Coverage Planning with Finite Resources

Grant P. Strimel
Computer Science Department
Carnegie Mellon University
gstrimel@andrew.cmu.edu

Manuela M. Veloso
Computer Science Department
Carnegie Mellon University
mmv@cs.cmu.edu

Abstract— **The robot coverage problem, a common planning problem, consists of finding a motion path for the robot that passes over all points in a given area or space. In many robotic applications involving coverage, e.g., industrial cleaning, mine sweeping, and agricultural operations, the desired coverage area is large and of arbitrary layout. In this work, we address the real problem of planning for coverage when the robot has limited battery or fuel, which restricts the length of travel of the robot before needing to be serviced. We introduce a new sweeping planning algorithm, which builds upon the boustrophedon cellular decomposition coverage algorithm to include a fixed fuel or battery capacity of the robot. We prove the algorithm is complete and show illustrative examples of the planned coverage outcome in a real building floor map.**

I. INTRODUCTION

We have extensively experienced mobile indoor robots that are capable of accurately navigating in buildings and performing service tasks, such as transporting items or accompanying people to locations [1]. Given their accurate localization and navigation, and their reliable motion planning, we investigate their service to further include a complete sweeping task. In this work, we address a robot space coverage problem.

Coverage planning has been commonly studied in robotics (e.g., [2]). The goal is to plan a path in which the robot covers all points in a given map, i.e., the robot's work space. Many approaches have been explored for a variety of applications, including item search [3], floor cleaning [4], large scale agriculture [5], mowing and milling [6], and painting [7]. In all of these applications, it is essential that the robot path or sensor paths are guaranteed to cover the surface in a robust and efficient manner to complete the objective.

These tasks inspire various methods in addressing the coverage problem. Techniques used can be categorized by how they address altering objectives. Some works only require the robot's sensors to cover an area while many force the robot base to pass over the entire region [8]. There is much work on randomized approaches without knowledge of the environment [9] as well as approaches with a predefined map like those which use cellular decomposition [10], [11], [12].

Often these approaches are motivated to minimize some objective. A common choice is to minimize the total distance traveled during the cover of the area. However, finding the optimal route in this regard is an NP-Hard problem. This can be seen by its close relation to solving the geometric Traveling Salesman Problem (TSP) with neighborhoods [13]. Hence, approximation heuristics are often used. Even through a cellular decomposition where the problem is broken down into smaller regions, planning an efficient tour through the regions requires some form of approximation. Other optimizations like minimizing the number of turns required in a decomposition have also been studied [14].

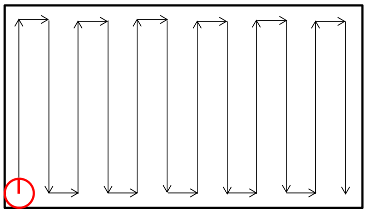


Fig. 1: Back-and-forth ox-plow motions.

In this work, motivated by our real robot, we investigate an additional component to the coverage problems by incorporating a consideration for a fixed battery or fuel source. Accounting for the limited battery life is important, as in many applications, the area to cover is too large for the robot to completely cover in a single non-interrupted charge of a battery. Coverage planning with energy constraints and timing restrictions has been addressed in previous works [15] [16] [17]. In [15] and [16] the works consider the problem of sensor based multi-robot coverage in narrow environments and use a hueristic algorithm to reduce the number of robots needed under energy constraints. [17] considers a multi-robot deployment problem to determine the number of groups unloaded by a carrier, the number of robots in each group and the initial locations of those robots for coverage tasks under both timing and energy constraints.

We focus here on coverage planning with a single robot and a fixed energy capacity and present a heuristic algorithm to reduce total distances traveled. To investigate this problem, we also assume that the space has a service station located in the map of the space. We contribute a new *battery-constrained* sweep algorithm (*BC Sweep*), which extends the boustrophedon cellular decomposition coverage algorithm to reason about a battery capacity constraint. Under the assumption that the robot charge is enough to at least navigate to any cell of the map and return back to the service station, the *BC Sweep* algorithm is guaranteed to return a complete sweep. We prove that this the case, and show illustrative examples of the fully-implemented algorithm in a realistic simulation

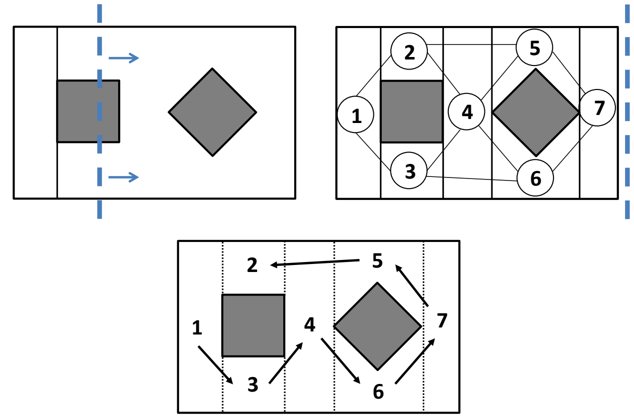


Fig. 2: A boustrophedon decomposition example with sweep line, graph construction from cells, and resulting path through regions.

environment.

II. BOUSTROPHEDON CELLULAR DECOMPOSITION

Our algorithm makes use of the boustrophedon cellular decomposition [11] [12] for bounded planar environments with obstacles. This decomposition breaks the map into disjoint regions called *cells*. The individual cells can be covered simply by back-and-forth or "ox-plow" motions. See Figure 1. To cover the entire free space or map, a tour through each region is constructed and the robot visits and covers the cells sequentially along this tour.

Specifically, the boustrophedon decomposition uses a verticle line sweep approach to construct the cells. The *slice* sweeps from left to right across the map. At any point, if the continuity of the sweep line changes count, then a new cell is spawned or two adjacent cells are merged. In the case of connectivity increasing, a new cell is added. For the instances when connectivity decreases then adjacent cells are merged [11] [12].

After decomposition, the algorithm constructs some graph (commonly a complete or adjacency graph) between the regions. Finding a minimal cost tour through all regions reduces to solving the TSP on the graph. An approximation algorithm is used as a heuristic to solve for a reasonable tour. Figure 2 shows an example decomposition.

Our algorithm makes use of this decomposition technique to create cellular regions and extends touring the regions to account for a fixed fuel or battery life.

III. PROBLEM FORMULATION

Our problem formulation and hueristic consider the total distance the robot travels and total energy used in relation to a fixed battery life or fuel capacity. We consider this extra constraint during path planning coverage.

We make the assumption that given some finite path for a robot, an estimate can be made on the energy used while executing the path. We denote this estimation function f.

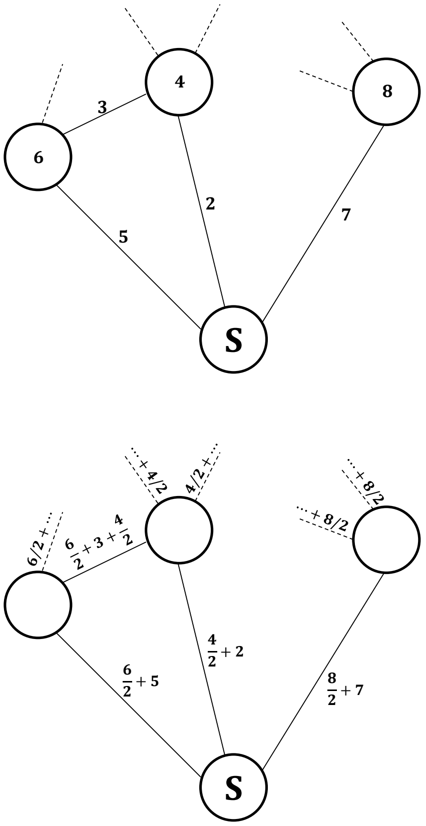


Fig. 3: *(Top)* graph cells (nodes) with intra & inter cell fuel costs *(Bottom)* modified graph G of *BC Sweep* algorithm

The problem formulation can be stated as follows: Given a two dimensional map M with a service center location **S**, a fuel capacity λ, and a fuel consumption function f, plan a route such that the robot covers the map and respects the robot's fuel capacity constraint.

IV. BC SWEEP

Again, we presume that given some route r of the robot that we have a function f such that $f(r)$ is an estimation of the energy used over that route. Often this is directly related to the length of r and could also incorporate the turns (could be expensive). f must also be linear (ie $f(r_1 \rightarrow r_2) = f(r_1) + f(r_{12}^*) + f(r_2)$) where r_{12}^* is the shortest direct route connecting r_1 to r_2. We present now the *BC Sweep* algorithm. The intuition of *BC Sweep* is straight foward. We construct a graph expressing the cellular decomposition of the space with a refueling location. The representation accounts for the costs of covering each cell and traveling between them. The algorithm then simply approximates a minimum cost walk through the graph which circles back to the service station when necessary to refuel. Our approach needs the minimum requirement our fuel capacity λ is large enough that the robot can depart from **S**, cover any cell and return without running out of fuel. Under these conditions we have the following algorithm.

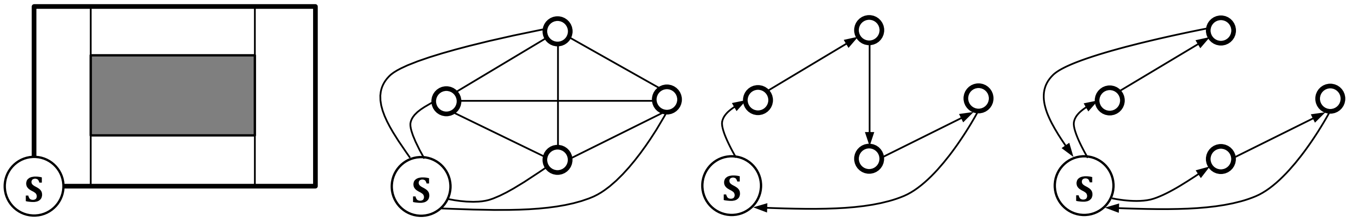


Fig. 4: Steps 1-4 (*left to right*) of the *BC Sweep* Algorithm

BC Sweep:

1) Perform boustrophedon decomposition on the map M into cell set X. Plan all back-and-forth cover-paths $\{r_1,...,r_n\}$ for respective cells $\{x_1,...,x_n\}$. For simplicity, we assume all cover-paths begin and end in the same location. Add a special cell x_0 of 0 size and a null cover-path r_0 representing the service station **S**.

2) Construct a complete graph $G = (V,E)$ between all cells including the service station x_0. Thus $V = \{x_0,x_1,...,x_n\}$. For every edge $e_{ij} \in E$ let $r^*_{ij} = r^*_{ji}$ be the shortest direct route on M between cell (or station) i and j. Assign the weight $w(e_{ij}) = f(r^*_{ij}) + \frac{1}{2}(f(r_i) + f(r_j))$ to each edge. This procedure is shown in Figure 3.

3) Use Christofides Algorithm [18] to generate a TSP tour $T = x^T_0 - x^T_1 - \cdots - x^T_n - x^T_0$ starting and ending at x_0 (ie $x_0 = x^T_0$).

4) We now *optimally* partition T into subroutes which meet the fuel capacity constraint. Define a cost matrix C as follows. $\forall\ i,j \in \{0,...,n\}$

$$C_{ij} = \begin{cases} w(x^T_0 - x^T_{i+1} - \cdots - x^T_j - x^T_0) & \\ & \text{if this cost} \\ & \text{is } \leq \lambda \text{ and } i < j \\ \infty & \text{otherwise} \end{cases}$$

This cost matrix defines a new directed graph H. We now use Dijkstra's algorithm on H to find the shortest path from *node* 0 to *node n*. Since each edge of this path represents a subroute, we append these subroutes together to get our tour. This gives us the shortest route using T which abides by the battery capacity contraint.

The *BC Sweep* steps are shown in Figure 4.

Steps 3) and 4) can be seen as an approximation of a reduction to the *distance contrained vehicle routing problem.* Vehicle routing with contraints is a variant of TSP and has been studied in several works including [19] [20] [21] [22]. The heuristic used here is described and analyzed in [19] and [21].

This heuristic we used to produce our route through G is an α-approximation of optimal with respect to total fuel used where

$$\alpha = 1 + (1.5)\frac{\lambda/d_m}{\lambda/d_m - 2}$$

and $d_m = \max\{f(r^*_{01}), f(r^*_{02}), ..., f(r^*_{0n})\}$ [21]. The α bound for the heuristic becomes ample when $\lambda \gg 2d_m$ [21].

We present the following theorem on completeness and correctness.

Theorem. *BC Sweep covers M and obeys the fuel capacity constraint.*

Proof. By boustrophedon decomposition if each cell is visited, it will be covered. We show that each cell is visited and obeys the fuel constraint. When G is constructed, we add half of every cell's covering fuel cost to all incident edges of that cell. See Figure 3. Hence any path which passes through the cell will pick up half the weight on the way in and the other half on the way out. Because of this, our TSP tour T accounts for all cell costs. With all fuel costs accounted for in T and H giving infinite weight to any subroutes violating the fuel constraint, our final route abides by the constraint. Note because we assumed a feasible solution exists, a finite cost path will always be possible. Since our final route is the concatenation of adjacent subroutes beginning and ending at x_0, the route visits all cells. □

V. ATOMIC REGIONS

One of the nice aspects of this algorithm as its stands is that if f does not underestimate energy usage and navigation is flawless, then each of the decomposed cells will be covered in an *atomic* nature. By atomic we mean the covering of a cell will not be interrupted by a need for a recharge. This is a useful feature in many applications where a room or designated area must be swept or covered all at once with no intermission guaranteed.

However, there are applications where every decomposed cell need not have an atomic covering. In cases like this, one can optimize the above algorithm to make less service trips and only perform them in designated regions. We extend the original *BC Sweep* to handle the scenario where there is a set of cells A which we want to be atomically covered and a set of cells B which does not have this constraint. Note $A \cup B = X$ and $A \cap B = \emptyset$.

We need only redefine $G = (V,E)$ and cost matrix C and the rest of the algorithm remains the same.

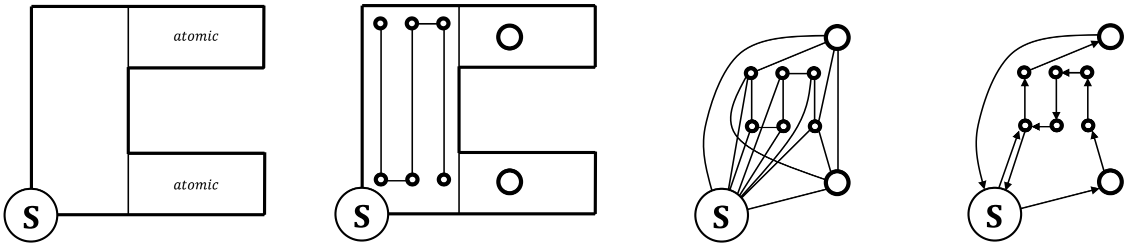


Fig. 5: Modified *BC Sweep* illustration for non-atomic regions. The environment contains two atomic regions and a single interruptable region.

Define $G_1 = (V_1, E_1)$ to be

- x_0 is a vertex in G_1.
- All cells $x_k \in A$ are vertices in G_1.
- For every ox-plow cover-path $y_1^k \to \cdots \to y_m^k = r_k$ corresponding to $x_k \in B$, the first and last nodes become vertices: $y_1^k, y_m^k \in V_1$. Like the depot, y_1^k, y_m^k will have a null cover-paths.
- There is an edge between all atomic cell vertices. There is an edge between all atomic cell vertices and start/finish cover-path vertices.
- Assign the weight $w(e_{ij}) = f(r_{ij}^*) + \frac{1}{2}(f(r_i) + f(r_j))$ to each edge.

Define $G_2 = (V_2, E_2)$ to be

- x_0 is a vertex in G_2.
- For all cover-paths $y_1^k \to \cdots \to y_m^k = r_k$ corresponding to $x_k \in B$, all nodes become vertices: $\{y_1^k, ..., y_m^k\} \subseteq V_2$.
- For every vertex y_i^k there is an edge e between y_i^k and y_{i+1}^k with weight $w(e) = f(y_i^k \to y_{i+1}^k)$.
- For every vertex y_i^k there is an edge e between x_0 and y_i^k with weight of the shortest path between the two.

Our final graph G is the union of these two graphs: $G = G_1 \cup G_2$.

We must now slightly change our cost matrix C after we approximate a TSP tour $T = v_0^T - v_1^T - \cdots - v_0^T$ on G.

$$C_{ij} = \begin{cases} w(v_0^T - v_i^T - \cdots - v_j^T - v_0^T) & \text{if this cost is } \leq \lambda \text{ and } i < j \text{ and } v_i^T \text{ is spawned from } B \\ w(v_0^T - v_{i+1}^T - \cdots - v_j^T - v_0^T) & \textbf{else} \text{ if this cost is } \leq \lambda \text{ and } i < j \\ \infty & \text{otherwise} \end{cases}$$

This change in C is necessary because in the non-atomic regions, the robot must return to the same spot it left off so that it may complete coverage in that area instead of advancing to the next region.

We now simply plan the entire route by running *BC Sweep* on the newly defined G and C. The route respects the λ-capacity constraint and guarantees regions $x_k \in A$ remain atomic. Figure 5 demonstrates the modified approach.

Theorem. *The modified BC Sweep covers M, obeys the fuel capacity constraint, and does not service in atomic regions.*

Proof. The algorithm behaves the same in the atomic regions so the proof follows through in the same maner as the original theorem for atomic regions. It is only necessary to argue that complete coverage occurs in non-atomic regions. Since the cost matrix C was constructed in a manner that if a non-atomic cover path was divided by a service trip that the tour would return to the same node after refueling, we know that no legs of the sweep will be skipped. And since we assumed a feasible solution, after resuming sweeping, progress will always be made in non-atomic regions until complete. Thus complete coverage occurs in non-atomic regions. □

VI. DYNAMIC BC SWEEP

The theoretic algorithms proposed so far are not entirely feasible in practice. *BC Sweep* leveraged unrealistic liberties when planning its covering route.

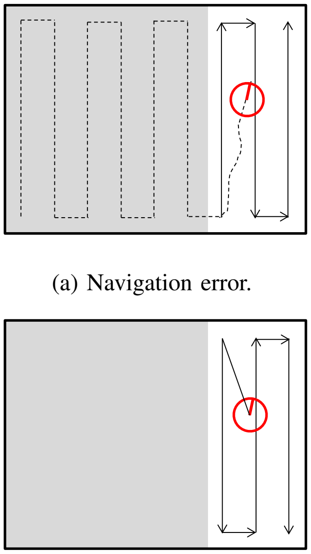
First, we assumed that when in operation, the robot has perfect navigation in the environment. Due to uncertainty in sensors, actuators and imperfect navigation algorithms, this is an impractical assumption.

Second, *BC Sweep* relies on a function f which does not underestimate the energy used over a given path. Though one could exaggerate f to meet this requirement, the more accurate f is, the more energy efficient and coverage effective *BC Sweep* becomes.

To relax some of these assumptions, we propose extensions to the *BC Sweep* algorithm to be more reliable in practice. The key here is that the algorithm needs to be adaptive while executing.

A. Variability in Navigation

To account for imperfection in navigation while executing a sweeping route, we dynamically adjust the route taken. Consider the scenario where while performing the back-and-forth motions in a cell, the robot drifts along one of the lengths. Figure 6a. If the navigation recognizes this, we can modify the route to cover the misssing area. We have seen robot navigation capable of this awareness using



(a) Navigation error.

(b) Navigation error recovery.

Fig. 6: Variability in navigation.

surrounding features (walls, tables, chairs, etc) [23].Thus we can dynamically adjust our *BC Sweep* route. Figure 6b. In such a case, we note that it is possible the atomic regions may need to be interrupted depending on how much rerouting is necessary. Denoting Q_λ to be the current fuel life, the online dynamic algorithm:

```
while covering cell $x_k$ do
    if off path then
        recalc. ox-plow path $r$ for the remainder of $x_k$;
        if $x_k$ is atomic and $f(r) + f(x_k \to x_0) > Q_\lambda$ then
            make $x_k$ interruptable;
        end
        rerun BC Sweep;
    end
end
```

B. Updating f

One of the more difficult aspects of *BC Sweep* is determining the energy consumption function f. Any offline theoretic function f estimate could change depending where you are in the route, map, or on any other factors. To account for a variable function f, it is possible to recalculate an f estimate dynamically while executing the sweeping. For example, one could consider a moving average approach evaluated on some past window size w for dynamically updating f. After each online update, *BC Sweep* can be rerun. Similar to accounting for variability in navigation, depending on how much f changes at any point, atomic regions may become interruptable or even revert back to being atomic.

VII. Simulations

We simulated *BC Sweep* on a test environment requiring covering. Figure 7 shows the floor plan of an academic building for which *BC Sweep* was run. The service depot/refueling station was designated to a small room in the lower-left corner. This is labeled in the figures.

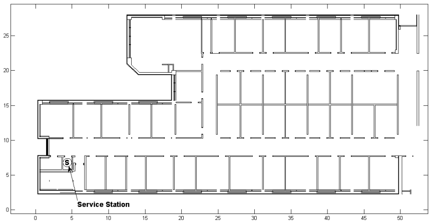


Fig. 7: *BC Sweep* test map.

We ran the *BC Sweep* algorithm with the following parameters. We used a circular robot of radius 0.25 *meters*. The fuel consumption function f was a one-to-one correspondence with the total distance traveled of a path. For example, if the robot traversed a path of 10 *meters* then the robot would have consumed 10 *units* of fuel. Each cell constructed from the boustrophedon decomposition was designated as an atomic region. All rooms in the map were assumed "closed" at the time of decomposition as to not unnecessarily increase the number of regions as the sweepline passes over doorways.

BC Sweep was run with two different fuel capacities λ. Figure 8 shows coverage with a battery capacity of 800 *units* ($\lambda = 800$). Each color represents the regions covered on the same fuel charge. The robot covered the space with two charges (a single recharge required). We also ran the algorithm with $\lambda = 350$ *units*. Figure 9 shows this. The run resulted in 4 recharges (5 total charges). For demonstration, a single charge subroute was overlayed through the magenta cover region on Figure 9. Additionally, we tested a different location for the service station on $\lambda = 350$ *units*. See Figure 10.

VIII. Conclusions

In this paper, we have introduced the *BC Sweep* algorithm to address the real problem of robot path coverage, with a battery or fuel capacity constraint. We build upon previous coverage research using boustrophedon decomposition, and contribute the *BC Sweep* planning that has the property of complete coverage, under the assumption that there is a limited amount of space that can be covered on a single battery charge, and the assumption that there is a recharging service station. We presented a proof of correctness that verifies the complete coverage under the resource constraint. *BC Sweep* runs on arbitrary geometrical physical layouts, and we have demonstrated it in simulation using a real world map and a real simulated coverage robot. We tested with varying parameters for the fuel capacity and service station location, and showed the multiple sweeps, one per charge, generated by our *BC Sweep* algorithm. After having addressed the real battery constraint, our future work includes to continue to bring coverage algorithms closer to real situations faced by real robots in the real world.

References

[1] J. Biswas and M. M. Veloso, “Localization and Navigation of the CoBots Over Long-term Deployments,” *Intelligent Robots and Systems*, vol. 32, no. 14, pp. 1679–1694, 2013.

[2] H. Choset, “Coverage for robotics A survey of recent results,” *Annals of Mathematics and Artificial Intelligence*, vol. 31, pp. 113–126, 2001.

[3] W. Sheng, N. Xi, M. Song, Y. Chen, and P. Macneille, “Automated CAD-Guided Robot Path Planning for Spray Painting of Compound Surfaces Ford Motor Company,” *Intelligent Robots and Systems*, pp. 1918–1923, 2000.

[4] K. L. Doty and R. R. Harrison, “Sweep Strategies for a Sensory-Driven, Behavior-based Vacuum Cleaning Agent,” *AAAI Fall Symposium*, pp. 42–47, 1993.

[5] S. Fabret, P. Soukrest, M. Tai, and L. Cordesses, “Farmwork Path-planning for Field Coverage with Minimum Overlapping,” *8th International Conference on Emerging Technologies and Factory Automation*, vol. 2, 2001.

[6] E. M. Arkin and J. S. B. Mitchell, “Approximation Algorithms for Lawn Mowing and Milling,” *Computational Geometry*, vol. 17, no. 1-2, pp. 25–50, 2000.

[7] W. Snyder and A. Pirzadeh, “A unified solution to coverage and search in explored and unexplored terrains using indirect control,” vol. 3, pp. 2113 – 2119, 1990.

[8] E. U. Acar, H. Choset, Y. Zhang, and M. Schervish, “Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods,” *The International journal of robotics research*, vol. 22, no. 7, 2003.

[9] D. W. Gage, “Randomized Search Strategies with Imperfect Sensors,” *SPIE Mobile Robots VIII*, vol. 2058, no. September, pp. 270–279, 1993.

[10] J.-C. Latombe, “Robot Motion Planning: Edition en anglais,” *Kluwer Academic, Boston, MA*, 1991.

[11] H. Choset, E. Acar, A. A. Rizzi, and J. Luntz, “Exact Cellular Decompositions in Terms of Critical Points of Morse Functions,” *IEEE International Conference on Robotics and Automation San Francisco, CA*, vol. 3, pp. 2270 – 2277, 2000.

[12] H. Choset and P. Pignon, “Coverage Path Planning : The Boustrophedon Cellular Decomposition,” *Proceedings of the International Conference on Field and Service Robotics, Canberra, Australia*, 1997.

[13] E. M. . Arkin and R. Hassin, “Approximation algorithms for the Geometric Salesman Problem *,” *Discrete Applied Mathematics*, vol. 55, no. 93, pp. 197–218, 1994.

[14] W. H. Huang and N. York, “Optimal Line-sweep-based Decompositions for Coverage Algorithms,” *IEEE International Conference on Robotics and Automation*, vol. 2, 2001.

[15] A. Sipahioglu, G. Kirlik, O. Parlaktuna, and A. Yazici, “Energy constrained multi-robot sensor-based coverage path planning using capacitated arc routing approach,” *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 529–538, May 2010.

[16] A. Yazici, G. Kirlik, O. Parlaktuna, and A. Sipahioglu, “A Dynamic Path Planning Approach for Multirobot Sensor-Based Coverage Considering,” vol. 44, no. 3, pp. 305–314, 2014.

[17] Y. Hu and C. Lee, “Deployment of mobile robots with energy and timing constraints,” *IEEE Transactions on Robotics*, vol. 22, no. 3, pp. 507–522, Jun. 2006.

[18] N. Christofides, “Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem,” *Report 388, Graduate School of Industrial Administration, CMU*, no. February, 1976.

[19] J. Beasley, “Route first - Cluster second methods for vehicle routing,” *OMEGA*, vol. 11, no. 4, pp. 403–408, Jan. 1983.

[20] C. Li and D. Simchi-Levi, “Worst-case analysis of heuristics for multidepot capacitated vehicle routing problems,” *ORSA Journal on Computing*, vol. 2, no. 1, 1990.

[21] C.-L. Li, D. Simchi-Levi, and M. Desrochers, “On the Distance Constrained Vehicle Routing Problem,” *Operations Research*, vol. 40, no. 4, pp. 790–799, 1992.

[22] V. Nagarajan and R. Ravi, “Approximation algorithms for distance constrained vehicle routing problems,” *Networks*, vol. 59, no. 2, pp. 209–214.

[23] J. Biswas and M. Veloso, “Episodic Non-Markov Localization: Reasoning About Short-Term and Long-Term Features,” *In Proceedings of ICRA’14, the IEEE International Conference on Robotics and Automation, , Hong Kong, China*, 2014.

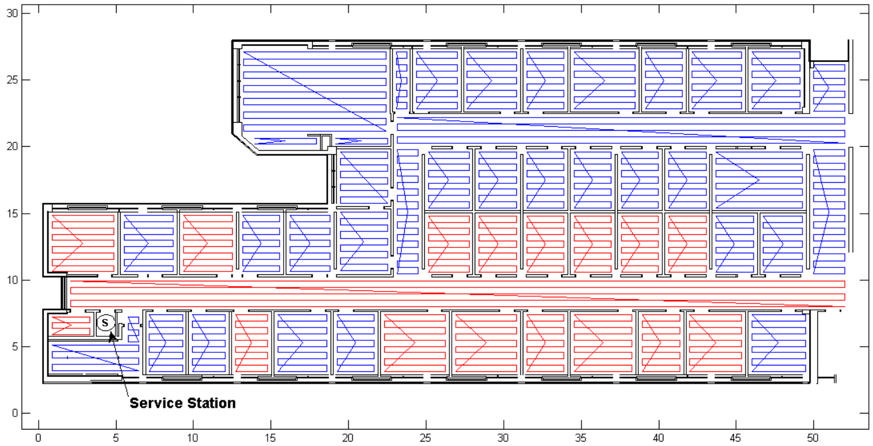


Fig. 8: *BC Sweep* coverage with $\lambda = 800$ *units*. Each colored region is covered by a single charge. Scale in meters.

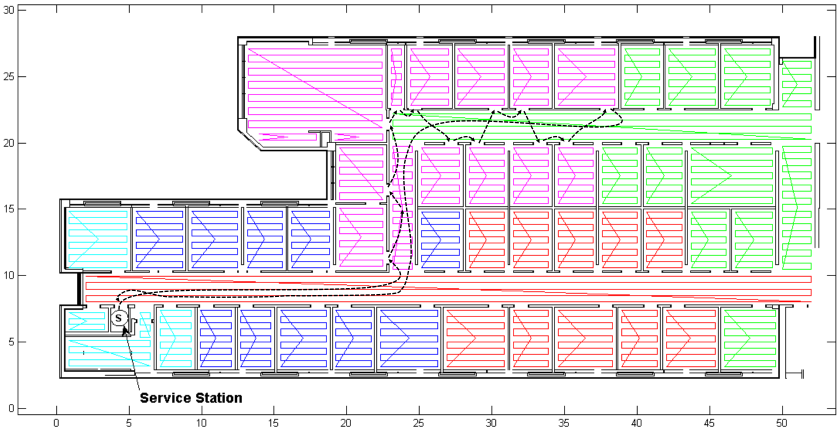


Fig. 9: *BC Sweep* coverage with $\lambda = 350$ *units*. Each colored region is covered by a single charge.

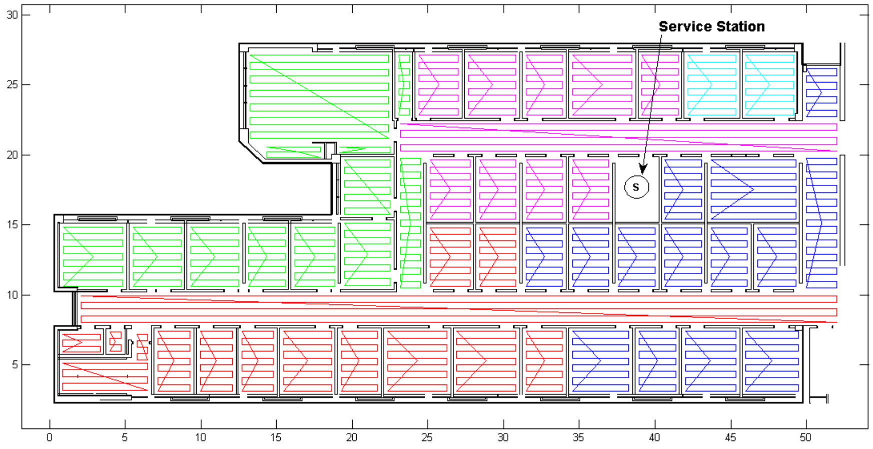


Fig. 10: *BC Sweep* coverage with alternate service station location ($\lambda = 350$ *units*).