The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

A Dynamic Path Planning Approach for Multi-Robot Sensor-Based Coverage Considering Energy Constraints

Ahmet Yazici, Gokhan Kirlik, Osman Parlaktuna, Aydin Sipahioglu

Abstract—**In this study, a novel dynamic path planning approach is proposed for multi-robot sensor-based coverage considering energy capacities of the mobile robots. The environment is assumed to be narrow and partially unknown. A Generalized Voronoi diagram-based network is used for the sensor-based coverage planning due to narrow nature of the environment. On the other hand, partially unknown nature is handled with proposed dynamic re-planning approach. Initially, the robots are assumed to be at the same depot with equal initial energy capacities. In this case, an initial complete coverage route is constructed considering robot energy capacities using classical capacitated arc routing problem (CARP) approach with some minor modifications related to coverage problem. But, due to partially unknown nature, the robots may face with blockage on routes, and a fast re-planning is required which considers remaining energy capacities and current positions of the robots. So, new plan is obtained by a modifying Ulusoy's algorithm that was developed for classical CARP. The developed algorithm is coded in C++ and implemented on P3-DX mobile robots in MobileSim simulation environment.**

I. INTRODUCTION

MULTI-ROBOT sensor-based coverage path planning is the determination of paths such that every point in a given workspace is covered at least once by one of the robot's sensor-range. In sensor-based coverage of narrow spaces, where obstacles lie within detector range, Generalized Voronoi Diagram (GVD) [1] can be used to model the environment. This model can be represented with a network of consisting edges and vertices. If some parts of the environment are not required to be covered, then the corresponding edges to these parts are called non-required edges. For complete sensor-based coverage of the given area, it is sufficient for the robots to follow the required edges of corresponding network. The method in [1] can be used to complete coverage of an indoor environment using single robot. However, if the map is known a-priori, more efficient coverage can be achieved as in [2]. On the other hand, if one robot's energy is not enough to completely cover a given environment, then using multiple robots is a necessity to increase efficiency. Additionally, using multiple robots may reduce the time required to complete the coverage task and enhances robustness compared to the single robot.

In the mobile robot literature, there are some approaches for multi-robot coverage path planning ([3], [4], [5]). In [3], the cost was evaluated in terms of the traveled distance by using the edges of the configuration space and Voronoi diagram. First a tour is generated for traversing all the paths, and then appropriate parts of the tour are assigned to each robot according to the cost evaluation. But, in this work, how to partition the path among the robots considering their energy capacities is not given. The approach in [4] an adaptation of the single robot cellular decomposition approach to multiple robots was presented. In that work also energy capacities of the robots were not considered. In another work [5], mobile robot deployment problem was considered for specific type of coverage problem. The deployment problem was described as determination of the number of groups unloaded by a carrier, the number of robots in each group, and the initial locations of those robots. Both timing and energy constraints of robots were considered for vast environments. However, the deployment was not considered for partially unknown narrow-complex environments. Constructing individual sensor-based coverage tours considering each robot's energy capacity becomes a challenging problem in narrow environments. From another point of view, this problem can be described as partitioning edges of the network among robots without exceeding their energy capacities. This problem resembles capacitated arc routing problem (CARP) [6]. CARP is a problem that aims to construct tours with minimum total distance for vehicles without exceeding their loading capacities. Due to this resemblance, the required energy to cover an edge can be considered as an edge demand, similarly robot's energy capacity can be considered as vehicle's capacity, in CARP.

This work is supported by the Scientific and Technical Research Council of Turkey (TUBITAK), Contract No 107E064, project title: "Mobile Robot Route Planning for Complete Coverage of Dynamic Indoor Environments".

A. Yazici is with the Computer Engineering Department, Eskisehir Osmangazi University, Eskisehir, 26480, Turkey (corresponding author: phone: +90-532-505 5169; fax:+90-222-2393613; e-mail: ayazici@ ogu.edu.tr).

G. Kirlik is with the Industrial Engineering Department, Eskisehir Osmangazi University, Eskisehir, 26480, Turkey (e-mail: gokhankirlik@ gmail.com).

O. Parlaktuna is with the Electrical Engineering Department, Eskisehir Osmangazi University, Eskisehir, 26480, Turkey (e-mail: oparlak@ ogu.edu.tr).

A. Sipahioglu is with the Industrial Engineering Department, Eskisehir Osmangazi University, Eskisehir, 26480, Turkey (e-mail: asipahi@ ogu.edu.tr).

However, there are two differences in coverage problem that was not addressed in classical CARP. The first one is related to edge demand that may vary regarding the edge's status. As the robot passes through a non-required edge, edge demand is determined by only traveling energy which is consumed by motors, embedded computer, microcontroller card and navigation sensors (sonar) [5]. On the other hand, for a required edge, robot spends energy both traveling and performing its coverage task and the coverage energy cannot be calculated by only considering traversal cost due high energy consumption of some sensors, as in this case. Because of this difference, CARP solution techniques can not be used directly for the sensor-based coverage problem with multiple robots. In [7], this difference between the classical CARP problem and the coverage problem explained and solved by some minor modification to Ulusoy's algorithm that is used for classical CARP. The method proposed in [7] was used to construct initial coverage routes for a robot team under the assumption that the robots start from the same depot point and they have equal energy capacities (all fully charged). But, if robots face with a blocking obstacle due to partially unknown nature of the environment, a fast re-planning is required. In this case, robots may be at different locations having different energy capacities. This is another problem that can not be handled using the classical CARP. In this paper, we have done major modification on Ulusoy's algorithm to handle this situation that may be faced in real applications. So, a novel dynamic path planning approach is proposed for solving multi-robot sensor-based coverage problem considering energy capacity of each robot.

The rest of this paper is arranged as follows. In Section 2, definitions and algorithms used in the proposed approach are given. In section 3, proposed approach is given. Applications of the proposed method are given in Section 4. Conclusions and discussions are given in the final section.

II. PRELIMINARIES AND ALGORITHMS USED IN THE PROPOSED APPROACH

A network (graph) can be modeled as *G(V,A)* where *V* is the set of vertices (nodes) and *A* is the set of edges (arcs) connecting the vertices. A network is called as directed or undirected if its edges have direction or not. If it is possible to reach all of the vertices through existing edges, network is called as connected, otherwise disconnected. If there is connection each pair of vertices, it is called complete network. If the network is not complete, the shortest path between any two vertices can be calculated using Floyd algorithm [8] to construct a distance matrix.

A tour *T* is defined as Eulerian if it is possible to return the starting vertex by passing through each edge exactly once [6]. If an Euler tour does not exist, some edges must be visited twice or more to return the starting point. In this case, Chinese Postman Problem (CPP) occurs. CPP is an edge visiting problem in a network with minimum cost or minimum distance. Since CPP is not an NP-hard problem, both mathematical model and heuristics such as Edmonds and Johnson's Minimum Perfect Matching algorithm [9] with Hierholzer algorithm [8] can be used efficiently. If only some edges of the network need to be visited, CPP turns into Rural Postman Problem (RPP). Unlike CPP, RPP is an NP-hard problem and finding an optimal RPP tour is really hard. If the required edge set is disconnected, Christofides' heuristic can be used to construct RPP tour. Constructing the new tour with only unvisited edges may result several visits of the same edges, which may result unnecessarily long tour. So, this tour can be improved by using Shorten Algorithm [10].

In CPP/RPP, there is only one vehicle (visitor) to visit all the edges. However, if there is more than one vehicle, the problem is called as k-Chinese problem [11]. Besides, if vehicles have capacity constraint such as loading or energy, the problem turns into CARP. Garbage collecting is a good application of CARP. In this problem, all vehicles have a certain loading capacity and each edge has an amount of garbage required to be collected which is called as edge demand. CARP is an NP-hard problem. Hence, different heuristic algorithms are developed in the literature such as Simple constructive methods (Construct-Strike, Modified-Construct-Strike, Path-Scanning, Augment-Merge algorithm, Parallel Insert Algorithm, Augment-Insert Algorithm), two-phase constructive methods (Ulusoy-Partitioning algorithm [12], Cut Algorithm and Cycle Assignment algorithm [13]), and metaheuristics (Tabu search-based algorithms [14] and genetic algorithms [15]). In [7], a minor modification was done on Ulusoy's partitioning algorithm to handle multi-robot sensor-based coverage path planning problem for the same depot point and equal energy capacity. Assume that e_{ij} denotes energy consumption during the travel through edge due to motors, micro controller units, etc., and q_{ij} denotes required energy for sensor coverage and traveling on the edge (v_i, v_j), R denotes the required edge set which must be serviced, and initially each robot has limited energy capacity (E_{cap}). The algorithm given in [7] is as follows:

Modified Ulusoy's Algortihm (MUA):

Inputs: G(V,A), R, T, depot, E_{cap}.

Outputs: Tour for each robot

Step 1: Re-label the vertices in *G* so that the given tour *T* is equal to $(v_0, v_1, ..., v_t = v_0)$, where v_0 is the depot. Let *r* be the largest index of a vertex incident to a serviced edge on *T*. Construct a directed network $G' = (V', A')$ with vertex set $V' = \{v_0, v_1, ..., v_r\}$ and introduce edge pairs (v_a, v_b) for $a, b = 1,2,3,\ldots, r$ in A that satisfy $b > a$. Remove all edges (v_a, v_b) for $a, b = 1,2,3,\ldots, r$ such that $b > a+1$ and (v_a, v_{a+1}) or (v_{b-1}, v_b) is not a serviced edge on *T*. Consider cases as follows.

• If $b = a + 1$ and (v_a, v_{a+1}) is not a required edge on *T*, then set $d'_{ab} = 0$.

• If $b > a + 1$ or (v_a, v_{a+1}) is a required edge and chain P_{ab} contains the depot on *T*. $P_{ab} = (v_a, ..., depot, ..., v_b)$. Then add to P_{ab} a shortest chain between v_b and v_a in *G*. And cycle $C_{ab} = (depot, ..., v_a, SP_{ab}, v_b, ..., depot)$ is held. Where SP_{ab} denotes shortest path between v_a and v_b in *G*.

• If $b > a + 1$ or (v_a, v_{a+1}) is a required edge and chain P_{ab} does not contain the depot on *T*. $P_{ab} = (v_a, ..., v_b)$. Then add to P_{ab} a shortest chain between the depot and v_a, and shortest chain between the depot and v_b in *G*. And cycle $C_{ab} = (SP_{depot\ a}, v_a, ..., v_b, SP_{depot\ b})$ is held.

• Calculate total energy load for C_{ab} using non-required edge demand e_{ij} and required edge demand q_{ij} for all edges in C_{ab}. If total energy load does not exceed E_{cap}, calculate distance cost d'_{ab} of (v_a, v_b) in G', defined as the total distance cost of C_{ab} in *G*. Else remove (v_a, v_b) from G'.

Step 2: Solve a shortest path problem from v_0 to v_r in G'. Each edge (v_a, v_b) used in the shortest distance path corresponds to a feasible vehicle route on *G*.

The logic behind the MUA is partitioning an initial Euler tour into feasible robot tours considering their energy capacities. This algorithm constructs these tours into two main steps. In the first step, the algorithm generates a directed graph (G') that's each arc represents a feasible robot tour. So, the elements of the distance matrix of this graph are the cost of the feasible tours. In the second step, a shortest path problem is solved on this directed graph from source vertex to terminal vertex. The result of the shortest path problem consist of is minimum cost feasible robot tours that cover all required edges.

The input of the algorithm is a single Euler tour *T* that can be generated using CPP/RPP. Then, the algorithm originally constructs an intermediate network by considering equal vehicle capacities. Later, a shortest path algorithm is used to determine both the minimum number of required vehicles to visit all the edges and tour for each vehicle. Since, the robots may have different energy capacities at some points (especially during re-planning), this algorithm can not handle this situation. So, in this paper, a new approach for dynamic re-planning is proposed. The details of the proposed approach combined with initial planning are given in the next section.

III. THE PROPOSED APPROACH

In multi-robot sensor-based coverage problem, since the edges are required to be covered and robots have limited energy capacity, this problem resembles CARP. But, there are some differences mentioned in the previous sections. In this paper, a new approach is proposed for dynamic re-planning of multi-robot sensor-based coverage problem considering robot energy capacities. A flow chart of the proposed approach is given in Fig. 1.

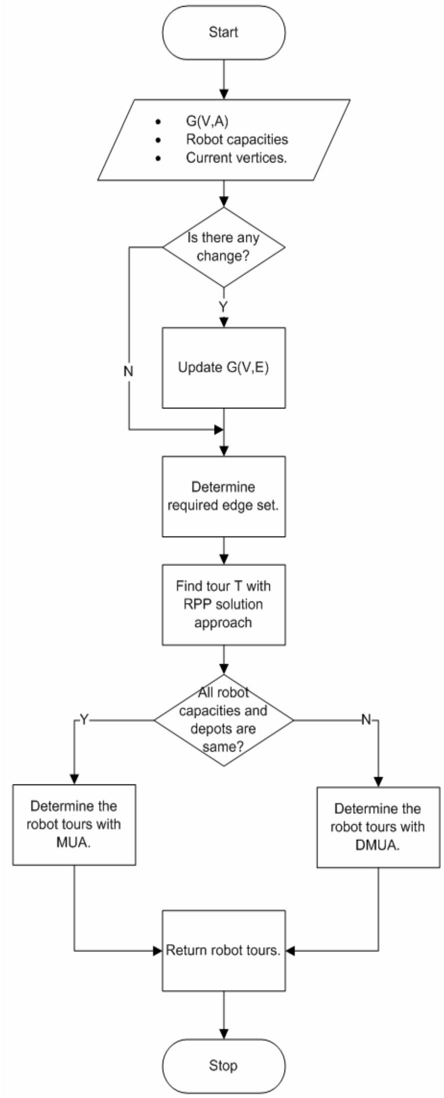

Fig. 1. Flowchart of the proposed approach

In the flow chart given in Fig. 1, initially a GVD-based network $G(V, A)$ of the environment is loaded. If there is any closed edge between any pair of vertices; Floyd's algorithm is used to find the minimum path between that pair of vertices. Using these minimum paths, the distance matrix *D* is updated such that all $d_{ij} > 0$. Then, the information about the robots: Number of available robots *m*, depot points ($Depot_k$) and energy capacity (E^k_{cap}) for $k = 1, \ldots, m$ of robots, are loaded.

If there is no change in the environment, the network of the initially loaded map is used, otherwise the network is updated. Then the required edge set *R* is determined. If complete coverage of the environment is required, then the required edge set includes all edges of the network *G*. But, in a changed environment, the required edge set includes the ones that are not covered yet. In the next step, a single Euler tour T is constructed that covers all required edges using the proposed approach in [2].

After determining an Euler tour T, one of the two cases is possible: all robots are in the same depot point with equal energy capacities or otherwise. The first case can be solved using with MUA proposed in [7]. This case usually can be faced during the initial plan where robots are fully charged and in the same depot.

But, if at least one robot is at a different depot ($Depot_k$), or has different energy capacity (E_{cap}^k) then the problem can be solved using with Dynamic Modified Ulusoy Algorithm (DMUA) as follows:

Inputs: $G(V,A)$, R, T, $Depot_k$, E_{cap}^k for $k = 1,\ldots,m$.

Output: Tour for each robot

Step1: For $k = 1:m$

Set $depot = Depot_k$, and $E_{cap} = E^k{}_{cap}$.

a. Using the step 1 of the MUA given in the previous section (determine C_{ab} and calculate the d_{ij}), construct G' for robot k.

b. Copy $G'(V',A')$ into a three dimensional network $G^*(V',A^*)$. (Noting that A^* is the edge set of G^* with elements of d_{ijk}).

Step 2: Solve the shortest path problem on G^* by using proposed mathematical model and construct tour for each robot with model results.

This algorithm is different than the Modified Ulusoy algorithm (MUA) since it repeats step 1 of MUA for each robot and constructs a new network which is denoted with $G^*(V',A^*)$. The vertices of G^* is same as the vertices of G'. But the edge set of G^* has a third dimension where the depth is determined by the number of robots (m). So it differs from the edge set of G'. As a result, there is a different shortest path problem than the former one in step 2. This shortest path problem is between the depot vertex and terminal vertex on network $G^*(V',A^*)$ and is solved with a new integer linear model. The model is given as follows:

Sets and parameters:

n The number of vertices in the network

m The number of robots

I, J $\{1, 2, \ldots, n\}$ denotes the numbers of vertices

K $\{1, 2, \ldots, m\}$ denotes the numbers of robots

c_{ijk} Distance between vertex i and vertex j for robot k

Decision variables:

$$x_{ijk} = \begin{cases} 1 & \text{If robot k visits from vertex } i \text{ to vertex } j \\ 0 & \text{otherwise} \end{cases}$$

Model:

$$min\left(\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{\substack{k=1 \\ (i,j,k)\in A^*}}^{m} c_{ijk} \cdot x_{ijk} \right) \quad (1)$$

subject to,

$$\sum_{k=1}^{m} \sum_{\substack{j=1 \\ (1,j,k)\in A^*}}^{n} x_{1jk} = 1 \quad (2)$$

$$\sum_{k=1}^{m} \sum_{\substack{i=1 \\ (i,n,k)\in A^*}}^{n} x_{ink} = 1 \quad (3)$$

$$\sum_{k=1}^{m} \sum_{\substack{i=1 \\ (i,p,k)\in A^*}}^{n} x_{ipk} - \sum_{k=1}^{m} \sum_{\substack{j=1 \\ (p,j,k)\in A^*}}^{n} x_{pjk} = 0 \quad p = \{2,\ldots,n-1\} \quad (4)$$

$$\sum_{i=1}^{n} \sum_{\substack{j=1 \\ (i,j,k)\in A^*}}^{n} x_{ijk} = 1 \quad k = \{1,\ldots,m\} \quad (5)$$

$$x_{ijk} \in \{0,1\} \quad (6)$$

In this model, the objective function (1) minimizes the sum of distances which are chosen depot vertex to terminal vertex. Constraint (2) ensures that exactly one sortie from depot vertex. Constraint (3) ensures that exactly one arrival to terminal vertex. Constraint set (4) is flow conservation constraints which means only one entry and one exit should be for intermediate vertices. Constraint set (5) ensures that exactly one edge should be chosen for each robot. In this study, LP-solver GAMS/CPLEX [16] is used to solve the model.

So, the overall proposed algorithm considers not only different energy consumption (i.e. during the coverage and the travel) but also handle dynamic re-planning for multi robot sensor-based coverage considering both different energy capacities and starting points for each robot in partially unknown environments.

IV. APPLICATIONS

The algorithms in the proposed approach are coded in C++ and tested in various environments. Firstly, the platform at ESOGU Artificial Intelligence & Robotic laboratory is used to show the effectiveness of the algorithms. This platform is also used to test various coverage algorithms on P3-DX robots [2]. Videos of the live performance of P3-DX mobile robots are recorded and can be reached from following web site [17]. Later, a larger indoor environment is used for the tests up to ten robots in simulation.

During the simulations, two types of energy consumption are defined: Deadheading energy which is consumed while robot is passing through a non-required edge and covering energy which is consumed while robot is performing coverage task. In this study laser range finder is used for coverage purposes. Assuming fixed velocity (ignoring turnings), and time to travel an edge as $t_{ij} = d_{ij} / velocity$, the energy consumption of travel is modeled with $e_{ij} = (17.49 + (7.4 \cdot (velocity/1000))) \cdot t_{ij}$ and Sick LMS-200 laser range finder's energy consumption is modeled as $q_{ij} = 20 \cdot t_{ij}$. During coverage energy is calculated as the sum of both energy types. Mobile robot's velocity is constant and 400mm/s. Therefore, mobile robot consumes 0.051 joule/mm for traversing and 0.05 joule/mm for coverage.

A. Application in the laboratory environment

First, the proposed algorithm is applied to sensor-based coverage of an indoor environment which is shown in Fig. 2. In this study, the platform is transferred to MobileSim simulation environment. A topological map of this platform with GVD-based network is given in Fig. 3. Covering all the area is achieved by following all the edges of this figure.

Fig. 2. A photograph of the test environment

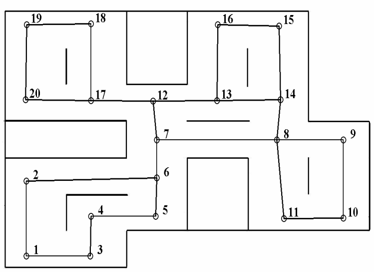

Fig. 3. Topological map of the test environment

In simulations, two mobile robots, initially at vertex 1, are assumed have initial energy capacity of 3800 joules. The proposed method generates the paths as in Fig. 4.

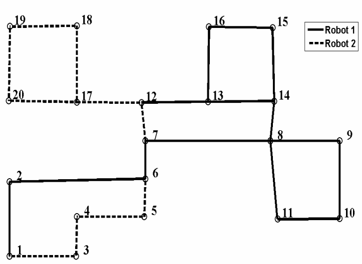

Fig. 4. Division of the tour among two robots

Robot tours are as follows; Robot 1 tour: **(1-2), (2-6), (6-7) (7-8), (8-9), (9-10), (10-11), (11-8), (8-14), (14-15), (15-16), (16-13), (13-14),** *(14,13),* **(13-12)**, *(12-7)*, *(7-6)*, *(6-2)* and *(2-1)*, and Robot 2 tour: *(1-2)*, *(2-6)*, *(6-7)*, *(7-12)*, **(12-17), (17-18), (18-19), (19-20), (20-17)**, *(17-12)*, **(12-7)**, *(7-6)*, **(6-5), (5-4), (4-3), (3-1)**. The robots performs coverage task while passing through bold-written edges and use the italic-written edges for passing (deadheading edges). If both robots were finished their tours without facing an obstacle, they would consume 2953 joules and 1944 joules energy, respectively. Tour lengths would be 33352 mm. and 23946 mm.

But, Robot 2 detects an obstacle between vertices 7-12 at 40th second, and unable to follow the planned path. Up to this moment, Robot 1 has covered the edges **(1-2), (2-6), (6-7) (7-8),** so remaining uncovered edges 1-3, 3-4, 4-5, 5-6, 8-9, 8-11, 8-14, 9-10, 10-11, 12-13, 12-17, 13-14, 13-16, 14-15, 15-16, 17-18, 17-20, 18-19 and 19-20 should be covered.

At this moment, consumed energies are different for each robot. Robot 1 consumed 422 joules for traversing and 413 joules for coverage. Totally, it consumed 835 joules of energy. Robot 2 consumed only 367 joules for traversing. Remaining capacities for robots are 2965 joule and 3433 joules, respectively. Considering these values, new tours are constructed using the proposed approach as in Fig. 5.

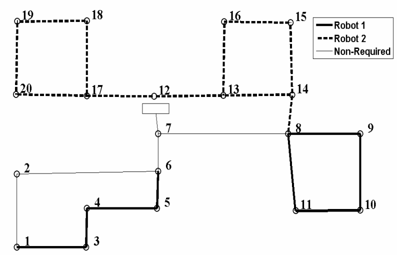

Fig. 5. Division of the tour among two robots in dynamic case

The re-assigned required edges to the robots by the proposed algorithm are as follows; Robot 1 tour: *(7-8),* **(8-9), (9-10), (10-11), (11-8),** *(8-7), (7-6) , (6-2), (2-1),* **(1-3), (3-4), (4-5), (5-6),** Robot 2 tour: *(7-8),* **(8-14), (14-15), (15-16), (16-13), (13-12), (12-17), (17-18), (18-19), (19-20), (20-17),** *(17-12), (12-13)*, **(13-14).** Note that the upper right edges (13-14, 14-15, 15-16, 13-16) are assigned to Robot 2 in the new tours. For these tours, Robot1 and Robot 2 consume 1763 joules and 2038 joules of their remaining energies, respectively. Tour lengths are 22645 mm. and 22464 mm. Trace of the mobile robots in MobileSim simulation environment for full coverage of the given environment is given in Fig. 6.

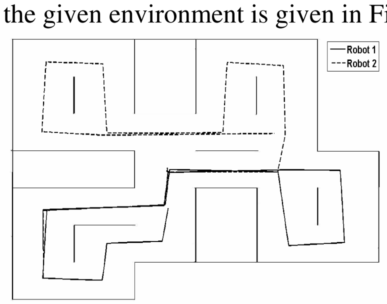

Fig. 6.. Trace of robots for complete sensor-based coverage

Remaining energy capacities for Robot 1 and 2 are 1202 and 1395 joules, respectively at the end of the coverage.

B. *Test of the algorithm on a larger graph*

In order to test the proposed algorithm on a large-number-vertices environment, the first floor of the Eskisehir Osmangazi University Electrical Engineering laboratory building is used as the test bed. In this floor, there are 4 laboratories. Inside each laboratory, there are three rooms, tables, storage cabinets, and columns. A corridor connects the laboratories. Topological map of the first floor is given in Fig. 7. Number of vertices in the graph is 90.

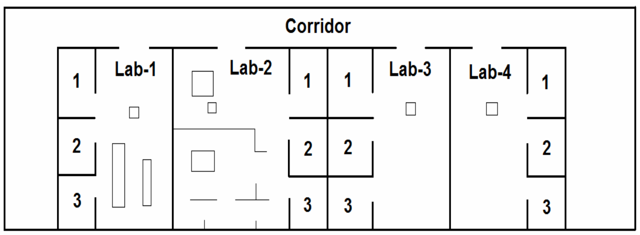

Fig. 7. Topological map of the first floor

Simulations are carried out up to ten robots using the proposed approach for different randomly selected robot initial locations. Fig. 8. shows a considerable decrease in the average tour length by each robot up to 6 robots. The solution time is less than 1 second up to 4 robots, and 4 seconds up to ten robots. There is a linear increase of solution time in terms of number of robots.

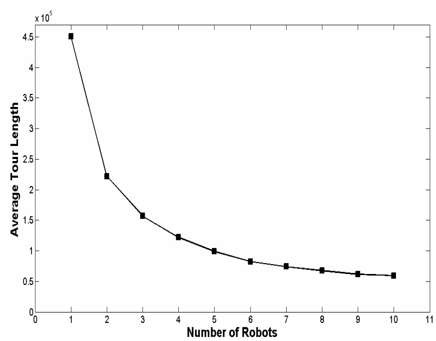

Fig. 8. The average tour length versus number of robots

As seen from Fig. 8, marginal utility of additional robot decreases in the case of average tour length per robot after 6 robots. Therefore, maximum six robots may be assigned to the above coverage task. This kind of analysis can be used to determine maximum number of robots to assign for a given coverage task.

V. CONCLUSION

In this study, a new approach, based on capacitated arc routing problem (CARP) was proposed and applied to multi-robot dynamic sensor-based coverage planning for interior environments. It constructs the sensor-based coverage paths both in known and partially unknown environments considering robot energy capacities. Partially unknown nature of the environment handled by adding re-planning which was not addressed in classical CARP solutions. The proposed method can also be used in the case of robot failures. A dynamic re-planning with existing robot locations and energies should be enough to handle the failure. This approach can also be used to determine maximum number of robots to be assigned to a given coverage task for efficient coverage which is very important for resource allocation. Another important contribution of this study is that the proposed algorithm is flexible and can be used in other robot application problems. For example, although the proposed algorithm is used for energy capacitated mobile robots, it can be extended to consider time-constraint sensor-based mobile robot coverage problem, as well.

REFERENCES

[1] E. U. Acar, H. Choset and J. Y. Lee, “Sensor-based coverage with extended range detectors”, IEEE Transactions on Robotics, vol. 22 (1), pp. 189-198, 2006.

[2] A. Sipahioglu, A. Yazici, G. Kirlik, O. Parlaktuna, “A sensor-based coverage approach in dynamic environments”, The 6th International Symposium On Intelligent & Manufacturing Systems, pp. 502-509, Sakarya, Turkey, 2008.

[3] D. Kurabayashi et al., “Cooperative sweeping by multiple mobile robots”, Proc IEEE Int. Conf Robotics Automat, vol. 3, pp. 1744–1749, 1994.

[4] D. Latimer et al., “Towards sensor based coverage with robot teams”, Robotics and Automation Proceedings ICRA '02, vol. 1, pp. 961-96, 2002.

[5] Y. Mei, Y.H. Lu,, Y.C. Hu, C.S.G Lee, “Deployment of mobile robots with energy and timing constraints”, IEEE Transactions on Robotics and Automation, vol. 22(3), pp. 507-522, 2006.

[6] H.A. Eiselt, “A historical perspective on arc routing”. In: Dror M., editor. Arc routing. Theory, solutions and applications. Boston, Kluwer Academic Publishers, pp. 1-16, 2000.

[7] O. Parlaktuna, A. Sipahioglu, G. Kirlik, A. Yazici, Multi-robot sensor-based coverage path planning using capacitated arc routing approach, the 24th IEEE International Symposium on Intelligent Control (ISIC), in MSC09, pp. 1146-1151, Russia, July 8-10 2009.

[8] K. H. Rosen, Handbook of Graph Theory, New Jersey: CRC Press, 2004.

[9] J. Edmonds, E.L. Johnson, “Matching, Euler tours and Chinese Postman”, Mathematical Programming vol. 5, pp. 88-124, 1973.

[10] A. Hertz, G. Laporte and P. N. Hugo, “Improvement procedures for the undirected rural postman problem”, INFORMS Journal on Computing, vol. 11, pp. 53-62, 1999.

[11] A. Osterhues and F. Mariak, “On variants of the k-Chinese postman problem”, Operations Research und Wirtschaftsinformatik, Universitat Dortmund, No. 30, 2005.

[12] G. Ulusoy, “The fleet size and mix problem for capacitated arc routing”, European Journal of Operation Research, vol. 22, pp. 329-337, 1985.

[13] A. Hertz, M. Mittaz, “Heuristic algorithms”. In: Dror M., editor. Arc routing. Theory, solutions and applications. Boston, Kluwer Academic Publishers, pp. 327-386, 2000.

[14] A. Hertz, G. Laporte, G. and M. Mittaz, “A tabu search heuristic for the capacitated arc routing problem”, Operations Research, vol. 48, pp. 129–135, 2000.

[15] P. Lacomme, C. Prins and W. Ramdane-Chérif, “Competitive memetic algorithms for arc routing problems”, Annals of Operations Research, vol. 131, pp. 159–85, 2004.

[16] CPLEX Optimization Inc., CPLEX Version 11.0, 2007.

[17] Eskisehir Osmangazi University, Artificial Intelligence & Robotic Laboratory, www.ai-robotlab.ogu.edu.tr, 2008.