ELECTRONICS AND ELECTRICAL ENGINEERING
ISSN 1392 – 1215 2011. No. 1(107)
ELEKTRONIKA IR ELEKTROTECHNIKA

T 120 SYSTEM ENGINEERING, COMPUTER TECHNOLOGY
SISTEMŲ INŽINERIJA, KOMPIUTERINĖS TECHNOLOGIJOS

Heuristic Methods in Vehicle Routing Systems

Ö. Kuşcu, E. U. Küçüksille
Department of Information Technology, Süleyman Demirel University,
Department of Computer Engineering, Faculty of Engineering and Architecture, Süleyman Demirel University,
32260 Çünür, Isparta/Türkiye, phone: +90 246 211 11 26, e-mails: okuscu@sdu.edu.tr, ecir@sdu.edu.tr

Introduction

The most popular algorithms are genetic algorithms and ant colony algorithm are used to solve NP-Hard Problems [1–4].

Vehicle Routing Problem is the task of finding the optimal set of routes for a fleet of vehicles with a minimum cost. A vehicle set of routes starts at on point and ends again at the same point. Methods of solution and difficulty of solutions vary with the placement of additional constraints on the Vehicle Routing Problem.

In this study, an application has been designed where the predetermined routes of a shuttle bus used in short-distance transportation will be entered into the system and this shuttle bus will be routed in the most effective manner, minimizing the cost. With this application, the optimum route will be devised, the route devised will be shown on the map, with the directions for the user to follow visually. In addition, it will allow for transferring rotation information to the users via email or different means if desired.

The Vehicle Routing Problem (VRP) was first introduced to the literature by Dantzig and Ramser in 1959. In this study, the authors focused on the problem of gasoline delivery to gas stations and designed the first mathematical programming model to solve the problem. Then in 1964, Clarke and Wright offered a heuristic solution for the problem, which led to an increased interest in VRP in the literature. VRP is one of the optimization models for which the highest number of methods were developed so far [5]. The purpose of a typical Vehicle Routing Problem is to design a minimum cost set of routes for a fleet of vehicles. Each route starts from a depot and ends again at the same depot following service to a set of customers whose demands are known. Each customer should be assigned to a single vehicle and the total demands of customers assigned to the vehicle should not exceed that vehicle's capacity. Assuming that the vehicle capacities are larger than the largest of all customer demands, it is ensured that one or more customers are assigned for each vehicle [6].

The literature contains many studies aimed at vehicle routing problems and solution of these problems. VRP itself is an NP-Hard problem and the researchers' experiences suggested that a Vehicle Routing Problem with Time Windows (VRPTW) was basically a harder problem than VRP [7]. Savelsbergh showed that even finding the feasible solutions for VRPTW at fixed fleet sizes had a NP-Hard structure [7]. So, many solution approaches are based on heuristics. Solomon designed a set of heuristics for VRPTW, namely extended savings, time-oriented nearest neighbor, time-oriented insertion and time-oriented sweep heuristics. The results showed that the sequential time-space insertion heuristic has been successful in designing cost-effective routes. Besides, a set of comparison problems has been created to compare the various procedures designed. [8]. Baker and Schaffer conducted a computational study on route development with procedures applied on heuristically created start solutions [9].

The results demonstrate that such developments required high computational efforts. Van Landeghem described a two-criterion heuristics for VRPTW, stating that the interaction between spatial and temporal issues made it much harder to gain insight into the underlying dynamics of VRPTW and hence more difficult to design good heuristics [10]. Kolen et al. suggested an optimal algorithm based on branch-and-bound strategies for VRPTW. However, the largest problem solved involved 4 vehicles serving 14 customers with narrow time windows. VRPTW was described as a set partitioning problem and column generation approaches were suggested for various types of the problem.

Desrochers et al. devised a new optimization algorithm employing branch-and-bound strategies for VRPTW's set-covering-based type of formulation. LP relaxation of the model was solved using the column generation method and the problem of shortest path with time windows and capacity constraints was solved to add feasible columns [11]. Dumas et al. employed a similar approach to solve the time-window pickup and delivery problem [12].

Fisher (et al.) proposed two methods for the optimal solution of VRPTW. The first is based on k-tree relaxation with time windows. And the other solves the problem of semi-assignment, followed by the shortest path with time windows capacity constraints [13]. Optimal solutions were obtained by comparison problems in both methods. Sexton and Choi focused on a single-vehicle pickup and delivery routing problem. In their study, each cargo has its respective origin and destination [14]. They tried to minimize a linear combination of total vehicle operating time and total customer penalty due to missing any of the time windows by applying Bender's decomposition procedure. Min handled the single-vehicle library delivery system with a single depot where the daily activities started and ended [15]. He included specific time window preferences of customers into the modelling process, proposing a target approach to solve the problem. The goals are the minimization of total travelling time and any deviations from time window preferences.

Koskosidis et al. developed a heuristic approach for Vehicle Routing Problem with Flexible Time Windows (VRPFTW) [16]. In their heuristic approach, a series of problems consisting of generalized assignment problem to assign customers to vehicles and a flexible time interval consisting of routing and scheduling were described as the time-constrained travelling salesman problem, creating a loop structure between these two approaches. Balakrishnan proposed three heuristics based the nearest neighbor, Clarke-Wright savings and spatio-temporal rules for VRPFTW with a homogeneous fleet of vehicles [17]. The cost of violating the time windows was addressed as a linear function of deviations from time windows. Taillard et al. proposed a tabu search heuristic minimizing the total penalty scores for failure to offer the services within time windows for the VRPFTW and the total travel distance [18]. Fagerholt addressed the multi-ship pickup and delivery problem with flexible time window, also including inconvenience costs imposed for servicing customers except their time windows [19]. The set partitioning approach with variable feasible schedules was used to solve the problem with the objective of minimizing the sum of transportation costs and inconvenience costs.

Heuristic Algorithms

An algorithm is an effective method for solving a problem expressed as a finite sequence of instructions. From another perspective, it can be said that an algorithm is the assembly of the algorithmic expression of a certain task with data structures [20].

Heuristic algorithms are criteria or computer methods defined to realize any purpose or decide on the effective ones of various alternative actions to achieve the target. Such algorithms have convergence properties, cannot guarantee a final solution but can only guarantee a solution near the final solution. Convergence by these algorithms to the optimum solution in the space of solutions is named as improvable algorithms. The reasons for requiring heuristic algorithms are as follows [21]:

- The optimization problem should have a structure where the operation of finding the exact solution cannot be described;
- Regarding understandableness, heuristic algorithms may be much more easier for the decision maker;
- Heuristic algorithms may be used for learning purposes and as part of an operation to find the final solution;
- In descriptions made using mathematical formula, generally the hardest parts of the real-time problems (which goals and constraints should be used, which alternatives should be tested, how the problem data should be collected) are ignored. Erroneous data used at the stage of determination of model parameters might cause problems bigger than the suboptimal solution that the approach can produce.

Cordeau et al. emphasized that a heuristic algorithm yielding good results for VRP should have four basic properties. [22].

1. *Accuracy.* The concept of accuracy in the literature is known as the proportion of a value produced by an algorithm to the optimum solution of a problem.

2. *Speed.* Highly increased speeds of computation in today's world made the realization of the solutions of combinatorial problems like VRP through iterative processes in a short time possible. Besides, the authors focused on the idea that quick solutions could be possible by heuristic methods for VRP should parallel programming be used in an appropriate manner.

3. *Simplicity.* One of the reasons why a majority of heuristic methods cannot be applied in real-time systems is the difficulty of its comprehension and coding. One of the characteristics needed in order that a heuristic algorithm can be highly applicable is the fact that the quality of solutions produced is not too much dependent upon the conditions at the start. The method being simple increases the applicability of the algorithm, and will make it possible to produce solution for VRP in a quick manner as will allow productive use of computer resources.

4. *Flexibility.* A good VRP heuristic algorithm should have the flexibility of application to real systems, integrated with various constraints. Despite the general availability of capacity and distance constrained problems in the VRP literature, it is often not mentioned as to what sort of changes should be made in the algorithm when a new constraint is inserted. Making such changes in the model generally results in a large-scale drop in the algorithm performance.

And metaheuristic algorithms can be supposed as a form of the classical heuristic algorithms, developed by inspiration from the nature [23].

Metaheuristic algorithms are general optimization techniques developed for combinatorial problems. The most important feature of these methods is that they have the flexibility of application not only for a certain type of problem, but for all combinatorial problems. Hence, many success stories of these techniques are encountered in the literature. The first thing to do in order that a metaheuristic algorithm can yield good results is the fine adaptation of the basic concepts of the method to the problem [24].

Blum and Roli made the following generalizations about metaheuristic methods [25]:

- Metaheuristic methods are strategies guiding during the search process.

- The general purpose is to effectively search the search space and obtain optimum or near-optimum results.
- Metaheuristic methods are common from local search techniques to complex learning procedures.
- Metaheuristic algorithms are approximate methods and are not generally deterministic.
- They contain mechanisms that will prevent getting stuck in local optimum points in the search space.
- Metaheuristic techniques are not problem-specific methods. They can generally be applied for all combinatorial problems.
- Today, memory-based processes are available to guide during the search in highly developed metaheuristic algorithms.

The Developed Application

In this study, vehicle routing operation is done by the help of both GA and ACO. Are there any benefits of making explanations about genetic algorithm and ant colony optimization as a subtitle? If an explanation is to be made, what should its extent be?

The application is accessible through the internet. Members may login to the system at any time over the internet and may make any desired computations by marking their stops on the system.

All users except the members are prevented from using the system. After login, the system offers to the user a map to make sure they can choose the points of origin and destination. The mapping system used in the system (Google Maps)is the free mapping system owned by Google.. The members can mark the desired regions using the page in Fig. 4.

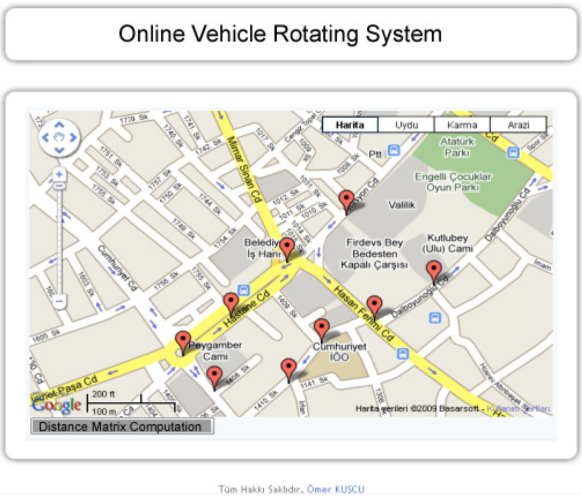

Fig. 1. Location marking on the system

The maps allow more than one marking. Following the markings, distances to other points are computed from the mapping system for each point. Taking these distances, a matrix of distances is created. The "Distance Matrix Computation" button seen in Fig. 4. 3 is used in order that the matrix of distances can be created.

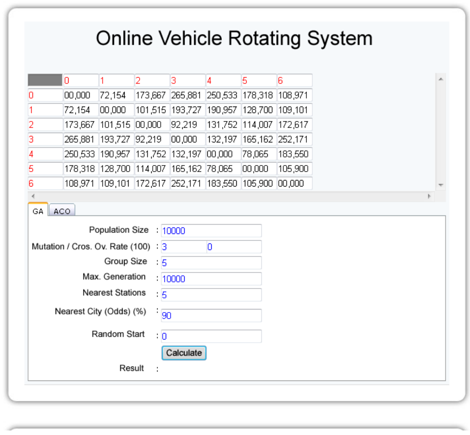

Fig. 2. Creation of the matrix of distances

A matrix of distances is created in Fig. 2. Each of the numbers in the horizontal and vertical axes represents a point and the distances of each point with one another is given in meters (m). For example, the distance between position no 2 and position no 5 is 114,007 meters.

After the matrix of distances created, the users can make the routing operation by two different algorithms. After the entry of required parameters, routing operation can be computed using the Genetic Algorithm (GA) or Ant Colony Optimization (ACO). In Fig. 3, routing result is obtained using the Ant Colony Optimization.

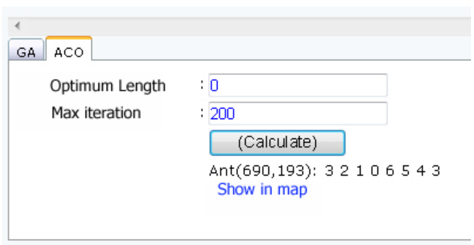

Fig. 3. Making the routing operation using Ant Colony Optimization

Optimum distance is given as 0 in Fig. 3. Optimum distance can be change if desired and terminates when the iterations catch the optimum distance. The operation can be repeated many times by changing the parameters. After completion of the computations, the routing operation is seen on the map using the "Haritada Gör (See on Map)" link in Fig. 3.

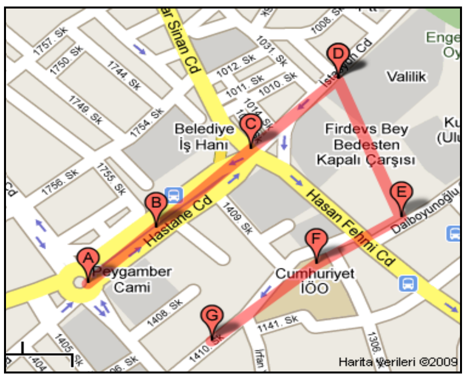

Fig. 4. Route drawing on the map

Fig. 4 shows the route drawing operation on the map. Here, a letter is assigned to each route to indicate its order. Starting at the point represented by A, the route terminates at point G and the routing operation is completed upon return to the start point after point G.

Experimental Result

In this section, an example was first applied the any colony optimization and then Genetic Algorithm and the results were compared. There are 16 stops in the problem and the minimum distance between these stops is found *432,23* using both algorithms. Route drawings are made according to 10, 50, 100 and finally 1000 steps (iteration) values. Details about the problem are shown in Table 1.

Table 1. Information about the example problem

Number of Stops	**16**
Maximum Number of Iterations	**1000**
Distance of Shortest Tour (meters)	**432,23**
Configurations of Computer Used	**P-4 1.9 Ghz Processor, 1 GB Ram**

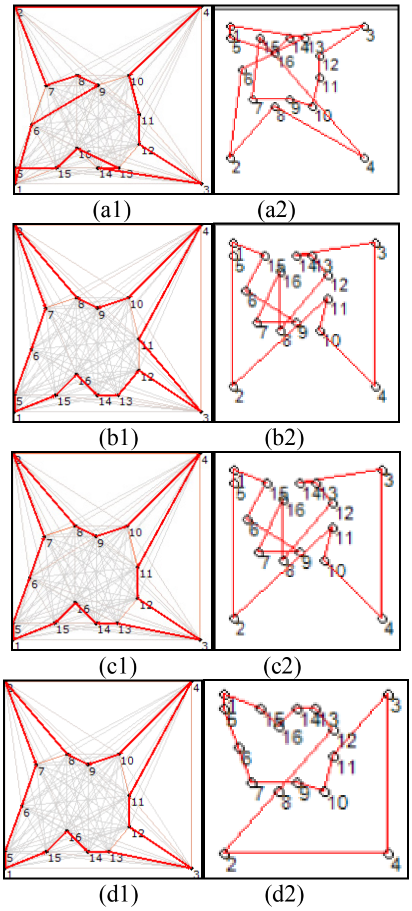

Fig. 5. Graphs showing routing drawings by numbers of iteration. Using ACO for a1 – 10 iteration; using GA for a2 – 10 iteration; using ACO for b1 – 50 iteration; using GA for b2 – 50 iteration; using ACO for c1 – 100 iteration; using GA for c2 – 100 iteration; using ACO for d1 – 1000 iteration; using GA for d2 – 1000 iteration

Fig. 5 shows a comparison of algorithms based on certain iteration values and the graphical expression of the best route found. The graphs expressed with a1, b1, c1, d1 are obtained using ACO, and the graphs expressed with a2, b2, c2, d2 are obtained using GA.

In order to be able to compare the performance of algorithms based on the number of steps, first the time values as compared to the numbers of iteration should be compared. Fig. 6 shows time-dependent iteration numbers between GA and ACO.

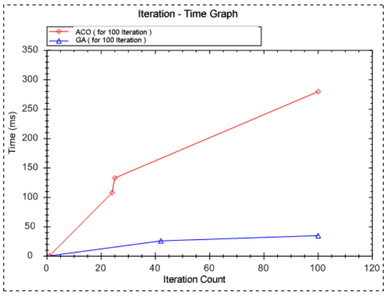

Fig. 6. Graph of time-dependent iteration numbers based on 100 iteration values for Ant Colony Optimization and Genetic Algorithm

In order to be able to solve the example problem in Table 1, although the algorithm developed using the ant colony optimization method can reach the result in fewer steps, it completes the operation in longer time compared with the algorithm developed by Genetic Algorithm. According to the experiment performed on a computer configuration of Pentium-IV 1.9 Ghz Processor and 1 GB Memory, as seen in Fig. 4.15, while the tour distance of *432,23* meters can be reached in *232* milliseconds by GA, the same distance is reached in *335* milliseconds by ACO. The *432,23* meters of distance here is the minimum possible distance. Both algorithms could find the minimum value after specified times.

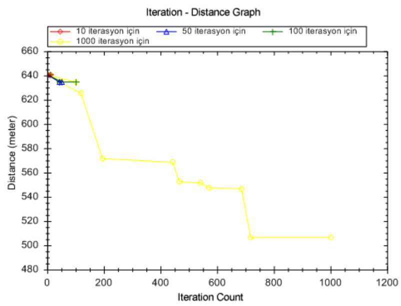

Fig. 7. Graph showing the time-dependent change of the distance difference for 10, 50, 100 and 1000 iteration values using Genetic Algortihm

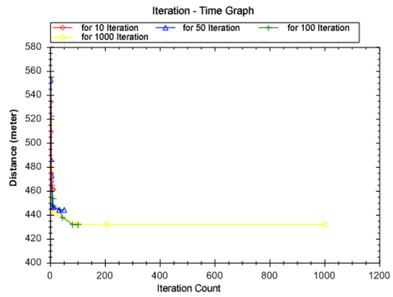

Fig. 8. Graph showing the time-dependent change of the distance difference for 10, 50, 100 and 1000 iteration values using Ant Colony Optimization

Conclusions

In this study, predetermined routes of a shuttle bus used in short-distance transportation are entered into the system and an internet-based software was developed to calculate the routing result using Genetic Algorithm or Ant Colony Optimization methods in order that this shuttle bus can be routed most effectively and with minimized costs.

The system can be used in long-distance transportation as well as short-distance transportation. Many applications in the literature compare only the performances of the developed algorithms. However, this study develops a software that enables making of internet accessible instant computations both to compare the performances and to make sure they are applicable in real life situations.

With the developed internet-based software, the results are computed using two different algorithms, and their differences were observed. Computations were made using Genetic Algorithm and Ant Colony Optimization to obtain the shortest route. Considering the results of the trials made using the developed application, it was observed that Genetic Algorithm achieved quicker results than the Ant Colony Optimization in this study.

The map used in the application allows the computation of the bird's eye distance between any two points, but does not allow the computation of the real distance. And the bird's eye distance taken has an error tolerance between 3% and 5%.

In future studies, it will be possible to achieve the desired purpose more clearly using a mapping system that can take the real distance between any two points more accurately and make the routing operation on street basis.

References

1. **Toemeh R., Arumugan S.** Breaking Transposition Cipher with Genetic Algorithm // Electronics and Electrical Engineering. – Kaunas: Technologija, 2007. – No. 7(79). – P. 75–78.
2. **Goranin N., Čenys A.** Malware Propagation Modeling by the Means of Genetic Algorithms // Electronics and Electrical Engineering. – Kaunas: Technologija, 2008. – No. 6(86). – P. 23–26.
3. **Laptik R., Navakauskas D.** Application of Ant Colony Optimization for Image Segmentation // Electronics and Electrical Engineering. – Kaunas: Technologija, 2007. – No. 8(80). – P. 13–18.
4. **Nirmala Devi M., Mohankumar N., Arumugam S.** Modeling and Analysis of Neuro–Genetic Hybrid System on FPGA // Electronics and Electrical Engineering. – Kaunas: Technologija, 2009. – No. 8(96). – P. 69–74.
5. **Toth P., Vigo D.** The Vehicle Routing Problem. – Society for Industrial and Applied Mathematics, 2002. – 224 p.
6. **Balakrishnan N.** Simple Heuristics for the Vehicle Routing Problem with Soft Time Windows // Journal of the Operational Research Society, 1993. – No. 44(3). – P. 279–287.
7. **Savelsbergh M.** Local Search in Routing with Time Windows // Annals Of Operations Research, 1985. – No. 4(6). – P. 285–305
8. **Solomon M**. Algorithms for the Vehicle Routing and Scheduling with Time Window Constraints // Operations Research, 1987. – No. 35. – P. 254–265.
9. **Baker E., Schaffer J.** Solution Improvement Heuristics for the Vehicle Routing and Scheduling Problem with Time Window Constraints // American journal of Mathematical and Management Sciences, 1986. – No. 6. – P. 261–300.
10. **Van Landeghem H. R. G.** Abi–criteria Heuristic for the Vehicle Routing Problem with Time Windows // European Journal of Operational Research, 1988. –No. 36. – P. 217–226.
11. **Desrochers M., Desrosiers J., Solomon M.** A New Optimization Algorithm For The Vehicle Routing Problem With Time Windows", Operations Research, 1992. –No. 40, –P. 342–354.
12. **Dumas, Y., Desrosiers J., Soumis, F.** "The Pickup and Delivery Problem with Time Windows // European Journal of Operational Research, 1991. – No. 54. – P. 7–22.
13. **Fisher M. L., Jornsten K. O., Madsen O. B. G.** Vehicle routing with time Windows – Preliminary Results–Research Report 4 // IMSOR. – Technical University of Denmark, Denmark, 1991. – No. 33. – P. 38–47.
14. **Sexton T. R., Choi Y. M.** Pickup and Delivery of Partial Loads with Soft Time Windows // American Journal of Mathematical and Management Sciences, 1986. – No. 6(3). – P. 369–398.
15. **Min H.** A Multiobjective Vehicle Routing Problem with Soft Time Windows: The Case of a Public Library Distribution System // Socio–Economic Planning Science, 1991. – No. 25(3). – P. 179–188
16. **Koskosidis Y. A., Powell W. B., Solomon M. M.** An Optimization–based Heuristic for Vehicle Routing and Scheduling with Soft Time Window Constraints // Transportation Science, 1992. – No. 26(2). – P. 69–85.
17. **Balakrishnan N.** Simple Heuristics for the Vehicle Routing Problem with Soft Time Windows // Journal of the Operational Research Society, 1993. – No. 44(3). – P. 279–287.
18. **Taillard E., Badezu P., Gendreau M., Guertin F., Potvin J. Y.** A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows // Transportation Science, 1997. – No. 31(2). – P. 170–186.
19. **Fagerholt K.** Ship Scheduling with Soft Time Windows: An Optimisation Based Approach // European Journal of Operational Research, 2001. – No. 131. – P. 559–571.
20. **Çölkesen, R.** Veri Yapıları ve Algoritmalar, –Papatya Yayıncılık, 2002. –392 p.
21. **Karaboğa D.** Yapay Zekâ Optimizasyon Algoritmaları. – Atlas Yayın Dağıtım, Cağaloğlu, 2004. – 210 p.
22. **Cordeau J. F., Gendreau M., Laporte G., Potvin J. Y., Samet F.** A Guide To Vehicle Routing Heuristics // Journal of the Operational Research Society, 2002. – No. 53.– P. 512–522.
23. **Laporte G., Gendreau M., Potvin J. Y., Samet F.** Classical And Modern Heuristics For The Vehicle Routing Problem // International Transactions in Operation Research, 2000. – No. 7. – P. 285–300.
24. **Hertz A., Widmer M.** Guidelines For The Use Of Meta–Heuristics In Combinatorial Optimization // European Journal of Operation Research, 2004. – No. 151. – P. 247–252.
25. **Blum, C., Roli, A.** Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison // ACM Computing Surveys, 2003. –No. 35(3). – P. 268–308.

Received 2010 05 21

Ö. Kuşcu, E. U. Küçüksille. Heuristic Methods in Vehicle Routing Systems // Electronics and Electrical Engineering. – Kaunas: Technologija, 2011. – No. 1(107). – P. 65–70.

In this study used in urban transport system with a service by passing the pre-determined route, and these services cost the most effective way to minimize an application for a route has been developed. Is close to the best route for vehicle detection with Genetic Algorithms or Ant Colony Optimization methods can be performed. Determination of the route will be shown on a map, driving directions, the user will be done visually. In addition, if requested via mail or other media rotation information can be transported to users. Internet users access the application and the position you want to go through the map by checking the system to process and route calculation and drawing during can see. The calculation process can be done with Genetic Algorithms or Ant Colony Optimization. Ill. 8, bibl. 25, tabl. 1 (in English; abstracts in English and Lithuanian).

Ö. Kuşcu, E. U. Küçüksille. Euristinių metodų taikymas transporto maršrutizavimo sistemose // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2011. – Nr. 1(107). – P. 65–70.

Aprašyta miesto transporto sistema su iš anksto numatytais maršrutais. Transporto valdymo sistema papildyta funkcijomis, gebančiomis apskaičiuoti trumpiausią atstumą. Trumpiausiam atstumui nustatyti gali būti taikomi genetiniai algoritmai ar skruzdžių kolonijos optimizavimo metodai. Nustatytas maršrutas gali būti pateiktas žemėlapyje su vairavimo nuorodomis. Taip pat yra galimybė pateikti informaciją vartotojams tiesiogiai, jei užklausa buvo atlikta elektroniniu paštu ar pan. Maršrutas gali būti apskaičiuotas ir nurodytas bei parodytas tiesiogiai jungiantis prie sistemos internetu. Il. 8, bibl. 25, lent. 1 (anglų kalba; santraukos anglų ir lietuvių k.).