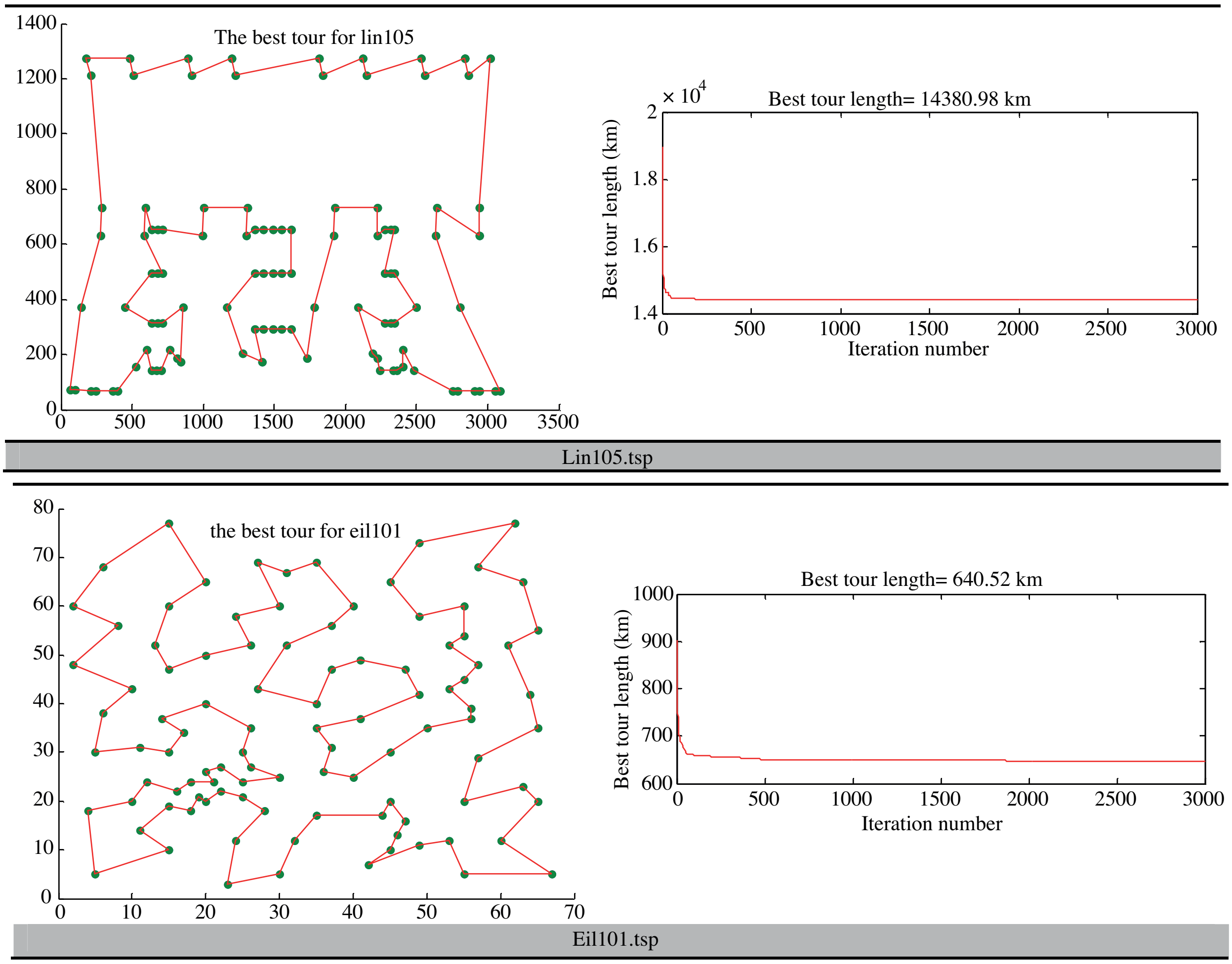

Figure 5. Solution of the Lin105 and Eil101 problems and the performance graphics.

We also compared the experimental results of the proposed method with the ones presented by Angeniol et al. [10], Somhom et al. [11], Alaykıran and Engin [12], Pasti and Castro [13], and Vallivaara [14]. The selection of these studies was based on the fact that the tests of these studies were implemented with the TSPLIB library and they are considered to be among the studies that display best performance. Angeniol et al. are the only ones who did not implement any experiments with the TSPLIB library. Our reason for choosing the algorithm of Angeniol et al. for comparison is because it is one of the pioneering algorithms in the literature and it also incorporates a growing mechanism in the network. Berlin52, Eil51, Eil76, Eil101, A280, KroA100, KroC100, Pr76, Lin105, Pr1002, and D1291 were selected from the TSPLIB library for comparison. The studies referenced in [12] and [14] are ant colony-based studies. In the present study, ant colony-based studies were examined in more detail, and the following conclusions were drawn:

Alaykıran and Engin [12] proposed a solution for the TSP that requires the use of ACO. In their study, the number of iterations was fixed at 2000 and the number of ants was fixed at 5. The authors employed experiments for parameters α, β, and ρ in order to determine the appropriate parameters. After they identified the border values for each parameter, the authors undertook experiments. The results of the experiments pointed to the following values: $\alpha = 0.6$, $\beta = 4.3$, and $\rho = 0.8$. The study did not provide any information about the value of parameter Q.

Vallivaara [14] proposed a TACO to solve the TSP. The innovative idea that was proposed includes the exchange of each ant in the ACO with other ant teams and providing opportunities for the teams to create solutions for the TSP. The iteration number in the study was fixed to the value of 150. The results of the experiments helped us to determine $\alpha = 1$, $\beta = 2$, and $\rho = 0.1$, and the ant number (m) as 10. The study did not provide any information about the value of parameter Q.

4.3.1. Comparison of the algorithms

The results presented are the minimum, average, and standard deviation of the cost (tour length) over 30 runs. The relative error (RE) values obtained according to the best known values and computing times for the algorithms are also displayed. Furthermore, the performances of other algorithms from the literature are considered when available.

Table 7 presents the comparative results acquired from the current study and the earlier solutions provided for problems regarding other heuristics in the literature. All of the results are presented for 30 runs. The best solutions are emphasized in bold. In Table 7, all of the algorithms were reimplemented using MATLAB 7.10. We chose to reimplement the algorithms in order to reduce the bias introduced by the implementation and programming language when comparing the computational cost of the different algorithms. All of the experiments required the completion of 3000 iterations to ensure a more impartial comparison. The values suggested by the authors in their original articles were used for the other parameter values. In general, better performances were observed with the proposed method in all of the cases compared with those of the other algorithms, with the exception of the Berlin52, Eil101, and KroA100 problems. The best solutions for Berlin52, Eil101, and KroA100 were obtained with our study and Somhom et al.'s method, with Somhom et al.'s method, and Pasti and Castro's methods, respectively.

Table 8 presents the comparison of computing times for each algorithm. It also displays the RE of the cost for each solution obtained by the algorithms in comparison with the best known solutions. The RE is a number that compares how incorrect a quantity is from a number considered to be true [69]. The relative percent error value given in Table 8 was obtained from the following equation:

$$Error\,(\%) = \frac{(A\, -\, B)}{B}\, \times\, 100, \tag{6}$$

Table 7. Obtained tour lengths and comparative analysis: best known solution (BKS), average solution (Mean), standard deviation (SD) and best solution found (Best).

Problem	Angeniol et al.'s method			Somhom et al.'s method			Pasti and Castro's method		
	Mean	SD	Best	Mean	SD	Best	Mean	SD	Best
Berlin52	8368.60	251.83	7778.3	7984.20	248.04	**7542.0**	8035.68	260.13	7543.9
A280	2672.22	25.16	2588.3	2640.41	21.94	2584.8	2604.58	19.68	2585.9
Eil51	443.41	4.52	432.1	438.62	3.34	433.0	438.76	3.82	429.0
Eil76	565.14	6.95	554.4	568.42	5.03	545.0	556.16	7.03	545.0
KroA100	24,670.65	859.04	23,009.5	21,835.53	184.21	21,641.1	21,868.41	245.76	**21,369.0**
Pr76	12,3310.76	5513.29	116,378.0	120,830.52	1290.24	112,372.0	112,566.42	1304.86	109,233.2
KroC100	23,485.16	884.30	22,340.8	21,408.01	198.92	20,924.2	21,231.63	230.91	20,915.6
Eil101	665.87	7.87	655.6	654.22	5.21	**637.0**	654.82	5.97	641.0
Lin105	16,111.44	632.10	14,995.5	14,771.95	108.95	14,466.5	14,702.22	325.30	14,382.1
Pr1002	345,195.01	2681.02	332,741.2	334,591.46	2467.03	328,451.1	329,857.01	2294.54	315,678.2
D1291	63,412.43	617.19	62,693.2	61,712.14	439.13	59,112.3	59,345.11	642.54	58,678.3

Table 7. Contunied

Problem	Vallivaara's method			Alaykıran and Engin's method			Our work			BKS
	Mean	SD	Best	Mean	SD	Best	Mean	SD	Best	
Berlin52	7543.33	278.10	7543.1	7650.30	245.21	7590.2	7635.02	234.52	**7542.0**	7542
A280	2626.44	29.98	2591.3	2800.25	22.03	2595.1	2650.01	19.09	**2583.1**	2579
Eil51	434.82	4.05	428.2	458.15	3.40	433.3	435.19	3.15	**426.0**	426
Eil76	563.12	8.03	547.8	608.02	5.37	552.3	565.28	5.92	**544**	538
KroA100	21,789.41	300.72	21,585.8	22,560.42	123.76	21,670.9	21,567.34	102.26	21,382.3	21,282
Pr76	110,025.90	1648.22	108,959.0	111,050.01	1650.07	109,382.1	110,420.00	1617.78	**108,785.4**	108,159
KroC100	21,368.00	295.19	20,904.5	22,109.45	178.16	21,035.2	21,850.18	201.64	**20,801.2**	20,749
Eil101	643.20	6.57	640.2	680.95	4.85	653.0	655.27	4.59	640.5	629
Lin105	14,745.32	357.37	14,389.5	15,030.39	103.03	14,750.2	14,475.05	98.38	**14,380.9**	14,379
Pr1002	298,191.82	2591.50	288,658.3	314,560.00	2317.70	306,755.4	287,500.11	2268.97	**280,001.1**	259,045
D1291	61,712.10	688.40	59,112.3	58,952.44	584.33	57,695.2	56,904.16	597.19	**54,915.2**	50,801

Table 8. Computing time for each algorithm and RE values obtained according to best known values.

Problem	City number	Angeniol et al.'s method		Somhom et al.'s method		Pasti and Castro's method		Vallivaara's method		Alaykıran and Engin's method		Our work	
		Time (s)	RE (%)	Time (s)	RE (%)	Time (s)	RE (%)	Time (s)	RE (%)	Time (s)	RE (%)	Time (s)	RE (%)
Berlin52	52	3.33	3.13	9.15	**0.00**	11.30	0.03	3.85	0.01	3.17	0.64	**3.15**	**0.00**
A280	280	54.49	0.36	59.80	0.22	68.55	0.27	80.40	0.48	**51.60**	0.62	52.50	**0.16**
Eil51	51	4.25	1.43	3.34	1.62	11.56	0.70	17.52	0.52	3.85	1.71	**3.32**	**0.00**
Eil76	76	6.91	3.05	6.77	1.28	17.12	1.30	8.43	1.82	5.95	2.66	**5.92**	**1.12**
KroA100	100	12.09	8.12	11.85	1.66	29.93	**0.41**	15.02	1.43	12.15	1.83	**11.52**	0.47
Pr76	76	**480.50**	7.60	1024.60	3.75	1795.24	0.99	2100.34	0.74	597.50	1.13	510.22	**0.58**
KroC100	100	**11.34**	7.67	11.94	0.84	31.55	0.80	16.90	0.75	15.80	1.38	15.50	**0.25**
Eil101	101	**11.37**	4.23	12.14	**1.26**	31.60	1.91	24.50	1.78	22.06	3.82	20.75	1.82
Lin105	105	**12.44**	4.29	13.27	0.60	35.54	0.02	28.58	0.07	20.40	2.58	21.45	**0.01**
Pr1002	1002	**969.18**	28.45	2124.20	21.13	1226.80	21.86	4000.07	11.43	1325.60	18.42	1330.51	**8.09**
D1291	1291	**1250.35**	23.41	1300.05	14.06	1450.90	15.51	5500.01	16.36	1480.54	13.57	1440.59	**8.10**

where A is the results obtained from this study and B is the optimum tour length. The best solutions are emphasized in bold in the table. All of the results are provided for 30 runs. When computing times in the table are considered, it is seen that the proposed method requires less computing time for problems involving less than 100 cities in general; however, for problems involving more than 100 cities, the method proposed by Angeniol et al. necessitates less computing time.

Figure 6 provides the results of the comparison graphically according to the RE values.

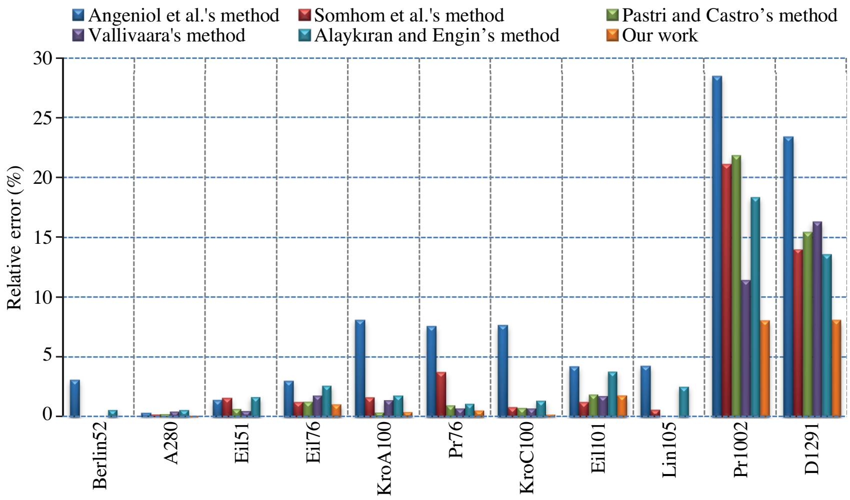

Figure 6. The results of the comparison graphically, according to the RE values.

Table 9 displays the comparison of the proposed method and the other methods suggested in the literature. The experimental results in this table are reported directly from the original articles of the authors, as opposed to Table 7. Table 9 does not carry the results obtained by Angeniol et al. because those authors did not utilize the TSPLIB for any experiments. Again, it is possible to observe that the proposed method is competitive when the target is to find the lowest cost routes for the TSP.

The comparison of results shows that the proposed method is only 1% worse than the best known solutions to date in instances involving less than 100 cities, a fact that proves the strength of the method. In bigger instances, deviations of up to 8% are observed compared to the best known results. This may be explained by the fact that the degree of difficulty increases exponentially with the increase of the number of cities. The fact that the algorithm does not obtain the best solutions in large-scale problems shows that the proposed method trips on the local minima, improvements in the algorithms are necessary, and the proposed method should be only applied to smaller scale problems.

The results obtained in light of the relative data show that the proposed method is superior to other approaches in terms of the attributes of the solution (tour length); however, more computing time is required for solving problems involving a higher number of cities. All in all, the average solution obtained with the proposed algorithm was satisfactory, generally better than those found by similar algorithms.

An important point to consider in the results analysis is that the algorithm (ACS) used by Alaykıran and Engin and the algorithm (ACS) used in the present study are identical. The most important difference between these 2 studies is the difference in the method used for identifying the parameters and the parameter values obtained as a result of the optimization. The differences in parameter values have affected the results significantly. This shows that algorithm parameter values are as important as the parameter selection.

Table 9. Computational results of the proposed method in comparison with other results directly taken from the literature (when available). All of the results for the proposed method are taken from 30 runs.

Problem	Somhom et al.'s method		Pasti and Castro's method		Vallivaara's method		Alaykıran and Engin's method		Our work		BKS
	Mean	Best	Mean	Best	Mean	Best	Mean	Best	Mean	Best	
Berlin52	7980.1	**7542**	8044.3	7544	N/A	N/A	N/A	7596	7635.4	**7542.0**	7542
A280	N/A	N/A	N/A	N/A	N/A	N/A	N/A	2608	2650.3	**2583.2**	2579
Eil51	440.6	433	N/A	N/A	443.8	440	N/A	435	435.4	**426.4**	426
Eil76	567.4	546	N/A	N/A	558.3	551	N/A	559	565.5	**544.1**	538
KroA100	21,840.4	**21,347**	21,356.5	21,875	N/A	N/A	N/A	21,677	21,567.1	21382.3	21,282
Pr76	N/A	N/A	N/A	N/A	N/A	N/A	N/A	109,544	110,420.1	**108,785.4**	108,159
KroC100	21,408.9	20,926	N/A	N/A	N/A	N/A	N/A	21,028	21,850.1	**20,801.2**	20,749
Eil101	655.2	**637**	N/A	N/A	655.1	650	N/A	667	655.0	640.5	629
Lin105	14,774.5	14,468	N/A	N/A	N/A	N/A	N/A	14,893	14,475.2	**14,380.9**	14,379
Pr1002	N/A	N/A	N/A	N/A	N/A	N/A	N/A	307,761	287,500.1	**280,001.1**	259,045
D1291	N/A	N/A	N/A	N/A	N/A	N/A	N/A	57,713	56,904.7	**54,915.2**	50,801

Note: 'N/A' means that there are not any studies related to that type of problem.

4.4. Future investigations

The performance analysis shows that the algorithm trips on local minima. As a future undertaking, the algorithm will be implemented on more complex test problems and hybridized with some local search heuristics to obtain better results. It is also aimed to obtain a system that generates faster solutions by reducing the computing time with the help of running the algorithm on graphic processors, such as CUDA-based GPU programming, whose popularity is increasing day by day.

5. Results

This study applies the ant colony algorithm inspired by the behavior of ant colonies to the TSP, a route planning problem. The results obtained are summarized below:

- The most interesting part of the study is the parameter optimization that affects the system performance to a great extent. It was seen in the system, utilizing the Taguchi method, that it is possible to provide fewer experiments and save time and costs. It is also possible through this system to undertake parameter optimization in ant colony metaheuristics in order to achieve optimum or close to optimum results. The best parameter set that can be achieved with 3125 trials was obtained with only 25 trials with this method.
- It is possible to observe the operations of the ant colony algorithm to achieve results with the help of the simulation-based features of this study.
- The most effective parameter in the parameter optimization of the study was found to be β, with the effect value of 29.09%. The percentage effect of the Q parameter, which is not regarded to be important in the literature and therefore is not included in parameter optimization, was found to be 8.35%. This value is a substantial value that points to the need for this parameter to be included in studies.

The Taguchi method can be used to realize parameter optimization in other heuristic algorithms to ameliorate performance.

Appendix. A brief survey of the ant-based algorithms and details of the system parameters.

Authors	Application field	Used ant algorithm	Determination method of system parameters	Parameters
Mazzeo and Loiseau [70]	Vehicle routing problem	Ant colony optimization (ACO)	Not mentioned	Not mentioned
Tsai et al. [71]	Data mining	ACODF (ACO with different favor algorithm)	Not mentioned	Not mentioned
Korosec et al. [72]	Mesh-partitioning problem	Multilevel ant colony algorithm	Not mentioned	Not mentioned
Dorigo and Gambardella [15]	TSP	ACO	Experiments	α, β, ρ,S,m
Chang et al. [73]	Fault section estimation	Ant system	Experiments	α, β, ρ,Q
Song et al. [74]	Solving combined heat and power economic dispatch	ACSA (ant colony search algorithm)	Experiments	α, β, ρ,S,m
Afshar [75]	Storm water network design	MMAS	Experiments and previous studies	α, β, ρ,p^{best},S,m
Randall and Lewis [76]	TSP	Parallel ACO	Previous studies	β, ρ,S,m,Q
Chu et al. [77]	TSP	PACS (parallel ant colony system)	Previous studies	α, β, ρ,S
Weng and Liu [78]	Time series data segmentation	RACOS (random ACOS)	Previous studies	α, β, ρ
Doerner et al. [79]	Project portfolio selection	P-ACO (Pareto ant colony optimization)	Experiments	α, β, ρ, q_0 (algorithm-specific)
Ying and Liao [80]	Flowshop scheduling	ACS	Previous studies	α, β, ρ
Rajendran et al. [81]	Flowshop scheduling	ACO and parallel ACO	Previous studies	α, β, ρ
Alaykıran and Engin [12]	TSP	ACS	Experiments	α, β, ρ
Shyu et al. [82]	Cell assignment problems in PCS networks	ACO	Experiments	α, β, ρ,S,m
De Campos et al. [83]	Learning Bayesian networks	ACO	Previous studies	β, ρ,m,S
Lim et al. [84]	Bandwidth minimization problem	Hybrid AS	Experiments and previous studies	α, β,m,S
Blum [8]	Open shop scheduling	Beam-ACO	Experiments	α, β, ρ,m
Gao and Wu [85]	Routing in Cognitive Radio Network	ACO	Experiments	α, β, ρ
Merkle et al. [86]	Resource-constrained project scheduling	ACO	Experiments	α, β, ρ,m
Benlian and Zhiquan [87]	Multitarget tracking	Multiobjective ACO	Experiments	α, β, ρ
Duan and Yu [88]	TSP	Hybrid ACO	Experiments	α, β, ρ
Tsai et al. [89]	TSP	ACOMAC algorithm	Experiments	α, β, ρ
Vallivaara et al. [14]	TSP	A team ant colony system (TACO)	Experiments	m, α, β, ρ
Ortiz and Requena [90]	Control rod pattern design	Ant system	Previous studies	α, β, ρ,m
Maier et al. [91]	Data clustering	ACO	Experiments	α, β, ρ
Kuo et al. [92]	Clustering analysis	Ant K-means	Experiments	α, β, ρ,Q
Gamez and Puerta [93]	Graph triangulation problem	ACO	Experiments	α, β, ρ,m
Kuo et al. [94]	Developing diagnostic system	Fuzzy ant colony system	Experiments	α, β, ρ
Yin [95]	Polygonal approximation	Ant colony search algorithm	Experiments	α, β, ρ,m
Stützle and Hoos [96]	TSP	MMAS	Experiments	α, β, ρ, m, p^{best} (algorithm-specific),S

References

[1] B. Özkan, U. Cevre, A. Uğur, "[Route planning with a hybrid optimization technique]", Akademik Bilişim, Vol. 44, pp. 173–183, 2008 (article in Turkish with an abstract in English).

[2] İ. Meşecan, İ.Ö. Bucak, Ö. Asilkan, "Searching for the shortest path through group processing for TSP", Mathematical and Computational Applications, Vol. 16, pp. 53–65, 2011.

[3] J.Y. Potvin, "The TSP: a neural network perspective", ORSA Journal on Computing, Vol. 5, pp. 328–348, 1993.

[4] L.M. Gambardella, M. Dorigo, "Solving symmetric and asymmetric TSPs by ant colonies", Proceedings of IEEE International Conference on Evolutionary Computation, pp. 622–627, 1996.

[5] S. Yigit, B. Tugrul, F.V Celebi, "A complete CAD model for type-I quantum cascade lasers with the use of artificial bee colony algorithm", Journal of Artificial Intelligence, Vol. 5, p. 76, 2012.

[6] T. Keskintürk, H, Söyler, "Global ant colony optimization", Journal of the Faculty of Engineering and Architecture of Gazi University, Vol. 21, pp. 689–698, 2006.

[7] M. Peker, B. Şen, Ş. Bayır, "Using artificial intelligence techniques for large scale set partitioning problems", World Conference on Innovation and Software Development, İstanbul, 2011.

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

[9] TSPLIB. University of Heidelberg. Available at http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/ (accessed 13.04.2011).

[10] B. Angeniol, G.S.L.C Vaubois, J.Y.L. Texier, "Self-organizing feature maps and the traveling salesman problem", Neural Networks, Vol. 1, pp. 289–293, 1988.

[11] S. Somhom, A. Modares, T. Enkawa, "A self-organizing model for the traveling salesman problem", Journal of the Operational Research Society, Vol. 48, pp. 919–928. 1997.

[12] K. Alaykıran, O. Engin, "Ant colony meta heuristic and an application on TSP", Journal of the Faculty of Engineering and Architecture of Gazi University, Vol. 1, pp. 69–76, 2005.

[13] R. Pasti, L.N.D. Castro, "A neuro-immune network for solving the traveling salesman problem", Proceedings of the International Joint Conference on Neural Networks, pp. 3760–3766, 2006.

[14] I. Vallivaara, "A team ant colony optimization algorithm for the multiple travelling salesmen problem with minmax objective", 27th IASTED International Conference on Modelling, Identification, and Control (MIC 2008), pp. 1–6, 2008.

[15] M. Dorigo, L.M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem", IEEE Transactions on Evolutionary Computation, Vol. 1, pp. 53–66, 1997.

[16] I. Ellabib, P. Calamai, O. Basir, "Exchange strategies for multiple ant colony system", Information Sciences, Vol. 177, pp. 1248–1264, 2007.

[17] H.D. Nguyen, I. Yoshihara, K. Yamamori, M. Yasunaga, "Implementation of an effective hybrid GA for large-scale traveling salesman problem", IEEE Transactions on System, Man, and Cybernetics – Part B, Vol. 37, pp. 92–99, 2007.

[18] X.F. Xie, J. Liu, "Multiagent optimization system for solving the traveling salesman problem", IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, Vol. 39, pp. 489–502, 2008.

[19] J. Yi, W. Bi, G. Yang, Z. Tang, "A fast elastic net method for traveling salesman problem", Proceedings of the 8th International Conference on Intelligent Systems Design and Applications, pp. 462–467, 2008.

[20] M. Saadatmand-Tarzjan, M. Khademi, M.R. Akbarzadeh-T, H.A. Moghaddan, "A novel constructive-optimizer neural network for the traveling salesman problem", IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, Vol. 37, pp. 754–770, 2007.

[21] J.G. Sauer, L. Coelho. “Discrete differential evolution with local search to solve the traveling salesman problem: fundamentals and case studies”, Proceedings of the 7th IEEE International Conference on Cybernetic Intelligence Systems, pp. 1–6, 2008.

[22] X. Shi, L. Wang, Y. Zhou, Y. Liang, “An ant colony optimization method for prize-collecting traveling salesman problem with time windows”, 4th International Conference on Natural Computation, Vol. 7, pp. 480–484, 2008.

[23] L. Li, S. Ju, Y. Zhang, “Improved ant colony optimization for the traveling salesman problem”, Proceedings of the International Conference on Intelligent Computation Technology and Automation, Vol. 1, pp. 76–80, 2008.

[24] F. Liu, G. Zeng, “Study of genetic algorithm with reinforcement learning to solve the TSP”, Expert Systems with Applications, Vol. 36, pp. 6995–7001, 2009.

[25] C.Y. Chien, S.M. Chen, “A new method for handling the traveling salesman problem based on parallelized genetic colony systems”, International Conference on Machine Learning and Cybernetics, pp. 2828–2833, 2009.

[26] C.B. Cheng, K.P. Wang, “Solving a vehicle routing problem with time windows by a decomposition technique and a genetic algorithm”, Expert Systems with Applications, Vol. 36, pp. 7758–7763, 2009.

[27] H.M. Naimi, N. Taherinejad, “New robust and efficient ant colony algorithms: Using new interpretation of local updating process”, Expert Systems with Applications, Vol. 36, pp. 481–488, 2009.

[28] Y. Marinakis, M. Marinaki, “A hybrid genetic–particle swarm optimization algorithm for solving the vehicle routing problem”, Expert Systems with Applications, Vol. 37, pp. 1446–1455, 2010.

[29] D. Karaboğa, B. Görkemli, “A combinatorial artificial bee colony algorithm for Traveling Salesman Problem”, International Symposium on Innovations in Intelligent Systems and Applications, pp. 50–53, 2011.

[30] X.M. You, S. Liu, Y.M. Wang, “Quantum dynamic mechanism-based parallel ant colony optimization algorithm”, International Journal of Computational Intelligence Systems, pp. 101–113, 2010.

[31] T.A.S. Masutti, L.N.D. Castro, “A self-organizing neural network using ideas from the immune system to solve the traveling salesman problem”, Information Sciences, Vol. 179, pp. 1454–1468, 2009.

[32] L. Bianchi, L.M. Gambardella, M. Dorigo, “An ant colony optimization approach to the probabilistic travelling salesman problem”, Lecture Notes in Computer Science, Vol. 2439, pp. 883–892, 2002.

[33] C.F. Tsai, C.W. Tsai, C.C. Tseng, “A new and efficient ant-based heuristic method for solving the traveling salesman problem”, Expert Systems, Vol. 20, pp. 179–186, 2003.

[34] M. Dorigo, L.M. Gambardella, “Ant colonies for the travelling salesman problem”, BioSystems, Vol. 43, pp. 73–81, 1997.

[35] A. Baykasoğlu, T. Dereli, I. Sabuncu, “An ant colony algorithm for solving budget constrained and unconstrained dynamic facility layout problems”, Omega: The International Journal of Management Science, Vol. 34, pp. 385–396, 2006.

[36] A.E. Eiben, R. Hinterding, Z. Michalewicz, “Parameter control in evolutionary algorithms”, IEEE Transactions on Evolutionary Computation, Vol. 3, pp. 124–141, 1999.

[37] F.T.S. Chan, R. Swarnkar, “Ant colony optimization approach to a fuzzy goal programming model for machine tool selection and operation allocation problem in an FMS”, Robotics and Computer-Integrated Manufacturing, Vol. 22, pp. 353–362, 2006.

[38] S.P. Simon, N.P. Padhy, R.S. Anand, “An ant colony system approach for unit commitment problem”, Electrical Power & Energy Systems, Vol. 28, 315–323, 2006.

[39] O. Karpenko, J. Shi, Y. Dai, “Prediction of MHC class II binders using the ant colony search strategy”, Artificial Intelligence in Medicine, Vol. 35, pp. 147–156, 2005.

[40] L. Shi, J. Hao, J. Zhou, G. Xu, “Ant colony optimization algorithm with random perturbation behavior to the problem of optimal unit commitment with probabilistic spinning reserve determination”, Electric Power Systems Research, Vol. 69, pp. 295–303, 2004.

[41] T. Keskintürk, M.B. Yıldırım, M. Barut, "An ant colony optimization algorithm for load balancing in parallel machines with sequence-dependent setup times", Computers & Operations Research, Vol. 39, pp. 1225–1235, 2012.

[42] M. Dorigo, L.M. Gambardella, "Ant colonies for the TSP", BioSystems, Vol. 43, pp. 73–81, 1997.

[43] A.O. Bozdoğan, A.E. Yılmaz, M. Efe, "Performance analysis of Swarm Optimization approaches for the generalized assignment problem in multi-target tracking applications", Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 6, pp. 1059–1076, 2010.

[44] Wikipedia. Ant Colony Optimization. Available at http://en.wikipedia.org/wiki/ant_colony_optimization (accessed 13.04.2011).

[45] K. Alaykıran, O. Engin, A. Döyen, "Using ant colony optimization to solve hybrid flow shop scheduling problems", Journal of Advanced Manufacturing Technology, Vol. 35, pp. 541–550, 2007.

[46] L. Xie, H.B. Mei, "The application of the ant colony decision rule algorithm on distributed data mining", Communications of the International Information Management Association, Vol. 7, pp. 85–94, 2007.

[47] A. Akdaglı, K. Güney, D. Karaboga, "Pattern nulling of linear antenna arrays by controlling only the element positions with the use of improved touring ant colony optimization algorithm", Journal of Electromagnetic Waves and Applications, Vol. 8, pp. 4–10, 2002.

[48] T. Stützle, H. Hoos, "The MAX-MIN ant system and local search for the TSP", Proceedings of the IEEE International Conference on Evolutionary Computation, pp. 309–314, 1997.

[49] D. Karaboga, K. Guney, A. Akdagli, "Antenna array pattern nulling by controlling both amplitude and phase using modified touring ant colony optimisation algorithm", International Journal of Electronics, Vol. 91, pp. 241–251, 2004.

[50] M. Dorigo, V. Maniezzo, A. Colorni, "Ant system: optimization by a colony of cooperating agents", IEEE Transactions on Systems, Man, and Cybernetics: Part B: Cybernetics, Vol. 26, pp. 29–41, 1996.

[51] J. Zhang, X. Hu, X. Tan, J.H. Zhong, Q. Huang, "Implementation of an ant colony optimization for job shop scheduling problem", Transactions of the Institute of Measurement and Control, Vol. 28, pp. 93–108, 2006.

[52] H. Zeng, T. Pukkala, H. Peltola, S. Kellomäki, "Application of ant colony optimization for the risk management of wind damage in forest planning", Silva Fennica, Vol. 41, pp. 315–332, 2007.

[53] M. Dorigo, G.D. Caro, L.M. Gambardella, "Ant algorithms for discrete optimization", Artificial Life, Vol. 5, pp. 137–172, 1999.

[54] A. Uğur, D. Aydın, "Visualization with Java of ant system algorithm", Akademik Bilişim, pp. 572–576, 2006.

[55] Z. Dan, H. Hongyan, H. Yu, "The optimal selection of the parameters for the ant colony algorithm with small-perturbation", International Conference on Computing, Control and Industrial Engineering, pp. 16–19, 2010.

[56] H. Duan, G. Ma, S. Liu, "Experimental study of the adjustable parameters in basic ant colony optimization algorithm", IEEE Congress on Evolutionary Computation, pp. 149–156, 2007.

[57] G. Wang, W. Gong, R. Kastner, "System level partitioning for programmable platforms using the ant colony optimization", International Workshop on Logic and Synthesis, 2004.

[58] J.H. Zhao, Z. Liu, M.T. Dao, "Reliability optimization using multi objective ant colony system approaches", Reliability Engineering and System Safety, Vol. 92, pp. 109–120, 2007.

[59] Y. Zhang, B. Feng, S. Ma, "Text clustering based on fusion of ant colony and genetic algorithms", Frontiers of Electrical and Electronic Engineering, Vol. 4, pp. 15–19, 2009.

[60] G. Taguchi, E. Elsayed, T. Hsiang, Quality Engineering in Production Systems, New York, McGraw-Hill, 1989.

[61] E. Canıyılmaz, F. Kutay, "An alternative approach to analysis of variance in Taguchi method", Journal of the Faculty of Engineering and Architecture of Gazi University, Vol. 18, pp. 51–63, 2003.

[62] S. Sağıroglu, N. Özkaya, "An intelligent face features generation system from fingerprints", Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 17, pp. 183–203, 2009.

[63] C. Göloğlu, N. Sakarya, "The effects of cutter path strategies on surface roughness of pocket milling of 1.2738 steel based on Taguchi method", Journal of Materials Processing Technology, Vol. 206, pp. 7–15, 2008.

[64] Minitab Inc. Minitab Statistical Analysis Software.

[65] G. Taguchi, S. Chowdhury, Y. Wu, Taguchi's Quality Engineering Handbook, New Jersey, Wiley-Interscience, 2004.

[66] Ö. Yeniay, "A comparison of the performances between a genetic algorithm and the Taguchi method over artificial problems", Turkish Journal of Engineering & Environmental Sciences, Vol. 25, pp. 561–568, 2001.

[67] C. Göloğlu, C. Mızrak, "Customer driven product determination with fuzzy logic and Taguchi approaches", Journal of the Faculty of Engineering and Architecture of Gazi University, Vol. 25, pp. 9–19, 2010.

[68] E. Bonabeau, M. Dorigo, G. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems, Oxford, Oxford University Press, 1999.

[69] K. Danisman, I. Dalkiran, F.V. Celebi, "Design of a high precision temperature measurement system", Lecture Notes in Computer Science, Vol. 3973, pp. 997–1004, 2006.

[70] S. Mazzeo, I. Loiseau, "An ant colony algorithm for the capacitated vehicle routing", Electronic Notes in Discrete Mathematics, Vol. 18, pp. 181–186, 2004.

[71] C.F. Tsai, C.W. Tsai, H.C. Wu, T. Yang, "ACODF: A novel data clustering approach for data mining in large databases", The Journal of Systems and Software, Vol. 73, pp. 133–145, 2004.

[72] P. Korosec, J. Silc, B. Robic, "Solving the mesh-partitioning problem with an ant-colony algorithm", Parallel Computing, Vol. 30, pp. 785–801, 2004.

[73] C.S. Chang, L. Tian, F.S. Wen, "A new approach to fault section estimation in power systems using ant system", Electric Power System Research, Vol. 49, pp. 63–70, 1999.

[74] Y.H. Song, C.S. Chou, T.J. Stonham, "Combined heat and power economic dispatch by improving ant colony search algorithm", Electric Power System Research, Vol. 52, pp. 115–121, 1999.

[75] M.H. Afshar, "Improving the efficiency of ant algorithms using adaptive refinement: application to storm water network design", Advances in Water Resources, Vol. 29, pp. 1371–1382, 2006.

[76] M. Randall, A. Lewis, "A parallel implementation of ant colony optimization", Journal of Parallel and Distributed Computing, Vol. 62, pp. 1421–1432, 2002.

[77] S.C. Chu, J.F. Roddick, J.S. Pan, "Ant colony system with communication strategies", Information Sciences, Vol. 167, pp. 63–76, 2004.

[78] S.S. Weng, Y.H Liu, "Mining time series data for segmentation by using ant colony optimization", European Journal of Operational Research, Vol. 173, pp. 921–937, 2006.

[79] K.F. Doerner, W.J. Gutjahr, R.F. Strauss Hartl, C. Stummer, "Pareto ant colony optimization with ILP processing in multi objective project portfolio selection", European Journal of Operational Research, Vol. 171, pp. 830–841, 2006.

[80] K.C. Ying, C.J. Liao, "An ant colony system for permutation flow-shop sequencing", Computers & Operations Research, Vol. 31, pp. 791–801, 2004.

[81] C. Rajendran, H. Ziegler, "Two ant-colony algorithms for minimizing total flowtime ant in permutation flow shops", Computers & Industrial Engineering, Vol. 48, pp. 789–797, 2005.

[82] S.J. Shyu, B.M.T. Lin, P.Y Yin, "Application of ant colony optimization for no-wait flow shop scheduling problem to minimize the total completion time", Computers & Industrial Engineering, Vol. 47, pp. 181–193, 2004.

[83] L.M. De Campos, J.M. Fernandez-Luna, J.A. Gamez, J.M. Puerta, "Ant colony optimization for learning Bayesian network", International Journal of Approximate Reasoning, Vol. 31, pp. 291–311, 2002.

[84] A. Lim, J. Lin, B. Rodrigues, F. Xiao, "Ant colony optimization with hill climbing for the bandwidth minimization problem", Applied Soft Computing, Vol. 6, pp. 180–188, 2006.

[85] R. Gao, J. Wu, "An improved ant colony optimization for routing in cognitive radio network", International Journal of Advancements in Computing Technology, Vol. 3, pp. 73–81, 2011.

[86] D. Merkle, M. Middendorf, H. Schmeck, "Ant colony optimization for resource-constrained project scheduling", IEEE Transactions on Evolutionary Computation, Vol. 6, pp. 333–346, 2002.

[87] X. Benlian, W. Zhiquan, "A multi-objective-ACO-based data association method for bearings-only", Communications in Nonlinear Science and Numerical Simulation, Vol. 12, pp. 1360–1369, 2007.

[88] H. Duan, X. Yu, "Hybrid ant colony optimization using memetic algorithm for TSP", Proceedings of the IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning, pp. 92–95, 2007.

[89] C.F. Tsai, C.W. Tsai, C.C. Tseng, "A new hybrid heuristic approach for solving traveling salesman problem". Information Science, Vol. 166, pp. 67–81, 2004.

[90] J.J. Ortiz, I. Requena, "Azcatl-CRP: An ant colony-based system for searching full power control rod patterns in BWRs", Annals of Nuclear Energy, Vol. 33, pp. 30–36, 2006.

[91] H.R. Maier, A.C. Zecchin, L. Radbone, P. Goonan, "Optimizing the mutual information of ecological data clustering using evolutionary algorithms", Mathematical and Computer Modelling, Vol. 44, pp. 439–450, 2006.

[92] R.J. Kuo, H.S Wang, T.L Hu, S.H. Chou, "Applications of ant k-means on clustering analysis", Computers & Mathematics with Applications, Vol. 50, pp. 1709–1724, 2005.

[93] J.A. Gamez, J.M. Puerta, "Searching for the best elimination sequence in Bayesian networks by using ant colony optimization", Pattern Recognition Letters, Vol. 23, pp. 261–277, 2002.

[94] R.J. Kuo, Y.P. Kuo, K.Y. Chen, "Developing a diagnostic system through integration of fuzzy case-based reasoning and fuzzy ant colony system", Expert Systems with Applications, Vol. 28, pp. 783–797, 2005.

[95] P.Y. Yin, "Ant colony search algorithm for optimal polygonal approximation of plane curves", Pattern Recognition, Vol. 36, pp. 1783–1797, 2003.

[96] T. Stützle, H.H. Hoos, "MAX–MIN ant system", Future Generation Computer System, Vol. 16, pp. 889–914, 2000.