Robotics and Autonomous Systems 60 (2012) 123–132

Contents lists available at SciVerse ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Multi-robot exploration and terrain coverage in an unknown environment

K.S. Senthilkumar *, K.K. Bharadwaj

School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi 110067, India

ARTICLE INFO

Article history:
Received 18 December 2010
Received in revised form
11 September 2011
Accepted 20 September 2011
Available online 29 September 2011

Keywords:
Multi-robot exploration
Terrain coverage
Spanning tree coverage
Cellular decomposition

ABSTRACT

Terrain exploration and coverage are required for a variety of applications such as mine clearing, intrusion detection and other humanitarian missions like search and rescue operations, for example, fire or blast in a building. During an emergency situation within a building it is crucial to explore the area as fast as possible in order to search and find the wounded people and other hazards. On account of the prevailing breakdown of communication in indoor environments in some situations, it is suggested that the robots can communicate indirectly with the use of markings in the environment. The Spanning Tree Coverage (STC) method, proposed for this problem, suffers in environments that have partially occupied cells and narrow door openings between rooms. In this paper, we consider an extension of the Simultaneous Multiple STC (S-MSTC) algorithm, which we proposed in our previous work on multiple autonomous robots used in exploration and coverage in an unknown terrain. The proposed extended S-MSTC (ES-MSTC) uses ant-type robots to cover the terrain leaving marks on the terrain, which can be sensed by the robots and allow them to cover the terrain, similar to the nature of ants. This algorithm can handle partially occupied cells and narrow door openings in the terrain and performs a complete coverage of the surface regardless of the shape of the environment by constructing multiple spanning trees simultaneously. We present a simulation study and compare the performance of the ES-MSTC algorithm with other existing algorithms.

1. Introduction

These days the Terrain Coverage problem is receiving considerable attention because of the current interest in competitive algorithms for Autonomous Robot systems with incomplete data. The terrain coverage algorithms have been studied well in the robotics literature [1–5], including on-line and off-line algorithms, and single-robot and multi-robot algorithms [6,7]. The Terrain Exploration and Coverage problem can be defined as a mobile robot, with an attached tool, visiting each and every location in a continuous bounded unknown terrain by avoiding obstacles to perform a task. Most of the approaches divide the environment into grid cells that are explored one cell at a time until the whole area is covered. Complete coverage planning for autonomous mobile robots guarantees that the robot's detector passes over all reachable points in the target terrain [8]. Exploration is a fundamental problem for multi-robots, and requires the team to acquire as much information about the terrain as possible within the shortest amount of time or distance traveled and limiting repeated coverage.

The so-called Spanning Tree Coverage (STC) technique is used to solve the single-robot coverage problem optimistically [9]. It is proved that STC is a polynomial time coverage algorithm that divides the terrain into grid cells, creates a spanning tree and the robot circumnavigates around it. The same idea was first applied using multiple robots by Hazon and Kaminka [10]. An ant-type algorithm which uses a sensor network infrastructure to provide robots with information about the visited areas and direct them to the least recently visited direction is proposed in [11]. Ant-type robots are simple robots with limited sensing and computational capabilities. The work on coverage using ant-type robots, which leave pheromone-like markers in the environment, has urged interest in the complex coverage [12]. It is shown that while there are several remarkable research attempts which have been made for coordination of multiple robots, much work yet remains to realize multiple robots' full potential. And also, there is not enough research being done addressing the effects of communication constraints between robots on co-operative terrain exploration and coverage for an unknown environment.

A terrain coverage algorithm must generate a coverage path, which is a chain of motion steps for a robot that sweeps the given sensor over a given terrain. An optimal coverage algorithm would return a coverage path with minimum time, and guarantee to cover the entire terrain and perform the task efficiently.

* Correspondence to: Department of Computer Science, College of Arts & Science, Wadi Al Dawasir, P.O. Box 54, King Saud University, Riyadh 11991, Saudi Arabia. Tel.: +966 17841841, +966 534554821 (Mob.).

E-mail addresses: ksskumar16@yahoo.com (K.S. Senthilkumar), kbharadwaj@gmail.com (K.K. Bharadwaj).

URL: http://faculty.ksu.edu.sa/ssomasundaram/default.aspx (K.S. Senthilkumar).

0921-8890/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2011.09.005

Many coverage applications utilize on-line coverage algorithms and the robots cannot depend on a priori knowledge of the terrain, and construct their movement trajectories step-by-step, by avoiding obstacles in their path. In our previous work, we proposed an algorithm [13], which is a variant of the TSP and finds shortest paths for multiple robots in the environment using genetic algorithms. In this work, we try to reduce the repeat coverage (defined as any robot covers previously covered space), by which we achieve an optimal solution. When multiple robots cover the terrain simultaneously, any reduction in the repeat coverage increases the performance of the task. In this study, we tried to address these important questions by considering recent research studies. We study the problem of multi-robot exploration and coverage with limited communication using ant-type robots. In this proposed extension study, we use heuristics such as obstacle avoidance, partially occupied cell coverage, and narrow opening area movements, which are simple rules-of-thumb that work well in different situations. Furthermore, our ant-type robots do not need to know the map of the terrain and the terrain can change over time. To achieve a complete coverage, we focus on an on-line multiple ant-type robot coverage algorithm based on an approximate cellular decomposition. That is the terrain is decomposed into equal square-sized cells, using a technique known as cellular decomposition. Provably complete and optimal coverage is achieved by ensuring that multiple robots simultaneously visit each cell in the decomposed terrain only once. Our algorithm (ES-MSTC) effectively coordinates the robots in cooperative covering by placing the makers on the covered environment.

The remainder of the paper is structured as follows: Section 2 discusses the background of the coverage problem and its related work, the proposed ES-MSTC scheme for multi-robot coverage is discussed in Section 3; Section 4 contains the simulation results from a variety of environments and analysis of the coverage algorithm and finally, Section 5 presents conclusions and some directions for future work.

2. Related work

In this section, we present a brief overview of a variety of techniques that are used to solve the terrain exploration and coverage problem using both a single robot and multiple robots. Gabriely and Rimon have worked on a coverage approach for single robots that provides optimal paths in a grid-like representation of the terrain [9]. The algorithm is called Spanning Tree Covering (STC), which subdivides the work-area into disjoint cells and then makes a spanning tree. They developed three versions of the STC algorithm. The first version is off-line, where the robot has perfect a priori knowledge of its environment. The second version of STC is on-line, where the robot uses its on-board sensors to detect obstacles and construct a spanning tree of the environment while covering the work-area. The third version of STC is an ant-like type. In this version, too, the robot does not have knowledge of the environment, but it may leave pheromone-like markers during the coverage process. STC was generalized to Multi-Robot Spanning Tree Coverage (MSTC) recently, to make an optimal polynomial-time multi-robot coverage [10]. This off-line MSTC algorithm first computes the same spanning tree as STC using the initial locations of the robots, and considers the tour that circumnavigates the spanning tree. Each robot follows the tour segment clockwise ahead of it. To improve the cover time, the longest segment is evenly divided between the two adjacent robots and only the backtracking version of the algorithm guaranties the optimal solution.

Another spanning tree construction algorithm using multi-robots which is based on approximate cellular decomposition is proposed in [14]. After creating multiple spanning trees, they tried to connect them to make one optimal tree which minimizes the time. Batalin et al. propose two multi-robot algorithms, which spread the robots in the terrain and to make the robots move away from each other when they are within sensing range [8]. In their work, they planned to achieve optimal terrain coverage, but failed to prove optimality of the coverage time. This algorithm is strictly off-line whereas our algorithm is an on-line one. Hazon et al. consider on the robustness of the solution, so that even if only one robot remains in operation, it will be able to carry on and complete the exploration task [15]. An algorithm that explores an area in forward and reverse phases using multiple robots and gradually builds a graph of the environment is proposed in [16]. Here the robots always know the uncovered cells and its position in the terrain and therefore they can position themselves in the area. Even though this algorithm gives an efficient coverage time, they need perfect wireless communication, which may be impossible in some hazardous situations. Howard et al. present an approach to explore a building and find objects and report them back to people outside [17]. This algorithm is not fully autonomous, because it requires some human support to solve problems like loop closures or merging of maps between the robots. Wagner et al. [12] have proposed discrete coverage algorithms for multiple robots using pheromone-like markers to guide the coverage process and also used to do the coordination between several robots. The continuous covering problem has been studied by Arkin et al., and it is also strictly off-line [18].

Several authors proposed multi-robot exploration strategies based on market principles, in which robots place bids on subtasks of the exploration attempt [4,5]. These bids are typically based on values such as expected information gain and travel cost to a particular location in the environment or terrain. Bids may be assigned in a distributed fashion among team members, or by a controlling agent. Zlot et al. presented a distributed approach for coordination using a market-based approach that requires no central agent [5]. The market architecture seeks to maximize the benefit while minimizing the cost. In this approach, each robot applies one or more of a set of goal point generation heuristics to produce a list of target points to visit. At startup and subsequently upon completion of a task, a robot generates new goal points and inserts them into its schedule until it reaches a predefined maximum length. Robots periodically call single-award multi-task auctions to relieve themselves of any goals that they may have generated which may be handled by another teammate in a better way. In Zlot's market economy strategy, the multi-robot system does not rely on perfect communication and it can still be carried out if some of the colony members lose communications. When strength of communication is included into the bid calculation, the robots avoid areas outside of communication range. For multi-robot teams with constraints on communication range, coordination under a limited communication situation must be taken into account. This approach does not rely on maintaining consistent maps and thus can run in more communication-limited situations.

Frontier-based, distributed bidding models for coordination of multiple robots were proposed in [19,20]. Similar to market-based methods, utilities of individual frontiers may include a factor related to the chance of communication success, so robots are less likely to explore areas that take them out of the team communication range. In [20], Sheng et al. essentially suggest several modifications and improvements to the approach of Simmons et al. One modification is that they use a distributed auction protocol that does not require a central auctioneer or map. In this algorithm, the robots attempt to reach their goals, and upon task completion a robot determines the highest-utility goal from a list of frontiers. The limited communication is considered in two ways in Sheng's work. Firstly, robots are guided to stay close to each other by considering the distances between robots. Secondly, a new coding method is developed for map representation that reduced the volume of exchanged data. Based on simulation results, it appears that including the nearness factor may improve exploration time slightly. Since there is no central auctioneer, the robots try to maintain consistent map information in order to reduce repeated coverage. There is much that still needs to be done in terms of providing a flexible market-based multi-robot coordination approach. Market-based multi-robot coordination approaches have only been implemented and tested in a few application domains to date.

Many of the multi-robot coverage algorithms using ant robots were also proposed in the past and they all interacted and planned only locally [21,7,12]. The ant-type algorithm divides the area into square grid cells on which the exploring agents leave traces of their passage, similar to real ants leaving pheromones [22]. An ant robot cannot use conventional planning or coordination methods due to their limited sensing and computational capabilities which limit their planning capabilities even for simple planning tasks such as path planning or the coverage of terrain. Thus, they might not be able to cover terrain as efficiently as robots with more powerful sensing and computational capabilities. On the other hand, groups of ant robots can take advantage of both their fault tolerance and their parallelism. A single robot algorithm was proposed in [23], in which there is no obstacle avoidance procedure. If an unexpected obstacle comes across the path, the robot stops and waits for the obstacle to move, whereas in our algorithm we have obstacle avoidance procedure to avoid an obstacle and move. The algorithms that compute spanning trees have discontinuous jumps between nodes. But in our algorithm, there is no jump and it will be a continuous graph. Real-time search methods have been developed using Artificial Intelligence techniques as an alternative to more traditional search methods [24,25,21,26]. The experiments show that real-time search methods robustly cover terrain even if the ant robots are moved without realizing that, some ant robots fail, and some markings get destroyed. Here the ant-type robots leave markings such as heat traces, or alcohol traces in the terrain, sense the markings at their neighboring locations, and change the marking at their current location. These results demonstrate that terrain coverage with real-time search methods is an interesting alternative to conventional terrain coverage methods.

In our previous work [27], we considered the problem of covering a continuous bounded planar terrain by multiple robots based on the STC algorithm. All of these works emphasized the importance of achieving on-line coverage based on the robot's sensors. We studied on-line multiple ant-like robots, which can take the advantages of both their fault tolerance and parallelism. The algorithm builds multiple spanning trees corresponding to multiple robots simultaneously using the depth first search (DFS) technique and named as the Simultaneous Multiple Spanning Trees Construction (S-MSTC) algorithm. We assumed that the robots are positioned in the initial positions ($S_1 \dots S_k$) in the terrain. The challenge is to find k spanning trees for k robots simultaneously such that once all the robots complete making their spanning trees, the entire work-area is covered. We assume that robots are homogeneous and all cells in the terrain are accessible from any starting cell. The execution of the algorithm is completely decentralized, as each robot executes its own independent copy concurrently. Starting from the respective initial position, each robot independently constructs its spanning tree and moves along the path which circumnavigates its spanning tree along a counterclockwise direction. Collision is a common problem with multiple robots. As we implemented obstacle avoidance heuristics in our approach, and there is no spanning tree which crosses itself or other spanning trees, there is no collision with

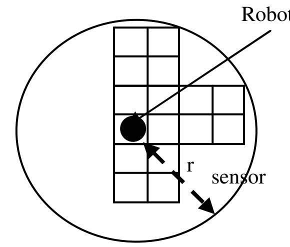


Fig. 1. Robot and sensor range.

other robots. The algorithm strived to embed a Hamiltonian cycle for each robot, having the largest number of cells in the given terrain. The resulting Hamiltonian cycle (union of k-spanning trees) is automatically an optimal covering path in terms of path length, since the tool covers every cell exactly once. In STC and MSTC algorithms, the construction of a spanning tree is done first and later it is divided up into sections, and each section is assigned to one robot. Hence, the output travel cost was unbalanced. But in our approach, spanning trees are created in parallel by each robot, and hence we overcame the travel cost unbalanced problem. We used the cellular decomposition method to represent the environment. This algorithm covers an environment whose geometry is not a priori known to the robots. The proposed simultaneous on-line coverage strategy for multi-robots is structured and it assures complete, optimum and robust coverage of the surface regardless of the shape of the unknown environment. The robots coordinate themselves through the marks on the terrain. The S-MSTC algorithm does not consider an area with partially occupied cells and narrow openings in the terrain.

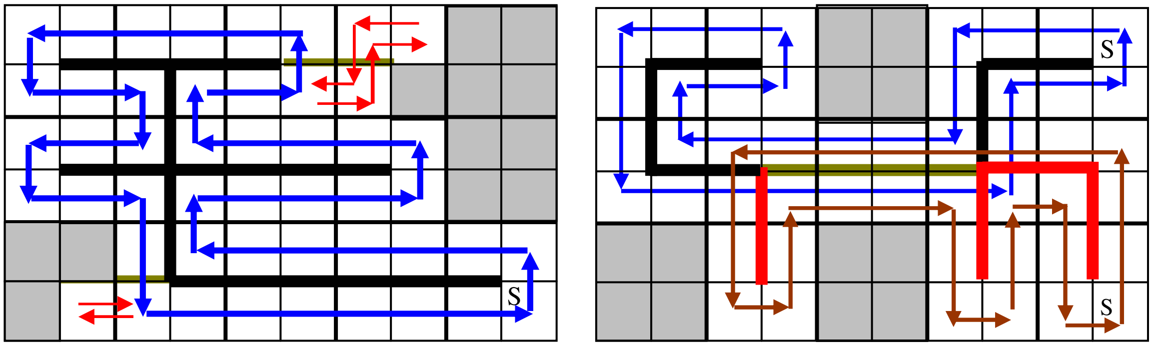
3. Proposed extended S-MSTC (ES-MSTC) scheme

In this paper, we present an on-line algorithm that covers every point of the bounded terrain by multiple robots for tasks such as lawn mowing, field de-mining and humanitarian missions. We study multiple ant-type robots, which can take advantage of both their fault tolerance and their parallelism. The goal of this algorithm is that all the ant-type robots must completely cover the obstacle free location near to them simultaneously, while minimizing a metric such as travel distance. In our previous work in [27], we proposed an algorithm based on STC, called Simultaneous Multiple Spanning Trees Construction (S-MSTC) and here we extend our work (ES-MSTC) to cover partially occupied cells and to share narrow openings in the terrain by multiple robots so that the work load will be shared among the robots almost equally.

3.1. The model and assumptions

The robots have no prior knowledge about the terrain except the initial position and must use its on-board sensors to detect the obstacles and the markings and construct their own spanning tree for the entire terrain. The terrain is divided into square shaped grid cells and the size of the cell is determined by the sensor range (r_{sensor}) of the robot as depicted in Fig. 1. When a robot is in the center of the sub-cell, it must be able to cover the entire area with its sensor. The size D must be equal to $2(r_{\text{sensor}})/7$. We also assume that the robot's sensor is capable of identifying obstacles in the four orthogonal neighboring cells to the robot's current cell. During the spanning tree construction, the robot subdivides every cell it encounters into four identical sub-cells of size D, each being identical to the tool size.

The robots are homogeneous and all cells in the terrain are accessible from any starting cell. It is to be noted that the execution



(a) Partially occupied cell.

(b) Narrow opening area.

Fig. 2. Two different situations in a terrain.

of the algorithm is completely decentralized, as each robot executes its own independent copy concurrently. Starting from the respective initial robot's positions, each robot independently constructs its spanning tree and moves along the path which circumnavigates their spanning tree along a counterclockwise direction. Each robot's initial cell location is given as the input to the algorithm, and output is the spanning tree of that robot. The robot scans orthogonal neighbors (up/down, left/right) in a counterclockwise direction starting from its parent cell. Also we assume that when this is implemented physically the robot has the ability to leave markers in the sub-cells it covers and there is a detection device in the physical robot, capable of inspecting the sub-cell markers in the current cell and its four orthogonal neighbors. To keep track of the marks created by robots in the terrain, here we use the matrix called visited, in our simulator.

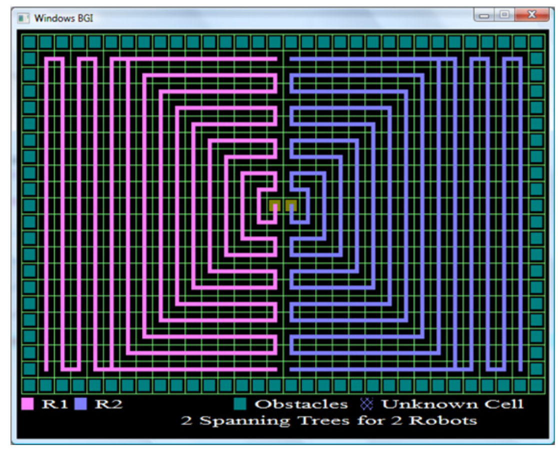


Fig. 3. Two spanning trees created by two robots.

3.2. ES-MSTC scheme

We try to explore a hazardous terrain by building multiple spanning trees by multiple robots simultaneously using the Depth First Search (DFS) technique. Since this is an on line algorithm, it is impossible to add weights to edges to make a true optimal solution (minimum spanning tree). Partially occupied cells are also considered for making a spanning tree in the ES-MSTC algorithm and the robot is forced to travel on one side of the tree by visiting the free sub-cells in partially occupied cells. And when a narrow opening area appears in the terrain, all the robots are allowed to visit those cells if it is needed to cross that narrow-opening area. By allowing repetition of coverage on those cells only, we try to balance the work load of the robots. Fig. 2 depicts the two situations, where one side of the tree is traveled to cover the sub-cells in a partially occupied cell and the narrow-opening area is crossed by two robots. In the narrow-opening case there is a possibility of repeat visits of sub-cells only. We assume that the robots are positioned in the initial positions ($S_{1...}S_k$) in the terrain which is known to the robots. Once all the robots complete their spanning trees, we can join all the trees to make the complete spanning tree for the entire work-area covered.

In the improved version of the algorithm, we categorized the cells as:

1. Unknown Cell — to be explored by a robot.
2. Explored Cell — it is explored, but allows other robots to visit this cell again if needed.
3. Visited Cell — it is explored and not allowed to be visited again.
4. Occupied Cell — cell occupied by obstacle or boundary (Wall).

We use different color marks for each category to uniquely identify them in the terrain. The *explored cell* category is used to allow other robots to revisit the cell, which is in the narrow opening area. In the S-MSTC algorithm, the robot creates the spanning tree by following a manner depicted in Fig. 3 that is thickening the visited cells boundaries by visiting their close neighboring cells first. This technique eliminates the blocking of the unexplored area to one robot by another robot.

When the robot enters a cell, it puts a mark on the cell as either visited or explored. It marks the current cell as visited, if the current cell does not block the path between two neighboring cells. Otherwise the current cell is marked as explored. This kind of situation may happen when one cell divides the two accessible regions or accessing the sub-cells in a partially occupied cell. The flow chart of the ES-MSTC algorithm is given below in Fig. 4.

In the ES-MSTC algorithm we add the following extensions:

- Partially occupied Cells are considered to make complete terrain coverage.
- Spanning tree is created for partially occupied cells, so that the free sub-cells in the partially occupied cells can be visited.
- The cells in between the two orthogonal occupied cells (considered as narrow paths) are marked as *explored*, not marked as *visited*.This operation will help robots to make optimal equal sized spanning trees and remove deadlock situations.

The *explored cell* category is introduced to allow the robot to enter the partially occupied cells and also to enter the cells in the narrow entrances to rooms. Sometimes a large room area and a small area are connected by a narrow door or corridor. In these kinds of situations, more than one robot needs to enter through the narrow door or corridor to complete the exploration task without one robot idling, leaving more work to another robot. If one robot fails, the other robots take the responsibility to cover the remaining terrain. We analyze the algorithm by proving completeness, robustness and optimality. The circumnavigating the spanning tree produces a closed curve (Hamiltonian cycle) which visits all the sub-cells exactly once.

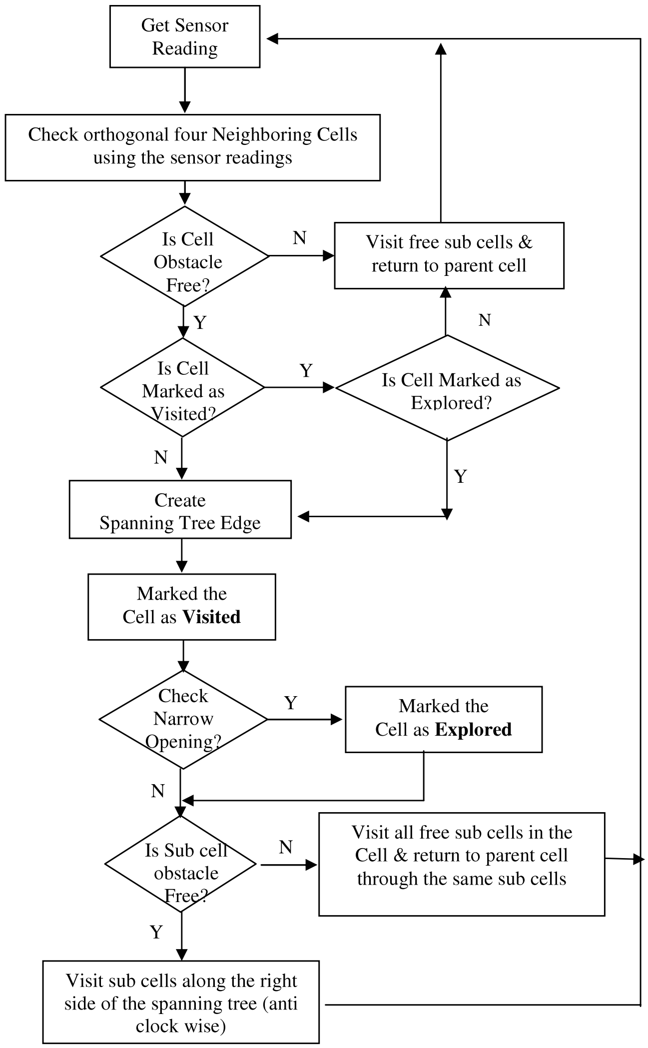


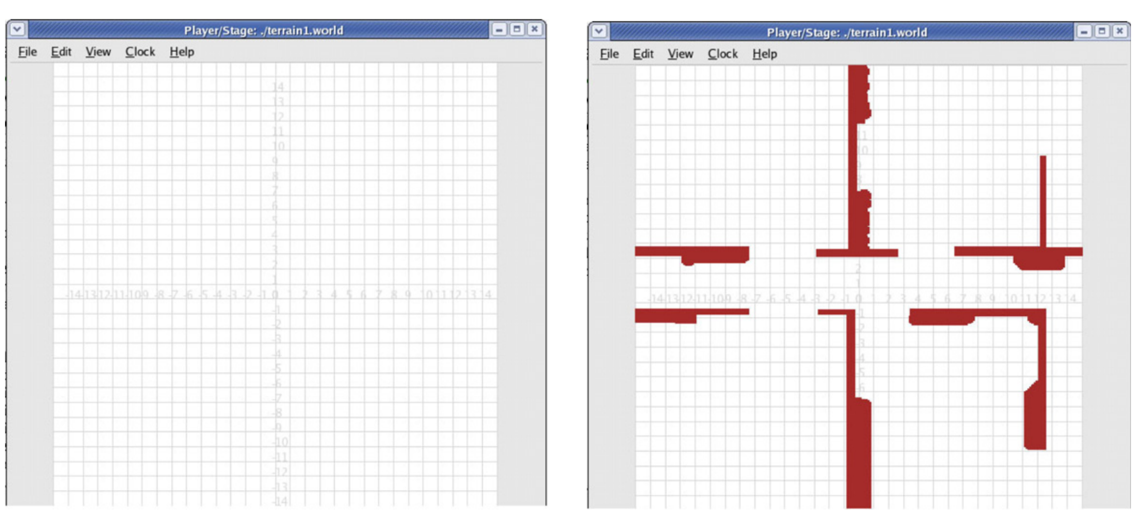
Fig. 4. ES-MSTC algorithm flow chart.

4. Simulation results

In this section, we provide simulation results and screen shots of the coverage paths of the robots. The dimension of the terrain is a 30 × 30 grid map as shown in Fig. 5 and has several obstacles to pose a challenge for the proposed method. The ant-type robots could move to each of the four orthogonal neighboring cells of their current cell provided that the next cell is unoccupied, unvisited and unexplored by any robot or explored and visited by other robots. Cells are not traversable if they contain either walls or obstacles. The terrain is transferred to the player/stage simulation environment and the results of the proposed method are compared to an existing grid based method.

An algorithm has been implemented using the C++ programming language in the Linux operating system. The ant-type robots do not even need to communicate with each other directly, but can communicate indirectly only through the markings. They do not need to know or learn the spanning tree in which they are operating. They only have to sense the markings at their neighboring cells and change the marking of their current cell. Robots were assigned initial positions and permitted to find the individual spanning tree concurrently. The spanning trees reflect the sensing ability of the robots and the shape of the terrain. We compared the simulation results of previous algorithms (MSTC, S-MSTC), with the result of the proposed new algorithm (ES-MSTC), which explore and cover the partially occupied cells for four robots. The simulation outputs of ES-MSTC and S-MSTC are depicted in Fig. 6. In Fig. 7, we display the simulation output for the terrain containing a narrow opening area. When we use four robots to explore the terrain with narrow openings, the S-MSTC algorithm could not equally distribute the exploration task between robots due to the blockage created by one robot with another. But after we include the heuristic to check for narrow openings in the terrain in the ES-MSTC algorithm, all the robots are able to explore the terrain almost equally. That helps us to balance the work load among the robots. We intend to prove that the ES-MSTC algorithm is complete, non-redundant and robust. This algorithm is complete if k spanning trees are created by k robots, so that the union of all k spanning trees covers the terrain completely. Non-redundancy is achieved by showing that none of the robot covers the same place more than once. Then we have to show that the terrain coverage is down by at least one robot, and then robustness is achieved.

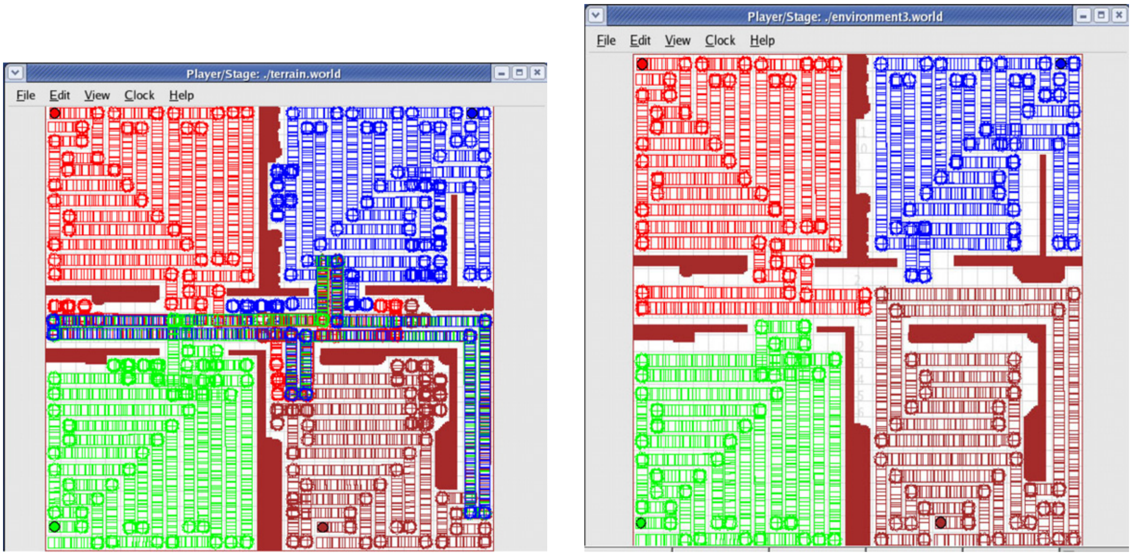
Completeness. Our approach generates k spanning trees that together cover every cell accessible from the respective starting cells ($S_1 \dots S_k$). Each robot visits each and every free cell that can be covered by it and makes its own spanning tree. If there are k robots then there are k spanning trees. Finally we join the entire spanning trees created by k-robots, and this gives the one complete spanning tree, which covers the whole terrain. Completeness is achieved by



(a) Obstacle free environment.

(b) Environment with partially occupied cells and narrow openings.

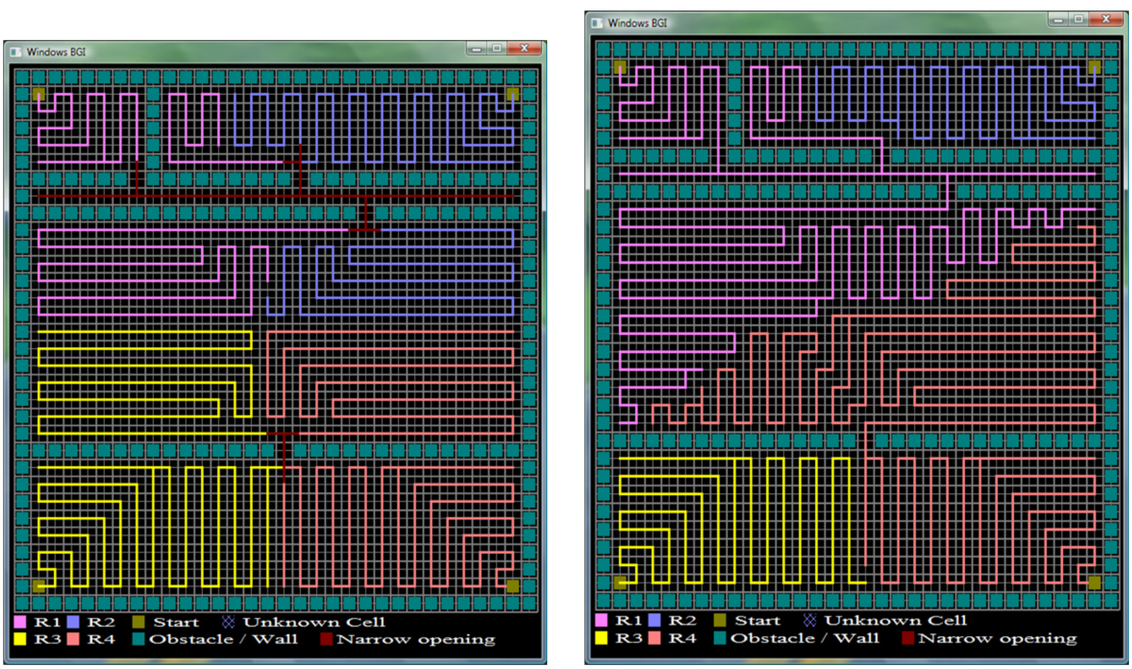
Fig. 5. Different types of terrain used for exploration and coverage.



(a) Partial cells are visited (ES-MSTC).

(b) Partial cells are not visited (S-MSTC).

Fig. 6. Snap shots of the simulation output for partially occupied cells.



(a) Proposed ES-MSTC algorithm.

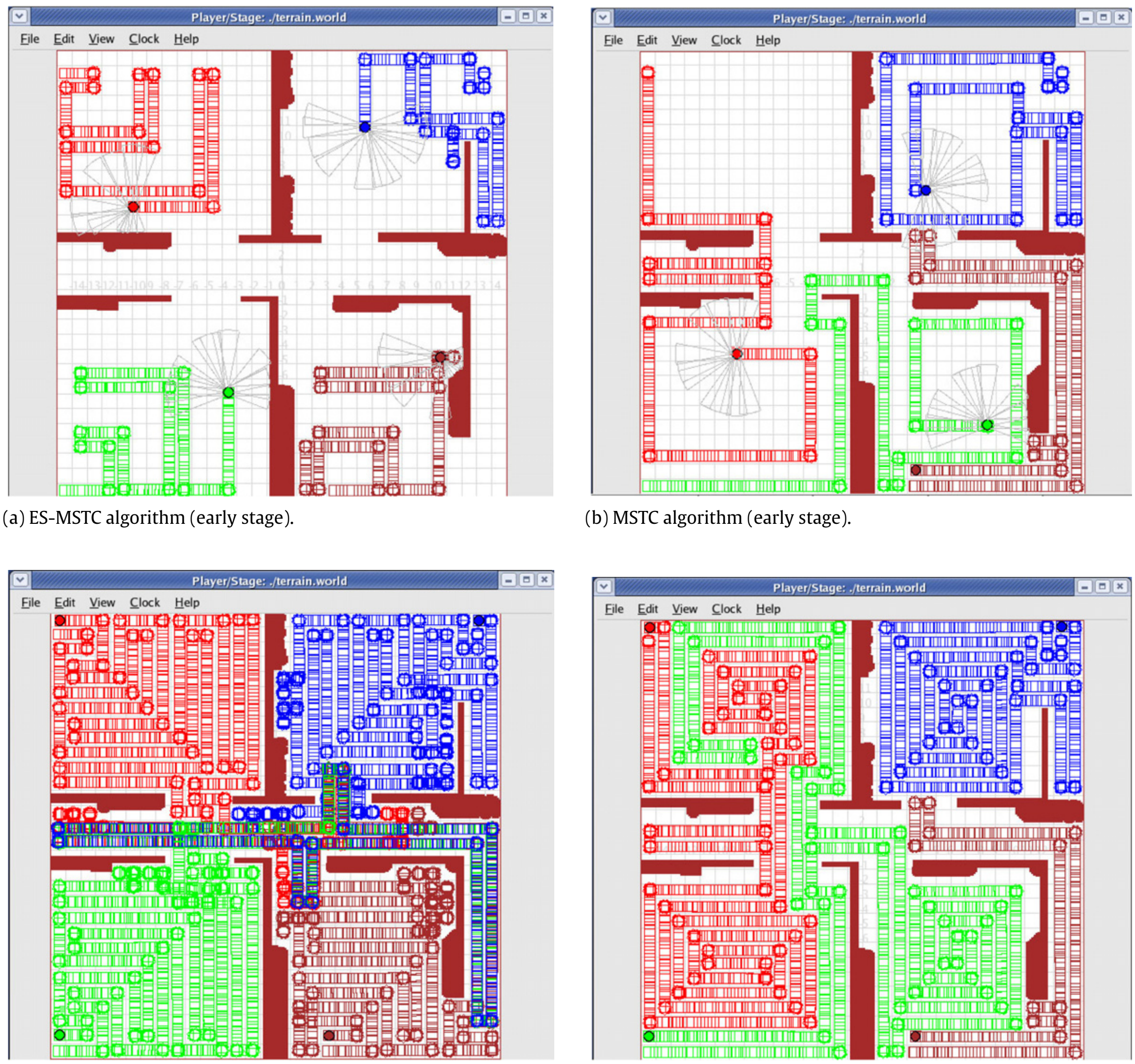
(b) Previous S-MSTC algorithm.

Fig. 7. Snap shot of the simulation output for narrow openings.

ensuring that every cell (within the area boundaries) will have a tree edge connection from one of the trees. Note that this approach to achieve completeness is different from the previous off-line approach: In the on-line algorithm, if not stopped from going into certain cells, the robot will expand its tree to cover the entire work-area. This is in contrast to the off-line algorithms where a single spanning tree is constructed, and every robot covers only a portion of this spanning tree path.

Robustness. Our algorithm guarantees that the coverage will be completed in finite time even with upto $k - 1$ robots' failure. The spanning tree creation is an on-line process for k robots in our algorithm. If one robot fails, the rest of the robots $(k - 1)$ will cover the terrain. In the algorithm, and the robot gets to know whether the adjacent (up/down, left/right) grid cells are visited by the *marks* made by the visited robot. So if any grid cell which is not visited by any robot, will be visited by the other robot. It does not matter which robot fails or how many robots failed at the same time. Hence, we prove that our algorithm is robust to catastrophic failures, where robots fail and can no longer move. This result relies on an assumption that robots which fail do not block live robots. This assumption is achieved by moving the robots to visit a cell close to the visited cell.

The main idea behind the algorithm is that every robot gradually builds a local spanning tree of the uncovered cells that it discovers. The spanning tree is built by a depth-first-like procedure: Scan for a non-occupied neighboring cell, build a tree edge to it, enter that cell and continue recursively with this cell. If there is not any free cell, the robot goes back along its local spanning tree to the previous covered cell. Each ES-MSTC instance



(a) ES-MSTC algorithm (early stage).

(b) MSTC algorithm (early stage).

(c) ES-MSTC algorithm (final stage).

(d) MSTC algorithm (final stage).

Fig. 8. Simulation results comparison.

generates a path for its controlled robot on-line, one step at a time. The union of these paths is guaranteed to be complete, non-redundant, and robust.

Fig. 8 depicts the comparison of simulation results of ES-MSTC and MSTC algorithms. We can clearly see by Fig. 8(c), that the ES-MSTC algorithm is complete, because the algorithm covered the entire terrain (all cells accessible by all the robots). And also the work loads of the four robots are almost balanced. By looking at Fig. 8(b), if the robot (red) and the other robot (green) are malfunctioned, then there is no way for another robot (blue) to cover the unexplored area in the terrain. But by looking at Fig. 8(a), we can say there is a possibility to cover the unexplored area by the robot (blue) to complete the task. There is no blockage for the robot (blue) to go across into the region which the two broken robots may investigate. So the ES-MSTC algorithm is more robust than the MSTC algorithm. One of the greatest strengths of ES-MSTC is its ability to deal successfully with changing conditions. Since the algorithm does not rely on a hierarchical structure for coordination and task assignment, the system is highly robust to changes in the environment, including malfunctioning robots.

We implemented obstacle avoidance heuristics in our approach in such a way that there is no spanning tree which crosses the other spanning trees, as there is no collision with other robots. The proposed algorithm is implemented in parallel to ensure the optimal efficiency. We assume that in each time frame, each robot moves one step at a time. The algorithm creates a Hamiltonian cycle for each robot, having the largest number of cells in the given terrain. The resulting Hamiltonian cycle (union of k-spanning trees) is automatically an optimal covering path in terms of path length, since the tool covers every cell exactly once. To analyze the number of steps required to complete the coverage, we define the *running time*as the maximum over the steps that each robot has to go, which is proposed in [4]. Hence running time is calculated by using *max* (*step* (i)), where *step (i)* is the total number of steps taken by robot $i \varepsilon k$. The best running time for the algorithm is $4\lceil \frac{n}{k} \rceil - 1$, where n is the number of grid cells and k is the number of robots. The best case is when the starting positions such as $S_1 \ldots S_k$ place the robots at almost equal distances from each other. It is proved that the running time is critically dependent on the initial positions of the robots. The worst running time for our algorithm is dependent on the number of robots and their initial positions. The worst-case scenario is where generally all the robots start at cells next to each other, or adjacent cells. If any robot blocks the growing

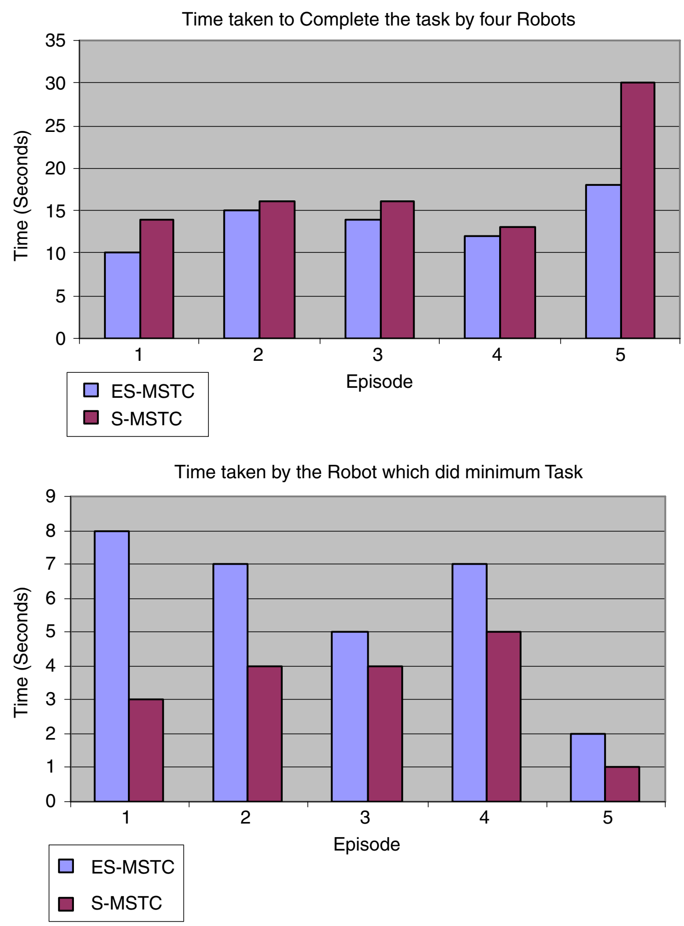


Fig. 9. Time comparison graph of five episodes.

process of the other robot's spanning tree then the situation will become worse.

We compare the simulation results of the ES-MSTC algorithm and the S-MSTC algorithm using four homogeneous robots in one environment shown in Fig. 5, with different starting positions [five different episodes]. The two graphs depicted in Fig. 9 show the time taken by the four robots to complete the exploration and coverage task and the time taken by the robot, which stops first (did minimum exploration and coverage) respectively. This time improvement is achieved due to the behavior known as identifying and sharing the narrow openings by multiple robots, which we proposed in our new algorithm (ES-MSTC).

In Fig. 10, four graphs are depicted to show the area covered by four robots using five different algorithms. In these algorithms, some do cover partially occupied cells and some do not. We compare the results of these five algorithms and prove that ES-MSTC shows the best performance compared to the other algorithms. In all graphs, we can notice that the ES-MSTC algorithm shows results close to the average outcomes for all the robots. This indicates that the work load is almost equally distributed among the robots which are in operation. Here the MSTC algorithm is developed by other researchers, while S-MSTC and ES-MSTC are developed by us with different versions. MSTC — partially occupied cells are not covered, MSTC (P) — partially occupied cells are covered, S-MSTC — partially occupied cells are not covered and cells in narrow opening area are not allowed to revisit, ES-MSTC (N) — only cells in narrow opening area are allowed to revisit, ES-MSTC (P) — only partial cells are covered, ES-MSTC — partially occupied cells are covered and cells in narrow opening area are allowed to revisit.

5. Conclusions and future works

In this paper, we propose an extended scheme (ES-MSTC) of our previous work (S-MSTC) to cover the terrain simultaneously by multiple ant-type robots in a complete and efficient manner. We use the cellular decomposition method to represent the continuous bounded terrain. This representation breaks down the terrain into simply connected cells that are easy to cover. Furthermore, the ES-MSTC algorithm covers an environment whose geometry is not known to the robots. The proposed decentralized simultaneous on-line coverage scheme for multi-robots is structured and it assures complete and robust coverage of the surface regardless of the shape of the terrain. This scheme considers partially-occupied cells and cells in narrow openings in the terrain are allowed to be revisited. Since we use ant-type robots, there is no need of a centralized component or any other global coordination and the robots effectively coordinate themselves through the marks in the terrain. As long as there is no blockage in the terrain by any robot, the exploration and coverage tasks are done in a balanced manner and this reduces the travel cost compared to the previous STC algorithms. We did experiments through simulations and the results were quite encouraging. As future work, we may consider generalizing this approach to robots with sensor uncertainty and other typical imperfections. The system performance is a very important issue, when we incorporate this algorithm in a situation such as dangerous or disaster situations. We will consider the system performance issue too, in our future extension of this work.

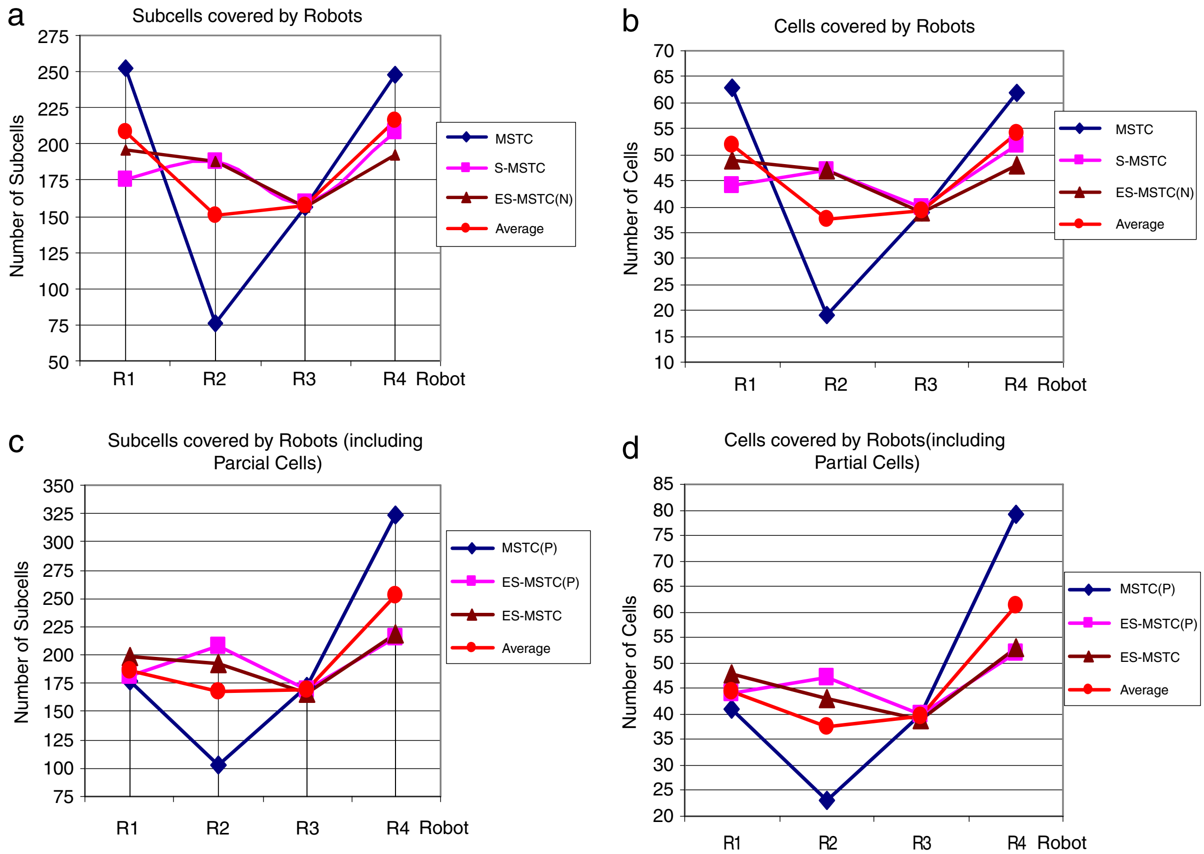


Fig. 10. Comparison graphs using five algorithms.

References

[1] N. Correll, A. Martinoli, Collective inspection of regular structures using a swarm of miniature robots, in: Proceedings of the 9th International Symposium on Experimental Robotics, ISER, 2006, pp. 375–385.

[2] Y. Gabrielly, E. Rimon, Competitive on-line coverage of grid environments by a mobile robot, Computational Geometry 24 (2003) 197–224.

[3] Pirzadeh A, W. Snyder, A unified solution to coverage and search in explored and unexplored terrains using indirect control, in: Proceedings of the IEEE International Conference on Robotics and Automation, 1990, pp. 2113–2119.

[4] W. Sheng, Q. Yang, J. Tan, N. Xi, Distributed multi-robot coordination in area exploration, Robot, Autonomous System 54 (12) (2006) 945–955.

[5] R.M. Zlot, A. Stentz, M.B. Dias, S. Thayer, Multi-robot exploration controlled by a market economy, in: Proc. IEEE Int. Conf. on Robotics and Automation, Washington, 2002, pp..

[6] D. Kurabayashi, J. Ota, T. Arai, E. Yoshida, Cooperative sweeping by multiple mobile robots, Proceedings of the IEEE International Conference on Robotics and Automation 2 (1996) 1744–1749.

[7] J. Svennebring, Koenig, Building Terrain-covering ant robots: a feasibility study, Autonomous Robotics 16 (3) (2003) 313–332.

[8] M.A. Batalin, G.S. Sukhatme, Spreading out: a local approach to multi-robot coverage, in: Proceedings of the 6th International Symposium on Distributed Autonomous Robotic Systems, 2002, pp. 373–382.

[9] Y. Gabrielly, E. Rimon, Spanning-tree based coverage of continuous areas by a mobile robot, Annals of Mathematics and Artificial Intelligence 31 (2001) 77–98.

[10] N. Hazon, G.A. Kaminka, On redundancy, efficiency and robustness in coverage for multi-robot, Robot Autonomous System 56 (2008) 1102–1114.

[11] M.A. Batalin, G.S. Sukhatme, The analysis of an efficient algorithm for robot coverage and exploration based on sensor network deployment, in: ICRA05: Proceedings of IEEE International Conference on Robotics and Automation, 2005, pp. 3478–3485.

[12] I.A. Wagner, M. Lindenbaum, A.M. Bruckstein, Distributed covering by ant-robots using evaporating traces, IEEE Transaction on Robotics Automation 15 (5) (1999) 918–933.

[13] K.S. Senthilkumar, K.K. Bharadwaj, A hybrid evolutionary approach for multi robot path exploration problem, in: Proceedings of IMCECS, Hong Kong, 2008, pp. 1310–1315.

[14] N. Agmon, N. Hazon, G.A. Kaminka, Constructing spanning trees for efficient multi-robot coverage, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2006, pp. 1698–1703.

[15] N. Hazon, F. Mieli, G.A. Kaminka, Towards robust on-line multi-robot coverage, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2006, pp. 1710–1715.

[16] C.S. Kong, N.A. Peng, I. Rekleitis, Distributed coverage with multi-robot system, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2006, pp. 2423–2429.

[17] A. Howard, L.E. Parker, G.S. Sukhatme, Experiments with a large heterogeneous mobile robot team: exploration, mapping, deployment and detection, The International Journal of Robotics Research 25 (2006) 431–447.

[18] E.M. Arkin, S.P. Fekete, J.S. Mitchell, Approximation algorithms for lawn mowing and milling, (CGTA) 17 (1–2) (2000) 25–50.

[19] W. Burgard, M. Moors, C. Stachniss, F. Schneider, Coordinated multi-robot exploration, IEEE Transactions on Robotics 21 (3) (2005) 376–378.

[20] W. Sheng, Q. Yang, S. Ci, N. Xi, Distributed multi-robot coordination algorithm for area exploration and coverage, in: Proc. IEEE Int. Conf. on Robotics and Automation Workshop on the State-of-the-Art of Mobile Robot Area Coverage, New Orleans, 2004.

[21] S. Koenig, B. Szymanski, Y. Liu, Efficient and inefficient ant coverage methods, Annals of Mathematics and Artificial Intelligence — Special Issue on Ant Robotics 31 (2001) 41–76.

[22] S. Koenig, Y. Liu, Terrain coverage with ant robots: a simulation study, in AGENTS01: Proceedings of the Fifth International Conference on Autonomous Agents, 2001, pp. 600–607.

[23] A. Zelinsky, R. Jarvis, J. Byrne, S. Yuta, Planning paths of complete coverage of an unstructured environment by a mobile robot, in: Proceedings of the International Conference on Advanced Robotics, 1993, pp. 533–538.

[24] G. Di Caro, M. Dorigo, Antnet: distributed stigmergetic control for communications networks, Journal of Artificial Intelligence Research 9 (1998) 317–365.

[25] S. Koenig, B. Szymanski, Value-update rules for real-time search, in: Proceedings of the National Conference on Artificial Intelligence, 1999, pp. 718–724.

[26] R. Sharpe, B. Webb, Simulated and situated models of chemical trail following in ants, in: Proceedings of the International Conference on Simulation of Adaptive Behavior, 1998, pp. 195–204.

[27] K.S. Senthilkumar, K.K. Bharadwaj, Multi-robot terrain coverage by constructing multiple spanning trees simultaneously, International Journal of Robotics and Automation ACTA Press 25 (3) (2010) 195–203.

K.S. Senthilkumar received his B.Sc. and M.Sc. Degrees in Computer Science from the University of Peradeniya, Sri Lanka in 1998 and 2003, respectively. Then, he received his M.Tech. Degree in 2006 and Ph.D. Degree in 2011, from Jawaharlal Nehru University, New Delhi, India. He did research on Multiple Robot Terrain exploration and coverage for his Ph.D. Degree in Computer Science. He is a lecturer in the Department of Computer Science, College of Art & Science, Wadi Al Dawaser, King Saud University, Kingdom Of Saudi Arabia. His research area covers Robotics, Artificial Intelligence and Soft Computing. He has published five scientific papers in international conference proceedings.

K.K. Bharadwaj is currently a professor in the School of Computer & Systems Sciences, Jawaharlal Nehru University (JNU), New Delhi, India. He joined JNU in 1985 as an Associate Professor and since 1990 he is a full Professor at JNU. Prior to JNU he has been a Faculty in the Computer Science Department, BITS, Pilani (Rajasthan), India. He received his M.Sc. Degree in Mathematics from the University of Udaipur (Rajasthan), India and his Ph.D. Degree in Computer Science from the Indian Institute of Technology (IIT), Kanpur, India. He has published widely in International journals and International conference proceedings. His current research interests include Machine Learning, Intelligent Systems, Knowledge Discovery in Databases (KDD), and Computational Web Intelligence.