2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Multi-Robot Area Patrol under Frequency Constraints

Yehuda Elmaliach, Noa Agmon and Gal A. Kaminka*
The MAVERICK Group, Computer Science Dept.
Bar Ilan University, Israel
{elmaley,segaln,galk}@cs.biu.ac.il

Abstract— **This paper discusses the problem of generating patrol paths for a team of mobile robots inside a designated target area. Patrolling requires an area to be visited repeatedly by the robot(s) in order to monitor its current state. First, we present frequency optimization criteria used for evaluation of patrol algorithms. We then present a patrol algorithm that guarantees maximal uniform frequency, i.e., each point in the target area is covered at the same optimal frequency. This solution is based on finding a circular path that visits all points in the area, while taking into account terrain directionality and velocity constraints. Robots are positioned uniformly along this path, using a second algorithm. Moreover, the solution is guaranteed to be robust in the sense that uniform frequency of the patrol is achieved as long as at least one robot works properly.**

I. Introduction

Robots can save human lives and costs by replacing humans in mundane, or dangerous tasks. For instance, robots may be used for cleaning [6], and hazardous waste removal [14]. One specific application of interest is surveillance and patrolling along perimeters [18] or in sensitive areas [4].

This paper discusses the problem of patrolling a target area. Patrolling involves repeatedly visiting a target location in order to assess environmental state by deploying sensors in those locations. If the entire terrain cannot be monitored at all times, each location in the target area is monitored once every f time cycles. The frequency is, then, $1/f$. Increased availability of multiple robots raises new opportunities for patrol missions. First and foremost, patrolling can be made more time-efficient in the sense that the frequency is potentially higher, i.e., f is smaller. In addition, robustness can be attained in the sense that if at least one robot is active, the patrol mission can still be accomplished.

Previous work has offered several approaches to surveillance of areas [10], [15], [17], [2]. However key challenges in surveillance have been left open. First, patrolling has mostly been done in ad-hoc fashion, without a formal analysis of the quality of the task in light of its principal goal. Second, the opportunity for increased robustness has not been investigated theoretically. Third, handling non-uniform terrains in terms of velocity and directional constraints was not addressed.

Hence, this paper deals with constructing patrol paths for a group of mobile robots that are required to patrol in a non-uniform continuous target area (divided into a grid). We base our solution on recent work in multi-robot coverage [1], [13], in which the authors suggest a family of algorithm for generating cyclic paths for covering the terrain *once*. We rely on their basic idea introduced by Gabriely and Rimon [8], in which spanning trees are used in order to generate the cyclic paths in uniform grid based terrains. In our solution, we guarantee that every point will be attended at the same frequency by creating one cyclic patrol path visiting all points in the target area (a Hamiltonian cycle in the grid), and instructing all robots to walk along this cycle in equidistant relative positions.

Robots have velocity limitations, which depends on both the *terrain* and *direction* in which they travel. For example, climbing a hill can typically be done in a lower velocity compared to climbing down the same hill. Therefore a cost should be associated with each point (and direction) of the terrain, making the terrain grid directionally non-uniform.

We consider directionally non-uniform terrains. We first provide an algorithm that finds the minimal cyclic path (minimal Hamiltonian cycle) given the terrain. We then find points along the path from which the patrol will start, and find an optimal assignment of robots to those locations in the sense that they will arrive at their starting points in minimal time. Finally, we evaluate our derived patrol algorithm using the frequency optimization criteria described in Section III. By basing our solution on the choice of minimal Hamiltonian cycle, we guarantee maximal uniform frequency in the cycle.

Similar to the robustness of the multi-robot coverage described in [13], our solution is robust, therefore guarantees maximal uniform frequency for any number of non-faulty robots greater or equal to one.

II. Background

The patrol task, also known as sweeping or repetitive coverage, was investigated previously in various approaches. Most approaches concerning multi-robot patrol partitions the area into sub-areas divided between the robots. Inside such sub-area, each robot patrols using some single-robot patrol algorithm. Ahmadi and Stone [2] describe a negotiation-based approach for dividing the area between the robots, dealing with events such as addition and removal of robots from the system. Guo et. al ([11], [12]) also divide the area between the robots while focusing on their localization and sensorial capabilities. Jung and Sukhatme describe in [15] a region based approach for tracking targets in a system with multiple robots and stationary sensors.

*This work was supported in part by Israel's Ministry of Science.

However, key challenges in surveillance have been left open. First and foremost, patrolling has mostly been done without a formal analysis of the quality of the task in light of its principal goal, i.e., frequency of visits to each target point. Second, the opportunity for examining patrolling in groups of robots for robustness has not been analyzed and theoretically proven.

Related work by Chevaleyre [5] offer the first theoretical analysis of the patrol problem. The author provides an analysis of partition-based versus cyclic patrol paths. He introduces the notion of *idleness*, which is the duration each point in the patrolled area is not visited. He provides an algorithm based on the Traveling Salesman Problem (TSP) to create a cyclic path for the robots and proves its optimality in the sense that the idleness is minimized. Finding the optimal path in the TSP problem is $\mathcal{NP}$-Hard, therefore the solution is an approximation to the optimal solution. The authors use the ST based approximation for creating the cycle, which assumes that the original graph is complete. Our approach, on the other hand, provides an *optimal* solution in the same sense, and moreover - provides *uniform* optimal solution for grid based graphs (under Gabrieli and Rimon's initial assumptions in [9]). A survey by Almeida et. al. [3] brings a discussion concerning different approaches towards patrolling with regards to the idleness criteria. They compare paths based on machine learning, agents using negotiation mechanisms, heuristic agents and ones going along one cycle (described in [5]). Their empirical results show great advantage to the cycle based approach in average idleness, therefore it strengthens our choice of choosing one cyclic path for the patrol.

The patrol problem resembles the coverage problem, a canonical problem in robotics, in the sense that both require the robot or group of robots to visit all points in the given terrain. However, while coverage seeks to minimize the number of visits to each point (ideally, visiting it only once), patrolling seeks to maximize it (while still visiting all points). Therefore solutions that are used for the coverage problem can be used as basis for patrolling.

Specifically, we chose to use the Spanning Tree Coverage (STC) method as base for our work. In this method, introduced by Gabriely and Rimon [8] the authors assume that a single robot is equipped with a square shaped tool of size D, hence the area was divided into cells of size D placed on grid. The grid was then coarsen such that each new cell is of size $2D\,X\,2D$, and a spanning tree was built over this new coarse grid. After such a tree was built, the robot follows the tree around, creating a Hamiltonian cycle visiting all cells of the original grid (see example in Figure 1).

The idea was first broadened for a multi-robot system by Hazon and Kaminka in [13] in the family of Multi Robot Spanning Tree Coverage (MSTC) algorithms. Their solution, along with decreasing the total coverage time, achieved robustness in the sense that as long as one robot works properly, the coverage of the terrain is guaranteed.

In most cases the spanning tree is built upon uniform grids. However, in [9] the authors considered a special case of non-uniform terrains, for a single-robot coverage case. In that work, each edge in the coarse grid was given a different weight in order to favor movement in certain directions, and a *minimal* spanning tree (MST) was found. However, the authors did not show the correspondence of the weights of the coarse grid edges to the fine grid edges, and minimality of the Hamiltonian cycle was not proven.

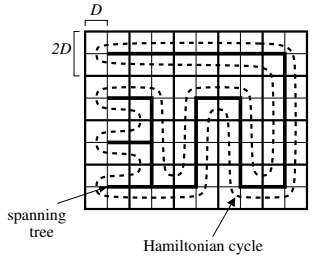

Fig. 1. An example for spanning tree based coverage. Coarse grid is in bold, and the spanning tree connects all coarse grid cells. The Hamiltonian cycle over the fine grid is the dotted line along the spanning tree.

III. Frequency optimization criteria in area patrol

Continuously monitoring a target location by a group of mobile robots can be done using various methods. Given a team of k robots and N target areas, each of size smaller or equal to the sensorial range of a robot. If $k \geq N$, then all target areas can be monitored at all times by the team simply by assigning a robot to each target area. Formally, $f = 1$ since each target area is monitored at each time cycle by at least one robot. In the more common case, $k \ll N$, each target area a_i, $1 \leq i \leq N$, is visited by a mobile robot at some frequency f_i. Note that the frequency in which each target area is visited is not necessarily uniform.

There are several possible target visit frequency criteria according to which patrol algorithms can be evaluated, and are described as follows:

Uniform frequency: The goal is to decrease the variance between the frequencies in which each target is visited, i.e., all targets should ideally be visited with uniform frequency f.

Average frequency: In the case where uniformity cannot be guaranteed, the goal is to increase the average frequency f in which targets are visited.

Under-bounded frequency: The goal is to increase the minimal frequency in which any target is visited, such that every target is visited with frequency of at least $1/f$. In other words, all targets should be monitored at least once every f cycles.

In our work, we guarantee optimal uniform, under-bounded average frequency through all cells in the target area. We do this by generating a cyclic path visiting all target areas (a Hamiltonian cycle) and then place the robots uniformly along the cyclic path. If all robots move at the same direction—either clockwise or counterclockwise along the cyclic path—then clearly each cell is visited at the same frequency (uniform frequency). Moreover, in uniform terrains each cell is visited at least once every $\lceil \frac{cycle\ length}{num\ robots} \rceil$ number of cycles, where cycle length is simply the number of nodes plus one.

We address a more realistic case of directional terrain, in which we have a grid over a continuous area, with velocity

constraints, depending on the position and direction of movement. In this case we have to find a minimal Hamiltonian cycle (Section IV), and then find an assignment of robots to their starting points along the path while considering the weighted directional terrain (Section V).

IV. Generating a minimal cyclic path

In this section we find a minimal circular path based on the spanning tree solution for coverage [8]. Previous work based on spanning tree for coverage dealt with uniform grids, i.e., the cost of going to either four directions from a cell is uniform along the entire grid (hence each spanning tree is minimal). We introduce a domain in which a cost is attached to a movement between any two adjacent cells in the fine grid. In order to use the known results using spanning trees, we must convert the directed edges of the fine grid to undirected edges in the coarse grid while preserving the properties of the edges such that a minimal spanning tree on the coarse grid will yield a minimal Hamiltonian cycle on the fine grid.

Since the patrol tour can be conducted in either clockwise (CW) or counterclockwise (CCW) directions along the spanning tree path, we divide our world to CW and CCW. In general, there are four directed edges entering and four leaving each cell in the fine grid. Since we follow some spanning tree path, the options decrease and each cell have up to two incoming and two outgoing edges in each world (CW and CCW), as described in Figure 2. We find a minimal spanning tree in each of the worlds separately, and choose the minimal between both as base for the patrol path.

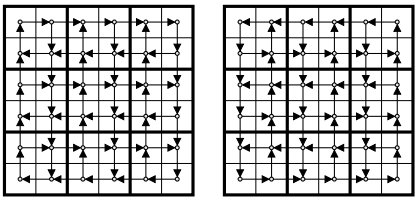

Fig. 2. Division of the area to clockwise (a.) and counterclockwise (b.) directions. The graphs are built such that the movement is suitable for traveling along a spanning tree. Union of the two graphs provide all possible movement options from each cell: up, down, right and left.

Note that Figure 2 clearly illustrates that the CW and CCW worlds are complementary in the following sense. First,the intersection between the worlds is empty, i.e., $\forall e \in E$, $e \in \mathsf{CW}(G)$ or $e \in \mathsf{CCW}(G)$. Second, together they provide all connections between adjacent edges in four possible directions: up, down, right and left.

Following, we describe an assignment of weights (Assignment Assign_Opt) to the undirected edges of the coarse grid based on the weights of the directed edges of the fine grid. We then argue that using this assignment, we indeed guarantee that finding a minimal spanning tree on the coarse grid representation yields an minimal Hamiltonian cycle on the fine grid. In order to do that, we first prove that in our scenario, a Hamiltonian cycle is created by each spanning tree and vice versa, i.e., each Hamiltonian cycle in the fine grid is translated to a spanning tree in the coarse grid. Based on that, we then prove, in Lemma 3, that Assignment Assign_Opt yields the minimality property we seek.

Assignment Assign_Opt**:** The cost assigned to the undirected edge (u, v) in the coarse grid (see figure 3) is the sum of the directed edges in the fine grid, parallel to (u, v) from its two sides minus the sum of the directed edges perpendicular to (u, v) and intersecting it, or: $(a+b)-(c+d)$. Note that this can generate edges with negative cost. In this case we shift the cost of all edges by the minimal negative value, and use Kruskal's algorithm ([7]) for finding an MST.

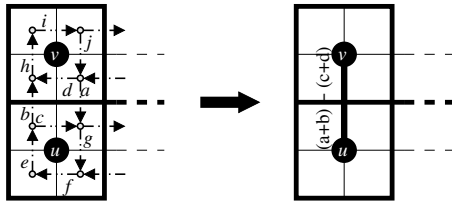

Fig. 3. The assignment of weights to the undirected edges of the coarse grid based on the directed edges of the fine grid (here in the CW direction).

Lemma 1: Every spanning tree on the coarse grid can be translated to a Hamiltonian cycle on the fine grid and vice versa, i.e., every Hamiltonian cycle on the fine grid can be translated to a spanning tree on the coarse grid.

Proof: The first part of the Lemma is shown in the initial algorithm of Gabriely and Rimon [8]. According to their algorithm, a Hamiltonian cycle is generated simply by walking along the spanning tree path in the fine grid.

In order to prove the second direction, we will first show that the existence of Hamiltonian cycle in the fine grid guarantees that only full edges are picked in the coarse grid. We choose, without loss of generality, the CW case. Figure 4 illustrates two adjacent vertices in the coarse grid, u and v, and their corresponding vertices in the fine grid. Denote the edge (u_2, v_1) by a, (v_3, v_4) by b, (u_2, u_4) by d and (v_3, v_1) by c. We must show that choosing edge a guarantees choosing also b and excludes c, d, and vice versa, i.e., choosing c forces choosing d and excludes a, b. Assume, towards contradiction, that a Hamiltonian cycle exists in the grid, but it uses only edge a and not b. Therefore in order to visit vertex u_4 d is chosen, contradicting the fact that they are all part in a Hamiltonian cycle, as u_2 cannot have two outgoing edges. The fact that c and d cannot be chosen along with a and b is proven similarly.

It is left to show that the Hamiltonian cycle on the fine grid creates a spanning tree on the coarse grid. Assume, towards contradiction, that there exists a Hamiltonian cycle that does not translate into a spanning tree on the coarse grid. This means that there exists a vertex in the coarse grid that is not covered by the spanning tree. This can happen only if not all fine vertices are visited, contradicting the fact that we have a Hamiltonian cycle. ∎

Corollary 2: A Hamiltonian cycle on the fine grid can include edges from either the CW world or the CCW world, but not from both.

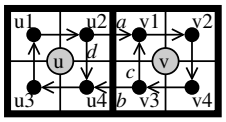

Fig. 4. Illustration of Lemma 1.

Figure 5 shows an example illustrating the corollary. A closed path is a Hamiltonian cycle if it covers all vertices of the graph once, meaning that the in-degree and out-degree of vertices in this path is 1. In Figure 5, all edges in the path are from the CCW world except for edge (v_1, v_2) (dashed), which is from the CW world. The resulting path is *not* a Hamiltonian cycle, as v_1 has in-degree 2 and v_2 has out-degree 2.

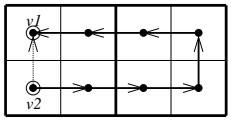

Fig. 5. Illustration of Corollary 2, demonstrating the problem of combining edges from the CW and the CCW world.

Lemma 3: Using Assignment Assign_Opt, an MST on the coarse grid representation yields a minimal Hamiltonian cycle (HC_Min) on the fine grid.

Proof: Assume again, toward contradiction, that there exists a Hamiltonian cycle, HC′ with total weight smaller than HC_Min. This can happen in one of two scenarios.

a. Its corresponding spanning tree has lower cost than the MST. This contradicts the minimality of the MST, hence this case is impossible.

b. Its corresponding spanning tree (ST′) has higher total weight than the MST's weight and still HC′ < HC_Min. Consider the case in which the trees differ by one edge, $e \in$ MST, $e \notin$ ST′ and $e' \in$ ST′, $e' \notin$ MST. Denote the directed edges forming e by a, b, c, d and the directed edges forming e' by a', b', c', d' (as described in Assign_Opt). Since ST′ > MST and based on Lemma 1, it follows that that $\text{weight}(e') > \text{weight}(e)$. Therefore, according to Assign_Opt, $a' + b' - (c' + d') > a + b - (c + d) \Rightarrow a' + b' - (a + b) > c' + d' - (c+d)$. Since we assume that HC′ < HC_Min and they differ only by e and e', if follows by the inclusion of e in HC and exclusion in HC_Min that $a' + b' + c + d < a + b + c' + d' \Rightarrow a' + b' - (a+b) < c' + d' - (c+d)$, leading to a contradiction. It can be shown similarly for every spanning tree greater than the MST that this case is impossible. ∎

As a corollary of Lemmas 1 and 3, the following algorithm finds the minimal Hamiltonian cycle over the given terrain.

Note that the construction of minimal Hamiltonian cycle in our domain applies to the coverage problem as well as for the patrol problem. Meaning, the minimal cycle as found here for non-uniform terrains can be used also for single-robot coverage, achieving minimal coverage path with respect to the constraints on the terrain as long as the robot moves in either CCW or in CW direction (as implied by the output).

Procedure Generate_Cycle

1) Divide the area into two CW and CCW scenarios.
2) For each scenario, create a graph on the coarse grid by assigning weights to edges as described in Translation A.
3) Find a minimal spanning tree in the coarse grid using Kruskal's algorithm.
4) Calculate the total length of the Hamiltonian cycle generated by the minimal spanning tree.
5) Report scenario (CW or CCW) and cycle with shorter total length.

V. Assigning Initial Locations to Robots

After establishing the minimal cyclic path for the patrol mission by the group of mobile robots, it is left to determine the position of the robots along the cycle from which they begin their patrol. Clearly, in order to achieve uniform frequency it is sufficient to spread the robots uniformly along the cyclic path. The distance between every two robots along the cyclic path should be the total weight of the cycle divided by the number of robots, yielding an equal distance between every two consecutive robots along the patrol path. Since there is more than one possible assignment of the robots to such positions, we want to find the assignment that requires minimal change from current positions of the robots. Therefore we describe herein the algorithm Initialization, which finds the locations from which the robots should start patrolling, while minimizing the maximal distance a robot should travel in order to arrive at its location. As the robots move simultaneously from their initial positions to their positions along the cycle, this corresponds to minimizing the time it takes all robots to be positioned and ready for the patrol mission.

We define the *Minimal Path Matching* (Min_Path_Match) problem as follows.

Min_Path_Match**:** Given a weighted graph $G = (V, E, W)$, a Hamiltonian cycle visiting all vertices in the graph, and a set of initial positions of k robots on vertices of G. Find an assignment of each robot to a position in the graph such that the following are fulfilled.

1) The distance between every two consecutive robots along the Hamiltonian cycle is equal.
2) The maximal distance traveled by a robot from its initial position to the assigned location is minimized.

We suggest the initialization algorithm Initialization for solving the Min_Path_Match problem. The input to the algorithm is: G - graph, HC - minimal HC found by Generate_Cycle, RI - set of initial locations of the robots on the graph and BW - basic weight of edge (generally equals 1 unless scale involves fractions, in which BW will be scaled accordingly). Define the length of a Hamiltonian cycle by $\text{len}(\text{HC})$. The algorithm works as follows. First, it generates HC′ by separating the edges of the cyclic path into sub-edges, each of size BW (see Figure 6. Each vertex in HC' represents an optional starting point. It then sets the initial positions of the robots along the path such that the distance between them is $\frac{\text{len}(\text{HC}')}{k}$, and finds the assignment of robots

to these locations such that the maximal distance traveled by a robot from its initial position to the assigned location is minimized. It does that by using procedure PMPM. Then, it checks the minimal maximal distance of all rotations of the positions along the cycle, and reports the positions yielding minimal maximal distance.

Algorithm Initialization(G,HC,RI,BW)

1) $L \leftarrow \emptyset$ {output optimal match}
2) $min = \infty$ {minimal match weight}
3) HC′ ← separation of HC by BW.
4) $VL \leftarrow k$ vertices from HC' with distance of $\frac{\text{len(HC}')}{k}$ between consecutive vertices along HC′.
5) Compute Dijkstra (shortest path) for each robot from its current location to all vertices in the graph (HC)
   a) **For** $i = 0$ to $\frac{\text{len(HC}')}{k}$
   b) $VL = VL$+1 {progress each of the VL vertices one optional starting point forward in HC′}
   c) Let BG be a full bipartite graph of the two sets RI and VL {the weights will be based on the above Dijkstra calculation}
   d) $ML = \emptyset$ {ML be the current match list}
   e) MatchValue ← PMPM(BG, ML)
   f) **If** MatchValue $< min$ **then**
      i) $L = ML$
      ii) $min =$ MatchValue
6) return L

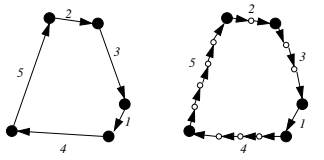

Fig. 6. On the left: The basic Hamiltonian cycle. On the right: The separated Hamiltonian cycle (in this example BW = 1)

In step 3 we create optional starting points along the spanning tree path. In step 4 we choose a specific set of starting points with equal weights from one to the next. In step 5b. we update the set of starting points by progressing each point to its following neighbor. Step 5 of the algorithm Initialization will be repeated for each possible set of starting points. In step 5e. we use Procedure PMPM, where it receives BG - a full weighted bipartite graph and an empty list ML. Procedure PMPM fills ML with the optimal possible match considering BG. An optimal match is one in which the minimal maximal weight of an edge in the bipartite graph over all possible permutations maximal edges. It also returns the value of this maximal edge named here MatchValue.

Procedure PMPM uses the *Hungarian algorithm* [16] which finds a match in bipartite graphs with minimal sum of edges. As illustrated in Figure 7, the Hungarian algorithm finds a match between $r1$ and $d1$ and between $r2$ and $d2$ since this is the minimal match sum. But in our application we would like to find the minimal biggest edge from all the possible permutations, i.e., we want to match here $r1$ to $d2$ and $r2$ to $d1$. In this match the maximal edge weight is 9 while in the previous it is 10. Step 3 in the PMPM algorithm construct

Procedure PMPM(BG, ML)

1) Let V be BG vertices
2) Sort the edges in BG
3) Construct BG' from BG with edge weights start from 1 and by the increased sorted order multiple by $\frac{|V|}{2}$ (see Figure 7).
4) Run the Hungarian-Algorithm(BG',ML)
5) Return biggest edge weight from the match ML in the corresponding BG edges.

BG' from BG. The BG' graph, by its construction, causes the *Hungarian algorithm* call (in step 4) to prefer the permutation in which the maximal edge is minimal.

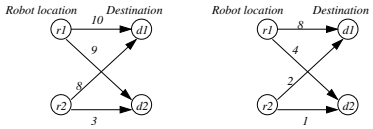

Fig. 7. Basic bipartite graph, and bipartite graph after conversion.

Lemma 4: The construction of BG' in step 3 of PMPM assures that the Hungarian algorithm returns a match with minimal maximal edge in BG.

Proof: First, we prove that the construction assures the selection of a match with minimal maximal edge in BG'. For that, assume, towards contradiction, that the Hungarian algorithm returns a minimal match with sum of edges M and maximal edge m but there exists another match with sum of edges M' that has a maximal edge m' such that $m > m'$. By the construction of BG' in step 3 of PMPM it follows that any edge in BG' is greater than the sum of all edges smaller than it. Specifically, by our assumption that $m > m'$ it follows that $m > M' \Rightarrow M > M'$ contradicting the minimality of M returned by the Hungarian algorithm.

It is left to show that the match found on BG' yields a match on BG with minimal maximal edge as well. This follows directly from the fact that the order of edges remains through construction, hence minimal maximal edge in BG transforms to the minimal maximal edge in BG', and back. ∎

The time complexity of Procedure PMPM is as the Hungarian algorithm [16] k^3 and Algorithm Initialization runs it $\frac{|V|}{k}$ times. It also run Dijkstra (shortest path) k times. Then this algorithm complexity is $O(KV(K + lgV)$

VI. Evaluation of the Derived Patrol Algorithm

In this section we evaluate the patrol algorithm that is based on procedure Generate_Cycle and Initialization according to the frequency optimization criteria described in Section III.

Theorem 5: The patrol algorithm which is derived by the combination of Procedures Generate_Cycle and Initialization guarantees: *a.* Uniform frequency *b.* Maximal average frequency *c.* Optimal under-bounded frequency.

Proof: a. The first part of the Min_Path_Match problem requires the robots to be placed initially, i.e., before they begin their patrol, in uniform distance along the cyclic path. This requirement is clearly fulfilled by step 4 in Procedure

Initialization, where the only positions considered are the ones where all robots are placed with equal distance from their neighbors along the cyclic path. Since the robots are homogeneous and all target areas are covered by the cyclic path, their movement along the cyclic path yields a uniform frequency of $\frac{1}{\mathrm{len}(\mathsf{HC})/k}$, where k is the number of robots and $\mathrm{len}(\mathsf{HC})$ is the total length of the minimal HC found by Generate_Cycle and the standard deviation is 0.

b. + *c.* The cyclic path found by Generate_Cycle was proven by Lemma 3 to be minimal. Therefore one robot traveling along this cycle has maximal frequency of $\frac{1}{\mathrm{len}(\mathsf{HC})}$, hence the maximal possible frequency by k robots is $k \times \frac{1}{\mathrm{len}(\mathsf{HC})}$, which is exactly the frequency guaranteed by our algorithm. All targets are monitored, then, exactly once every $\frac{\mathrm{len}(\mathsf{HC})}{k}$ cycles, and by that optimal under-bounded frequency is guaranteed. Since we have proven uniform and under-bounded frequency, maximal average frequency is straightforward. ∎

VII. Guaranteeing Robustness

In [13], Hazon and Kaminka have demonstrated the benefit of using spanning trees as base for multi-robot spanning tree coverage. In their work they used the circular path generated by the spanning tree as base for *robust* multi robot coverage. Therefore in our work we use the circular path not only for assuring uniform frequency while patrolling, but for robustness as well. In our case, we refer to robustness in the sense that as long as at least one robot remains intact, the patrol mission is guaranteed to be performed successfully.

Robustness is clearly guaranteed, as if one robot fails the other robots simply divide the circular path again between them by re-running Procedure Initialization. Theorem 5 is, then, guaranteed for the new number of robots. In this statement we have a hidden assumption that the system is stable in the sense that the uniform, maximal-average, optimal under-bounded frequency is guaranteed as long as the system performs properly, and if a failure occurs it again guarantees the above properties after a short reorganization time. This reorganization time is the period of time necessary for the robots to execute the algorithm and arrive at their new initial positions. If all robots are to walk along the cyclic path following the current direction, then this period of time will not exceed $\frac{\mathrm{len}(\mathsf{HC})}{6}$, see Lemma 6 for the proof. If the system is unstable, i.e., robots fail one after the other, then Theorem 5 is guaranteed for the final number of robots after stabilization.

Lemma 6: The reorganization time required when decreasing the number of robots from k to $k-1$ is maximum the time required to travel the distance $\frac{\mathrm{len}(\mathsf{HC})}{6}$ if robots follow the circular path on their way towards their new initial positions.

Proof: Consider the case in which there are three robots, and one fails. The length of the HC is divided by the three robots prior to the failure, and is divided by two after the failure. Therefore, if only one robot travels along the path, it has to travel from $\mathrm{len}(\mathsf{HC})/3$ to $\mathrm{len}(\mathsf{HC})/2$, which is exactly $\mathrm{len}(\mathsf{HC})/6$. For any other k, the distance traveled is smaller: $\frac{\mathrm{len}(\mathsf{HC})}{k-1} - \frac{\mathrm{len}(\mathsf{HC})}{k} < \frac{\mathrm{len}(\mathsf{HC})}{2} - \frac{\mathrm{len}(\mathsf{HC})}{3}$ for any $k > 2$. Clearly, for $k = 2$ the remaining robot has no reorganization phase, as it simply patrols along the circular path alone. ∎

VIII. Conclusions and Future work

In this work we have discussed the patrol problem and its frequency aspects. First, we have suggested frequency parameters according to which a patrol mission can be evaluated. Next, we described an algorithm for finding a minimal Hamiltonian cycle in a non-uniform, directional, terrain. Based on this cyclic path, we have shown an algorithm that assigns locations to the robots along the path such that the time necessary to arrive to those locations is minimal, and patrolling from those locations create a uniform maximal-frequency patrol. Last, we have shown that this algorithm is robust in the sense that it guarantees patrol at uniform frequency as long as at least one robot works properly.

There are still several areas we plan to pursue in future work. First, we would like to take into consideration at the cycle generation phase also other aspects, for example minimizing number of turns. Second, we would like to examine the case in which the robots are heterogeneous. In addition, we would like to consider task allocation during patrol missions, for example extracting robots to handle events while satisfying some frequency constraints. Last, we wish to consider also non-uniform terrains having also prioritized requirements.

References

[1] N. Agmon, N. Hazon, and G. Kaminka. Constructing spanning trees for efficient multi-robot coverage. In *ICRA*, 2006.

[2] M. Ahmadi and P. Stone. A multi-robot system for continuous area sweeping tasks. In *ICRA*, 2006.

[3] A. Almeida, G. Ramalho, H. Santana, P. Tedesco, T. Menezes, V. Corruble, and Y. Chevaleyr. Recent advances on multi-agent patrolling. *Lecture Notes in Computer Science*, 3171:474–483, 2004.

[4] D. Carrolla, C. Nguyena, H. Everetta, and B. Frederickb. Development and testing for physical securtiy robots. In *SPIE*, Orlando, 2005.

[5] Y. Chevaleyre. Theoretical analysis of the multi-agent patrolling problem. In *Proceedings of Intelligent Agent Technology (IAT)*, 2004.

[6] J. Colegrave and A. Branch. A case study of autonomous household vacuum cleaner. In *AIAA/NASA CIRFFSS*, 1994.

[7] T. Corman, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. MIT Press, 1990.

[8] Y. Gabriely and E. Rimon. Spanning-tree based coverage of continuous areas by a mobile robot. *Annals of Mathematics and Artificial Intelligence*, 31:77–98, 2001.

[9] Y. Gabriely and E. Rimon. Competitive on-line coverage of grid environments by a mobile robot. *Comp. Geometry*, 24:197–224, 2003.

[10] D. W. Gage. Command control for many-robot systems. In *The nineteenth annual AUVS Technical Symposium (AUVS-92)*, 1992.

[11] Y. Guo, L. Parker, and R. Madhavan. Towards collaborative robots for infrastructure security applications. In *CTS04*, pages 235–240, 2004.

[12] Y. Guo and Z. Qu. Coverage control for a mobile robot patrolling a dynamic and uncertain environment. In *WCICA*, June 2004.

[13] N. Hazon and G. Kaminka. Redundancy, efficiency, and robustness in multi-robot coverage. In *ICRA*, 2005.

[14] S. Hedberg. Robots cleaning up hazardous waste. *AI Expert*, pages 20–24, May 1995.

[15] B. Jung and G. Sukhatme. Tracking targets using multiple robots: The effect of environment occlusion. *Autonomous Robots*, 13(3), 2002.

[16] H. W. Kuhn. The hungarian method for the assignment problem. In *Naval Research Logistics Quarterly*, volume 2, pages 83–97, 1995.

[17] S. Kumar, T. Lai, and A. Arora. Barrier coverage with wireless sensors. In *ACM MobiCom*, pages 284–298, Cologne, Germany, 2005.

[18] K. Williams and J. Burdick. Multi-robot boundary coverage with plan revision. In *ICRA*, 2006.