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

Ahmet Yazici, Gokhan Kirlik, Osman Parlaktuna, and Aydin Sipahioglu

***Abstract*—Multirobot sensor-based coverage path planning determines a tour for each robot in a team such that every point in a given workspace is covered by at least one robot using its sensors. In sensor-based coverage of narrow spaces, i.e., obstacles lie within the sensor range, a generalized Voronoi diagram (GVD)-based graph can be used to model the environment. A complete sensor-based coverage path plan for the robot team can be obtained by using the capacitated arc routing problem solution methods on the GVD-based graph. Unlike capacitated arc routing problem, sensor-based coverage problem requires to consider two types of edge demands. Therefore, modified Ulusoy algorithm is used to obtain mobile robot tours by taking into account two different energy consumption cases during sensor-based coverage. However, due to the partially unknown nature of the environment, the robots may encounter obstacles on their tours. This requires a replanning process that considers the remaining energy capacities and the current positions of the robots. In this paper, the modified Ulusoy algorithm is extended to incorporate this dynamic planning problem. A dynamic path-planning approach is proposed for multirobot sensor-based coverage of narrow environments by considering the energy capacities of the mobile robots. The approach is tested in a laboratory environment using Pioneer 3-DX mobile robots. Simulations are also conducted for a larger test environment.**

***Index Terms*—Agent-based robot control architecture, capacitated arc routing problem, dynamic replanning, multirobot, sensor-based coverage, Ulusoy algorithm.**

I. Introduction

THE MULTIROBOT sensor-based coverage (MRSBC) path-planning problem can be defined as the construction of a tour for each robot in a team such that every point in a given workspace is covered by at least one robot using its sensors. Some examples of situations exhibiting this problem are landmine detection, foraging, patrolling, and search-rescue [1].

Manuscript received June 16, 2012; revised January 26, 2013; accepted March 7, 2013. Date of publication April 26, 2013; date of current version February 12, 2014. This work was supported in part by the Scientific and Technical Research Council of Turkey under Contract 107E064, Project Title, "Mobile Robot Route Planning for Complete Coverage of Dynamic Indoor Environments." This paper was recommended by Associate Editor R. Selmic.

A. Yazici is with the Department of Computer Engineering, Eskisehir Osmangazi University, Eskisehir 26480, Turkey (e-mail: ayazici@ogu.edu.tr).

G. Kirlik is with the Graduate School of Science and Engineering, Koç University, Istanbul 34450, Turkey (e-mail: gkirlik@ku.edu.tr).

O. Parlaktuna is with the Department of Electrical and Electronics Engineering, Eskisehir Osmangazi University, Eskisehir 26480, Turkey (e-mail: oparlaktuna@gmail.com).

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

Digital Object Identifier 10.1109/TCYB.2013.2253605

In sensor-based coverage of narrow spaces, where obstacles lie within the sensor range, environment can be represented by using generalized Voronoi diagram (GVD)-based graph [2]. For complete sensor-based coverage of the given area, it is sufficient for the robots to pass through the required edges of the graph. Acar *et al.* [2] described a method for complete coverage of an indoor environment using a single robot. If the map is known in advance and the objective function is minimization of the traversed distance of a robot, coverage path plan can be achieved using the rural postman-based path planning approach proposed in [3]. On the other hand, if the energy of a single robot is not sufficient, for complete coverage of the environment, a multirobot team is preferred. In addition, using multiple robots may reduce the time required to complete the coverage task, and using multiple robots enhances the robustness compared to the single-robot case [4].

In the literature, some approaches for multirobot coverage path planning are described [4]–[6]. In [5], the edges of the configuration space and the Voronoi diagram were used to compute paths over the entire area. First, a tour was generated for traversing all the paths, and then appropriate parts of the tour were assigned to each robot according to the cost evaluation. The cost was evaluated in terms of the traversed distance by each robot. However, in that method the energy capacity of robots was not considered during the partitioning of the path among the robots. The approach in [4] presented algorithmic solutions, for the complete coverage path planning problem, using a team of mobile robots. The algorithms used the same planar cell-based decomposition as the Boustrophedon single-robot coverage algorithm but provided extensions to account for the way in which robots covered a single cell, and the way in which robots were allocated among the cells. In that work, the energy capacities of the robots were not considered either. In [6], the mobile robot deployment problem was considered as a specific type of coverage problem. The deployment problem was described as a determination of the number of groups, the number of robots in each group, and the initial location of those robots. Both the timing and energy constraints of the robots were considered. However, this paper only considers simple rectangular scan lines as the coverage route, and solves the deployment problem using a smaller number of robots in vast environments. The deployment was not considered for partially unknown and narrow environments.

Constructing a sensor-based coverage tour for each robot in the team by considering the energy capacity of each robot is a challenging problem. This problem can also be described as partitioning the edges of the graph of the environment among the robots without exceeding their energy capacities [7]. This problem resembles the capacitated arc routing problem (CARP). The CARP is a problem that aims to construct tours for vehicles to minimize the total traversed distance without exceeding the energy capacity of the vehicles [8]. The vehicle capacity in the CARP corresponds to the energy capacity of a robot in the coverage problem. In the CARP, the amount of material to be collected (or distributed) on an edge is defined as the edge demand. However, in the coverage problem, two types of edge demands occur depending on the robots energy consumption. The robot passes through an edge with or without covering. In the case of covering an edge, robot consumes energy both for the travel and the coverage at the same time. In the case of no coverage, robot only consumes energy for the traverse. These cases require different energy consumption for any edge in the graph. This is the main difference between the CARP and the coverage problem. Therefore, solution techniques for CARP cannot be used directly for the sensor-based coverage problem with multiple robots. In [7], two types of edge demands were considered by modifying Ulusoy partitioning algorithm [9] that was developed originally for CARP. The modified Ulusoy algorithm (MUA) constructs coverage routes for a robot team assuming that all robots start from the same depot with equal energy capacities. Due to the partially unknown nature of the environment, any robot in the team may face a blockage during the execution of the planned paths. In realization of this case, the robots may be at different location with different energy capacities. Therefore, the MUA cannot be used directly for the problem, and this requires another rapid replanning algorithm.

In this paper, the MUA is extended to solve the dynamic path-planning approach for multirobot sensor-based coverage (DPP-MRSBC) with the energy capacities taken into account. Some preliminary results of the proposed algorithm are given in [10] with simulations. Unlike the study in [10], the proposed approach considers the following extensions: 1) open tour (the tour has different start and end points), 2) multidepot (there is more than one depot), and 3) heterogeneous fleet (each robot energy capacity may differ from one another). In CARP literature, some studies consider multidepot, heterogeneous fleet, and open tour cases separately. However, to the best knowledge of the authors, there is no study that includes these three cases together. In addition, the proposed approach includes these three cases together with different edge demands related to the sensor-based coverage problem.

Golden and Wong [8] showed that even obtaining the 1.5-approximate solution for any CARP instance is $\mathcal{NP}$-hard. DPP-MRSBC considers traversing demand with alternative depots and heterogeneous fleet. Hence, DPP-MRSBC is more general form of CARP, i.e., CARP is special case of the DPP-MRSBC problem. This implies that DPP-MRSBC is also $\mathcal{NP}$-hard.

The remainder of the paper is arranged as follows. Sensor-based coverage problem and solution algorithms are described in Section II. In Section III, the proposed approach is described. To demonstrate the effectiveness of the proposed approach, experiments in a laboratory environment and simulations with a larger network are described in Section IV. Conclusion and discussions are provided in the final section.

II. Sensor-Based Coverage Problem and Solution Approaches

Sensor-based coverage aims to have every point in a given workspace covered by a sensor of a robot. There are many methods [2] that can be used for sensor-based coverage of vast environments, where the sensor range is small compared to the size of the environment. However, if narrow spaces are present, such that the detector's range is limited by the presence of obstacles (e.g. most of the indoor environments), these methods are not applicable or do not produce an efficient path [2]. In this case, the environment can be represented with a graph that is constructed by GVD. In the GVD-based graph approach, visiting all the edges of this graph would be sufficient for complete sensor-based coverage of the environment. For the efficient sensor-based coverage, redundant edge visits should be minimized. Then the problem turns into the Chinese postman problem (CPP).

The CPP is defined as the determination of minimum cost tour that visits each edge at least once. The CPP is solved in polynomial time by using a minimum perfect matching (MPM) algorithm [11]. However, if some edges of the graph need not to be visited, e.g., sensor-based coverage is not required, then the CPP turns into the rural postman problem (RPP) [12]. Unlike the CPP, the RPP is in the class of $\mathcal{NP}$-hard problems [13]. In the RPP, the edges are divided into two groups: required edges (the edges that need to be serviced) and nonrequired edges. The partial graph that is induced from required edges of the original graph is called required subgraph. If the required subgraph is connected, the RPP tour can be obtained optimally by means of CPP solution method. However, the required subgraph may be disconnected, then the Frederickson heuristic [14] can be used to determine the 1.5-approximate RPP tour. These methods can find efficient coverage routes for single robots. However, the energy capacity of a single robot may not be sufficient for complete sensor-based coverage of the given environment. Multiple robots may solve this problem and enhance the robustness compared to the single-robot case. In addition, the use of multiple robots may reduce the time required to complete the coverage task.

The MRSBC problem turns into visiting required edges of the graph by a robot team. Similar to the CARP, in the MRSBC problem there are more than one robot having certain capacities and the aim is to obtain the minimum total distance without exceeding capacity constraint for each robot. Since the CARP is an $\mathcal{NP}$-hard problem, several heuristic algorithms were developed in the literature, simple constructive methods and two-phase constructive methods are explained comprehensively in [15]. Together with constructive heuristics, various metaheuristic approaches have been proposed so far. One of the most recent metaheuristic is a hybrid ant-colony optimization algorithm, which was presented in [16] for the extended version of the CARP, where the total service time and fixed investment cost were considered. A comprehensive CARP survey was presented in [17] for both CARP solution approaches (lower bounds, exact methods, and meta-heuristics) and its variations. Although many studies refer the CARP solution methods, they cannot be applied directly to solve the MRSBC of narrow environments.

In the MRSBC problem, two types of edge demands are defined that are traversing energy and task-performing energy (coverage energy). When the robot passes through a nonrequired edge, there is no need to consume coverage energy for this edge, and the edge demand is determined only by the traversing energy, which arises from motors, embedded computer, microcontroller card, and navigation sensors (sonar). On the other hand, in a required edge, a robot consumes energy for both traversing and performs its coverage task. Namely, a robot consumes additional energy for covering the environment with its coverage sensor (e.g., camera, laser range finder, thermal sensor, etc.). The required edge demand is calculated by adding the traversing energy and the coverage energy. In CARP, the demand is defined only for the required edges that corresponds to consumed energy during the coverage task in the MRSBC problem. However, traversing energy type of demand is undefined in CARP. Due to this difference, the Ulusoy partitioning algorithm was modified to solve the multirobot sensor-based coverage problem with two different edge energy demands [7]. Nevertheless, the MUA also cannot be applied directly to the partially unknown environments because of the uncertain nature of the environment. For example, an open corridor into the the map may be blocked after a period of time, and one of the robots may face it assuming the path is clear. This situation requires us to consider the DPP-MRSBC problem for narrow environments.

In the DPP-MRSBC problem, the robots may be at different locations with different energy capacities, and the new tours should be constructed quickly. Moreover, any robot in the team finishes its tour at one of the alternative depots, where the alternative depots represent the charging points of the environment. The planning procedure, that determines the new tours, has to consider positions, capacities of the robots, and alternative depots of the environment. Hence, a new algorithm, which is is named as dynamic modified Ulusoy algorithm (DMUA), is proposed to handle these cases. The preliminary results of this algorithm were described in [10]. In our setting, it is assumed that the robot team is homogeneous, i.e., all robots have the same abilities and equipments. Nevertheless, energy level of robots may be different at any planning time. It is also assumed that all robots move with a constant velocity during the task. Both the MUA and the DMUA are the extensions of Ulusoy's heuristic [9]. The details of the DMUA are provided in the following section.

III. Proposed Approach

Let an undirected graph $G = (V, E)$ be a GVD-based graph of the environment, where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of vertices, E is a set of edges, and $R \subseteq E$ is set of required edges. Each edge incurs positive edge cost ($c_{ij} > 0, \forall(v_i, v_j) \in E$) and two types of edge demands that are traversing energy consumption for each edge, $e_{ij} > 0\ \forall(v_i, v_j) \in E$, and sensor-coverage energy for all required edges, $q_{ij} > 0\ \forall(v_i, v_j) \in R$. Robot team contains m homogeneous robots with different initial energy capacities, Q_k for $k = 1, \ldots, m$. The kth robot in the team starts its tour at $v_c^k \in V$ and completes the tour at one of the alternative depots $v \in V_d$, where V_d represents set of alternative depots, $V_d \subseteq V$. DPP-MRSBC problem aims to find the minimum total cost robot tours that serve all required edges R without exceeding capacity Q_k. Depending on the initial and the terminal vertices of the robot tour, the tour may be open or closed.

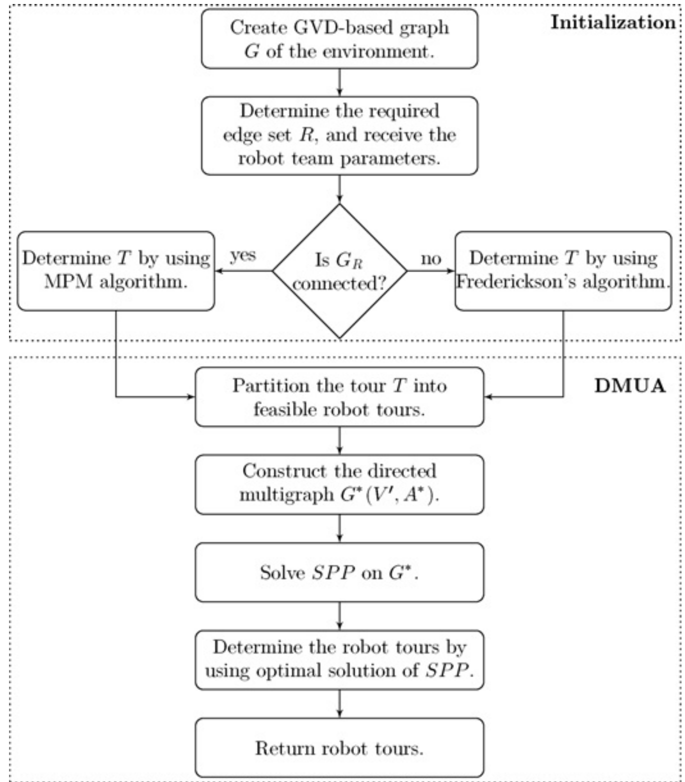


Fig. 1. Flowchart of the proposed method to solve DPP-MRSBC problem.

DPP-MRSBC is an $\mathcal{NP}$-hard problem, therefore, a heuristic algorithm DMUA is proposed to solve the problem. The flow chart of the proposed method is given in Fig. 1. An initialization phase is required to prepare the inputs of the DMUA. In this phase, firstly the graph $G = (V, E)$ of the environment is constructed based on GVD, and the required edge set R is determined by using regions that need to be covered into the environment. The other parameters of the DMUA are related to the robot team, i.e., the set of alternative depots (V_d), the number of robots, the starting vertices, and the initial energy capacities of robots. By using these information, an Eulerian tour T is constructed that covers R. Depending on R, the solution procedure is changing. If the subgraph G_R, that is induced from R is connected, the optimal tour T is obtained by using MPM algorithm [11] and Hierholzer's algorithm [18]. If G_R is disconnected, then Frederickson's approximation algorithm [14] is used to obtain T. The initialization phase is followed by the DMUA to construct the robot tours.

Several functions are needed to construct robot tours in the DMUA. These functions incur some basic calculations that are shortest path, length calculation, and total load calculation. Shortest path function represented with $\mathtt{SP}(v_i, v_j)$. This function determines the shortest path between the vertices v_i and v_j. Length calculation function represented with $\mathtt{L}(X)$. This function calculates the length of the X, where X could be a complete tour, tour partition, or path. Final procedure is total load calculation procedure that is represented with $\mathtt{TotalLoad}(T)$. This function takes a tour T as an input, and calculates total traversing and coverage energy on T. The steps of the DMUA are given in Algorithm 1.

Initially, each tour partition in T is turned into robot tour, where the tour starts with the current vertex of the kth robot (v_c^k) and ends with one of the alternative depots (V_d). After the tour is constructed, the total load on this tour is calculated by using the TotalLoad procedure. If the total load is less than or equal to the capacity of kth robot (Q_k), tour is feasible, then this tour is represented as an arc (directed edge) in G'_k, where the cost of the arc is the length of the tour. These steps of the algorithm are repeated for each robot in the team. Then a directed multigraph G^* is constructed by using G'_k for $k = 1, \ldots, m$. While vertex set of G^* and G'_k, for $k = 1, \ldots, m$ are same, the arc set of G^* is the combination of m different directed graphs' arc set, i.e., $A^* = A_1 \cup A_2 \cup \ldots \cup A_m$. Since $G^*(V', A^*)$ is constructed by using m different graphs, any arc in A^* is represented with $(v_i, v_j)_k$ notation, where $k \in \{1, \ldots, m\}$, $v_i, v_j \in V'$, and $1 \le i < j \le r$.

After directed multigraph G^* is constructed, the shortest path problem with a side constraint is solved on G^* via a mathematical model. Unlike well-known form of the shortest path problem, our shortest path problem formulation includes set packing constraint that guarantees at most one edge is used from each $k \in \{1, \ldots, m\}$. A directed multigraph (G^*) with source (v_s) and terminal (v_t) vertices are the parameters of the mathematical model. The decision variable x_{ijk} is binary that determines whether arc $(v_i, v_j)_k$ is used or not on the path. Along with these definitions the shortest path model $SPP(G^*, v_s, v_t)$ is given below

$$\min \sum_{k=1}^{m} \sum_{\forall (v_i, v_j)_k \in A^*} c_{ijk} x_{ijk} \tag{1}$$

$$\text{s.t.} \sum_{k=1}^{m} \sum_{\substack{\forall v_j \in V' \\ (v_i, v_j)_k \in A^*}} x_{ijk} = 1 \quad v_i = v_s \tag{2}$$

$$\sum_{k=1}^{m} \sum_{\substack{\forall v_i \in V' \\ (v_i, v_j)_k \in A^*}} x_{ijk} = 1 \quad v_j = v_t \tag{3}$$

$$\sum_{k=1}^{m} \left(\sum_{\substack{\forall v_i \in V' \\ (v_i, v_p)_k \in A^*}} x_{ipk} - \sum_{\substack{\forall v_j \in V' \\ (v_p, v_j)_k \in A^*}} x_{pjk} \right) = 0 \tag{4}$$

$$\forall v_p \in V' \tag{5}$$

$$\sum_{\forall (v_i, v_j)_k \in A^*} x_{ijk} \le 1 \quad k = 1, \ldots, m \tag{6}$$

$$x_{ijk} \in \{0, 1\} \quad \forall (v_i, v_j)_k \in A^*. \tag{7}$$

In this model, the objective function (1) minimizes the distance between the source and the terminal vertex. Constraint (2) and (3) ensure that there is exactly one leaving arc from the source vertex and there is exactly one arrival to the terminal vertex. Constraint (4) ensures the flow conservation at each vertex. Constraint (6) is additional side constraint to the shortest path problem, which ensures that at most one arc should be chosen for each robot. Equation (7) is binary integrality constraint. In this paper, the MIP-solver GAMS/CPLEX [19] is used to solve the model.

In solution of this mathematical model, at most m decision variables can take positive value due to (6). Hence, required number of robots (m') to cover R is less than or equal to number of robots ($m' \le m$). In case of $m' < m$, at least one robot is unnecessary to complete the coverage task. In the proposed algorithm, the optimal solution of the model is the minimum-cost path from v_1 to v_r on G^*. All edges on the path correspond to a robot tour, and these tours are obtained by using Step 11–14 in Algorithm 1. Basically, DMUA partitions the Euler tour T into feasible robot tours considering their energy capacities. Then the algorithm selects some of these tours (T_k for $k = 1, \ldots, m'$) that serve R with the minimum total cost.

The DMUA considers the open tour, multidepot, and heterogeneous fleet cases that arise in the DPP-MRSBC problem. In addition, the DMUA incurs different energy consumption cases. In the following section, the results of the application of the proposed method on real robots are presented.

IV. Applications

Solution of the DPP-MRSBC problem on mobile robots require integration of the DMUA into the robot control architecture. Therefore, the DMUA is coded and integrated to an open-agent-based robot control architecture that allows interface among robot behaviors, human, and the DMUA. Experiments and simulations are conducted to show the effectiveness of the proposed approach. Then experiments were conducted in a small laboratory environment using Pioneer 3-DX (P3-DX) mobile robots. Simulations are conducted for a larger test environment.

A. Robot Control Architecture

Mobile robot applications require the coordination of several modules such as perception, localization, low-level control, planning, and human robot interface. Depending on the application, different robot control architectures are used to coordinate these modules [20]. In this paper, agent-based control architecture is proposed for the multirobot sensor-based coverage path planning and execution. The agents in the robot control architecture are fully interconnected. It allows several features such as reusability of developed modules without modification and resource sharing among the robots in the group. It also permits a more natural integration of the human into the robotic system. Therefore, for a human–robot integrated application the use of a multiagent framework seems more appropriate than deliberative, reactive, or hybrid robotic architectures.

Algorithm 1 Dynamic Modified Ulusoy algorithm (DMUA)

Input : Graph of the environment $G = (V, E)$, required edge set R, Eulerian tour T that covers R, alternative depots set V_d, number of robots m, starting vertices v_c^k and initial energy capacities Q_k of robots for $k = 1, \ldots, m$.
Output: Robot tours T_k.

1 Relabel the vertices on $T = \{v_1, v_2, \ldots, v_t = v_1\}$ where v_1 is the depot.
2 Let r be the largest index of a vertex incident to a serviced arc on T.
3 Construct a directed multigraph $G^* = (V', A^*)$ where $V' = \{v_1, v_2, \ldots, v_r\}$ and $A^* = \emptyset$.
4 **for** $k = 1$ **to** m **do**
5   Construct a directed graph $G'_k = (V', A_k)$ where $V' = \{v_1, v_2, \ldots, v_r\}$ and $A_k = \emptyset$.
6   Introduce (v_i, v_j) into A_k for $i, j = \{1, 2, \ldots, r\}$ and $i < j$.
7   Remove (v_i, v_j) from A_k such that (v_i, v_{i+1}) or (v_{j-1}, v_j) is non-serviced edge where $j > i + 1$.
8   **foreach** $((v_i, v_j) \in A_k)$ **do**
9   **if** $((v_i, v_{i+1})$ *is a non-serviced edge)* **then** $d^k_{i,i+1} \leftarrow 0$.
10   **if** $(j > i + 1$ ***or*** (v_i, v_{i+1}) *is a serviced edge)* **then**
11             $P'_{ij} \leftarrow P_{ij}$, $P''_{ij} \leftarrow P_{ij}$.
12   Add to P'_{ij} the shortest path between v_c^k and v_i, and the shortest path between v_j to $v \in V_d$, i.e. $P'_{ij} = \{\mathtt{SP}(v_c^k, v_i), v_i, \ldots, v_j, \mathtt{SP}(v_j, v)\}$.
13   Add to P''_{ij} the shortest path between v_c^k and v_j, and the shortest path between v_i to $v \in V_d$, i.e. $P''_{ij} = \{\mathtt{SP}(v_c^k, v_j), v_j, \ldots, v_i, \mathtt{SP}(v_i, v)\}$.
14   **if** $((\mathtt{L}(P'_{ij}) < (\mathtt{L}(P''_{ij}))$ **then** $P_{ij} \leftarrow P'_{ij}$. **else** $P_{ij} \leftarrow P''_{ij}$.
15         $load \leftarrow \mathtt{TotalLoad}(P_{ij})$
16   **if** $(load \leq Q_k)$ **then** $d^k_{ij} \leftarrow \mathtt{L}(P_{ij})$. **else** $A_k \leftarrow A_k \setminus (v_i, v_j)$.
17     $A^* \leftarrow A^* \cup (v_i, v_j)_k \quad \forall (v_i, v_j) \in A_k$.
18 $x^* \leftarrow \mathtt{SPP}(G^*(V', A^*), v_1, v_r)$.   /* Obtain optimal shortest path from v_1 to v_r on G^*. */
19 $k' \leftarrow 1$
20 **for** $k = 1$ **to** m **do**
21     $(v_a, v_b) \leftarrow \{(v_i, v_j) \in A_k : \; x^*_{ijk} = 1\}$.
22   **if** $((v_a, v_b) \neq \emptyset)$ **then**
23         $T_{k'} \leftarrow$ Apply Step 11-14 with (v_a, v_b).
24         $k' \leftarrow k' + 1$.
25 $m' \leftarrow k'$.
26 **return** T_k for $k = 1, \ldots, m'$.

The architecture in use consists of software agents performing specific tasks required for the overall control of the robot team (Fig. 2). In the system, there are m homogeneous robots. Dashed square in Fig. 2 shows a robot (MR_i, where $i \in \{1, \ldots, m\}$) in the team, and each block inside the dashed square represents a software agent. Hence, the control of mobile robots is performed by coordinated operation of different mechanisms, each agent in the robot may have a different functionality. These functionalities may be communication with other robots, perception, action, negotiation, planning, human interaction, and so on.

In the proposed architecture, each robot has four functional software agents: a user interface agent, planner agent, action agent, and communication agent. The functional details of the agents are defined as follows:

The user interface agent (UIA): The main task of this agent is to provide interaction between the robot and users. It provides a user-friendly environment to define tasks and provide information to the user about the internal state of the robot using a terminal or graphical interface.

The planner agent (PA): This agent is responsible for generating plans, which may include motion planning, path planning, task planning, and so on. It generates plans when either a task is received from the UIA or a new plan is necessary. The plans may be generated for one or more robots. It is assumed that all robots have a GVD-based representative network of the environment. The PA has an ability to construct the path plans for each robot in the team by using the DMUA.

The action agent (AA): The action agent handles perception, localization, and action. This agent is responsible for tasks related to the hardware of robots (actuators and sensors). This agent performs behavior-based motions according to the motion plans provided by the PA. The task-oriented behaviors are implemented in conjunction with survival behaviors, such as obstacle avoidance, by using the subsumption [21] approach. Moreover, the perception module detects any change in the environment using the previously known GVD-based network of the environment. Localization is realized by using ARNL's laser-based localization module in the ARIA software architecture [22]. After the action agent receives the motion plan, it starts to follow the path described by the motion plan.

The communication agent (CA): This agent is responsible for interaction with the other robots. The interaction takes

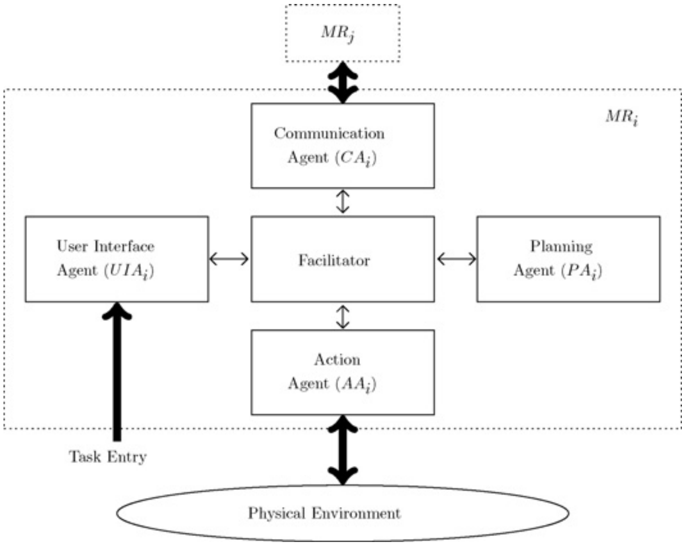


Fig. 2. Overall block diagram of the architecture.

place over the infrastructure realized using TCP/IP. Each robot explicitly communicates with others via the CA. As seen in Fig. 2, MR_i communicates with MR_j through CA_i, where $i, j \in \{1, \dots, m\}$ and $i \neq j$.

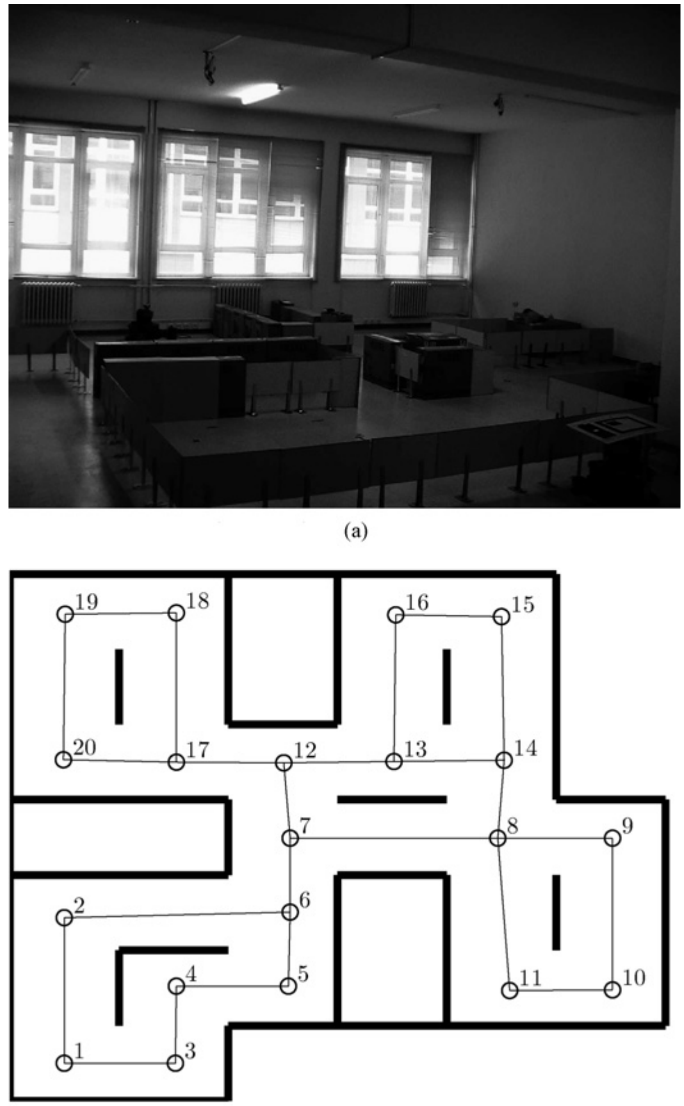
All of the agents are separate computational processes. While the agents perform their regular tasks, they also response to messages received from the other agents. We use open agent architecture (OAA) [23] to facilitate the messaging between the agents. This framework provides content-based message transfer, which enables the agents to receive the relevant messages. The facilitator is a computational process provided by the framework. While the facilitator provides messaging between intra-robot agents communication, another messaging unit is used for communication among the robots. This unit is only responsible for directing the messages to their receivers. Mobile robots communicate with each other via CA over this unit. First, all CAs are registered to this unit with a unique ID and planning priority. Planning priorities are assigned to each robot to avoid planning conflicts. The details of messaging between robots in the proposed architecture are provided with sequence diagrams in [24].

B. Experiments in the Laboratory Environment

A platform at the Eskisehir Osmangazi University (ESOGU) Artificial Intelligence & Robotic Laboratory [25] (Fig. 3(a)) was used as a test bed. This platform was also used to test various coverage algorithms on P3-DX robots. A topological map of this platform with a GVD-based graph is shown in Fig. 3(b). Covering the entire area is achieved by following all the arcs in this figure.

P3-DX robots are used in the applications. These robots have an onboard P3-800 computer with Linux OS. The sensors on the robot are: a SICK LMS laser range finder, sonar sensors, a camera, and a compass. A wireless network is set to communicate with the other robots and with the computers. The robots use the open-agent software architecture as explained above.

During the experiments, as in the problem definition of DPP-MRSBC two types of energy consumption are defined.



(a)

(b)

Fig. 3. Test environment. (a) A photograph of the test environment. (b) Topological map of the test environment.

The first type of consumption is traversing consumption (motor, microcontroller, sonar, embedded computer) that occurs while the robot passes through a nonserviced edge. Given robot velocity $\mathcal{V}$ in m/s, the power model of P3-DX robot was given $P_T(\mathcal{V}) = 17.49 + 7.4\mathcal{V}$ *Watts* [6]. The second one is coverage energy consumption that is observed while the robot performs the coverage task. The coverage was assumed to be achieved by a SICK LMS laser range finder that has sufficient sensing range for narrow indoor environments. The power of the laser range finder is 20*Watts*. Hence, the power model of coverage becomes $Q_C(\mathcal{V}) = 37.49 + 7.4\mathcal{V}$ *Watts*. During the applications, the velocity of the robots is set to $0.4m/s$. With constant velocity ($\mathcal{V}$) assumption the time to traverse an edge $(v_i, v_j) \in E$ is determined by $t_{ij} = c_{ij}/\mathcal{V}$ where c_{ij} is the length of the edge.

In the applications, two alternative depots were used, $V_d = \{v_1, v_{15}\}$. Initially, two mobile robots were assumed at v_1 with 3200 J of energy, i.e., $v_c^k = v_1$ and $Q_k = 3200$ J for $k = 1, 2$. Each robot in the team had a planning priority such that

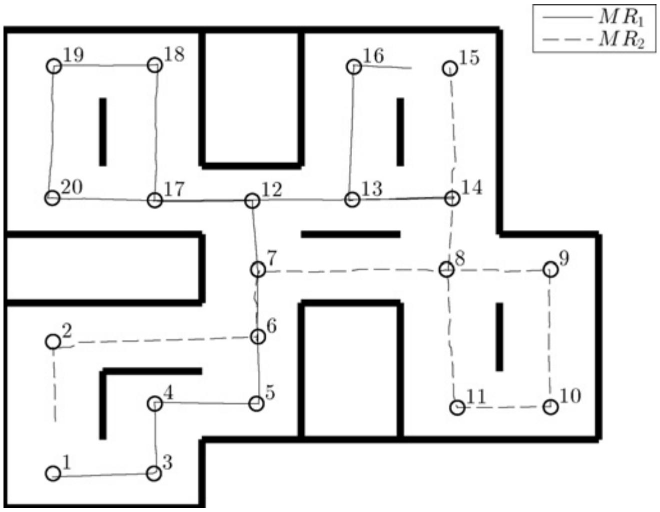


Fig. 4. Traces of the robots for complete sensor-based coverage.

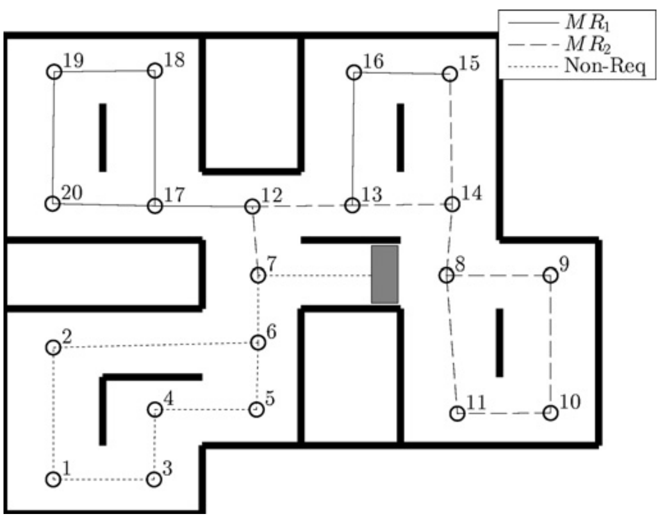


Fig. 5. Division of the edges among two robots in changing environment.

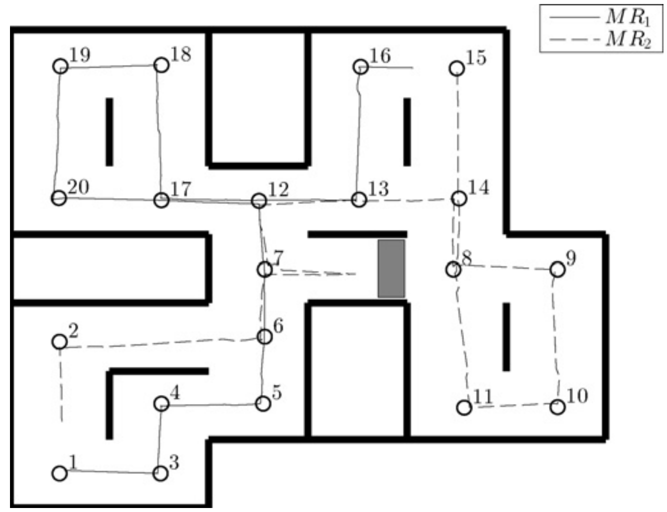


Fig. 6. Traces of the robots for complete sensor-based coverage in changing environment.

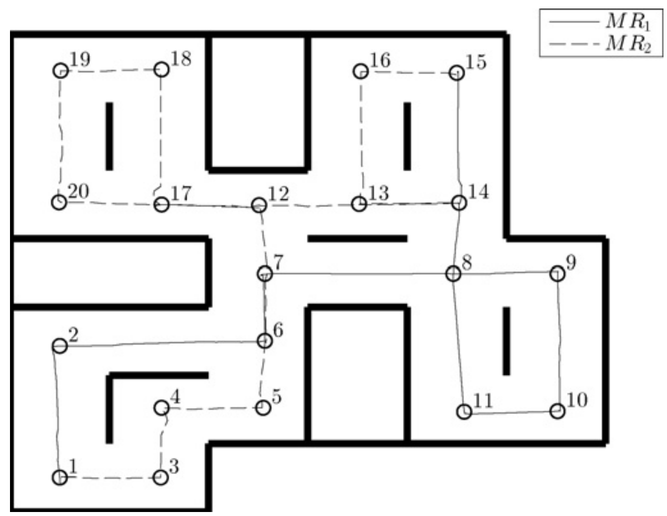


Fig. 7. Traces of the robots for complete sensor-based coverage when robots are in different initial start vertices and have different energy capacities.

highest-priority robot generates new plans in dynamic case. In these applications, MR_1 had higher planning priority than MR_2. The user enters the mission input to MR_2's user interface agent that is UIA_2 cover(A1) via the user terminal. Then PA_2 generates the path plans for each robot. The proposed method generated the paths as follows. Planned path for MR_1: v_1, v_3, v_4, v_5, v_6, v_7, v_{12}, v_{17}, v_{20}, v_{19}, v_{18}, v_{17}, v_{12}, v_{13}, v_{14}, v_{13}, v_{16}, v_{15}. Planned path for MR_2: v_1, v_2, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_8, v_{14}, v_{15}. In these tours, MR_1 covered $(v_1 - v_3)$, $(v_3 - v_4)$, $(v_4 - v_5)$, $(v_5 - v_6)$, $(v_7 - v_{12})$, $(v_{12} - v_{13})$, $(v_{12} - v_{17})$, $(v_{13} - v_{14})$, $(v_{13} - v_{16})$, $(v_{15} - v_{16})$, $(v_{17} - v_{18})$, $(v_{17} - v_{20})$ $(v_{18} - v_{19})$, and $(v_{19} - v_{20})$, MR_2 covered $(v_1 - v_2)$, $(v_2 - v_6)$, $(v_6 - v_7)$, $(v_7 - v_8)$, $(v_8 - v_9)$, $(v_8 - v_{11})$, $(v_8 - v_{14})$, $(v_9 - v_{10})$, $(v_{10} - v_{11})$, and $(v_{14} - v_{15})$. The robots perform the coverage task while passing through the serviced edges and the remaining edges for traverse without coverage.

If both robots finished their tours without facing an obstacle, robots MR_1 and MR_2 would consume 2237.79 J and 1925.15 J of energy over tour lengths of 23.89m and 19.03m, respectively. During the application, the (x, y) positions of each robot are logged. Fig. 4 shows the traces of both robots. It is possible to view the video of this application from [25].

In the second application, a blockage is placed between v_7 and v_8 to test the proposed approach in a changing environment. This application starts as in the first application, but MR_2 detects an obstacle between v_7 and v_8 at the 50^{th} second via AA_2. At the instant MR_2 has detected the obstacle, MR_1 is between v_6 and v_7. Up to this time, MR_1 has covered $(v_1 - v_3)$, $(v_3 - v_4)$, $(v_4 - v_5)$, $(v_5 - v_6)$, and MR_2 has covered $(v_1 - v_2)$, $(v_2 - v_6)$, $(v_6 - v_7)$, $(v_7 - v_8)$. The remaining uncovered edges are $(v_7 - v_{12})$, $(v_8 - v_9)$, $(v_8 - v_{11})$, $(v_8 - v_{14})$, $(v_9 - v_{10})$, $(v_{10} - v_{11})$, $(v_{12} - v_{13})$, $(v_{12} - v_{17})$, $(v_{13} - v_{14})$, $(v_{13} - v_{16})$, $(v_{14} - v_{15})$, $(v_{15} - v_{16})$, $(v_{17} - v_{18})$, $(v_{17} - v_{20})$, $(v_{18} - v_{19})$, and $(v_{19} - v_{20})$. In addition, MR_1 and MR_2 have consumed 835.46 and 551.82 J and the remaining energy capacities are 2364.54 and 2648.18 J, respectively.

After AA_2 detects the obstacle, it stops the robot and informs PA_2 about the blockage. Then PA_2 determines the end points of the blocked edge and updates the representative

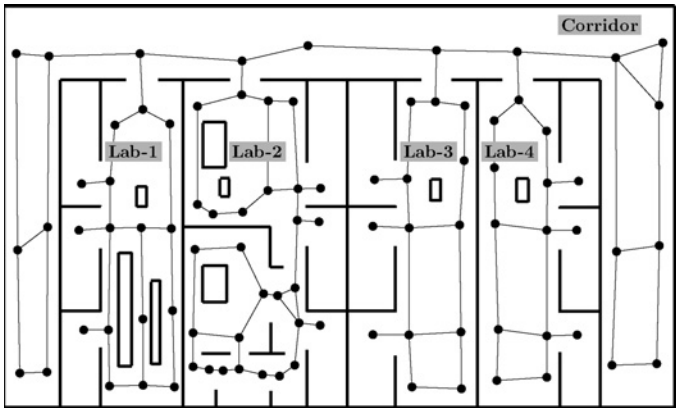


Fig. 8. GVD-based graph of the test environment with large number of vertices.

graph. Since MR_1 is the highest-priority robot in the team, PA_1 generates the path plans for each robot with the current vertices ($v_c^k = v_7$ for $k = 1, 2$), remaining energy capacities ($Q_1 = 2364.54$ J, $Q_2 = 2648.18$ J), alternative depots ($V_d = \{v_1, v_{15}\}$), and nonserviced edge list parameters by using the proposed planning approach. The resulting robot path plans are as follows. Planned path for MR_1: v_7, v_{12}, v_{17}, v_{20}, v_{19}, v_{18}, v_{17}, v_{12}, v_{13}, v_{16}, v_{15}. Planned path for MR_2: v_7, v_{12}, v_{13}, v_{14}, v_8, v_9, v_{10}, v_{11}, v_8, v_{14}, v_{15}. The edge partitions among the robots in the changing environment experiment are shown in Fig. 5.

After replanning, MR_1 and MR_2 consumed 1381.12 J and 1519 J of their remaining energies and the tour lengths were 15.43m and 15.64m, respectively. Traces of the real robots for complete coverage of the given environment are shown in Fig. 6.

In the third application, the robots are initially placed at different vertices with different energy capacities. MR_1 was placed at v_{15} with an energy of 3000 J and MR_2 was placed at v_1 with an energy of 3200 J. The path plans were generated by using the proposed planning approach with the current vertices ($v_c^1 = v_{15}$, $v_c^2 = v_1$), robot energy capacities ($Q_1 = 3000$ J, $Q_2 = 3200$ J) and alternative depots ($V_d = \{v_1, v_{15}\}$) parameters. The resulting paths are as follows. Planned path for MR_1: v_{15}, v_{14}, v_8, v_{11}, v_{10}, v_9, v_8, v_7, v_6, v_2, v_1. Planned path for MR_2: v_1, v_3, v_4, v_5, v_6, v_7, v_{12}, v_{17}, v_{20}, v_{19}, v_{18}, v_{17}, v_{12}, v_{13}, v_{14}, v_{13}, v_{16}, v_{15}. On these paths, while MR_1 covers $(v_1 - v_2)$, $(v_2 - v_6)$, $(v_6 - v_7)$, $(v_7 - v_8)$, $(v_8 - v_9)$, $(v_8 - v_{11})$, $(v_8 - v_{14})$, $(v_9 - v_{10})$, $(v_{10} - v_{11})$, and $(v_{14} - v_{15})$. MR_2 covers $(v_1 - v_3)$, $(v_3 - v_4)$, $(v_4 - v_5)$, $(v_5 - v_6)$, $(v_7 - v_{12})$, $(v_{12} - v_{13})$, $(v_{12} - v_{17})$, $(v_{13} - v_{14})$, $(v_{13} - v_{16})$, $(v_{15} - v_{16})$, $(v_{17} - v_{18})$, $(v_{17} - v_{20})$, $(v_{18} - v_{19})$, and $(v_{19} - v_{20})$.

For these paths, MR_1 and MR_2 consumed 1925.15 J and 2237.79 J of energy and the tour lengths were 19.03m and 23.89m, respectively. The traces of the real robots for complete coverage of the given environment are shown in Fig. 7.

C. Test of the Algorithm on a Larger Graph

To test the proposed algorithm on an environment with large number of vertices, the first floor of the Eskisehir Osmangazi University Electrical Engineering laboratory building was used as the test bed. In this floor, there are four laboratories. Inside each laboratory, there are three rooms and some tables, storage cabinets, and columns. A corridor connects the laboratories. The topological map of the first floor, and the corresponding GVD network is shown in Fig. 8. The number of vertices in the graph is 90.

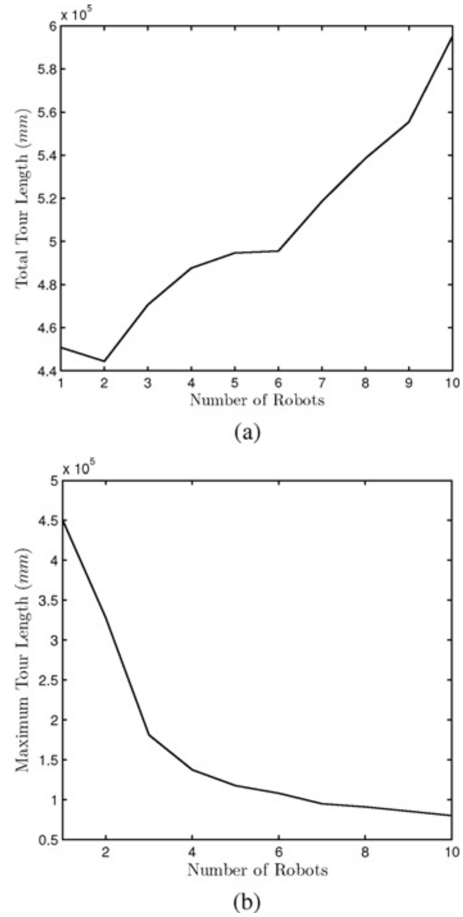


Fig. 9. Tour length results for increasing number of robots. (a) Total tour length versus number of robots. (b) Maximum tour length versus the number of robots.

The corresponding network, shown in Fig. 8, is transferred to the MobileSim environment [26] that can be simulate the P3-DX mobile robot with the proposed robot control architecture. Simulations were performed using the proposed approach for up to ten robots. Table I shows the results of the proposed planning for the larger test environment. The first column of this table shows the number of robots (m). The second column stands for the initial energy capacity of each robot (Q). The third and the fourth columns represent total length ($\sum_{k=1}^{m} TL_k$) and total energy consumption ($\sum_{k=1}^{m} CE_k$), respectively. The fifth and the sixth columns indicate tour length and energy consumption per robot. The maximum tour length and maximum energy consumption are shown in the seventh and the eighth columns. The last column denotes the CPU time of the proposed algorithm in seconds. Five different randomly selected robot initial locations were used to determine the results for each number of robots. Then the average of these results is used to construct each row in Table I. The solution time increases linearly with increase in the number of robots. The average and maximum consumed energy decrease as the number of robots increases.

TABLE I
THE RESULTS OF THE PROPOSED PLANNING FOR LARGER TEST ENVIRONMENT

m	Q	$\sum_{k=1}^{m} TL_k$	$\sum_{k=1}^{m} CE_k$	$\sum_{k=1}^{m} TL_k/m$	$\sum_{k=1}^{m} CE_k/m$	$\max_{k=1,\ldots,m} TL_k$	$\max_{k=1,\ldots,m} CE_k$	CPU Time (s)
1	40000	450788.2	37800.2	450788.2	37800.2	450788.2	37800.2	1.2
2	30000	444410.4	37474.1	222205.0	18737.0	328141.0	27089.0	1.6
3	15000	470669.2	38816.6	156889.6	12938.9	181016.2	14806.4	1.9
4	11000	487665.0	39685.5	121916.0	9921.4	137289.0	10911.1	2.2
5	9000	494679.2	40044.1	98935.8	8008.8	117438.4	8806.0	2.5
6	8000	495512.2	40086.7	82585.4	6681.1	107890.5	7951.8	2.8
7	7000	518544.6	41264.2	74077.8	5894.9	94775.0	6931.5	3.1
8	6000	538532.4	42286.1	67316.5	5285.8	90784.5	5943.8	3.4
9	5500	555408.4	43148.9	61712.1	4794.3	85385.1	5446.2	3.8
10	5100	595034.4	45174.8	59503.4	4517.5	79954.5	5067.9	4.3

As seen from Fig. 9(a), the total tour length increases as the number of robots increase. The increment of the total tour length is almost linear if the number of robots is greater than six. Fig. 9(b) shows the maximum tour length taken by any robot in the team. Note that, under constant travel velocity assumption, the maximum tour length shows the time span of coverage for the robot team. The maximum tour length decreases exponentially up to six robots, and remains almost constant if the number of robots greater than six. Therefore, using seven or more robots for the coverage would not be meaningful. This type of analysis can be used to determine the maximum number of robots to assign for a given coverage task.

V. CONCLUSION

In this paper, a new approach based on the CARP was proposed and applied to the multirobot dynamic sensor-based coverage planning for indoor environments. The partially unknown nature of the environment was accounted, for by adding a replanning algorithm, which was not achievable using the classical CARP solution algorithms. The proposed algorithm was developed by modifying Ulusoy's partitioning algorithm. Agent-based robot control architecture was developed to implement the proposed algorithm, and experiments in the laboratory and simulation for large environments were conducted.

The proposed method constructs the sensor-based coverage paths that account for robot energy capacities both in known and partially unknown environments. This approach could also be used to determine the maximum number of robots to assign for a given coverage task for efficient coverage, which was very important for resource allocation. The proposed method could also be used in the case of robot failures. A dynamic replanning with existing robot locations and energies should be sufficient to handle for the failure.

Another important contribution of this paper was that the proposed algorithm was flexible and could be used in other robot application problems. For example, it could be extended to consider the time-constraint sensor-based mobile robot coverage problem.

REFERENCES

[1] C. Trevai, J. Ota, and T. Arai, "Multiple mobile robot surveillance in unknown environments," *Adv. Robot.*, vol. 21, no. 7, pp. 729–749, 2007.

[2] E. U. Acar, H. Choset, and J. Y. Lee, "Sensor-based coverage with extended range detectors," *IEEE Trans. Robot.*, vol. 22, no. 1, pp. 189–198, Feb. 2006.

[3] A. Sipahioglu, A. Yazici, G. Kirlik, and O. Parlaktuna, "A sensor-based coverage approach in dynamic environments," in *Proc. 6th Int. Symp. Intell. Manuf. Syst.*, 2008, pp. 502–509.

[4] I. Rekleitis, A. P. New, E. S. Rankin, and H. Choset, "Efficient boustrophedon multi-robot coverage: An algorithmic approach," *Ann. Math. Artif. Intell.*, vol. 52, no. 2, pp. 109–142, 2008.

[5] D. Kurabayashi, J. Ota, T. Arai, and E. Yoshida, "Cooperative sweeping by multiple mobile robots," in *Proc. IEEE Int. Conf. Robot. Automation*, vol. 2. Apr. 1996, pp. 1744–1749.

[6] Y. Mei, Y. H. Lu, Y. C. Hu, and C. S. G. Lee, "Deployment of mobile robots with energy and timing constraints," *IEEE Trans. Robot.*, vol. 22, no. 3, pp. 507–522, Jun. 2006.

[7] A. Sipahioglu, G. Kirlik, O. Parlaktuna, and A. Yazici, "Energy constrained multi-robot sensor-based coverage path planning using capacitated arc routing approach," *Robot. Autonomous Syst.*, vol. 58, no. 5, pp. 529–538, 2010.

[8] B. L. Golden and R. T. Wong, "Capacitated arc routing problems," *Networks*, vol. 11, no. 3, pp. 305–315, 1981.

[9] G. Ulusoy, "The fleet size and mix problem for capacitated arc routing," *Eur. J. Oper. Res.*, vol. 22, no. 3, pp. 329–337, 1985.

[10] A. Yazici, G. Kirlik, O. Parlaktuna, and A. Sipahioglu, "A dynamic path planning approach for multi-robot sensor-based coverage considering energy constraints," in *Proc. Int. Conf. Intell. Robots Syst.*, 2009, pp. 5930–5935.

[11] J. Edmonds and E. L. Johnson, "Matching, euler tours and the chinese postman," *Math. Programming*, vol. 5, no. 1, pp. 88–124, 1973.

[12] H. A. Eiselt, M. Gendreau, and G. Laporte, "Arc routing problems, part II: The rural postman problem," *Oper. Res.*, vol. 43, no. 3, pp. 399–414, 1995.

[13] J. K. Lenstra and A. H. G. Kan, "On general routing problems," *Networks*, vol. 6, no. 3, pp. 273–280, 1976.

[14] G. N. Frederickson, "Approximation algorithms for some postman problems," *J. ACM*, vol. 26, no. 3, pp. 538–554, 1979.

[15] A. Hertz and M. Mittaz, *Heuristic Algorithms*. Boston, MA, USA: Springer, 2000, ch. 9, pp. 327–386.

[16] L. N. Xing, P. Rohlfshagen, Y. W. Chen, and X. Yao, "A hybrid ant colony optimization algorithm for the extended capacitated arc routing problem," *IEEE Trans. Syst. Man Cybern., B, Cybern.*, vol. 41, no. 99, pp. 1–14, Aug. 2011.

[17] S. Wøhlk, "A decade of capacitated arc routing," *The Vehicle Routing Problem: Latest Advances and New Challenges* (Operations Research/Computer Science Interfaces, vol. 43), B. Golden, S. Raghavan, and E. Wasil, Eds. Berlin, Germany: Springer, 2008, pp. 29–48.

[18] J. L. Gross and J. Yellen, *Graph Theory and Its Applications*. Boca Raton, FL, USA: CRC press, 2006.

[19] Cplex, *IBM ILOG Cplex Optimization Studio 12.1*, 2012.

[20] B. Siciliano and O. Khatib, *Handbook of Robotics*. Berlin, Germany: Springer-Verlag, 2008.

[21] R. Murphy, *Introduction to AI Robotics*. Cambridge, MA, USA: The MIT Press, 2000.

[22] ARIA. (2009). *MobileRobots' Advanced Robot Interface for Applications* [Online]. Available: http://robots.mobilerobots.com/wiki/ARIA

[23] D. L. Martin, A. J. Cheyer, and D. B. Moran, "The open agent architecture: A framework for building distributed software systems," *Appl. Artif. Intell.*, vol. 13, nos. 1–2, pp. 91–128, 1999.
[24] M. Ozkan, G. Kirlik, O. Parlaktuna, A. Yufka, and A. Yazici, "A multi-robot control architecture for fault-tolerant sensor-based coverage," *Int. J. Adv. Robot. Syst.*, vol. 7, no. 1, 2010.
[25] ESOGU AIRLAB. (2009). Eskisehir Osmangazi University Artificial Intelligence Robotic Laboratory [Online]. Available: www.ai-robotlab.ogu.edu.tr
[26] MobileSim. (2009). *MobileRobots' MobileSim* [Online]. Available: http://robots.mobilerobots.com/wiki/MobileSim

Ahmet Yazici received the B.S. degree from Osmangazi University, Eskisehir, Turkey, in 1998, and the M.S. and Ph.D. degrees in control systems in 2000 and 2005.

He was a member of the OSU-ACT Team in 2007 Darpa Urban Challenge. He has been with the Computer Engineering Department, Eskisehir Osmangazi University since 2005. He is also a co-founder of the Osmangazi University Artifcial Intelligence and Robotic laboratory. His current research interests include control theory, artificial intelligence, mobile robots, and intelligent vehicles.

Gokhan Kirlik received the B.Sc. degree in industrial engineering, the second B.Sc. degree in computer engineering, and the M.Sc. degree in industrial engineering from Eskisehir Osmangazi University, Eskisehir, Turkey, in 2007, 2008, and 2009, respectively. He is currently pursuing the Ph.D. degree in industrial engineering and operation management program at Koç University, Istanbul, Turkey.

His current research interests include multiobjective optimization, integer programming, scheduling, and heuristic methods.

He was a recipient of the IBM Ph.D. Fellowship Award in 2013.

Osman Parlaktuna received the B.S. and M.S. degrees in electrical engineering from Middle East Technical University, Ankara, Turkey, and the Ph.D. degree from Vanderbilt University, Nashville, TN, USA.

He is currently a Professor of electrical-electronics engineering at Eskisehir Osmangazi University, Eskisehir, Turkey. His current research interests include mobile robotics, automation, and control.

Aydin Sipahioglu received the Ph.D. degree from the Industrial Engineering Department, Eskisehir Osmangazi University, Eskisehir, Turkey.

He is currently an Associate Professor at the Industrial Engineering Department, Eskisehir Osmangazi University. His current research interests include integer programming (especially routing problems), logistics management, and project management.