Check for updates

INTECH
open science | open minds

OPEN ACCESS
ARTICLE

International Journal of Advanced Robotic Systems

A Smooth Tour Construction Approach for a Mobile Robot with Kinematic Constraints

Regular Paper

Ahmet Yazici[1,*]

1 Computer Engineering Department, Eskisehir Osmangazi University, Batimeselik, Eskisehir, Turkey
* Corresponding author E-mail: ayazici@ogu.edu.tr

Received 10 Dec 2012; Accepted 27 Aug 2013

DOI: 10.5772/56942

© 2013 Yazici; licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract Mobile robots are increasingly used for service-like applications in which the service points are known and the mobile robot starts from a starting location, visits all the service points requested and returns to the starting location. The tour construction problem in these applications can be treated as a Travelling Salesman Problem (TSP). Classical tour construction algorithms that are proposed for the TSP find tours do not consider robot kinematic constraints. These tours may have sharp turns at some service points. When a mobile robot follows such a tour, it stops, turns and speeds up again. Therefore, the robots waste a considerable amount of power and time. In these cases, tour smoothing can be used to overcome this problem. However, smoothing an existing tour may result in unnecessarily long tours. In this study, a Smooth Tour Construction (STC) approach is proposed for mobile robots with kinematic constraints. The STC approach considers tour construction and tour smoothing concurrently. The logic behind the tour construction part of the approach is based on the Savings Algorithm (SA). The tour smoothing is based on Dubins' arc-line approach. Experiments are conducted for P3-DX robots in a laboratory environment. Comparisons are also drawn with various tour smoothing algorithms in simulation environments to demonstrate the effectiveness of the proposed STC approach.

Keywords Mobile Robot, Kinematic Constraints, Smooth Tour Construction, TSP

1. Introduction

Nonholonomic mobile robots are increasingly used in environments such as offices, hospitals, houses and outdoors ([1], [2]) for service-type applications. In these applications, the service points are known, and the mobile robot starts from a starting location, visits all the service asking points and returns to the starting location. This problem is a tour construction problem and can be treated as a Travelling Salesman Problem (TSP) [3] that finds a tour for a salesman who starts from a home location, visits a prescribed set of cities and returns to the original location via the minimum possible distance [4].

In the literature, there are studies which apply TSP to mobile robot path planning problems ([5], [6]). In their

work, Yu et al. [5] described how to apply a genetic algorithm to find a globally sub-optimal path for a robot group. They formulized the path planning problem for the robot group as a multiple Travelling Salesman Problem that employs either total-path-shortest or longest-path-shortest as the evaluating criterion. In [6], a heuristic-based TSP approach is applied to the tour construction problem of a mobile robot in a dynamic environment. In this study, a method that is a combination of the Savings and Dijkstra algorithms is proposed to construct tours in real time. It was shown through simulations and experiments that tours are constructed in real time. The proposed method also constructs a tour in cases of changes in the network structure. However, the works mentioned above do not consider mobile robot kinematic constraints during tour construction. Because the resulting tour of a classical TSP has sharp turns, when the mobile robot follows such a tour, it must slow down, orient itself and then accelerate. These turns take a significant amount of time. Therefore, smooth tours are preferred for real applications. Smooth tour construction which considers kinematic constraints is a difficult problem.

A similar problem is widely studied ([8], [9], [10]) for classical start-goal oriented path planning problems, which differ from the TSP. In [8], a smooth path planning method for a car-like vehicle is presented. In [9], a smooth motion planning for differential driving mobile robots is presented. In [10], a smooth path planning approach is proposed and compared with path smoothing for a differential driving mobile robot with kinematic constraints. As mentioned before, start-goal oriented path planning differs from the tour construction problem considered in this paper.

There are a few approaches that consider mobile robot kinematic constraints for a Travelling Salesman Problem. In [11], the authors defined tour construction problems for mobile robots with kinematic constraints. They proposed an algorithm to smooth a constructed tour for a mobile robot. The tour smoothing is achieved using the circular arcs-lines [7] approach based alternating algorithm. Therefore, tour construction and smoothing are decoupled.

In this study, tour construction and smoothing are coupled for service-like problems, considering mobile robot kinematic constraints. Finding the optimal solution of a TSP is time-consuming because of the NP-Hard nature of the TSP. For this reason, the heuristic approach used in [6] is extended to take the mobile robot kinematic constraints into consideration.

In the following section, the mobile robot kinematic model, the Travelling Salesman Problem and the tour smoothing are briefly presented. In section 3, the proposed smooth tour construction approach is explained. An application of the proposed method for dynamic environments is also described. Experimental results for P3-DX robots in a laboratory environment and comparisons with various tour smoothing algorithms in simulation environments are described in Section 4. The conclusions are given in the last section.

2. Preliminaries

The problem considered in this paper is composed of two main components: the mobile robot kinematic constraints and the tour construction problem. In the following subsection, the kinematic model of a differential-drive mobile robot used in our laboratory is presented. Next, the basics of the tour construction problem, the Travelling Salesman Problem and the solution approach are given.

2.1 Mobile Robot Kinematic model and path tracking problem

Most Wheeled Mobile Robots (WMRs) have differential drives. The P3-DX mobile robot is an example of this type of robot (Fig. 1 (a)); the P3-DX has two main wheels attached to its motors and a caster wheel placed in the rear. Let (x_R, y_R) be the position of the robot, and θ_R the heading of the robot.

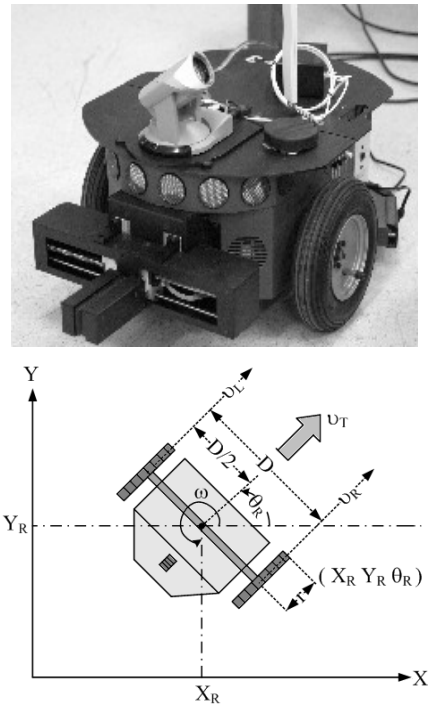


Figure 1. a) Pioneer P3-DX mobile robot **b)** Representation of velocities and robot position

In the compact form, the kinematics of the robot can be modelled by the equation

$$\begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{bmatrix} = \begin{bmatrix} \cos\theta_R & 0 \\ \sin\theta_R & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (1)$$

where, in general, v and ω are the linear and the angular velocity of the mobile robot, respectively. These velocities are related to the wheel velocities (v_R, v_L) and can be expressed as

$$v = r\frac{(v_R + v_L)}{2}$$
$$\omega = \frac{r}{D}(v_R - v_L) \quad (2)$$

where D is the width of the mobile robot from the centre of the left wheel to the centre of the right wheel and r is the wheel radius. Based on the kinematics model, it is obvious that the location and the orientation of the WMR body changes according to a differential drive by means of the voltages applied to the right and left wheels. If both the left and right wheels of the WMR move at the same speed ($\upsilon_R = \upsilon_L > 0$), the robot follows a straight line. If $\upsilon_R \neq \upsilon_L$, then the robot turns in a clockwise or counter-clockwise direction based on the velocities of the two wheels. In this way, this rotation generates a curvature, the centre of which is at the ICC (Instantaneous Centre for Curvature), as shown in Fig. 2.

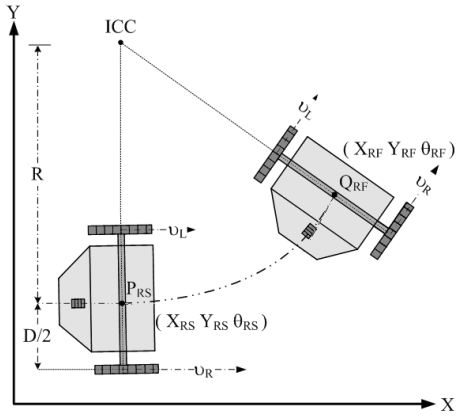


Figure 2. A curvature trajectory

In Fig. 2, R is the radius of a path followed by the mobile robot. The relationship between this radius and the two wheel velocities is given as follows:

$$R = \frac{D}{2}\left(\frac{\upsilon_R + \upsilon_L}{\upsilon_R - \upsilon_L}\right) \quad (3)$$

Note that under the assumption $\upsilon_R, \upsilon_L \geq 0$, $D/2$ is also the minimum turning radius $R_{\min}$ of the mobile robot.

If a tour and its first derivative are continuous (e.g., combinations of lines and arcs of circles with a minimum turning radius that are tangent at some defined points), then a car-like or differential-drive mobile robot can track such a tour with zero tracking error [12]. However, if the tour is continuous without a continuous derivative (e.g., just a combination of straight lines), only a differential-drive mobile robot can track the tour with zero tracking error. However, in the case of such a tour, a differential mobile robot should stop, reorient itself and then accelerate at line-intersection points. During the application, these turns take a significant amount of time [13]. Therefore, for differential-drive mobile robots, a continuous first derivative is preferable for tracking. The differential-drive mobile robot can track such a smooth tour using the following kinematic controller [14]:

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} v_{ref}\cos\theta_e + K_x x_e \\ w_{ref} + v_{ref}(K_y y_e + K_\theta \sin\theta_e) \end{bmatrix} \quad (4)$$

where K_x, K_y are positive controller parameters and v_{ref}, w_{ref} are the linear and angular reference velocities, respectively. The linear reference velocity can be defined by the user, and the reference angular velocity can be calculated using the reference path curvature. The right and left wheel velocities can be found using the results of equation (4) and equation (2).

2.2 The Travelling Salesman Problem and tour smoothing

In the tour construction problems, the robot is required to visit a set of points and return to the starting location. This can be considered as a Travelling Salesman Problem (TSP). The TSP involves finding a route for a salesman who starts from a home location, visits a prescribed set of cities and returns to the original location in such a way that the total distance travelled is minimized and each city is visited exactly once [4]. The TSP is one of the most widely studied combinatorial optimization problems [15]. It requires a network modelling of the problem. A network can be represented as a graph *G=(N, A)*, where *N* is a set of *n* nodes (vertices) and *A* is a set of arcs (edges). Nodes describe places to visit, and arcs describe links between nodes in the network. The distance from node *i* to node *j* is denoted by d_{ij}. The *nxn* matrix is composed of distances between all the pairs of nodes. It is called the distance matrix where n is the number of nodes. This matrix is symmetric if $d_{ij}=d_{ji}$, otherwise it is asymmetric. The distance matrix is complete if each node is connected to the others, i.e. $d_{ij}>0$ i=1,..,n, j=1,…,n otherwise it is sparse. The number of possible tours exponentially increases by *n*; therefore, TSP is an NP-hard problem. Due to the NP-hard nature of the TSP, finding the optimal solution is either impossible or takes a very long time. In problems in which the real-time solution of the problem is necessary, heuristics, rather than exact-solution methods, are preferred.

The Savings Algorithm (SA) [16] is a typical tour construction method. It is flexible enough to handle a wide range of practical constraints [17]. Although the Savings Algorithm does not guarantee the optimal solution, it constructs a tour very fast. It is shown that, on average, the results of the Savings Algorithm are 6.71% away from the optimal solution [18]. It starts with a subset of points linked in a cycle and adds the others one by one until the cycle is complete. Therefore, it is a complete algorithm. The complexity of the algorithm is $O(n^3)$ [19]. The aim of the SA is to maximize the cost saved over a previous tour at each step. The SA is flexible because of its applicability to various types of problems. In [6], the SA was used to solve a real-time dynamic tour construction problem for a differential-drive mobile robot. A typical tour, T, constructed using SA for a given set of fully connected nodes is shown in Figure 3.

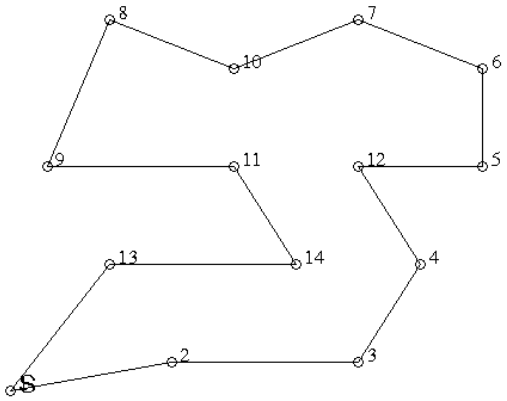


Figure 3. A typical TSP tour

Note that a differential-drive mobile robot can follow the tour in Figure 3 with zero tracking error. However, it will decelerate, stop and accelerate at each node, as in [6]. These turns take a significant amount of time and cause errors in the sensors' perception of the environment. Note that the initial and final orientations of the mobile robot are also not considered in this tour. Assuming that the robot starts with an initial orientation of θ_{init} and finishes the tour with a final orientation θ_{final}, a typical tour of a SA can be smoothed using the following Tour-Smoothing (TS) algorithm , as in Figure 4.a.

Tour-Smoothing (TS) Algorithm:

Inputs of the Algorithm: A planned tour $T = [t_1, \ldots, t_n]$ with node x-y coordinates, Robot initial orientation θ_s, Robot final orientation θ_f, Radius of smoothing arc R_{SA}

Output of the Algorithm: smooth tour ST, smooth tour cost $ST^{R_{SA}}_{cost}$.

Step 1: Set $\theta_1 = \theta_s$ and $\theta_n = \theta_f$.

Step 2: If not defined, set θ_i for $i = 2, \ldots, n-1$.

Step 3: Construct the smooth tour $ST^{R_{SA}} = [(t_1, \theta_1), \ldots, (t_n, \theta_n)]$ using (t_i, θ_i) pairs, R_{SA} and the arc-line approach in [7] and calculate the smooth tour cost $ST^{R_{SA}}_{cost}$.

Note that the approach in [7] finds the minimal length path between two points with described initial and terminal positions and tangents. In step 2, the orientations at each node can be directly assigned by the user. These can also be set to the value of the slope of the line between before-current nodes, current-after nodes or before-after nodes.

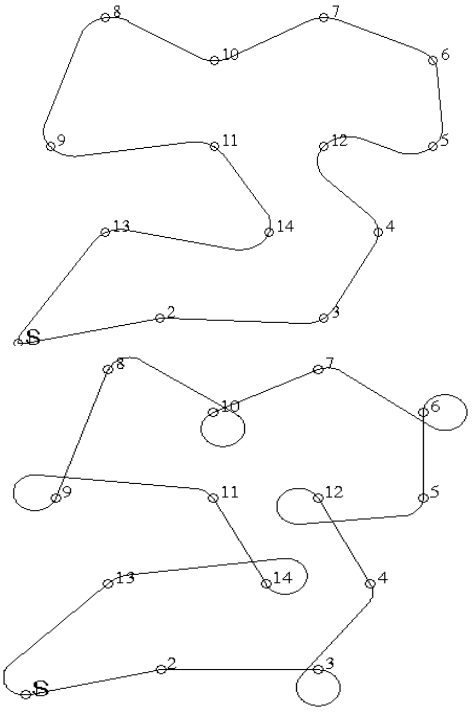


Figure 4. Smoothing of a tour of a SA using a) the TS Algorithm or b) The alternating algorithm [11]

The resulting tour obtained by the SA (Figure 3) is smoothed using the TS algorithm as in Figure 4.a. In figure 4.b., the Alternating Algorithm (AA) [11] is also used to smooth the tour constructed by the SA. In either case, smoothing a constructed tour may result in a longer tour depending on the radius of the smoothing arc. In this study, a coupled tour construction and smoothing are proposed for mobile robots with kinematic constraints. The proposed approach is given in the next section.

3. The proposed smooth tour construction approach

A mobile robot with kinematic constraints (i.e., a minimum turning radius) that starts from a node, visits

the required nodes and returns to the same location can be treated as a Dubins TSP; it is still NP-hard [11]. It can be solved using the approach in [11], but smoothing an existing tour may result in an unnecessarily long tour. A coupled tour construction and tour smoothing may reduce the tour. Therefore, a Smooth Tour Construction (STC) approach is proposed for mobile robots with kinematic constraints. The STC approach considers tour construction and tour smoothing concurrently. The logic behind the tour construction part is based on the SA tour construction method [6]. Additionally, it uses the TS algorithm given in section 2.2. The proposed approach also considers the mobile robot's initial and final orientation.

Assume a mobile robot starts from a node with an initial orientation of θ_{start}, is required to visit a set of nodes N and return to the same location with a final orientation of θ_{final}. Let T_s denote a whole tour or part of a tour and $ST^{R_{SA}}_{\cos t}(T_s)$ denote the cost of the smoothing that results after applying the TS algorithm to tour T_s. A detailed explanation of the STC algorithm that combines SA and TS is given below.

Initialization Step:

i- Set the robot's initial orientation $\theta_s = \theta_{start}$, the robot's final orientation $\theta_f = \theta_{final}$, the radius of the smoothing arc R_{SA} for smoothing, and the service requested active nodes N_{active}. Remove the robot's current node from N_{active} and denote it as node *0*.

ii- Calculate the savings $s_{ij} = d_{0i} + d_{j0} - d_{ij}$ for all pairs of i,j $\in N_{active}$, where $i \neq j$.

iii- Sort the savings in a non-increasing order. Select the first i,j pair from the list and construct an initial smooth tour $ST^{R_{SA}}$ using $T = [0, i, j, 0]$, θ_s and θ_f. Remove nodes i and j from N_{active} and set the tour size as $T_{length} = 4$.

Main Step:

1. Insert each node j $\in N_{active}$ between all the sequential nodes (t_k, t_{k+1}) of tour T and calculate the savings for each inserted node *j using one of the following cases:*
   - If $k = 1$,
   $$KS_{jk} = ST^{R_{SA}}_{\cos t}([t_1, t_2, t_3]) + ST^{R_{SA}}_{\cos t}([0, j, 0]) - ST^{R_{SA}}_{\cos t}([t_1, j, t_2, t_3])$$
   - If $2 \le k \le T_{length} - 2$,
   $$KS_{jk} = ST^{R_{SA}}_{\cos t}([t_{k-1}, t_k, t_{k+1}, t_{k+2}]) + ST^{R_{SA}}_{\cos t}([0, j, 0]) - ST^{R_{SA}}_{\cos t}([t_{k-1}, t_k, j, t_{k+1}, t_{k+2}])$$
   - If $k = T_{length} - 1$,
   $$KS_{jk} = ST^{R_{SA}}_{\cos t}([t_{k-1}, t_k, t_{k+1}]) + ST^{R_{SA}}_{\cos t}([0, j, 0]) - ST^{R_{SA}}_{\cos t}([t_{k-1}, t_k, j, t_{k+1}])$$
2. Sort the savings in non-increasing order, remove the node j with the maximum saving value from N_{active} and add it to the corresponding part of tour T. Set $T_{legth} = T_{legth} + 1$ and update smooth tour $ST^{R_{SA}}$.
3. If N_{active} is not empty, go to Step 1; otherwise, return to $ST^{R_{SA}}$

In this algorithm, the Initialization Step is used to determine an initial smooth tour segment $ST^{R_{SA}}$ for the Main Step. In the Main Step, the savings values are calculated by considering smooth transitions between the nodes and one of the eligible nodes is added to the tour. Note that there are three different types of savings calculations in Step 1 of the Main Step, depending on the insertion location of node *j*. If it is inserted just after the first node (case $k = 1$), the orientation of nodes t_1 and t_2 are kept as before, but the orientation of node t_2 changes. Therefore, the cost of the smooth tour segment $ST^{R_{SA}}_{\cos t}([t_2, t_3])$ is considered in the savings calculation. If it is inserted before the last node (case $k = T_{length} - 1$), the orientation of nodes t_{k-1} and t_{k+1} are kept as before. If node j is inserted between any remaining node pairs (case $2 \le k \le T_{length} - 2$), the orientations of nodes t_k and t_{k+1} changes, but the orientations of node t_{k-1} and t_{k+2} are not affected. This means the smooth tour segment costs are the same both before node t_{k-1} and after node t_{k+2}, so the cost $ST^{R_{SA}}_{\cos t}([t_{k-1}, t_k, j, t_{k+1}, t_{k+2}])$ is considered in the savings instead of in the whole tour cost. The node with the maximum-saving-value is selected in Step 2 of the Main Step. It is also removed from the N_{active} list. If there is no node left in this list, the algorithm returns to the $ST^{R_{SA}}$ in Step 3 of the Main Step.

The STC algorithm adds one of the eligible nodes to the tour at each step by considering the smooth transitions in the tour. This is repeated until there are no nodes left in N_{active}. It means that all the service requested active nodes were added in the planned smooth tour. Because of this property, the STC algorithm is complete. The STC algorithm uses the logic behind the Savings Algorithm in

the tour construction part. As mentioned before, it is shown that, on average, the results of the Savings Algorithm are 6.71% away from the optimal solution [18]. The STC algorithm differs from the Savings Algorithm during the cost calculations in Step 1 of the Main step. Note that the cost calculation of these smooth-sub-paths is independent of the problem dimension n. Therefore, this does not change the complexity of the algorithm; the STC is still $O(n^3)$, like the SA [19].

When the robot starts the tour from the home location, the proposed STC algorithm is used to construct the tour. If changes in service requests and/or arc status occur while following the tour, the current position of the robot is assumed to be connected to the home location. This is firstly handled in the Initialization Step. In Step 1, the unvisited remaining nodes are stored in N_{active} and the home location is denoted by node *0*. The value of index *i* is fixed to the current node of the robot in Step 2 and the initial smooth tour segment $ST^{R_{ss}}$ is found in Step 3. Later, in the Main Step, the index value is kept as a second term in the smooth tour $ST^{R_{ss}}$. This is realized by omitting the case of k=1 in Step 1 of the Main Step. Therefore, the proposed approach can also be used for smooth tour construction in a dynamic environment as in [6].

4. Applications

Experiments and simulations were performed to show the effectiveness of the proposed approach. Experiments were conducted of the smooth tour construction of Pioneer 3-DX (P3-DX) mobile robots. Simulations were performed for test problems.

4.1. Experimental results for the tour construction of P3-DX Mobile Robots

A platform at the Eskisehir Osmangazi University (ESOGU) Artificial Intelligence & Robotic Laboratory [20] (Figure 5) was used as a test bed. The nodes and a picture of the laboratory environment are given in Figure 5.a-b. The laboratory is a 25-node test environment and is assumed to be fully connected without loss of generality. This platform was also used to test various planning algorithms on P3-DX robots ([6], [13]).

The P3-DX robot used in the studies has an on-board P3-800 computer with Linux OS. The sensors on the robot include a SICK LMS laser range finder, sonar sensors, a camera, and a compass. A wireless network is set for communication with other robots and computers. There are cameras on the ceiling for the localization. The location of each robot is calculated and sent to the robots via the wireless network. The low-level control of the robot is carried out using a behaviour-based control approach. The necessary behaviours, such as obstacle avoidance, wandering, moving toward the goal, tracking a reference path, etc. are developed.

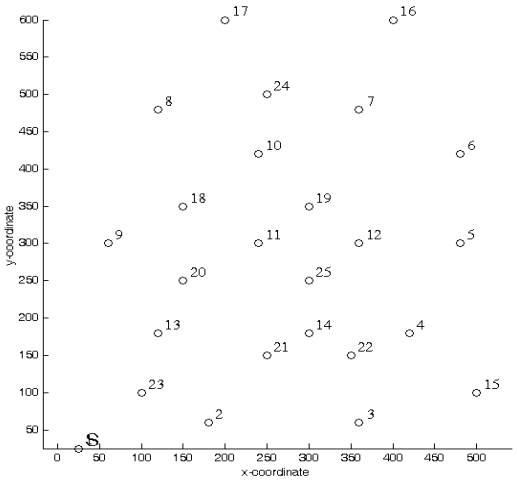




Figure 5. The ESOGU-AIRLAB: a) The placement of the nodes and b) A representative photo.

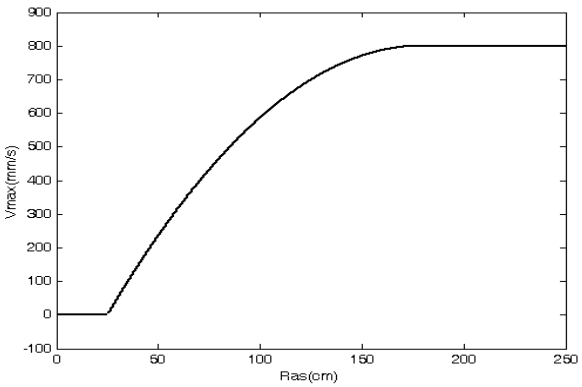


Figure 6. Maximum translational speed versus the smoothing arc radius

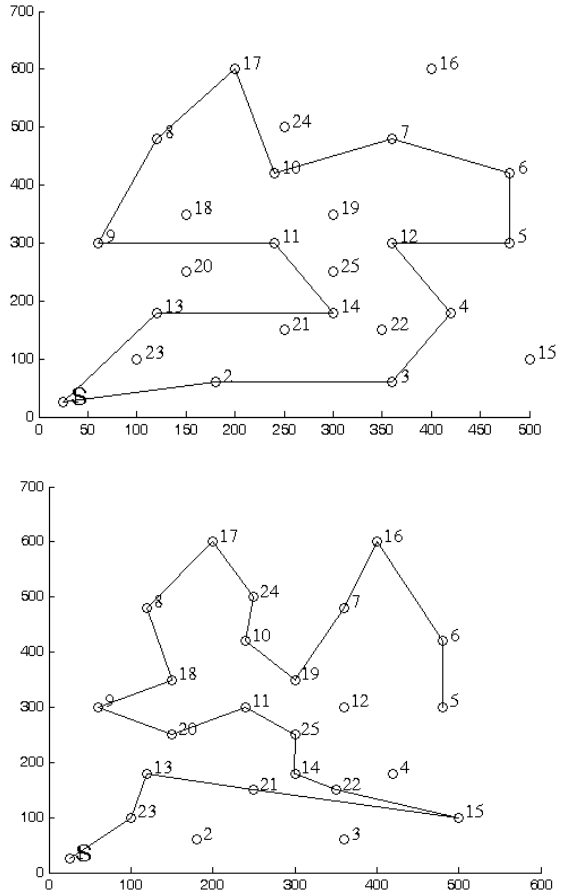


Figure 7. Test of the SA a) start Node is 1 and b) the start node is 5

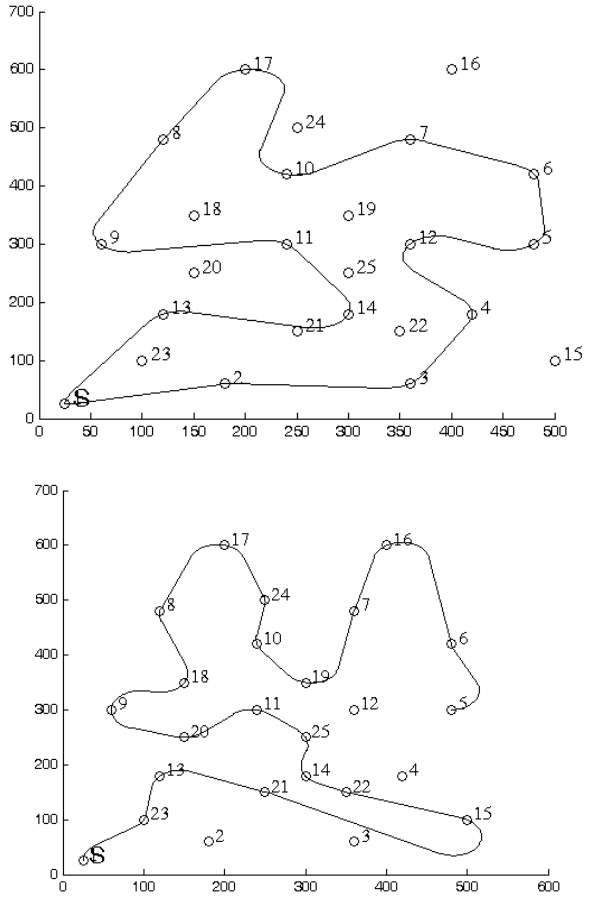


Figure 8. Tours for SA-TS and STC for $R_{SA} = 35cm$ a) start node=1 and b) start node=5

The mobile robot uses the "moving toward the goal" behaviour to track any tour with sharp turns. In this case, the robot decelerates, stops and accelerates at each node. The acceleration and deceleration values are $300\,mm/s^2$, the translational velocity is $100\,mm/s$, and the angular velocity is $30°/s$ during the tracking of such a path. The robot uses the "tracking a reference path" behaviour to track a given smooth tour. This behaviour uses the controller given in equation (4). During the experiments, it is observed that the mobile robot can track a smooth path with a radius of 35 cm with a translational speed of $100\,mm/s$ without tracking errors. Higher translational speeds require a higher circular arc radius to move without slipping. A maximum translational speed versus the smoothing arc radius graph is observed as shown in Figure 6 for nonslip conditions.

In the experimental part, the algorithms SA, SA-TS and the proposed STC are compared in terms of travel distance and travel time. In the first application, the robot is assumed to be at node 1 (starting Node S) with an initial orientation angle of 0° and visits only nodes 1-14 and 17 and then returns to the charging point with an orientation angle of 90°. The SA plans the tour and the robot visits the nodes in the following order T=[1, 2,3,4,12,5,6, 7,10,17,8,9,11, 14,13,1] as shown in Figure 7.a. The length of this tour is 2309.89 cm, and the robot can complete this tour in 271.06 s. In the second application, when the robot arrives at node 5, nodes 15-16 and 18-25 request service after the robot has visited nodes 2,3,4,1 and 2. The cost of the completed tour is 727.23 cm, which was travelled in 82.6 s. In this case, the SA given in [6] is able to find a tour in such a dynamic case. The required remaining nodes are used to plan a tour T=[1, 2,3,4,12,5,6,7,10,17,8,9,11,14,13,1] as shown in Figure 7.b. The length of this tour is 2399.25 cm and the robot can complete this in 288.9 s. Note that these tours are followed by the mobile robot using the "moving toward the goal" behaviour; therefore, the WMR decelerates, stops and accelerates at each node.

In the third application, the SA-Tour Smoothing (TS) and Smooth Tour Construction (STC) are compared. The experimental settings are the same as in the previous applications (see Figure 7). Both SA-TS and STC have the same results as in Figure 8.a for a smoothing radius of $R_{SA} = 35\,cm$. In this case, the length of this tour is 2417.21 cm and the robot can complete this tour in 241.7 s. For the case shown in Figure 8.b., the length of this tour is 2639.5 cm and the robot can complete this tour in

263.9 s. Although the tour lengths are increased compared with Figure 7, there is a decrease in travel time. Note that the tours are followed with the mobile robot using the "tracking a reference path" behaviour.

In the fourth application, the effect of the tour smoothing radius is examined. Considering the same experimental settings of Figure 7.a, the SA-TS and STC have different results, as shown in Figure 9.a-b, for a smoothing radius of $R_{SA} = 45cm$. The tour lengths are 2734.12 cm and 2677.07 cm for the SA-TS and STC, respectively. The travel times are 141.4 s and 138.4 s for SA-TS and STC, respectively. Note for the radius of 45 cm that the linear translational velocity is increased to 193 mm/sec. For this linear velocity, the completion time of the tour produced by SA in Figure 7.a would be 164.2 s, which is still higher than SA-TS or STC.

4.2. Comparisons of the SA, SA-ST, SA-AA, and STC methods

In this subsection, the results of the SA, SA-TS, SA-AA and STC are compared for all possible initial start nodes for the environment in Figure 6. During the simulations, the start node is changed and the planner is asked to visit all the nodes in Figure 6 and then return to node 1 again. Comparisons are made for the smoothing radius of 35 cm and 45 cm in Table 1 and Table 2, respectively. In these tables, the first column shows the initial start node, and TD and TT after the methods denotes the travel distance and travel time, respectively. The last row shows the average values of these values for all possible initial start nodes.

Note that in each table, the SA-TS, SA-AA or STC approaches increase the total travel distance. The average travel distance and travel time of SA-TS and SA-AA are higher than the proposed STC approach. The STC also has a better average travel time than the SA approach. For the 25 node problem, the SA calculation time is approximately 3ms for all the initial start nodes. The STC method calculates the tours at approximately 0.3s. The travel time cost benefit is acceptable.

Start Node	SA TD (cm)	SA-TS TD (cm)	SA-AA TD(cm)	STC TD (cm)	SA TT (sec)	SA-TS TT (sec)	SA-AA TT (sec)	STC TT (sec)
1	2934.3	3925.6	6059.7	3925.6	363.7	392.6	606,0	392.6
2	2604.7	3264.4	5369.3	3264.4	319.2	326.4	536,9	326.4
3	2695.9	3765.8	5657.3	3087.8	335.2	376.6	565,7	308.8
4	2713.3	3739.5	5417.4	3586.6	334.7	374	541,7	358.7
5	2745	3672.8	6090.5	3227.8	338.9	367.3	609,1	322.8
6	2816.1	3944.7	5755.6	3587.6	357.9	394.5	575,6	358.8
7	2723.7	4443.5	5819.4	3366.8	348	444.3	581,9	336.7
8	2719.2	3974.1	5777.6	3162.6	349.9	397.4	577,8	316.3
9	2748	3972.7	5630.8	3300.3	349.4	397.3	563,1	330
10	2653.2	4096.9	5823.9	3364.7	342.3	409.7	582,4	336.5
11	2844.7	3417	6186.5	3417	356.5	341.7	618,7	341.7
12	2766.9	3935.6	5188.4	3290.6	346.8	393.6	518,8	329.1
13	2783.5	4132.6	5353.6	3594.1	347.7	413.3	535,4	359.4
14	2766.2	3655.7	5372.6	3372.3	345.4	365.6	537,3	337.2
15	2646.4	3661.8	5550.9	3450	327.3	366.2	555,1	345
16	2660.9	4426.5	6105.8	3340.2	342.6	442.6	610,6	334
17	2728.6	3765.3	5360.2	3227.6	346.8	376.5	536,0	322.8
18	2711.9	3957.9	5907.6	3333.5	345.7	395.8	590,8	333.3
19	2806.2	3521.5	5840.4	3505.1	340.6	352.2	584,0	350.5
20	2771.9	3815	5914.1	3545.2	351	381.5	591,4	354.5
21	2724.6	3248.8	5317.2	3248.8	337.3	324.9	531,7	324.9
22	2738.8	3824.1	5603.3	3419	342.6	382.4	560,3	341.9
23	2832.1	3839	5321.3	3602.2	352	383.9	532,1	360.2
24	2748.3	4014.5	5491.0	3442.7	348.3	401.4	549,1	344.3
25	2725.1	3873.5	5836.0	3502	347.1	387.4	583,6	350.2
Aver.	**2744.4**	**3835.5**	**5670.0**	**3406.6**	**344.7**	**383.6**	**567,0**	**340.7**

Table 1. The results of the methods for a smoothing radius of $R_{SA} = 35cm$

Start Node	SA TD(cm)	SA-TS TD(cm)	SA-AA TD(cm)	STC TD(cm)	SA TT(sec)	SA-TS TT(sec)	SA-AA TT (sec)	STC TT(sec)
1	2934.3	5232.8	6531,2	5232.8	229.9	270.7	337,8	270.7
2	2604.7	4951.9	7154,3	3857.7	200.9	256.1	370,1	199.5
3	2695.9	6127	6943,2	3820.4	212.5	316.9	359,1	197.6
4	2713.3	5919.6	6266,7	3894.4	211.1	306.2	324,1	201.4
5	2745	6378.2	6672,4	4142.6	213.9	329.9	345,1	214.3
6	2816.1	6951.9	7227,9	3982.2	229.5	359.6	373,9	206
7	2723.7	7234.1	7065,8	4193.5	224	374.2	365,5	216.9
8	2719.2	7000.4	6685,3	3565	226.1	362.1	345,8	184.4
9	2748	6942.5	7210,3	3588.3	224.2	359.1	373,0	185.6
10	2653.2	7100.4	7066,7	4036.7	221.7	367.3	365,5	208.8
11	2844.7	4986.6	6834,7	4271.1	226.6	257.9	353,5	220.9
12	2766.9	6462	6516,9	3864	220.7	334.2	337,1	199.9
13	2783.5	5620.3	6408,5	4367	220.8	290.7	331,5	225.9
14	2766.2	5323	6767,2	4037.7	219.4	275.3	350,0	208.8
15	2646.4	5553	6451,0	3744.2	207.1	287.2	333,7	193.7
16	2660.9	7497.3	7408,1	3551.3	221.6	387.8	383,2	183.7
17	2728.6	6747.8	6609,8	3481.1	222.6	349	341,9	180.1
18	2711.9	6395.7	7301,3	4147.2	222.3	330.8	377,7	214.5
19	2806.2	4882.8	5790,2	3656.9	212.6	252.6	299,5	189.2
20	2771.9	6474.3	7938,4	3723.2	224.6	334.9	410,6	192.6
21	2724.6	4804.1	6709,3	4071.1	213.3	248.5	347,0	210.6
22	2738.8	6278.5	6759,9	4038.3	217.9	324.8	349,7	208.9
23	2832.1	5747	6631,5	4184.5	222.7	297.3	343,0	216.4
24	2748.3	6778	6602,9	3584.1	223.1	350.6	341,5	185.4
25	2725.1	6587.6	6498,5	4115.8	223	340.7	336,1	212.9
Aver.	**2744.4**	**6159.1**	**6802,1**	**3966.1**	**219.7**	**318.6**	**351,8**	**205.1**

Table 2. The results of the methods for a smoothing radius of $R_{SA} = 45cm$.

Simulations are also conducted to find the effect of the radius of the smoothing arc (R_{SA}) on the tour construction quality of SA-TS, SA-AA and STC for R_{SA} values between 35 cm and 250 cm. The average travel distance and travel time values of the SA-TS, SA-AA and STC methods are calculated by considering all possible initial start nodes, as in the last row of Table 1-2. The velocity values for the travel time are calculated using Fig 6. The average travel distance and time versus smoothing radius graphs are drawn in Fig. 10.

For the lower values of the smoothing radius (R_{SA}), the travel distance and travel time are almost the same for SA-TS and STC. The SA-AA has a higher travel distance and travel time. As the value of the R_{SA} increases, the SA-AA values become better compared to SA-TS. It is the expected result, as given in [11]. For the higher values of the R_{SA}, the travel distance increases in SA-AA more than in the STC method. In terms of the travel time, the SA-AA has higher values compared with STC (Fig. 10.b). Although all the methods use the same translational speeds for the same smoothing radius, the STC outperforms the other methods for the travel time criterion. This is due to the logic behind the methods. In the SA-TS and SA-AA approaches, an existing tour is smoothed. Therefore, the same tour (i.e., the same order of the nodes is also used for a larger smoothing radius. On the other hand, the STC approach calculates different tours depending on the smoothing radius, i.e., the order of the nodes to be visited changes.

The STC has the minimum average travel time value for a smoothing radius of approximately 90 cm. This can also be used to find the optimum tour smoothing radius value for the minimum time completion of a task in a given environment.

4.3. Comparison of SA-TS, SA-AA and STC on test problems from TSPLIB

The methods are compared with Eil51 and Eil76 TSP-test problems from TSPLIB [21] in terms of the average travel distance (Fig 11). Considering the dimensions of the test

problems, the range of the radius is selected between 0-5 units. Because they are test problems, no velocity is defined.

Considering Figure 11.a.b, the SA-TS and STC distances are almost the same at a radius interval of 0-2 units. They have the same cost values when $R_{SA} \to 0$. The SA-TS has a better travel distance compared to SA-AA, the average travel distance of SA-TS increases more than the average travel distance for the STC method. This is also expected for the travel time in real applications. For Eil76, the tour calculation time is 0.1 s and 10 s for the SA-TS/SA-AA and STC methods, respectively. The drawback of the STC method is the tour calculation time cost. However, considering the acceptable durations of high-level planning [2] and the cost benefit of the STC, the calculation time is not a drawback for real-time applications.

As a result, the proposed STC method has an advantage over smoothing an existing tour for higher smoothing radius (R_{SA}) values. The proposed approach can also be used as a decision support tool to determine the tour smoothing radius for a given specific environment for better travel time.

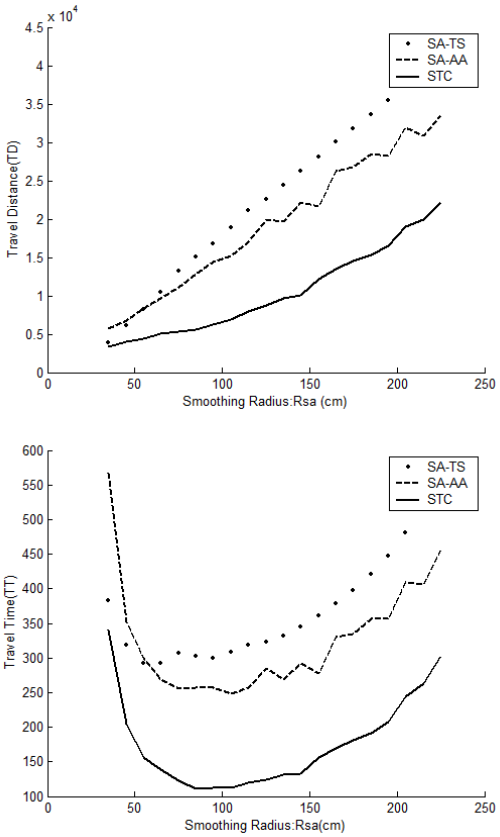


Figure 10. a) Average travel distance versus R_{SA} graph, b) Average travel time versus R_{SA} graph.

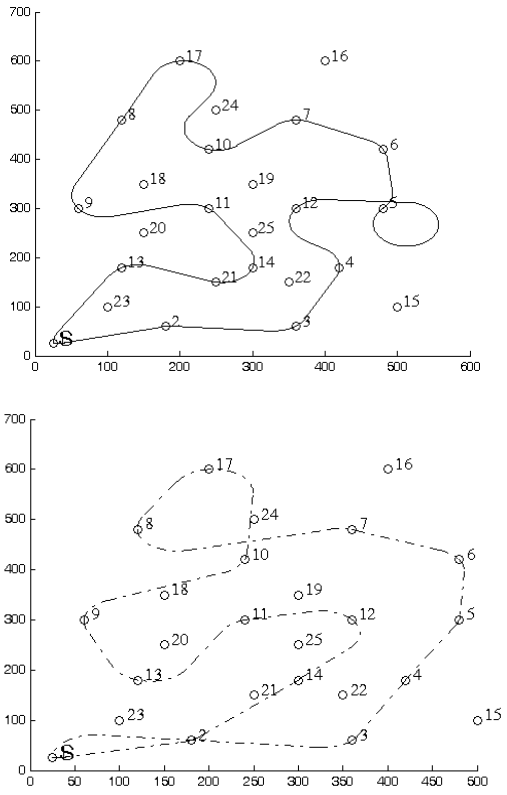


Figure 9. The result of the tours for $R_{SA} = 45cm$ and start node=1: a) SA-TS and b) STC

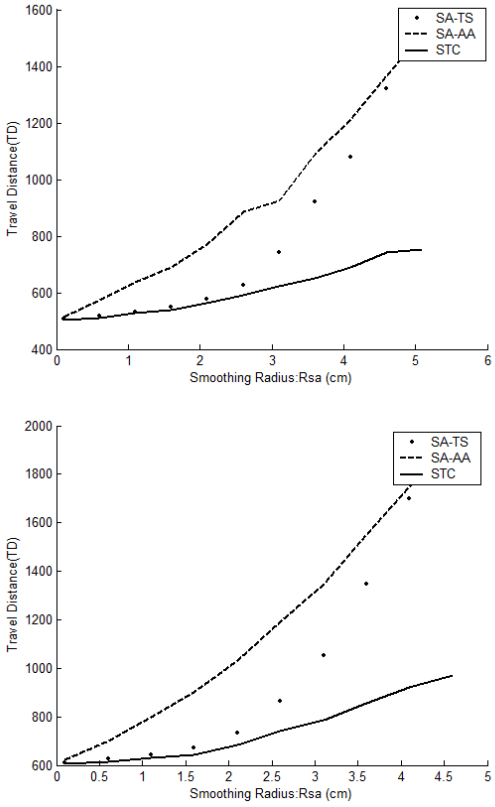


Figure 11. Travel Distance versus R_{SA} graph for test problems: a) Eil 51 and b) Eil 76

5. Conclusion

In this study, a smooth tour construction algorithm is proposed and tested for P3-DX robots and TSPLIB test problems. As the radius of the smoothing-arc increases, smoothing an existing tour leads to longer travel distances and travel times. Therefore, smooth tour construction is needed for efficient travel distance and travel time tours. The proposed approach can be used to find the optimal tour radius of the smoothing-arc for a minimum time tour for a given task environment. The proposed approach is also tested for dynamic cases. In future works, multi-robot applications will be considered.

6. Acknowledgements

This work was partially supported by the Research Fund of Eskişehir Osmangazi University under Contract 201215013.

7. References

[1] Y. Hada, H. Gakuhari, K. Takase, E. I. Hemeldan, "Delivery service robot using distributed acquisition, actuators and intelligence, Proceedings of the International conference on Intelligent Robots and Systems", Vol. 3, pp. 2997–3002, 2004.

[2] L. Fu, A.Yazıcı, U. Ozguner, "Route Planning For OSU-ACT Autonomous Vehicle in DARPA Urban Challenge", Proceedings of 2008 IEEE Intelligent Vehicles, pp. 781-786, 2008.

[3] O. Parlaktuna, A. Sipahioglu, A. Yazıcı, U. Gurel, "TSP Approach For Mobile Robot Dynamic Path Planning", in abstracts of the 1st International Conference on Control and Optimization with Industrial Application, pp. 79, 2005.

[4] A. Punnen, "The Traveling Salesman Problem Applications, Formulations and Variations", in G. Gutin, and A. P. Punnen, eds. The Traveling Salesman Problem and its Variations, Kluwer Academic Publishers, pp. 1-28, 2002.

[5] Z. Yu, L. Jinhai, G. Guochang, Z. Rubo, Y. Haiyan, "An implementation of evolutionary computation for path planning of cooperative mobile robots", Proceedings of the 4th World Congress on Intelligent Control and Automation, pp. 1798-1802, 2002.

[6] A. Sipahioglu, A. Yazıcı, O. Parlaktuna, U. Gurel, "Real-time tour construction for a mobile robot in a dynamic environment", Robotics and Autonomous Systems Vol. 56, Issue 4, pp. 289-295, 2008.

[7] L. E. Dubins, "On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents," Amer. J. Math., Vol. 79, pp. 497–516, 1957.

[8] F. Lamiraux, J. P. Laumond, "Smooth motion planning for car-like vehicles", IEEE Transactions on Robotics and Automation, Vol. 17 (4), pp. 498-502, 2001.

[9] M. Pivtoraiko, R. A. Knepper, A. Kelly, "Differentially Constrained Mobile Robot Motion Planning in State Lattices", Journal of Field Robotics, Vol. 26(3), 308–333, 2009.

[10] S. On, A. Yazıcı, "A comparative study of smooth path planning for a mobile robot considering kinematic constraints", Proceedings of International Symposium on Innovations in Intelligent Systems and Applications, pp. 565-569, 2011.

[11] K. Savla, E. Frazzoli, F. Bullo, "Traveling Salesperson Problems for the Dubins vehicle", IEEE Transactions on Automatic Control, Vol. 53, No. 6, pp.1378-1391, 2008

[12] L. Gracia, J. Tornero, "Characterization of zero tracking error references in the kinematic control of wheeled mobile robots", Robotics and Autonomous Systems Vol. 57, pp.565-577, 2009.

[13] A. Yufka, A. Yazıcı, O. Parlaktuna, "A Smooth Path Generation Approach for Sensor-based Coverage Path Planning", Proceedings of 6th International Conference on Electrical and Electronics Engineering, pp. 375-379, 2009.

[14] Y. Kanayama, Y. Kimura, F. Miyazaki, T. Noguchi, "A stable tracking control method for an autonomous mobile robot", Proceedings of IEEE International Conference on Robotics and Automation, Vol. 1, pp. 384 - 389, 1990.

[15] G. Laporte, "The Traveling Salesman Problem: An Overview of Exact and Approximate Algorithms", European Journal of Operational Research, Vol. 59, pp. 231-247, 1992.

[16] G. Clarke, J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points", Operations Research, Vol. 12, 568-581, 1964.

[17] R. H. Ballou, "Business Logistics / Supply Chain Management", 5th ed., Pearson and Prentice Hall, 2004.

[18] J. F. Cordeau, M. Gendreau, G. Laporte, J. Y. Potvin, F. Semet, "A Guide to vehicle routing heuristics", Journal of Operational Research Society, Vol. 53, 512-522, 2002.

[19] O. Parlaktuna, A. Sipahioglu, A. Yazıcı, "A VRP-Based Route Planning for a Mobile Robot Group", Turkish Journal of Electrical Engineering, Vol. 15, 187-197, 2007.

[20] ESOGU-AIRLAB, Eskisehir Osmangazi University Artificial Intelligence and Robotic Laboratory, http://www.ai-robotlab.ogu.edu.tr, 2013.

[21] TSPLIB, http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95, 2012.