Ann Math Artif Intell (2008) 52:109–142
DOI 10.1007/s10472-009-9120-2

Efficient Boustrophedon Multi-Robot Coverage: an algorithmic approach

Ioannis Rekleitis · Ai Peng New · Edward Samuel Rankin · Howie Choset

Published online: 17 March 2009
© Springer Science + Business Media B.V. 2009

Abstract This paper presents algorithmic solutions for the complete coverage path planning problem using a team of mobile robots. Multiple robots decrease the time to complete the coverage, but maximal efficiency is only achieved if the number of regions covered multiple times is minimized. A set of multi-robot coverage algorithms is presented that minimize repeat coverage. The algorithms use the same planar cell-based decomposition as the Boustrophedon single robot coverage algorithm, but provide extensions to handle how robots cover a single cell, and how robots are allocated among cells. Specifically, for the coverage task our choice of multi-robot policy strongly depends on the type of communication that exists between the robots. When the robots operate under the line-of-sight communication restriction, keeping them as a team helps to minimize repeat coverage. When communication between the robots is available without any restrictions, the robots are initially distributed through space, and each one is allocated a virtually-bounded area to cover. A greedy auction mechanism is used for task/cell allocation among the robots. Experimental results from different simulated and real environments that illustrate our approach for different communication conditions are presented.

I. Rekleitis (✉)
School of Computer Science, McGill University, Montreal, Canada
e-mail: yiannis@cim.mcgill.ca

A. P. New · E. S. Rankin
DSO National Laboratories, Singapore, Singapore
e-mail: naipeng@dso.org.sg

E. S. Rankin
e-mail: erankin@dso.org.sg

H. Choset
Robotics Institute, Carnegie Mellon University, Pittsburgh, USA
e-mail: choset@cs.cmu.edu

Keywords Coverage · Multi-robot · Algorithmic robotics

Mathematics Subject Classifications (2000) 68T40 · 11Y16

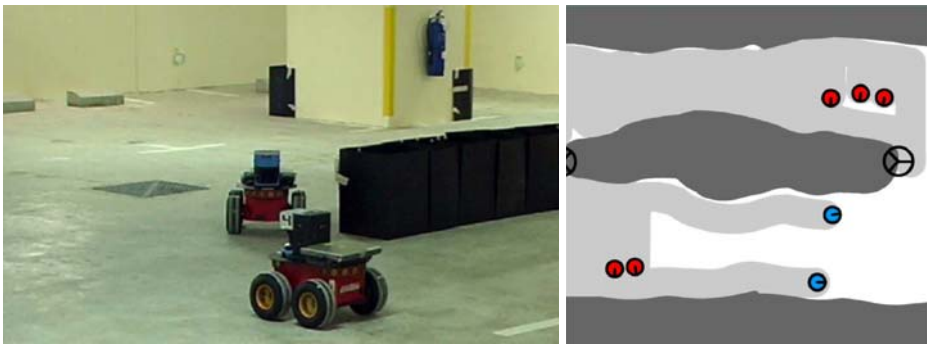
1 Introduction

The task of covering an unknown environment, common in many applications, is of great interest in a number of industries like automated vacuum-cleaner, carpet-cleaner, or lawn-mower manufacturing; chemical or radioactive spill detection and cleanup; and humanitarian de-mining. Many of those applications require complete coverage. The goal of complete coverage is to plan a path that will be used to guide one or more robots to pass an end-effector, in our case equivalent to the footprint of the robot, over every accessible area of the target environment. Complete coverage guarantees that no accessible area is left uncovered. In all cases we assume that the target area is of known dimensions and for simplicity's sake, we assume it is a rectangle. The interior of the area to be covered is unknown and partially occupied by obstacles.

In the single robot case, previous work has produced algorithms that guarantee complete coverage of an arbitrary unknown environment [1]. Introducing multiple robots provides advantages in terms of efficiency and robustness but increases the algorithmic complexity. Central in the multi-robot approach is the issue of communication. When communication is restricted to close proximity [2] or line of sight[1] [3], the robots have to remain together in order to avoid covering the same area multiple times. When unrestricted communication is available then the robots can disperse through the environment and can proceed to cover different areas in parallel, constantly updating each other on their progress. In this paper, three different algorithms for the two different modes of communication, restricted and global, are presented. For limited communication, we outline an algorithm called *team-based coverage*. For unlimited communication, we present two algorithms that go under the name of *distributed coverage*, which differ in the area-size each robot covers at a single step.

Firstly, we examine the problem of multi-robot coverage path planning for a team of robots with limited communication. Information sharing between the robots is restricted to line-of-sight communication (Fig. 1b). When communication is limited, the robots that drop out of communication cannot stay up-to-date on where the other robots have been, and end up covering again areas already covered; see [3]. It is easy to construct environments where lack of communication can result in the majority of space being covered repeatedly, thus, resulting in very little improvement in efficiency compared to a single-robot approach. To avoid repeat coverage and thus improve efficiency, we propose a multi-robot algorithm where the robots stay together as a team. Thus, the robots start together at an entry position in the area to be covered as a team, and the complete team can split into two sub-teams to cover around a single obstacle, with no further splitting of the sub-teams. When the area

[1]Line-of-sight restriction: communication is available *iff* two robots have an unobstructed line of sight between them.



(a) (b)

Fig. 1 **a** Two Pioneer robots performing coverage. **b** Seven robots in two teams covering around an obstacle

around the obstacle that caused the split is covered, the two sub-teams join again into a single team. Only after the two sub-teams re-unite, the team can split again. This approach can be considered "conservative"; however, when communication is limited, a conservative approach works best at reducing repeat coverage for a wide variety of environments.

Next, we examine the problem of complete coverage when unrestricted communication is available. In this case, the team of robots is deployed at regular intervals along one side of the field to be covered, as in Fig. 2. Thus, the environment is divided into a number of virtual stripes equal to the number of robots. Each robot is allocated initially the responsibility of the stripe it is placed at, and the coverage starts. The challenge is to allocate different uncovered areas to each robot such that no robot stays idle, thus, all finish covering around the same time, and the amount of time spent commuting among different regions is minimized. Providing an optimal solution for minimizing the travel time is an NP-hard problem;[2] see Section 6 for the proof. As such, an auction mechanism is used in order to reallocate regions to be covered between robots in such a way that the path travelled between regions is reduced. The auction mechanism is a greedy heuristic based on the general market approach [4]. The size of the tasks auctioned or negotiated among the covering robots is another parameter that can be changed: a variation of the second algorithm (full communication) has been implemented with real robots using smaller tasks and a simple negotiation protocol, instead of an auction; see Fig. 1a.

The proposed algorithms are based on prior single-robot coverage methods that guarantee complete coverage. To achieve provable completeness, most single robot coverage planners use a cellular decomposition of the environment. Exact cellular decompositions represent the free configuration space by dividing it into non-overlapping cells such that adjacent cells share a common boundary; the interior of

[2] In other words, finding an optimal path would require computation time exponential to the number of cells to be covered.

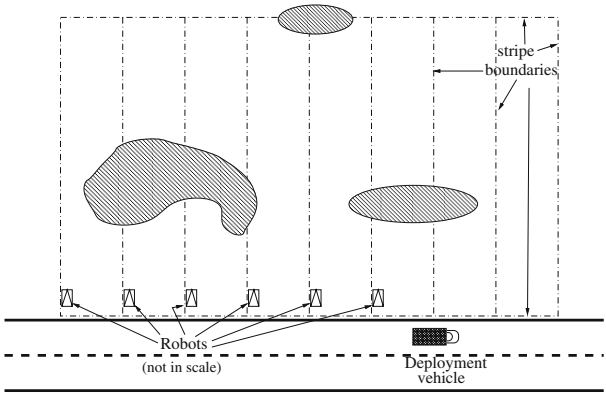


Fig. 2 A large unknown area is divided up in *vertical stripes*. Each covering robot is assigned a stripe to cover. A deployment vehicle is utilized that distributes the robots at the beginning of the stripes. The robots do not know the layout at the interior of each stripe. The union of all the stripes represents the area to be covered

each cell intersects no other cell; and the union of all the cells covers the free space. Covering each cell is simple, and complete coverage provably results from ensuring that the robot visits every cell.

Our algorithms are based on the sensor-based approach which covers the unknown space while simultaneously constructing a cellular decomposition, which, in turn is used to guarantee complete coverage. In the case of team-based coverage, a robot team consists of all robots within line-of-sight to each other. Each distinct team maintains its own internal representation of the world, which is shared and updated whenever two teams come within line-of-sight of each other and join again into one team. In the case of distributed coverage, each robot autonomously covers the area (stripe) it is assigned, keeping track of all the covered, uncovered, and unknown space by communicating with the other robots. In the second algorithm of distributed coverage, the granularity of each covering task is much finer. Each robot covers one thin stripe with width equal to twice the footprint of the sensor and, when done, selects the uncovered stripe that is closest to it.

Regardless of the communication capabilities, we assume that the robots know their position and orientation with respect to a global reference frame. For our hardware implementation, the localization problem is abstracted by allowing the robots to localize themselves using a laser range finder and retro-reflective targets placed in the environment. In all cases, the robot sensors are able to detect both static obstacles and mobile robots, and differentiate between the two. The sensors have limited range and good angular resolution.

This paper is divided into the following sections: Section 2 discusses related work on multi-robot coverage and market-based mechanisms; it also presents some necessary terminology related to the Boustrophedon cellular decomposition. Section 3 describes the limited communication multi-robot coverage algorithm. The global communication, auction-based algorithm is presented in Section 4. A variant of the full communication algorithm is outlined in Section 5; the proposed algorithm divides the coverage problem to finer granularity tasks. Section 6 contains the proof that solving optimally the multi-robot coverage problem is NP-hard. Section 7 contains qualitative experimental results from a variety of environments for the different

algorithms. Finally, Section 8 presents our conclusions and the directions of ongoing and future research.

2 Related work

This work employs the fundamental principles of a single-robot coverage algorithm for each individual robot. When global communication is available, an auction mechanism is used to negotiate among robots which areas each robot would cover. Firstly, we will examine relevant background on multi-robot coverage. Then, we will discuss related work on market-based mechanisms in mobile robotics. Finally, we present a brief overview of relevant terminology used in coverage and exact cellular-decomposition literature.

2.1 Multi-robot coverage

Previous work can be grouped using two different criteria: determinism and communication ability. First, we examine deterministic approaches, which guarantee complete coverage; then, we present a brief overview of a variety of techniques which use a stochastic approach—sometimes referred to as swarm, biologically inspired, or behavior-based techniques—to solve the multi-coverage problem.

Our work takes root in the Boustrophedon decomposition [5], which is an exact cellular decomposition whereby each cell can be covered with simple back-and-forth motions. The cells are defined by sweeping a slice [6, 7] (a one-dimensional line) through the configuration space and noting where the connectivity of the slice changes in the free configuration space. These connectivity changes occur at critical points. A method that uses simple sonar range sensors to detect critical points was introduced in [8]. Using this method, the robot can simultaneously cover an unknown space while looking for critical points to ensure complete coverage. An alternative approach based on a triangular tiling of the free space was presented in [9].

Butler et al. [10] achieved complete coverage of unknown rectilinear environments using a square robot with contact sensing. They performed an on-line decomposition where each cell, in the shape of a rectangle, was formed such that it could be covered completely by back-and-forth motions performed parallel to one of the walls of the environment. They have also extended their work to multiple robots [11]. The basic concept of the multi-robot algorithm is that cooperation and coverage are algorithmically decoupled. This means that a coverage algorithm for a single robot can be extended to a cooperative setting. To produce cooperative coverage, an overseer algorithm is added to the single robot algorithm, which takes incoming data from other robots and integrates it into the cellular decomposition. In their approach, unrestricted communication is assumed among the robots. It can be shown that the overseer indeed performs this operation in such a way that coverage can continue under the direction of the single robot algorithm without the algorithm even knowing that cooperation occurred.

Early work by Kurabayashi et al. proposed an off-line planning algorithm for sweeping a known area with the ability to plan for relocating objects. The algorithm acts in two stages: first the complete path is planned off-line, and the area is divided

among robots [12]; then, the location of the movable object is examined, and the optimal relocation path is estimated [13]. Experimental results showed an improvement in the efficiency due to smart relocation of movable objects. Recently, Latimer et al. [2] employed an algorithm based on the single robot cellular decomposition approach. Dispersing the robots through the environment was emphasized and encouraged to allow parallel coverage with finer granularity [2]. However, because this approach only allowed communication between robots in physical contact with each other, many robots ended up covering the same space.

Tao and How [14] used a decentralized approach to select cells in a grid world to be covered. They used a limited motion and sensor model and assumed global communication. Negotiation between agents was used in order to facilitate cooperation. More recently, Luo and Yang [15, 16] employed a neural network to represent the environment. Each cell/neuron corresponds to a cell in the occupancy grid, and the activity in each neuron represents the belief that the cell is occupied, unknown, or covered. They have demonstrated their approach for two [15] and four [16] collaborating robots covering an environment in simulation.

Ichikawa and Hara [17] proposed a multi-robot coverage behavior that emerges from simple obstacle avoidance for large number of robots. Their approach is simple, does not guarantee full coverage, and the same area is covered multiple times. Wagner et al. [18] use robots that employ traces to mark the covered areas in a biologically inspired approach termed ant-robotics. The environment is represented as a graph where each node is a cell in a grid-world. Communication among agents is done implicitly via the trace of pheromones each robot leaves. In later work [19], both deterministic and probabilistic approaches where discussed. The deterministic case used a depth-first-search algorithm and as such every place was covered twice on average. Finally, the existence, or not of dirt was used as an implicit method of communication in [20]. Menezes et al. [21] use pheromones simulations to disperse the robots in a random fashion, thus achieving good coverage in the probabilistic sense. Ant like robots were also used in [22]. Ge and Fua [23] proposed an algorithm that reduces repeat coverage by ensuring that each robot leaves paths around the obstacles and the area they cover. Another biologically inspired approach was used by Bruemmer et al. [24] for covering an area using a swarm of small robots. The complete coverage was determined by a human observer for different swarm sizes. More recently, Batalin and Sukhatme [25] proposed two behavior based algorithms for multi-robot coverage. The communication is limited to visual contact and the robots disperse through the environment to ensure maximum coverage with no guarantee for completeness. A physics base approach was introduced by Spears et al. [26], where a swarm was modelled like a gas in order to sweep through a corridor.

Kaminka et al. [27–29] proposed a variety of algorithms for covering a grid-like environment using a spanning tree in order to traverse the connectivity graph of a grid representation of the free space. In their work, the issues of redundancy and robustness to robot attrition were considered. The environment to be covered could be either known [27] or unknown [28]. Solanas and Garcia [30] addressed the problem of first exploring an unknown space and then dividing the area to a number of cells equal to the number of robots. A different version of the coverage problem involves the coverage of the boundaries of free space. Williams and Burdick [31] proposed an algorithm for solving the boundary-coverage problem using a graph structure.

Finally, when the terrain is known, existing algorithms guarantee a performance of at most eight times the optimal cost [32].

2.2 Market-based approach in robotics

Cooperation and task allocation among mobile robots is crucial in multi-robot applications. To facilitate task re-allocation, a new methodology based on a market economy has gained popularity. The main goal of this technique is to provide an efficient mechanism for utilizing a variety of resources that are distributed among multiple robots. There have been different variations of market-based schemes; for a comprehensive survey, please refer to [4, 33]. The main idea behind most of them is for one agent, either a robot or an operator, to put a task up for an auction. Then, each of the robots "bids" with a value that can represent, in different cases, a cost or an utility gain for that robot. The auctioneer collects all the bids and then awards the task to the robot that had the "best" bid. In our case, we are using the estimated cost for moving to a cell and covering it; the auctioneer thus is awarding the task to the robots with the least cost, see Section 4.3. Different variations exist with respect to the number of tasks that an agent can bid for, combinations of tasks or one task per robot, or the ability to incorporate constrains, such as sequencing requirements, into the bids.

Market-based approaches have been used to solve the multi-robot task allocation problem [34] in the domains of exploration [35, 36], failure/malfunction detection and recovery [37], box pushing [38], and also for more complex tasks [39].

2.3 Boustrophedon/Morse decomposition

To better describe the multi-robot coverage task, the following terms from single robot coverage: *slice*, *cell*, *sweep direction*, and *critical point* [40, 41] are borrowed; see Fig. 3. The Boustrophedon decomposition [5] is a type of Morse decomposition where the slice is a line; the robot follows the intersection of the slice and the area to be covered, thus allowing the robot to cover the area with vertical back and forth motions. A *cell* is a region defined by the Boustrophedon decomposition where slice connectivity is constant. *Sweep direction* refers to the direction in which the slice is swept. Lastly, a *critical point* represents a point on an obstacle, which causes a change in the slice connectivity. Thus, the free space is divided into regions (cells) of constant connectivity, each of which can be covered with a vertical back and forth motion.

Critical points are further divided into four categories (as in Fig. 4a) based on sweep direction and convexity/concavity of the critical points. The sweep direction is relative to a robot, and can change during the process of coverage. A critical point is characterized as convex/concave if it is introduced by the convex/concave region of an obstacle, i.e., the convexity of the point where the slice contacts the obstacle; see Fig. 4a for an illustration of convex and concave critical points. It is worth noting that, if the sweep direction reverses, a forward convex critical point (*FCV*) would become a reverse convex critical point (*RCV*). Concave critical points are also referred to as terminating critical points. No forward concave critical points are ever encountered, as by definition the sweep direction guides robots away from them.

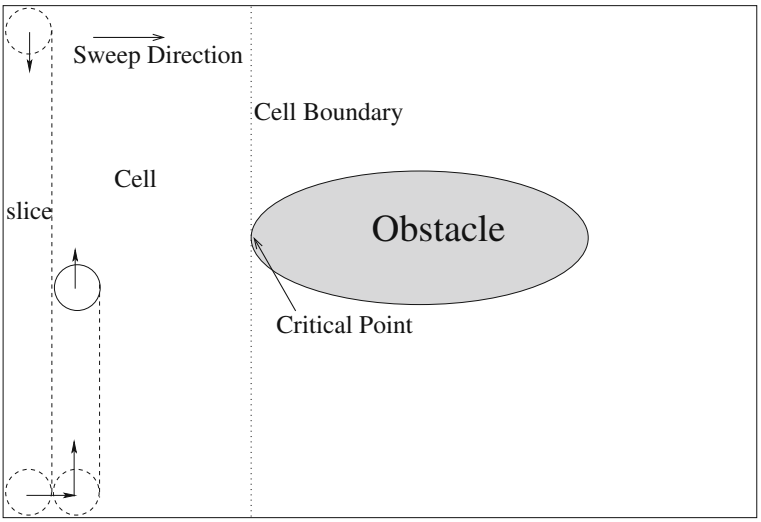


Fig. 3 Illustration of the terms borrowed from single robot coverage with a single robot and one obstacle in the target environment. The robot is performing coverage with simple back-and-forth motions

Another concept used here is the concept of a Reeb graph [41, 42]. A Reeb graph is a graph representation of the target environment where the nodes represent the critical points and the edges represent the cells; see Fig. 4b. Due to the nature of the Boustrophedon decomposition, all concave critical points are connected to exactly one cell, i.e., a node of degree one in the Reeb graph. Similarly, all convex critical points are connected to exactly three cells, i.e., a node of degree three in the Reeb graph.

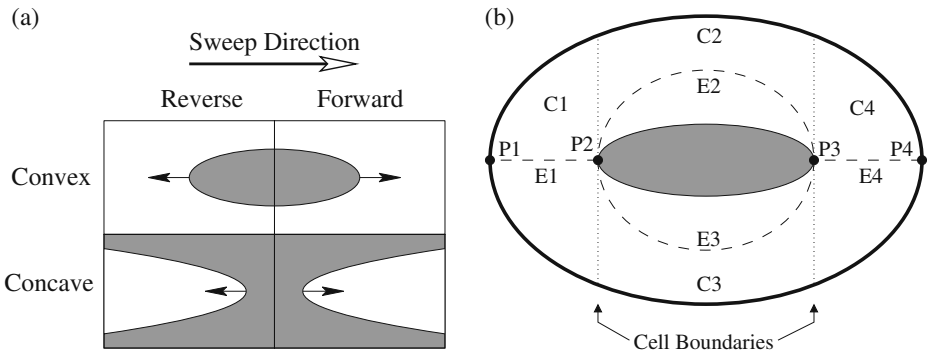


Fig. 4 **a** Depiction of the four types of critical points, based on concavity and the surface normal vector parallel to the sweep direction. 'Forward' indicates parallel sweep and normal vector directions, whereas 'Reverse' indicates opposite directions. Note that the *shaded areas* are obstacles and the *arrows* represent the normal vectors. **b** Here a simple Reeb graph is overlaid on top of a simple elliptical world with one obstacle. P1–P4 are critical points which represent graph nodes. E1–E4 represent edges which directly map to cells C1–C4

Next, the multi-robot coverage algorithm for the restricted communication case is presented.

3 Team-based multi-robot coverage algorithm

As mentioned earlier, all robots start in a horizontal formation; see for example the five robots that started at the upper left corner of Fig. 5. Although the world is not known a priori, we do assume a static environment of bounded dimensions.

The proposed algorithm is based on two main ideas. First, during the coverage of a single cell, the boundaries of the cell are covered by two robots. These robots follow the top and the bottom boundary, with the same lateral speed, until they are either no longer within line of sight of each other, or their distance becomes less than a safety threshold. We refer to these two robots as *explorers* to distinguish them from the rest of the team, termed *coverers*, which just cover the free space. The two explorers use any break in the line of sight to detect critical points and thus determine the termination of the cell. The line of sight has already been successfully used as an extended sensor by a number of different authors [43–45]. It is worth noting that prior work in single robot coverage actually had to rely on sophisticated obstacle detecting motion strategies to determine the location of critical points; now because multiple robots are used, simple wall following allows for critical point detection.

Second, in order to maintain the cohesiveness of the team and to avoid redundant coverage, our approach allows the team of robots to divide in two sub-teams only once. When an obstacle induces a convex critical point that terminates the current cell and introduces two new cells, for example, the point P_2 in Fig. 4b, then, the team of robots is allowed to split into two sub-teams. The two sub-teams cover only the cells adjacent to the obstacle that caused the split until the two sub-teams meet again and rejoin into a single team. Each of the two sub-teams requires at least two robots for top and bottom cell-boundary exploration. Thus, the algorithm we present requires a minimum of four robots. After the two sub-teams have rejoined into a single team, if they encounter another convex critical point with two uncovered cells attached, then the team will split again.

For the team-based coverage algorithm, the following terminology is also used: *union cell*, *top cell*, *bottom cell*, and *complete* and *incomplete critical points*; Fig. 6 illustrates these definitions. The *union* cell is defined as the cell pointed to by the normal vector at a convex critical point. Without loss of generality, we consider a horizontal, from left to right, sweep direction. The other two cells associated with a convex critical point are relative to the global reference frame and are defined as the

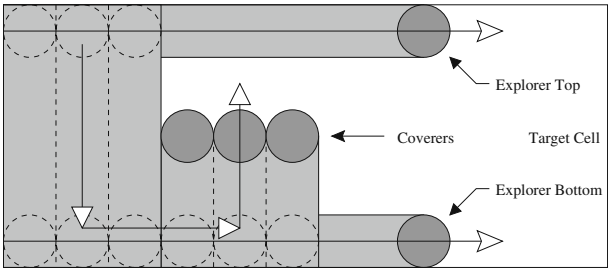


Fig. 5 Illustrates the *explorer/coverer* approach, where two robots *explorers* outline the top and bottom boundaries while the remaining robots (*coverers*) execute simple back-and-forth coverage

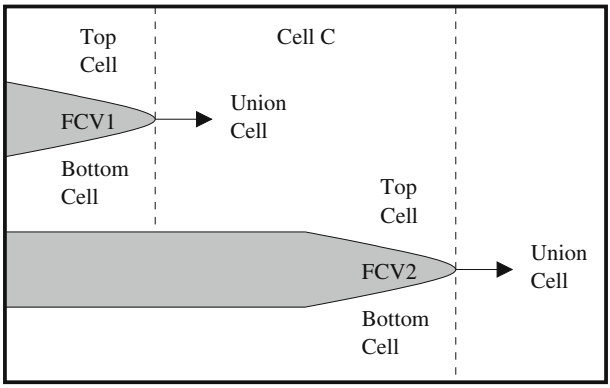


Fig. 6 Depicts cell naming adjacent to two forward convex critical points (*FCV1*, *FCV2*). The *union cell* is pointed at by the critical point normal and the *top cell* and *bottom cell* are above and below the critical point respectively. The *dashed lines* represent cell boundaries. Note that Cell C is both the union cell of *FCV1* and the top cell of *FCV2*

top (higher *y*-coordinate) and *bottom* (lower *y*-coordinate) cells respectively. Note that cell naming is relative to a critical point; therefore, a cell may have different names with respect to different critical points. A critical point is characterized as *complete* if all of the cells adjacent to it are completely covered; otherwise, the critical point is referred as *incomplete*.

The team, or the divided sub-teams, covers the cells, one cell at a time, and the Reeb graph determines which cell to cover next. The overall algorithm is depicted in Fig. 7a and described in detail in the following sub-sections. Initially, the robots cover the starting cell as one large team, using the *"Cover a Single Cell"* algorithm; see Fig. 7a, simultaneously creating a cellular decomposition and the corresponding Reeb graph. After a cell is covered, the team proceeds to the closest critical point with uncovered cells, the *"Go to Closest Incomplete Critical Point"* block in Fig. 7a. We call that goal critical-point "target critical point". If no such critical point exists, the space has been fully covered and the algorithm terminates.

As we saw earlier, the number of cells connected to a critical point must be exactly one (concave) or three (convex). Also, every known critical point is adjacent to at least one covered cell, namely, the cell at the end of which the critical point was discovered. Therefore, the target critical point, which is incomplete, has to be convex and has one or two uncovered cells attached to it. If the number of uncovered cells at a target critical point is one, the team invokes the procedure to cover a single cell, *"Cover Single Cell"*.

If the target critical point is forward convex (*FCV*), the entire team proceeds to cover the union cell; see Fig. 6. However, if the target critical point is reverse convex (*RCV*), the team splits into two sub-teams and starts covering the cells adjacent to the obstacle that introduced the critical point. One sub-team covers cells in a clockwise fashion around the obstacle. Conversely, the other sub-team covers cells in a counter-clockwise fashion. This is the only case where a robot team splits into sub-teams. The encircling process further guarantees that the two sub-teams will meet again; see Fig. 7b. After the two sub-teams have rejoined, if they encounter another *RCV*, they would split again. The main constraint in the proposed algorithm is that a sub-team would never split.

After rejoining, the two sub-teams update their internal representations (Reeb graphs). Then, the unified team travels to the closest incomplete critical point according to their merged Reeb graphs. This continues until no incomplete critical points

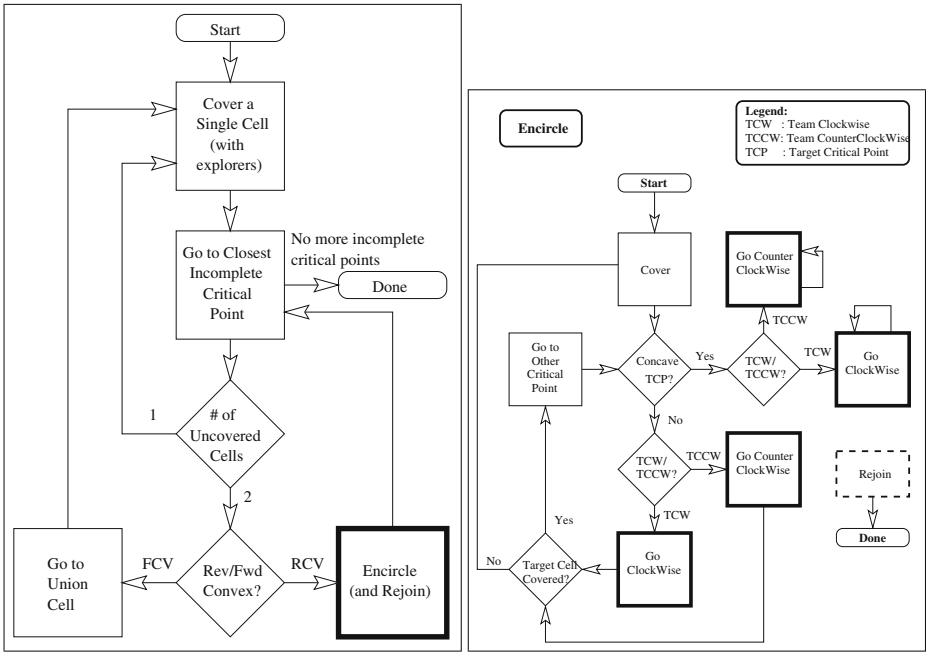


Fig. 7 **a** Illustrates the top level of our algorithm. Team splitting and rejoining only occurs in the *Encircle* state, which is delineated with a thick border to show that is has its own sub-diagram. **b** Illustrates the *Encircle* procedure. While travelling if a sub-team encounters, either a concave critical point, or the other sub-team, then the two sub-teams exchange information and rejoin into a single team

exist, which implies all cells have been covered. The following sections describe the major parts of the team-based algorithm.

3.1 Cover a single cell

Cover a Single Cell is the fundamental primitive behavior of our algorithm. As we saw earlier the robots act as explorers or, as coverers. The first two robots to enter the cell are designating as *explorers*. The roles of the explorers are to detect critical points, to cover the boundaries of the cell, and to facilitate rejoining. The coverers extend the normal back-and-forth sweeping technique, see Fig. 3, from single robot coverage into a group behavior.

First, we present some terminology; see Fig. 5. *Target cell* refers to the current cell being covered. *Top explorer* refers to the first robot to enter the target cell, which explores the top boundary of the cell; *bottom explorer* refers to the second robot which explores the bottom boundary. *Coverers* is the collective term that addresses any remaining robots.

Upon entering the target cell, the two explorers wall-follow along the top and bottom cell boundaries respectively maintaining the same lateral speed thus guaranteeing line-of-sight communication within the cell. Simultaneously, the coverers

move in the target cell, as space permits, and begin simple back-and-forth covering; see top of Fig. 5. Figure 8 presents examples of the different types of critical point detection by the explorers. In all cases, the simulation package Player/Stage [46] was used to illustrate the different scenarios. The explorers continue wall-following until one of these two situations occurs: either the line-of-sight is broken—thus a convex critical point is detected; see Fig. 8a, d, and Fig. 8b, e—or the explorers approach each other, resulting in the detection of a reverse concave critical point; see Fig. 8c, f.

When the line of sight is interrupted then a critical point is detected. If just prior to the interruption one of the robots travelled opposite to the sweep direction, a forward convex critical point was detected, as in Fig. 8a, d; otherwise, a reverse convex critical point was detected, as in Fig. 8b, e. In both situations, the explorers retrace their steps until the line of sight is once again restored. The bottom explorer then proceeds to travel vertically upwards until it joins with the top explorer. Note that in both situations a robot will pass by the detected critical point, allowing its position to be recorded.

When the bottom explorer is next to the top explorer, the explorers proceed to cover the cell with simple back-and-forth motions, sweeping opposite to the sweep direction; see Fig. 9a. In this manner, the coverers and explorers are moving toward each other, ensuring they will meet.

In the second situation, where the two explorers approach each other (Fig. 8c, f), the top explorer backs off and allows the bottom explorer to wall-follow the remaining unknown surface. If a reverse concave critical point is found,

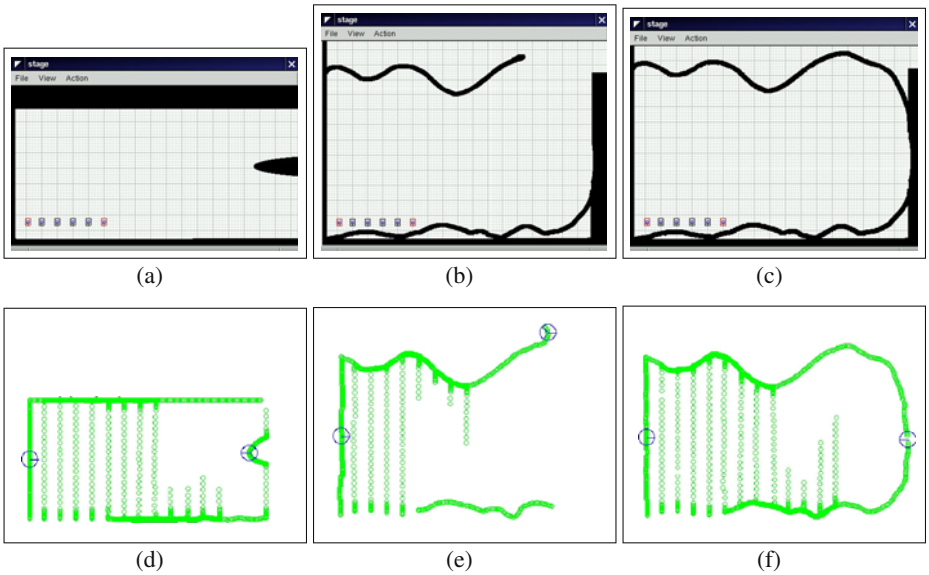


Fig. 8 Illustrates all the environmental possibilities *explorers* encounter. The *top row* (**a**, **b**, **c**) presents the environment in the Stage simulation package with the robots at their starting positions; the *bottom row* (**d**, **e**, **f**) presents the robot traces. **a**, **d**, **b**, **e** For convex critical points, losing line-of-sight signals the end of a cell. **c**, **f** Two explorers approaching each other marks a reverse concave critical point

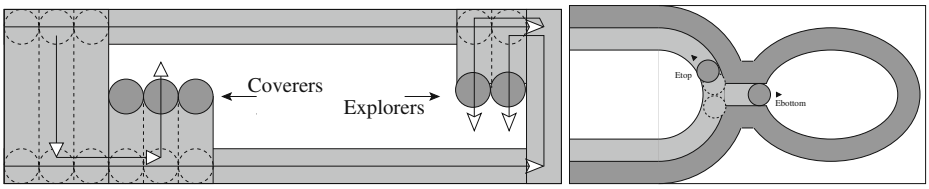


Fig. 9 **a** Shows *explorers* covering opposite the sweep direction after encountering a critical point. **b** Illustrates the second situation, when the remaining surface reveals a corridor where only one explorer can fit

it is recorded and the explorers pair up to cover in the opposite direction as in the previous situation.[3]

3.2 Encircle

Every time an obstacle introduces a reverse convex critical point, there are exactly two cells to be covered. For example, in Fig. 10a, seven robots cover Cell A; they encounter a reverse convex critical point; thus, two new cells, B and C, are introduced. Our algorithm covers the cells that are adjacent to that obstacle by splitting the team of robots into two sub-teams. The overall goal of the encircle stage is to take advantage of the finer granularity of smaller teams by splitting a larger team while ensuring that no cell is going to be covered twice due to the lack of communication/coordination between sub-teams (line-of-sight limitation).

Executing an encircle action splits the team into two sub-teams (Fig. 10a). Provided that each sub-team has a minimum of two robots, any partition is acceptable. A reasonable heuristic is to allocate robots proportional to the distance from the critical point to the top and bottom boundaries. As mentioned earlier, our team of robots splits only once; when the two sub-teams rejoin, they continue coverage as a single team, and only then they can split again. Thus, a sub-team never splits.

We name the two sub-teams team-clockwise and team-counter-clockwise because the sub-teams cover the cells around the obstacle in opposite directions. For both sub-teams, if the target cell is already covered, sub-teams travel to the opposite critical point[4] without covering, executing the procedure *Go to Other Critical Point*, and the algorithm continues. The two sub-teams cover around the obstacle in opposite directions until one of the following two terminating conditions are met:

- Case 1: The two sub-teams encounter each other, i.e., establish line-of-sight communication, within a cell. The two sub-team rejoin and then proceed to jointly finish covering the remainder of the cell in which they found each other.

[3]It is worth noting that there is special case when the remaining surface revealed an opening to a traversable corridor; see Fig. 9b. In such case, both robots would move inside the narrow corridor one after the other and continue exploring as soon as the space is wide enough, or find a terminating critical point.

[4]Note that all cells have exactly two bounding critical points by the nature of the Boustrophedon decomposition.

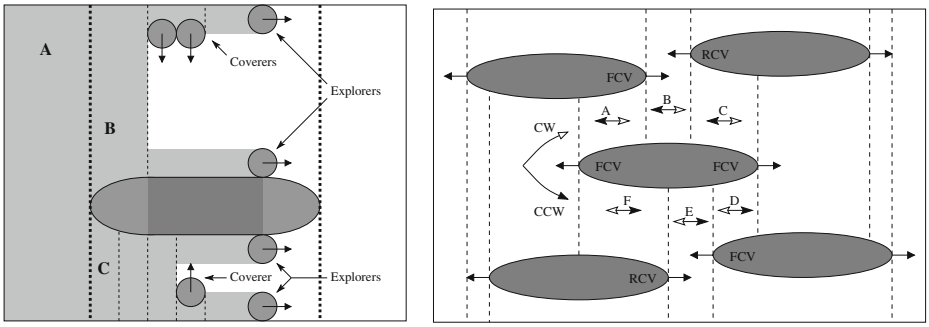


Fig. 10 **a** Shows two sub-teams encircling an obstacle. The *thick dashed lines* represent cell boundaries and the *thin dashed lines* represent slice boundaries. **b** Illustrates the clockwise and counter-clockwise paths taken during the encircle action. The *double-headed arrows* represent the sweep direction, *white* for clockwise, *black* for counter-clockwise. The *single black arrows* represent critical point normal vectors and forward convex (*FCV*) / reverse convex (*RCV*) critical points are labelled appropriately for a clockwise path. Cell boundaries are represented by *vertical dashed lines*. A clockwise path would travel through the cells *A* through *F* alphabetically, while a counter-clockwise path would travel through the cells in the reverse order [*F*,..., *A*]

- Case 2: One sub-team encounters a concave (terminating) critical point. This terminates one branch of the Reeb graph and the sub-team backtracks. The robots of this sub-team travel through the cell(s) they had already covered, in the direction opposite to the original direction (clockwise if counter-clockwise and vice versa), counting all encountered cells as covered, until they reach the other sub-team. We call this sub-team terminated.

It is important to note that the terminated sub-team does not cover before rejoining the other sub-team. Instead it follows the boundary of the obstacle being encircled. Similar to Case 1, once the line of sight is established and the two sub-teams rejoin, they cooperatively cover the target cell and conclude the encircle process. Next we briefly discuss the two procedures, *Go Clockwise* (*GCW*)/ *Go Counterclockwise* (*GCCW*), which guide the two sub-teams around the obstacle.

The two procedures *GCW* and *GCCW* are deterministic algorithms that guide the two sub-teams to cover the cells adjacent to a single obstacle. That obstacle is the one that introduced the critical point that forced the split. The team moving clockwise (*GCW*) follows a right-hand rule and every time selects the adjacent cell that has the obstacle to the right, when facing toward the sweep direction. For example, in Fig. 10b, the team going clockwise after covering cell A proceeds to cover cell B, which is the cell that keeps the obstacle at the right side with respect to the sweep direction. If the team continues and has to cover the cells below the obstacle, for example cell D, the sweep direction is now from right to left thus the obstacle remaining on the right-hand side. Ignoring the terminating conditions, the team moving clockwise in Fig. 14 would cover the cells A,..., F in order. The team moving counterclockwise follows a left-hand rule. The two sub-teams continue covering cells until one of the terminating conditions that we discussed earlier occurs. When the two teams establish line of sight they rejoin to a single team.

Next we examine the advantages and disadvantages of the use of explorers during team-based coverage.

3.3 Explorer/coverer discussion

In the case of restricted communication, there are clear benefits to an explorer/coverer split approach within a cell. Covering both the top and bottom boundaries simultaneously ensures complete coverage of the boundaries without the need for elaborate motion strategies such as the reverse wall-following Acar and Choset proposed in [41]. Line-of-sight communication enables the robots to detect critical points at a distance. Furthermore, in the case of two sub-teams covering from different ends of the same cell, the explorers enable both sub-teams to rejoin earlier, and simplify coordination.

Outlining the region also leaves coverers more space to turn. This space permits robots to take wider, more gradual turns, thus increasing the maximum acceptable speed. In other words, a wider turn can be executed safely at a higher speed than a sharp turn. An appropriate analogy would be outlining before filling regions in a coloring book.

Additional savings may be possible depending on robot configuration. For example, with homogeneous circular robots and the explorer/coverer technique, the coverer robots need not even touch the obstacle to guarantee complete coverage. They only need to come within a robot radius of the obstacle to wall-follow.

As for disadvantages, coverers will still generate repeat coverage by overlapping with some regions previously covered by the explorers. Furthermore, the explorer/coverer approach requires line-of-sight communication and introduces a considerable amount of complexity. However, we believe the advantages are well worth the cost of such complexity.

Finally, when the cells are small, and the whole team does not fit inside them, the robots have to wait, thus reducing the efficiency. Such disadvantages disappear when unrestricted communication is available.

3.4 Completeness

From the described algorithm. Any environment is decomposed into exact cells using the Boustrophedon cellular decomposition. There are a finite number of cell, encoded as edges of the Reeb graph. Every cell is accessible by any other cell by following the Reeb graph. The team of robots never covers a cell twice. The team of robots continues covering on cell at a time until all cells are covered.

Lemma 3.1 *The proposed algorithm always terminates.*

Proof Let us assume that the algorithm does not terminate.

Then, the team of robots after finishing (in finite time) covering some cells, they will have to attempt to cover an already covered cell. A contradiction with the algorithm description. □

Lemma 3.2 *The proposed algorithm performs complete coverage.*

Proof Let us assume that the algorithm terminates without covering an accessible area.

The not-covered area is either part of a cell, or a complete cell.

If the not-covered area is part of a cell then that means that the team of robots interrupted coverage, a contradiction to the description of the algorithm.

If the not-covered area is a complete cell, then the robots would move, via the Reeb graph, to cover it next, as it is accessible and marked as not covered in the Reeb graph. □

4 Distributed multi-robot coverage algorithm

In this section, we present the first efficient algorithm for guiding a group of robots performing complete coverage of an unknown environment with exact cell-decomposition with set boundaries. The robots do not alter the environment and communication among them is unrestricted. The robots spread out, and each one is assigned an area to cover; this area is called a stripe; see Fig. 11b. The proposed algorithm operates under the assumption that the team of N robots has to cover a bounded area of known dimensions. The height h and the width w of the area to be explored are provided to the robots. Consequently, each stripe has height h and width $w_s = w/N$. Our approach consists of two behaviors: exploration and coverage. The robots initially try to trace the outline of the areas assigned to them in order to become more knowledgeable about the general layout of the free space. The connectivity of the free space is recorded in a graph that consists of the Reeb graph augmented with extra vertices, termed Steiner points.[5] The Steiner points are introduced to the Reeb graph at the beginning, and they are placed at the boundaries of the assigned stripes for each robot. The edges of the graph represent areas of accessible, unexplored space, and each edge belongs to a robot. During the exploration phase, the robots exchange information; if the stripe a robot was assigned is not fully explored, then that robot calls an auction for the task of exploring the remaining area of that stripe. When an area is explored, the robot that owns it, proceeds to cover it using a single-robot coverage algorithm [41].

4.1 Cooperative exploration

During the exploration of the stripe boundary, the robot uses the cycle algorithm developed in single-robot Morse Decomposition [41]. The cycle path is a simple closed path, which mean that, when executing the cycle algorithm, the robot always comes back to the point where it started. This same cycle algorithm is used for both exploration and coverage. Before describing the cycle algorithm, we need to define two terms: lapping and wall following. Lapping is the motion along the slices while wall following is the motion along obstacle boundaries. A simple cycle algorithm execution will consist of forward lapping, forward wall following, reverse lapping,

[5] A point that is not part of the input set of points is called a Steiner point (http://www.nist.gov/dads/HTML/steinerpoint.html).

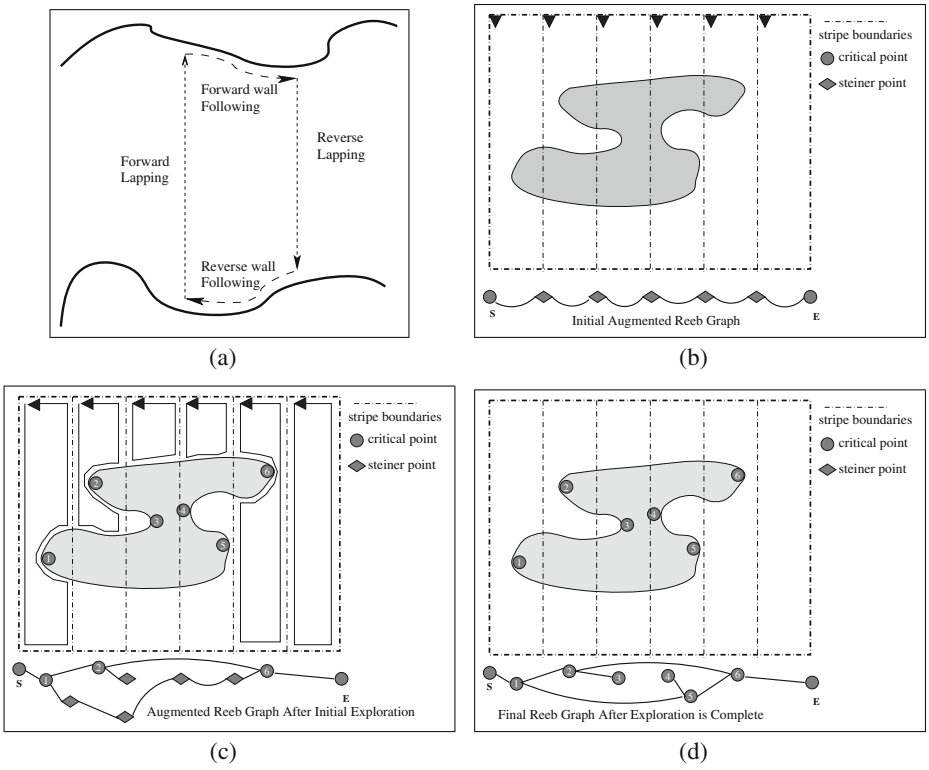


Fig. 11 **a** A simple cycle path consisting of forward lapping, forward wall following, reverse lapping and reverse wall following. **b** Simple environment with initial Augmented Reeb Graph. **c** Initial exploration of stripes. **d** The final Reeb Graph after exploration is complete

and reverse wall following, as shown in Fig. 11a. This is sufficient for exploring the stripe boundary.

To explain the cooperative exploration algorithm, we will look at an example. Figure 11b shows an unknown space with a single obstacle divided into six stripes. The Reeb graph of each robot is initialized with two critical points, labelled as **S**tart and **E**nd, and five Steiner points representing stripe boundaries.

The robots access their respective stripes and perform initial exploration using the cycle algorithm. During exploration, the robots modify their knowledge of the environment by updating the Reeb graph as they discover critical points and new information about Steiner points. After completing a cycle, each robot shares its updated partial Reeb graph with the rest of the robots. At the end of the initial exploration, the updated global Reeb graph is as shown in Fig. 11c.

During the exploration phase, some robots will detect that there are spaces in their stripe that they are not able to reach by moving only inside their stripe. These robots will formulate the unreachable portion of the stripe as an auction task and then call an auction to re-allocate the unreachable part of their stripe to other robots. In this manner, cooperative exploration is achieved. Figure 11d shows the completed Reeb Graph after exploration is complete. When the boundaries of a

stripe are fully explored, the Steiner points are replaced with the actual critical points and edges that represent the current knowledge of free space. Robots that do not have any exploration task can start performing partial coverage of known stripes in order not to waste time. Coverage of a cell is considered an atomic task; thus, a robot that has started covering a cell would finish covering it before starting another task. The global Reeb graph is updated to represent the increased knowledge of the environment.

4.2 Cooperative coverage

After all the stripe boundaries are completely explored, the Reeb graph is updated to represent the connectivity of free space. Even though the inside of each stripe is unknown, the cell connectivity across the stripes is known. Cells that span across the stripe boundaries are divided into two or more, and each cell or partial cell is owned by the robot that explored it or to the robot that won an auction for covering that cell. Therefore, no two robots can start covering the same cell, thus avoiding repeat coverage. The environment is fully represented by the Reeb graph; hence, it is decomposed into a set of connected cells, the union of all cells will represent the free space, and all free space is allocated to the robots. Next, the robots proceed to cover the cells under their charge. Coverage of a single cell is performed using the single robot Morse Decomposition algorithm. While a robot covers an area which is considered a single cell, with the current level of information, obstacles can be discovered. The discovery of obstacles in what was considered a single cell is treated as a single robot coverage task, and the robot responsible for that area, now no more a cell, continues the coverage updating on the fly the Reeb graph and broadcasting the information to the other robots. If there is a robot that is without a task or has completed its coverage task, it calls an auction to offer its service to the other robots. If all the robots have completed their cell coverage, and there are no uncovered cells in the Reeb graph, then the robots return to their starting positions and declare the environment covered.

4.3 Auctioning tasks

A simple auction mechanism is used to investigate the feasibility of auction to enable cooperation among robots. At any auction, a single task is auctioned out. In general, the auction mechanism operates as follows: (a) A robot discovers a new task—e.g. an unexplored part of its stripe or a cell far from the robots current position that needs to be covered. (b) The robot calls an auction for that task with an initial estimated cost; the robot calling the auction is called the auctioneer. (c) Other robots that are free to perform the task at a lower estimated cost, bid for the task. (d) When the auction time ends, the auctioneer selects the robot with the lowest bid and assigns the task. The winning robot adds the task into its task list and confirms that it accepts the task by sending an accept-task message back to the auctioneer. The auctioneer deletes the auction task, and the task auction process concludes. As stated in the previous sections, auction is used in two separate ways: for cooperative exploration and for cooperative coverage.

During exploration, a robot can encounter a situation where the stripe it is exploring is divided into two (or more) disconnected parts because of an obstacle;

see for example the second stripe from left in Fig. 16c. The robot starts with forward lapping, encounters the obstacle, and performs wall following. The wall following behavior brings it to the stripe boundary associated with reverse lapping. As a result, the robot infers that there is an unreachable part of the stripe from the obstacle to the top boundary, which was given to the robot at the beginning. At this point, it will formulate a new stripe-to-be-explored task and calls an auction for this new exploration task.

Please note that the robots generally do not have sufficient information to know accurately the cost of performing the exploration task. It can only estimate the cost based on the available information. Cost is the only parameter that decides the winning robot, and it is thus the factor that determines the quality of cooperation. The estimation of the cost can be potentially a complex function of many variables such as the time spent, fuel expended, priorities of the task, and the capabilities of the robot.

For this investigation, the task cost for the bidder is estimated based on two components: (a) access cost, based on the bidder's estimated current end-point—the point where its currently executing atomic task will end—which is the shortest Manhattan distance to access the new stripe; (b) exploration cost, assuming that the robot can access the desired point in the stripe, exploration cost is the minimum distance that the robot needs to travel in order to explore the stripe completely; as parts of the stripe could have already been explored, the starting point of the exploration could result in different costs for different robots.

When an initial estimate of the cells is available, i.e., exploration is complete, the robot that has explored a cell is initially responsible for covering it. A robot without any tasks will also offer its service by calling an auction. Any robot that has extra cells, less the cell that it is currently covering, will offer one of the cells, based on the auctioneer's position. The robot auctioning its services will use the estimated distance to access each cell as a selection criteria if there are more than one cell on offer.

It is worth noting that, depending on the environment, tasks obtained from Morse Decomposition can be quite coarse. As such, while one robot covers a large cell, other robots, after finishing their tasks, could be idling. The performance of the multi-robot system will be improved if the idling period can be reduced or eliminated. In the next section, we examine an alternative algorithm that decomposes the covering problem to smaller tasks shared among the robots; each task corresponds to a smaller coverage area (two slices).

5 Finer granularity distributed coverage algorithm

When communication is unlimited and the robots are distributed in space, the size of the area each robot covers, in an atomic action, defines the granularity of the algorithm. In the previous section, we presented an algorithm that uses the cells resulting from the Boustrophedon decomposition [8, 41], to assign tasks to the robots. At the other extreme, one can decompose the area using a grid-based decomposition and thus represent the area using a fine resolution occupancy grid. In this case, the task would be for the robots to move over all the unoccupied grids [27, 32]. One logical grid resolution will be size of the robot, as reaching the centroid of the grid cell implies the coverage of that cell. However, besides the problem of partially occupied

grid cells, the coverage paths of multiple robots are also difficult to coordinate and optimize. We believe that using such a fine grid granularity is not an efficient way to decompose the coverage task for multi-robot systems.

In this section, we present a new algorithm, which decomposes the coverage problem into finer resolution tasks than the algorithm described in the previous section. These tasks are "big" enough for a robot to perform uninterrupted for a period of time, yet "small" enough so no robot remains idle while other robots are busy covering a large area. In particular the area is decomposed into fixed width cells; the height of each cell is determined by the boundaries of the area to be covered or by obstacles, and the width of each cell is twice the width of the robots footprint. Thus each cell can be covered by an atomic "cycle algorithm" behavior. This algorithm was used in the previous section during exploration and is illustrated in Fig. 12a, it consists of a simple up and down motion. The robots again are distributed as in Fig. 2, but each robot is just assigned a single cell and not a full stripe. The adjacency graph is used to represent the decomposition of free space. The adjacency graph is of finer resolution than the Reeb graph, and it is conceptually its dual. As such, the nodes in the adjacency graph represent the small two-slice cells, and the edges represent the connectivity. At each node the coordinates of the four corners of the cell are being stored together with a label indicating if the cell is covered or not. In general, each robot after covering a cell updates its internal representation of the adjacency graph, broadcasts the information to the other robots, then, selects the closest uncovered cell relative to its current position and inform the other robots for the selection. Consequently, the robot moves to the selected cell and starts covering again. Please note, in this section the term cell is used as defined above and is different from the definition of the previous sections.

5.1 Cycle algorithm—coverage and detecting changes in connectivity

As we saw in Section 4 the cycle algorithm, proposed by Acar and Choset in [41], consists of lapping and wall following. The difference between [41] and the proposed algorithm is that the robot will not look for Critical Points. Instead, each robot will determine, by the presence of obstacles, if the cell being covered is divided into

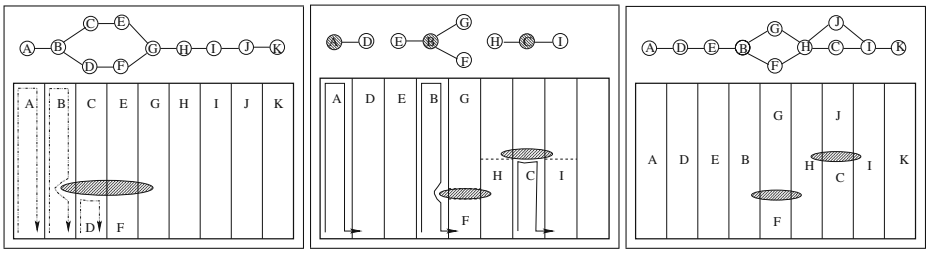


Fig. 12 **a** Cellular Decomposition with fixed size cell width. The *arrow line* shows the path that the robot will take to cover the next cell. *On top* is the Adjacency Graph representation. **b** A different environment, adding new cells to the model after completing the coverage of cells (*A*), (*B*) and (*C*). **c** The model of the entire area of **b** when the algorithm completes

disconnected parts. There are 3 possible scenarios, as can be seen in Fig. 12b, that can occur while the robot is executing the cycle algorithm:

- **Cell with no obstacles** The robot performs the forward and reverse phases without meeting any obstacle. Therefore the robot concludes that the current cell coordinates are correct and updates the graph to indicate that the cell is covered, e.g. cell (A) in Fig. 12b. In addition, the covering robot can also infer that the adjacent cells will have the same height as the current cell. As such, if the new cells are within the area to be covered, and they are not already in the graph, they can be added to the adjacency graph. For example, in Fig. 12b, the cell (D) is added to the right of cell A by the first robot. The current cell (A) and the new cell (D) are connected on the graph. The updated graph is shared with the other robots in the team.
- **Cell with obstacles blocking part of the cell's width** The robot encounters an obstacle partially obstructing its lapping path while performing forward or reverse phase. At the obstacle's location, the robot uses its sensor to determine that the obstacle is blocking only part of the cell's width, e.g., cell (B) in Fig. 12b. After passing the obstacle, the robot will record its location. Then, the obstacle's location will be used as the starting point of the adjacent cell to be added later. The robot continues the coverage of the cell. At the boundary of the bounded area, the robot updates the state of cell (B) to *covered*, and adds cells (E), (F), and (G) to the graph.
- **Cell divided by obstacle** The robot encounters the obstacle during the forward phase. By following the obstacles boundary, the robot moves laterally one cell width. This means that there is an obstacle dividing the cell into two disconnected parts. At the end of the cycle algorithm, the robot updates the state and the size of the current cell, and adds one or two new cells to the adjacency graph. For example, in Fig. 12b, the new cells (H) and (I) added are assumed to have the same height as the current cell (C).

After covering their current cell, each robot will select the closest uncovered cell from the adjacency graph, and will continue coverage. If the closest uncovered cell is not adjacent, the robot would move through the covered space via the shortest path.

When a new cell is added to the adjacency graph, the dimensions of the new cell are assumed to be similar to the current cell if no obstacles are detected. If obstacles are present, the dimensions of the new cell are considered to be limited by the robot's perception of the obstacle's location and size. For example, in Fig. 12c, the heights of cells (H) and (I) are considered to be the same as cell (C) from the second robot's point of view. Therefore, the robot must be able to rectify any wrong assumptions that were made when the cells were first added to the graph. More analytically, after the complete coverage of cell (C), cell (I) is added with a smaller height than actual. Let us assume that the robot moved then to cover cell (I). When the robot reaches the obstacle at cell I, the robot is able to detect that the obstacle is only covering part of cell (I)'s width. The Cycle Algorithm will bring the robot to the upper boundary of the region, and, when the coverage is completed, the robot will be able to correct the height of cell (I) and update the adjacency graph.

Furthermore, when adding a new cell, the robot may find that the cell was already added. In that case, the position and dimensions of the current new cell may be

different from the ones already in the graph. The merging of information about the versions of the same cell is handled as follows:

1. Information of a COVERED cell will be considered as accurate and final. Therefore, information on a COVERED cell will take precedence.
2. If both the cells are in the UNKNOWN state, the information from the two cell versions will be combined, and the final cell in the model will cover the same size as the two cells. An example can be seen for cell (H) of Fig. 12c. The robot that covered cell (C) will insert a new cell that is of the same size as cell (C), and the robot covering cell (G) will insert a new cell that is of the same size as cell (G). The final combined cell in the model will be the union of (C) and (G).

5.2 The cooperation mechanism of the robots

When a robot completes the coverage of a cell, the robot will broadcast the new graph information to the rest of the team. Upon receiving the new graph information, the robots will synchronize their own graphs with the new information. Therefore, all members of the team have a common global picture of the state of the world. Thus, they are able to coordinate their effort to complete the coverage in the shortest possible time. In the event of attrition, the remaining team is able to continue with the coverage without been affected. The loss of communication, however, will result in repeat coverage.

6 Complexity for multi-robot coverage solutions

The task of multi-robot coverage of an unknown environment can be divided into two stages: first, the exploration and partial coverage of the unknown environment that results in the construction of the Reeb graph, and second, the collaborative coverage of the remaining area that corresponds to the uncovered cells in the Reeb graph. Even if there exists an efficient algorithm that provides the Reeb graph required for the second stage, in polynomial time, or the Reeb graph is known, the complexity of computing an optimal solution for the second stage is at least NP-hard. In order to prove that the distributed coverage is at least an NP-hard problem we show that a special case with reduced complexity is equivalent to the multiple Traveling Salesman Problem (mTSP). First, two necessary definition and two lemmas are presented. The TSP is defined using a set of points on a plane and the distances between points, that is the traveling salesman can go from any point to any other point. The equivalent notation of a complete weighted graph is used where the points are vertices and the weight on an edge is equivalent to the distance between the two points connected by the edge.

Definition 6.1 Traveling Salesman Path Problem (TSPP). Given a fully connected weighted graph[6] we want to find the shortest path that visits every vertex in the graph exactly once.[7]

[6] A fully connected graph is a graph for which every vertex is adjacent to every other vertex.

[7] The TSPP differs from the classical TSP because the resulting path finishes at a different vertex than the starting vertex.

Lemma 6.2 *The TSPP is NP-hard.*

Proof We show that the Traveling Salesman Problem (TSP), a classic NP-complete problem, can be reduced to TSPP using a polynomial transformation. Given a fully connected weighted graph $G =< V, E >$ of N vertices (see Fig. 13a for an example of a five vertex graph), the TSP is defined as finding a minimal cost cycle that starts at an arbitrary vertex v_s, visits every vertex in the graph exactly once, and returns to v_s. To show the reduction, we create a new graph:

$$G' =< V - v_s, E - \{e_{s,j}\} : j = 1 \ldots N, j \neq s >$$

that has the same vertices except of the starting vertex v_s and it is still fully connected (see Fig. 13b for an example where $v_s = V1$). Then we calculate the TSPP in the G' $N-1$ times starting at a different vertex every time. Let $P_{be} = [v_b, \ldots v_e]$ denote the TSPP starting at vertex v_b and ending at vertex v_e (see Fig. 13c where $P_{5\,3} = [V5, V4, V2, V3]$) and w_{be} denote the total cost of P_{be}. For every TSPP P_{be} create a new cost:

$$w'_{be} = w_{be} + C(e_{s,b}) + C(e_{e,s})$$

where $C(e_{i,j})$ is the cost of traversing the edge $e_{i,j}$ (see Fig. 13d where $C(e_{1,5}) = C15$ and $C(e_{3,1} = C31)$). From the $N-1$ paths P_{be} we select the one P_{gl} with the minimal cost w'_{gl}. The cycle $[v_s, v_g, \ldots, v_l, v_s]$ has total cost $w_{ss} = w'_{gl}$ and is the solution to the TSP in G.

If we could solve the TSPP in polynomial time, then for every start vertex v_i in G' the TSPP solution P_{ik} and the cost w'_{ik} for every cycle $[v_s, P_{ik}, v_s]$ can be also calculated in polynomial time. Thus the TSP is reduced to the TSPP in polynomial time. As no polynomial solution to the TSP is known, the TSPP must also be NP-hard. □

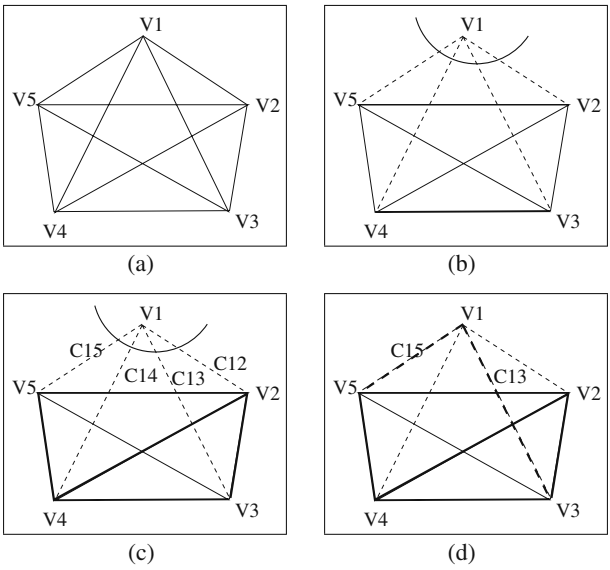
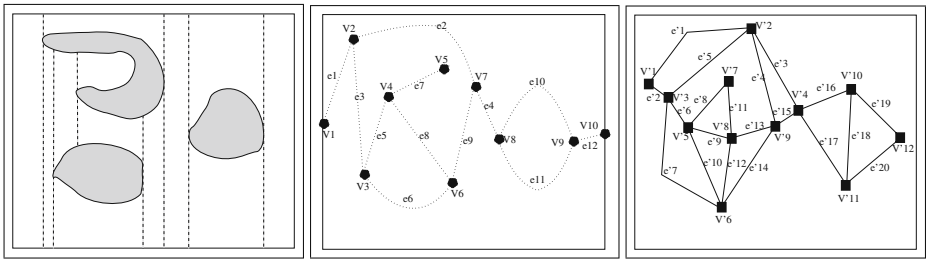


Fig. 13 **a** A fully connected graph with five vertices. **b** Let the starting vertex be V_1. We remove V_1 and all the edges the connect V_1 with the rest of the vertices in the graph. **c** Estimate the TSPP starting at vertex $V5$ (*thick line*) and the cost from $V1$ to every other vertex (V12,. . .,V15). **d** A complete TSP tour [V1,V5,V4,V2,V3,V1]



(a) (b) (c)

Fig. 14 **a** The cellular decomposition of an environment. **b** The corresponding Reeb Graph $G =< V, E >$. **c** The *line graph* of the graph G ($L(G) =< V' E' >$)

Definition 6.3 Multiple Traveling Salesman Path Problem (mTSPP). Given a fully connected graph with N vertices and k starting vertices ($k \leq N$) we want to find k paths such that every vertex in the graph appears in only one of the paths exactly once and the highest cost is minimized.

Lemma 6.4 *The mTSPP is NP-hard.*

Proof We show that the Traveling Salesman Path Problem (TSPP) can be reduced to mTSPP using a polynomial transformation. Given a fully connected graph $G =< V, E >$ of N vertices with a cost function $C(v_i, v_j)$ for every edge $e_{i,j}$ and a starting vertex v_s, we create first a new graph $G' =< V', E' >$ that contains k copies of G. $V' = [V_1, \ldots, V_k]$ and $V_i : i = 1 \ldots k$ is a copy of V. Every vertex $v_i^j \in V_i$ is a copy of the jth vertex of V. $E' = [E_1, \ldots, E_k, E_{\text{br}}$ where $E_i : i = 1 \ldots k$ is a copy of E, and E_{br} is a set of bridging edges that connect every vertex $v_i^j \in V_i$ with every vertex $v_l^q \in V_l$ for $i \neq l$. More formally:

$$e' = [v_j^i, v_q^l] = \left\{ \begin{array}{l} e_{jq} \in E \text{ if } i = l \\ e_{jq}^{il} \in E_{\text{br}} \text{ if } i \neq l \end{array} \right\}$$

The new graph G' is fully connected. The edges between vertices of the same sub-graph G_i maintain the cost they had in G and the cost of the bridging edges is a very large number $C_{\max}$ The cost function $C'(v_i, v_j)$ is then:

$$C'[v_i^j, v_l^q] = \left\{ \begin{array}{l} C(e_{jq}) \text{ if } i = l \\ C_{\max} \text{ if } i \neq l \end{array} \right\}$$

G' is then a fully connected graph whose construction is done in polynomial time.

Let the set of starting vertices for the mTSPP be $v_s^i : i = 1 \ldots k$. Due to the extremely high cost $C_{\max}$ of the bridging edges between sub-graphs, the bridging edges would never be selected in the minimum-cost path; thus, each one of the k TSPP solution paths $P_{ik} : i = 1 \ldots k$ would contain only edges and vertices that belong to the same sub-graph as the starting vertex. If there was an optimal solution of polynomial complexity to the mTSPP then the optimal path of a single sub-graph would be the optimal solution to the TSPP. From Lemma 6.2, TSPP is NP-hard thus mTSPP must also be NP-hard. □

Theorem 6.5 *Finding a minimum-cost path solution to the distributed multi-robot coverage problem as stated is at least NP-hard.*

Proof Let us assume that every cell has the same size; therefore, only the path travelled to visit each cell is affecting the cost. The problem then is reduced to finding the minimum cost (shortest) path each robot must follow in order for all the cells to be visited exactly once.

Given a Reeb graph $G =< V, E >$, let $G' = L(G)$ be the line graph[8] where every cell corresponds to a vertex $v'_i : i = 1 \ldots M$. Without loss of generality we consider the upper left point of each cell as the location of the corresponding vertex of the line graph. We augment the line graph $G' =< V', E' >$ with extra edges such that every vertex is connected with every other vertex; thus G' becomes fully connected. Each edge $e'_i = v'_k v'_l$ is assigned a weight w_i that is proportional to the shortest path through free space that connects the corresponding vertices (v'_k, v'_l). The complexity of assigning the shortest path weights is at worst $O(N^2 M \log M)$[9] where N is the number of graph vertices and M is the number of vertices in a polygonal approximation of the obstacles. Therefore, the complexity of computing all the pairwise shortest paths is polynomial. Finding the minimum-cost path in the fully connected graph for N robots involves solving the multiple Traveling Salesman Path Problem discussed above, which is NP-hard. Therefore, finding an optimal solution to the multi-robot coverage problem is NP-hard. □

In the next section, we present experimental results from different real and simulated environments employing the different algorithms described in this paper.

7 Experimental results

Numerous experiments were conducted for a variety of environments using the robotic simulation package Player/Stage [46, 49] and for different number of robots. Moreover, experiments were conducted using two Pioneer 3AT robots in an underground garage. Next, we briefly present the software architecture used for our implementation; then, we discuss the results from the different algorithms.

7.1 Architecture

For our implementation we adopted a highly distributed system architecture because it can quickly respond to problems involving one or several robots, and it is more robust to point failures and also to the changing dynamics of the system. Our architecture is based on the layered approach that has been used for many single-agent autonomous systems [50, 51]. We are employing two layers for each robot instead of the traditional three layers: planning and behavior. The upper layer

[8] A *line* graph $G' = L(G) =< V', E' >$ of another graph $G =< V, E >$ is a graph with a vertex v'_i for every edge e_i of G, and two vertices v'_i, v'_j are connected *iff* e_i, e_j are adjacent [47] (see Fig. 14b, c).

[9] During the exploration, the boundaries of the obstacles could be mapped using a polygonal approximation with M vertices. The complexity of finding the shortest path for any two points in the free space is $O(M \log M)$ [48] and there are $N(N-1)/2$ paths to be calculated.

consists of the processes *Planner* and *Model* and the lower layer of the process *Behavior*. The process Model maintains the Reeb graph and any other data-structures, e.g. local maps that model the environment. The Planner process implements the Morse Decomposition, auction mechanism, task scheduling, and task monitoring algorithms. The Behavior process serves the same function as in traditional layered architectures, i.e., controlling the robots to perform atomic tasks such as *goto*, *wall following*, and *forward motion* (*Lapping*).

Inter-process communication is facilitated by the Real Time Communication (RTC) package [52]. Finally, the GTL package [53] was used as a base for the Reeb Graph implementation.

7.2 Limited communication experiments in simulated environments

Next, we present an illustrative example for the limited-communication complete-coverage algorithm. Seven robots start together at the lower left corner of an environment, two of them start exploring while the remaining five start covering the free space; see Fig. 15a. When the first obstacle interrupts the line of sight between the two explorers, the existence of a critical point is deduced, see Fig. 15b, and the bottom explorer moves toward the top explorer in order to accurately record the position of the critical point; see Fig. 15c. After the first cell is fully covered, the robots split into two teams. The top cell is covered by a team of three robots, two explorers and one coverer, while the bottom cell is assigned to a team of four robots, two explorers and two coverers; see Fig. 15d, e. While the bottom explorer follows the bottom boundary of the top cell the direction of motion is reversed due to a forward convex critical point. The two explorers mark the end of the top cell and join the one coverer in covering the remaining area; see Fig. 15e. Next, the top team moves in the bottom cell where they rejoin with the bottom team and complete the coverage of the bottom cell; see Fig. 15f. Finally, the unified team proceeds to cover the final cell. It is worth noting that, as the two explorers meet, they detect the final critical point, reverse concave, in Fig. 15h. Figure 15i presents the final step of the completely covered environment.

7.3 Auction-based global-communication experiments in simulated environments

The distributed coverage algorithm was tested in a variety of simulated environments for two to five robots. Next, we are going to examine experimental results for the unrestricted communication scenario with four robots. A sample environment for testing the multi-robot coverage algorithm with unrestricted communication is shown in Fig. 16a. Each of the four robots is allocated a stripe and the planner of each robot receives the stripe information. The planner determines the point where the robots should access the stripe and sends the target-point to the Behavior process for execution. After accessing the stripe, the Behavior process sends a message to the Planner informing the Planner that access to the stripe is completed. Based on the stripe information and the robot pose, the Planner starts the exploration phase and plans for Forward Lapping, then sending this task to the Behavior where it is executed; see Fig. 16c. For this task, the four robots experience different terminating conditions due to the obstacle location in the environment: The left and the two right robots complete the exploration of their stripes without any problems. The

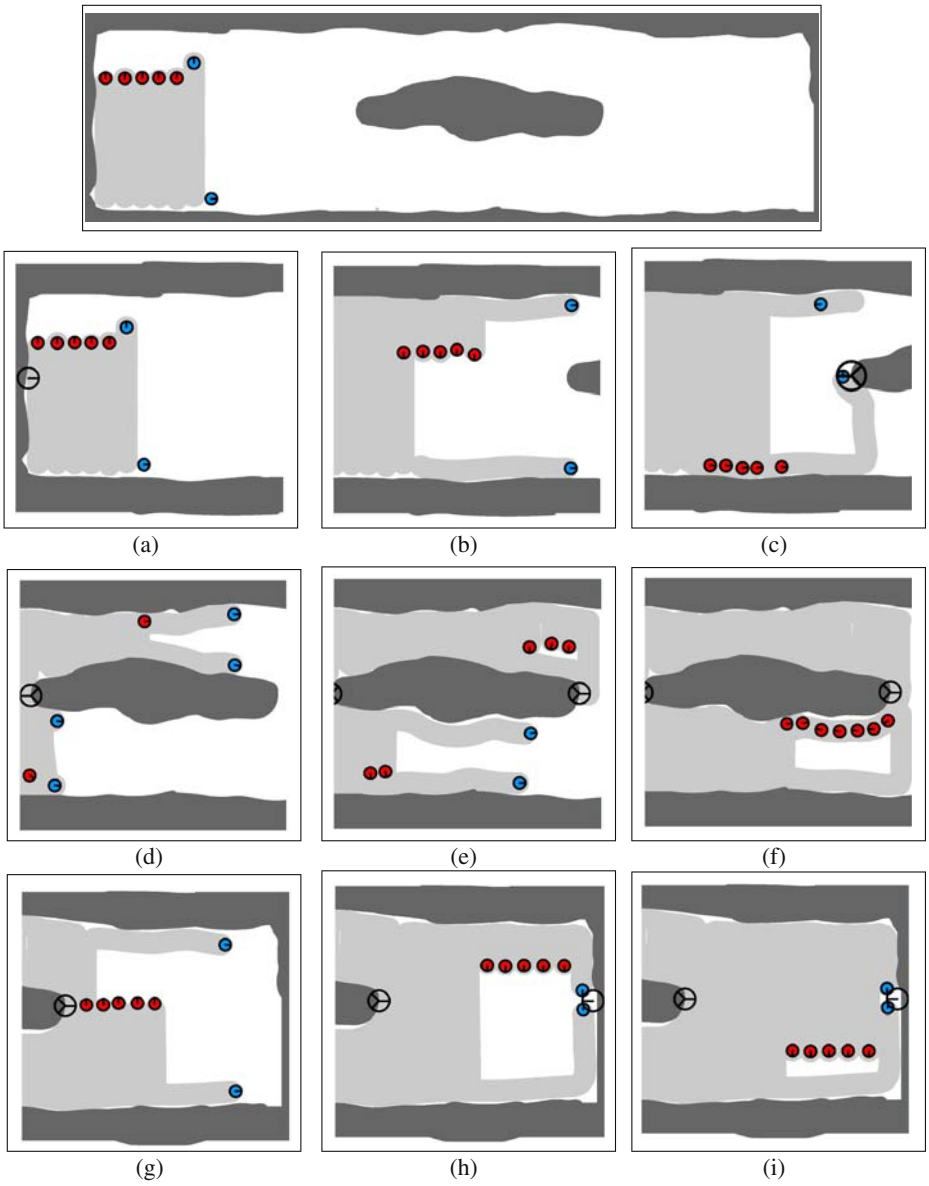


Fig. 15 Formation based coverage of an unknown/unstructured environment: Top figure: the complete environment. **a** The robots start covering and exploring. **b** The two explorers just before line of sight is interrupted. **c** Bottom explorer discovers the reverse convex critical point. **d** Top team has three robots, bottom has four (only three visible). **e** Bottom explorer of the top team discovers forward convex critical point (reversal of x direction) and joins top explorer in coverage task. **f** Rejoining of the two teams to finish covering the bottom cell. **g** One team, explorers and coverers are covering the final cell. **h** The two explorers meet, thus discovering a reverse concave critical point; no more cells to be covered. **i** Completed coverage

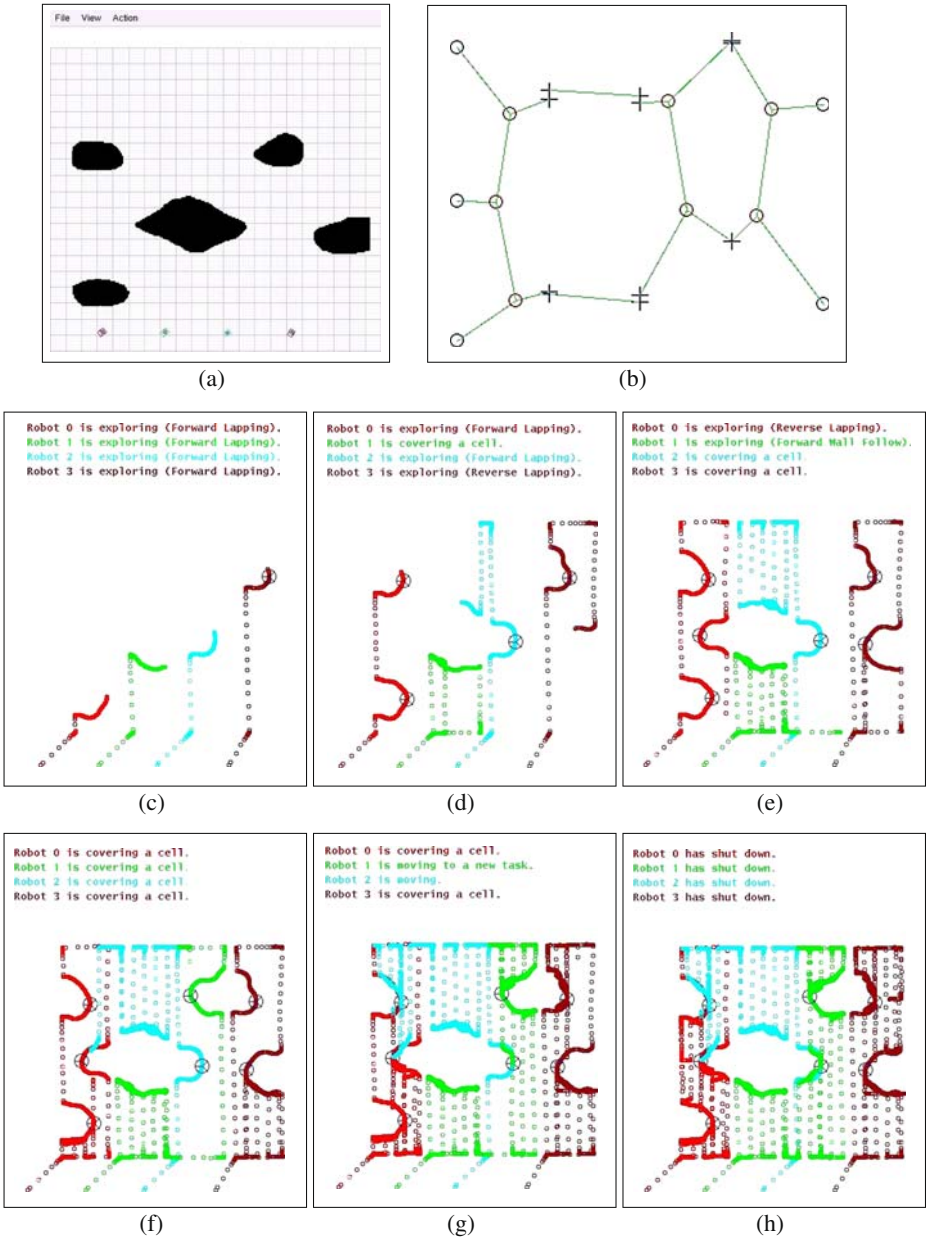


Fig. 16 **a** The environment and the four robots at the starting position in Stage. **b** The augmented Reeb graph with the critical points (*circles*) and the Steiner points (*crosses*). **c**–**h** The traces of the four robots (marked as *circles* which are smaller than the footprint) exploring and covering the environment, and the critical points encountered

middle robot realizes that it can not complete the exploration of its stripe and calls an auction. The robot to its right wins the auction and proceeds to explore the remaining part of the middle stripe. In the mean time, the second left robot starts

partial coverage. Finally, when exploration is completed, the robots exchange cells via auctions and proceed to completely cover the environment. Figure 16b shows the Reeb graph after exploration is completed. Figure 16c–h shows the trace of the four robots plotted as circles. The trace is smaller than the robot footprint for illustration purposes.

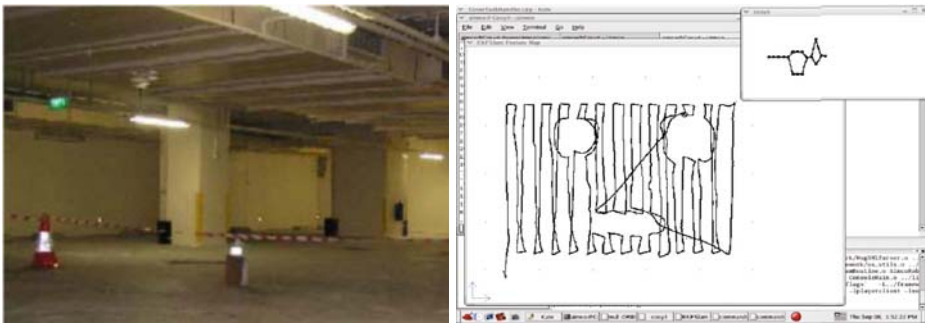
The same environment was used for experiments with three, four and five robots, the respective times to complete coverage were 30, 25.8 and 23.3 min. During our experiments, the robots continuously explored and covered the environment. After a few auctions, it was impossible to predict which task was scheduled next by each robot. It is worth noting, however, that the distance travelled by each robot was approximately the same, thus showing that the workload was distributed evenly.

7.4 Global communication experiments in a real environment

The finer granularity algorithm was tested in simulated environments and with real robots. Two Pioneer 3AT robots equipped with a SICK Laser range finder were used to cover an unknown environment. The laser range finder performed obstacle detection and localization via retro-reflective targets placed in the operating environment.

A series of experiments in a basement parking space were set-up; see Fig. 17a, and [54]. Localization beacons were placed three meters apart around the perimeter of the area to be covered. The area covered by the robot was rectangle 12 m by 9 m. There were three obstacles in the environment, two of which were concrete pillars, and the other one was an artificially added obstacle.

For comparison purposes, a single-robot experiment was performed first. A single Pioneer 3AT robot covered the environment presented in Fig. 17a in 25 min; the path can be seen in Fig. 17b. Then, the multi-robot coverage experiment was conducted using two robots in the same environment, shown in Fig. 17a. Figures 18a–f show a sequence of screen-shots taken during the experiment. In the figures, both the adjacency graph and the coverage trace are shown. The ones on the left of the image belong to Robot 1 while those on the right belong to Robot 2. The robots covered the same area in 13 min and 55 s, with one robot idling for 20 s.



(a) (b)

Fig. 17 **a** The testing environment used for single and multi-robot experiments. **b** The coverage trace and the adjacency graph from a single-robot experiment

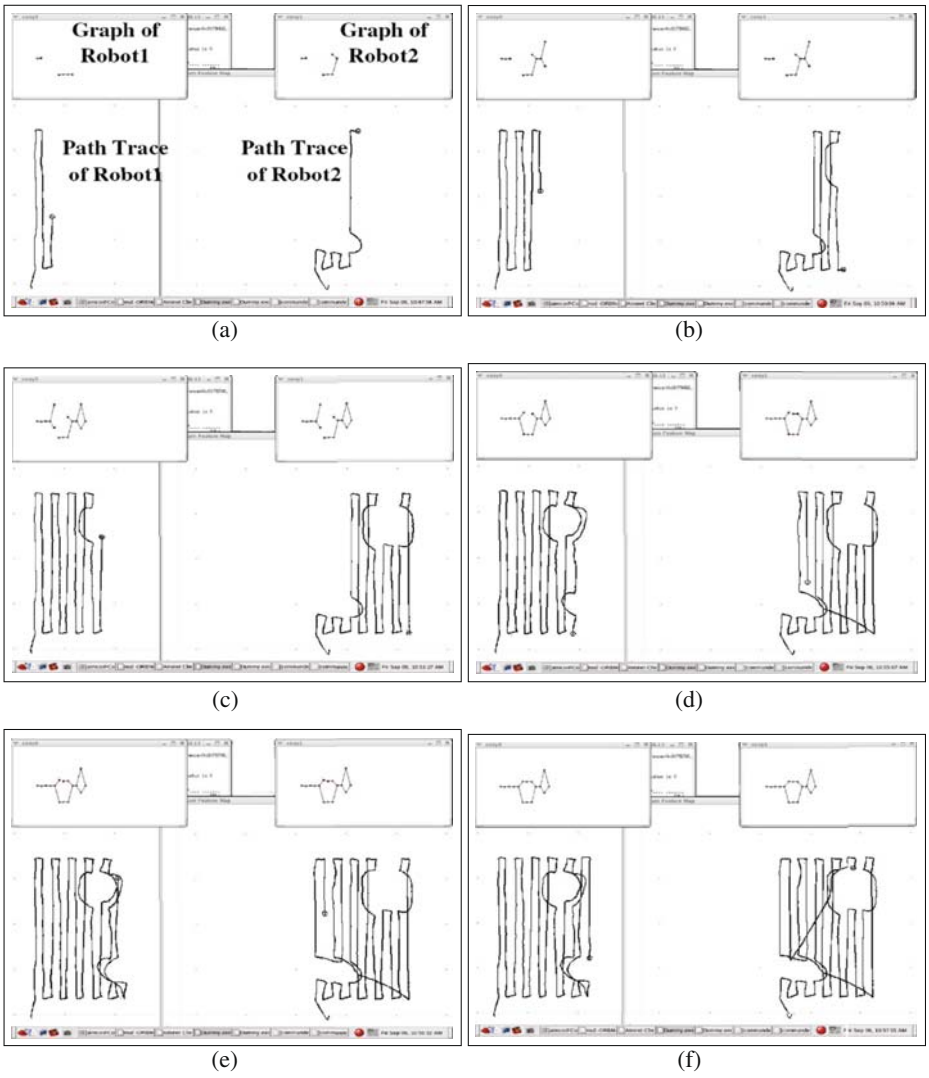


Fig. 18 **a–f** Screen captures from the two robot experiment using the Pioneer 3AT robots. Each screen-shot contains the paths of both robots and their model of the environment in the form of the Reeb Graph. The legend appears only in **a**

Figure 18a shows Robot 1 completing its first cell as Robot 2 completed the two cells below the large obstacle. While covering the second cell, Robot 2 passed by the end of the obstacle and introduced a new cell to the top of the large obstacle, Fig. 18a, b. Then, Robot 2 continues exploring to the right encountering a new obstacle, Fig. 18c. Figure 18d illustrates how the graph for both robots is updated, and the two disconnected branches join when Robot 1 reached the area Robot 2 started covering. In addition, Robot 2 has reached the right boundary of the area and is moving back to the next nearest uncovered cell which was discovered earlier.

Figure 18f shows that Robot 2 completed the last cell, while Robot 1 remained idle for a short period of time.

8 Conclusions

In this paper, we described three new multi-robot coverage algorithms. Our approach employs the Boustrophedon decomposition thus guaranteeing complete coverage. When information sharing is restricted to line-of-sight communication in an unknown environment the robots stay together in teams and cover the free space. Under these communication constraints, we contributed algorithmic multi-robot solutions for both single-cell coverage and Reeb-graph traversal, while trying to minimize repeat coverage. Under the assumption of global communication among the robots, each robot is allocated a bounded area of the unknown environment to cover. An auction mechanism is employed in order to facilitate cooperative behavior among the robots, thus improving their performance. In our approach no robot remains idle while there are areas to be covered. Finally, when unrestricted communication is available, a third algorithm which decomposed the coverage problem to smaller tasks was presented.

The communication ability is central in multi-robot collaboration tasks. In multi-robot coverage, the lack of communication traditionally resulted in repeat coverage that reduced the efficiency of the approach. Maintaining the cohesiveness of the team by allowing only minimal splitting, greatly reduced repeat coverage. The performance of the proposed algorithm depends on the average cell length; if, in most cells, the team does not "fit" then the remaining robots idle, a form of dynamic repeat coverage. When unrestricted communication is available, the robots spread through the environment and coordinate with each other, using an auction mechanism or a simple negotiation protocol. Repeat coverage is thus minimized and all robots are kept occupied. Even though no robot idled for any significant amount of time, in all cases the robots had to travel through covered space in order to reach uncovered areas. This can be seen as a form of repeat coverage, but it differs in two ways: First, a robot can travel through free, known, space in higher speeds than the ones used during coverage; and second, the path travelled through a cell is shorter, to the order of the square root, than the path to cover the same cell.

For future work, we would like to compare the performance between the distributed approach [55] with the team-based approach with limited communication [3]. Also, when there are no communication restrictions, we would like to improve the performance of the auction mechanism. Augmenting the cost function to take into account individual robot capabilities, especially in heterogeneous teams, is an important extension. Accurate localization is a major challenge in mobile robotics; we would like to take advantage of the meeting of the robots in order to improve the localization quality via cooperative localization [56]. Finally, developing more accurate cost estimates for the different tasks is one of the immediate objectives. Coverage is among the most pragmatically significant tasks in mobile robotics. The sale of over two million ROOMBAS, a vacuum cleaning robot developed by iRobot, as well as the interest in humanitarian de-mining clearly highlight the importance of coverage. Our algorithms provide a first principled approach for an efficient and robust solution to the multi-robot complete-coverage problem with

exact cell-decomposition, and open the field for realistic multi-robot applications in the coverage domain.

Acknowledgements We would like to thank Sze Kong Chan for his crucial help in the implementation of the full communication algorithms. Vincent Lee-Shue, Sam Sonne, and Ben Hollis for their valuable input during the early stages of this project. Bernadine Dias, Danni Goldberg, Rob Zlot, and Marc Zink for their help with the Market based approach and the multi-process architecture. Finally, Luiza Solomon provided valuable insights on the software design and an implementation of the graph class. Furthermore, we would like to acknowledge the generous support of the DSO National Laboratories, Singapore; the Office of Naval Research; and the National Science Foundation.

References

1. Choset, H.: Coverage for robotics—a survey of recent results. Ann. Math. Artif. Intell. **31**, 113–126 (2001)
2. Latimer-IV, D., Srinivasa, S., Lee-Shue, V., Sonne, S.S., Choset, H., Hurst, A.: Toward sensor based coverage with robot teams. In: Proc. 2002 IEEE International Conference on Robotics & Automation (2002)
3. Rekleitis, I., Lee-Shue, V., New, A.P., Choset, H.: Limited communication, multi-robot team based coverage. In: IEEE International Conference on Robotics and Automation, pp. 3462–3468. New Orleasn, LA (2004)
4. Dias, M.B., Stentz, A.T.: A market approach to multirobot coordination. Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI -TR-01-26 (2001)
5. Choset, H., Pignon, P.: Coverage path planning: the Boustrophedon cellular decomposition. In: International Conference on Field and Service Robotics, Canberra, Australia (1997)
6. Canny, J.: Constructing roadmaps of semi-algebraic sets i : completeness. Artif. Intell. **37**, 203–222 (1988)
7. Canny, J., Lin, M.: An opportunistic global path planner. Algorithmica **10**, 102–120 (1993)
8. Acar, E.U., Choset, H.: Critical point sensing in unknown environments. In: Proc. of the IEEE International Conference on Robotics & Automation (2000)
9. Oh, J.S., Park, J.B., Choi, Y.H.: Complete coverage navigation of clean robot based on triangularcell map. In: IEEE International Symposium on Industrial Electronics, vol. 3, pp. 2089–2093. Pusan, South Korea (2001)
10. Butler, Z.: CC R: a complete algorithm for contact-sensor based coverage of rectilinear environments. The Robotics Institute, Carnegie Mellon University, Tech. Rep. CMU-RI-TR-98-27 (1998)
11. Butler, Z., Rizzi, A., Hollis, R.: Distributed coverage of rectilinear environments. In: Proc. of the Workshop on the Algorithmic Foundations of Robotics (2001)
12. Kurabayashi, D., Ota, J., Arai, T., Yoshida, E.: An algorithm of dividing a work area to multiple mobile robots. In: 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems, 5–9 Aug 1995, vol. 2, pp. 286–291 (1995)
13. Kurabayashi, D., Ota, J., Arai, T., Yoshida, E.: Cooperative sweeping by multiple mobile robots. In: 1996 IEEE International Conference on Robotics and Automation, 22–28 April 1996, vol. 2, pp. 1744–1749 (1996)
14. Min, T.W., Yin, H.K.: A decentralized approach for cooperative sweeping by multiple mobile robots. In: 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems, 13–17 Oct. 1998, vol. 1, pp. 380–385 (1998)
15. Luo, C., Yang, S.: A real-time cooperative sweeping strategy for multiple cleaning robots. In: IEEE Internatinal Symposium on Intelligent Control, pp. 660–665 (2002)
16. Luo, C., Yang, S., Stacey, D.: Real-time path planning with deadlock avoidance of multiple cleaning robots. In: Proceedings 2003 IEEE International Conference on Robotics and Automation (ICRA), 14–19 Sept. 2003, vol. 3, pp. 4080–4085 (2003)
17. Ichikawa, S., Hara, F.: Characteristics of object-searching and object-fetching behaviors of multi-robot system using local communication. In: IEEE International Conference on Systems, Man, and Cybernetics, (IEEE SMC '99), vol. 4, pp. 775–781 (1999)
18. Wagner, I., Lindenbaum, M., Bruckstein, A.: Distributed covering by ant-robots using evaporating traces. IEEE Trans. Robot. Autom. **15**(5), 918–933 (1999)

19. Wagner, I.A., Lindenbaum, M., Bruckstein, A.M.: Mac vs. pc—determinism and randomness as complementary approaches to robotic exploration of continuous unknown domains. Int. J. Rob. Res. **19**(1), 12–31 (2000)
20. Wagner, I.A., Altshuler, Y., Yanovski, V., Bruckstein, A.M.: Cooperative cleaners: a study in ant robotics. Int. J. Rob. Res. **27**(1), 127–151 (2008)
21. Menezes, R., Martins, F., Vieira, F.E., Silva, R., Braga, M.: A model for terrain coverage inspired by ant's alarm pheromones. In: Proceedings of the 2007 ACM symposium on Applied computing (SAC07), pp. 728–732. New York, NY, USA: ACM (2007)
22. Koenig, S., Szymanski, B.K., Liu, Y.: Efficient and inefficient ant coverage methods. Ann. Math. Artif. Intell. **31**(1–4), 41–76 (2001). [Online]. Available: citeseer.ist.psu.edu/553035.html
23. Ge, S.S., Fua, C.: Complete multi-robot coverage of unknown environments with minimum repeated coverage. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA), 18–22 April 2005, pp. 715–720 (2005)
24. Bruemmer, D.J., Dudenhoeffer, D.D., Anderson, M.O., McKay, M.D.: A robotic swarm for spill finding and perimeter formation. International Conference on Nuclear and Hazardous Waste Management, Spectrum (2002)
25. Batalin, M.A., Sukhatme, G.S.: Spreading out: a local approach to multi-robot coverage. In: 6th International Symposium on Distributed Autonomous Robotics Systems, Fukuoka, Japan (2002), June
26. Spears, D., Kerr, W., Spears, W.: Physics-based robot swarms for coverage problems. Int. J. Intell. Control Syst. **11**(3), 124–140 (2006)
27. Hazon, N., Kaminka, G.: Redundancy, efficiency and robustness in multi-robot coverage. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA), 18–22 April 2005, pp. 735–741 (2005)
28. Hazon, N., Mieli, F., Kaminka, G.: Towards robust on-line multi-robot coverage. In: Proceedings 2006 IEEE International Conference on Robotics and Automation (ICRA), 15–19 May 2006, pp. 1710–1715 (2006)
29. Agmon, N., Hazon, N., Kaminka, G.: Constructing spanning trees for efficient multi-robot coverage. In: Proceedings 2006 IEEE International Conference on Robotics and Automation (ICRA), 15–19 May 2006, pp. 1698–1703 (2006)
30. Solanas, A., Garcia, M.: Coordinated multi-robot exploration through unsupervised clustering of unknown space. In: Proceedings 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1, 28 Sept–2 Oct. 2004, pp. 717–721 (2004)
31. Williams, K., Burdick, J.: Multi-robot boundary coverage with plan revision. In: Proceedings 2006 IEEE International Conference on Robotics and Automation (ICRA), 15–19 May 2006, pp. 1716–1723 (2006)
32. Zheng, X., Jain, S., Koenig, S., Kempe, D.: Multi-robot forest coverage. In: Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), pp. 3852–3857. Edmonton Alberta, Canada (2005)
33. Dias, M.B., Zlot, R., Kalra, N., Stentz, A.: Market-based multirobot coordination: a survey and analysis. Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, Tech. Rep. CMU-RI-TR-05-13 (2005)
34. Goldberg, D., Cicirello, V., Dias, M.B., Simmons, R., Smith, S.F., Stenz, A.: Market-based multi-robot planning in a distributed layered architecture. In: Multi-Robot Systems: From Swarms to Intelligent Automata: Proceedings from the 2003 International Workshop on Multi-Robot Systems, vol. 2, pp. 27–38. Kluwer Academic, Boston (2003)
35. Berhault, M., Huang, H., Keskinocak, P., Koenig, S., Elmaghraby, W., Griffin, P., Kleywegt, A.: Robot exploration with combinatorial auctions. In: IEEE/RSJ Int. Conference on Intelligent Robots and Systems, vol. 2, pp. 1957–1962 (2003)
36. Dias, M.B., Stentz, A.T.: Traderbots: a market-based approach for resource, role, and task allocation in multirobot coordination. Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-03-19 (2003)
37. Dias, M.B., Zinck, M., Zlot, R., Stentz, A.T.: Robust multirobot coordination in dynamic environments. In: International Conference on Robotics & Automation, pp. 3435–3442. New Orleans, LA (2004)
38. Gerkey, B., Mataric, M.: Sold!: auction methods for multirobot coordination. IEEE Trans. Robot. Autom. **18**(5), 758–768 (2002)
39. Zlot, R., Stentz, A.: Market-based multirobot coordination for complex tasks. International Journal of Robotics Research Special Issue on the 5th International Conference on Field and Service Robotics, vol. 25, no. 1, (2006)

40. Acar, E.U., Choset, H., Rizzi, A.A., Atkar, P.N., Hull, D.: Morse decompositions for coverage tasks. Int. J. Rob. Res. **21**(4), 331–344 (2002)
41. Acar, E.U., Choset, H.: Sensor-based coverage of unknown environments: Incremental construction of morse decompositions. Int. J. Rob. Res. **21**(4), 345–366 (2002)
42. Fomenko, A., Kunii, T.L.: Topological Modeling for Visualization. Tokio: Springer, New York (1997)
43. Rekleitis, I.M., Dudek, G., Milios, E.: Multi-robot collaboration for robust exploration. Ann. Math. Artif. Intell. **31**(1–4), 7–40 (2001)
44. Dellaert, F., Alegre, F., Martinson, E.B.: Intrinsic localization and mapping with 2 applications: diffusion mapping and marco polo localization. In: Proceedings of the 2003 IEEE International Conference on Robotics & Automation, Taipei, Taiwan, 14–19 September 2003, pp. 2344–2349 (2003)
45. Grabowski, R., P.Khosla, Choset, H.: Autonomous exploration via regions of interest. In: IEEE/RSJ Int. Conference on Intelligent Robots and Systems, vol. 1 (2003)
46. Vaughan, R.T.: Stage: a multiple robot simulator. Institute for Robotics and Intelligent Systems, School of Engineering, University of Southern California, Tech. Rep. IRIS-00-394 (2000)
47. Beineke, L.W., Wilson, R.J.: Selected Topics in Graph Theory 3. Academic, Boston (1988)
48. Hershberger, J., Suri, S.: An optimal algorithm for Euclidean shortest paths in the plane. SIAM J. Comput. **28**(6), 2215–2256 (1999)
49. Gerkey, B.P., Vaughan, R.T., Sty, K., Howard, A., Sukhatme, G.S., Mataric, M.J.: Most valuable player: a robot device server for distributed control. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2001), October 29–November 3 2001, pp. 1226–1231. Wailea, Hawaii (2001)
50. Schreckenghost, D., Bonasso, P., Kortenkamp, D., Ryan, D.: Three tier architecture for controlling space life support systems. In: IEEE Int. Joint Symposia on Intelligence and Systems, 21–23 May 1998, pp. 195–201 (1998)
51. Wagner, M., Apostolopoulos, D., Shillcutt, K., Shamah, B., Simmons, R., Whittaker, W.: The science autonomy system of the nomad robot. In: IEEE International Conference on Robotics and Automation, vol. 2, pp. 1742–1749 (2001)
52. Rtc, RTC, CMU, Pittsburgh, Pa (2000)
53. Raitner, M.: GTL—Graph Template Library Documentation, 1st edn. University of Passau–FMI–Theoretical Computer Science (2002)
54. Kong, C.S., New, A.P., Rekleitis, I.: Distributed coverage with multi-robot system. in Proc. of the IEEE International Conference on Robotics and Automation, pp. 2423–2429. Orlando, Florida (2006)
55. Rekleitis, I.M., New, A.P., Choset, H.: Distributed coverage of unknown/unstructured environments by mobile sensor networks. In: Schultz, A.C., Parker, L.E., Schneider, F. (eds.) 3rd International NRL Workshop on Multi-Robot Systems, 14–16 March 2005, pp. 145–155. Kluwer, Washington, D.C. (2005)
56. Roumeliotis, S.I., Rekleitis, I.M.: Propagation of uncertainty in cooperative multirobot localization: analysis and experimental results. Auton. Robots **17**(1), 41–54 (2004)