

The Multi-robot Coverage Problem for Optimal Coordinated Search with an Unknown Number of Robots

Hyeun Jeong Min and Nikolaos Papanikolopoulos
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455
Email: {hjmin|npapas}@cs.umn.edu

Abstract—This work presents a novel multi-robot coverage scheme for an unknown number of robots; it focuses on optimizing the number of robots and each path cost. Coverage problems traditionally deal with how a given number of robots covers the entire environment. This work, however, presents solutions of not only (i) how to cover the area (locations of interest) within the minimum time, but simultaneously (ii) how to find the optimal number of robots for a given time. Also, we consider the worst but realistic case of all robots starting at the same location instead of assuming randomly initialized positions. The minimum coverage time depends upon the number of robots used. Our research specifies (iii) how to find the minimum coverage time without knowing the number of robots. Finally, we present a deterministic coverage algorithm based on finding the shortest paths in order to optimize the number of robots and corresponding paths.

I. INTRODUCTION

This work investigates a multi-robot coverage problem, in which a group of an unknown number of robots needs to visit a series of search positions (locations of interest). In our work we assume that the robots aim to cover a given area in order to search for Objects Of Interest (OOIs). When considering multiple robots in real-world environments, we should manage the resources with respect to variables including the number of robots, required processing time, sensing ranges, among others. In addition, having a large number of robots is also not always a fast, effective, and efficient way to solve the problem. The Centibots [1] project that deals with approximately a hundred robots, has presented an exploration and mapping algorithm that uses one or multiple robots, but the problem of how many robots should be used at a given time is still an on-going challenge.

The coverage problem has been actively researched with respect to maximizing the coverage area and minimizing the total coverage time. The research has focused on optimizing the coverage area and time while assuming a known number of robots. Multi-robot coverage problems minimize the time with a given number of robots [2]. Multi-agents path search problems also assume a given number of agents to solve the search problem [3]. On the other hand, we consider the impact of the number of robots when the coverage cost (i.e., time) is given. We assume that the robots have constant velocity and we only consider the travel cost (excluding the search cost). Large numbers of robots may not help in the reduction of the coverage cost. Fig. 1 shows an example

comparing the optimal paths and the corresponding path costs for the given numbers (one, two, and three) of robots. Robots start at the same location (the upper left corner), they then visit every cell represented in the 3×3 grid map. Note that the *minimum coverage time* is the maximum time among the times associated with the travel paths covered by each robot. The minimum coverage times in Fig. 1 are 8; 4; and 4 in each case. It shows that having many robots does not necessarily guarantee faster coverage of the area. Conversely, when there are insufficient robots as shown in Fig. 1(a), the problem may be solved in longer time than expected. The number of robots used directly affects the minimum coverage time, which has to be at least the cost from the start to the point farthest away. This motivates us to take into account how many robots will be optimal within a given time frame.

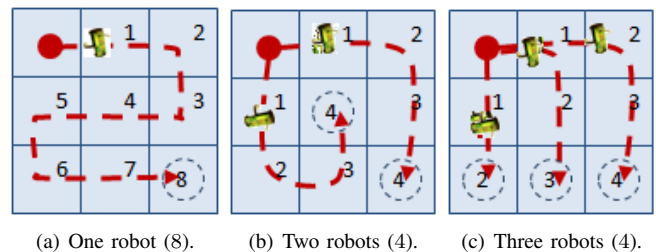


Fig. 1. The comparison of the coverage times when different numbers of robots are used under the same condition. The environment is represented by the 3×3 grid map. When one, two, and three robots are given, the coverage times are 8; 4; and 4 in (a), (b), and (c), respectively. The start position is marked by the red circle, and the numbers in each cell represent the travel cost for each robot in order to visit the cell.

Deciding the initial placement of the robots is also important. This work assumes that the robots start at the same location (area). It may be the worst scenario for multiple robots to start at the same location because they have the same task specifications (i.e., set of positions to visit). Randomly initializing the locations of robots, however, is less practical because it forces robots to divert time and resources to localization and movement to the desired starting location. No work has addressed this scenario yet.

This work not only addresses a new coverage problem but also highlights some properties involving the minimum coverage time and the optimal number of robots within a given time frame. We also propose an optimal coverage algorithm. The algorithm finds the optimal number of robots

and the corresponding paths (and tours) with the minimum cost. Since we cannot minimize all resources - the number of involved robots and path costs - at the same time, we try to minimize one resource first and simultaneously find the minimal value associated with the other resource (the number of robots). We emphasize the fact that our work focuses on finding the required number of robots for a given time frame.

II. RELATED WORK

This work considers a graph-based coverage problem in order to search for stationary OOIs without a pre-specified number of agents. A typical search problem involves finding the shortest route to a goal position, while minimizing search costs. An agent-centered search problem is investigated by Koenig [4]. The author has adapted a small planning cost between plan executions. A search problem for a target in an unknown environment based on visibility metrics is considered by Datta *et al.* [5].

A multi-robot coverage problem in order to minimize the coverage time is investigated by Agmon *et al.* in [2]. The authors have presented a tree construction algorithm based on distance metrics represented by cells on a map, but they have used a pre-specified number of robots. Searching for dynamic OOIs may require robots to visit some areas more than once. The intruder detection problems such as predator-prey [6] and multi-robot patrolling [7] fall in this category. Moors *et al.* have presented a graph-based search algorithm for the intruder detection [8]. The authors have used a randomized optimization algorithm to generate positions in order to cover the entire area based on the specific sensing ability. Konolige *et al.* have also worked on an intruder detection project using almost a hundred robots, and discussed a graph-based mapping algorithm [1].

The Traveling Salesman Problem (TSP) and the k-agents Traveling Salesman Problem (k-TSP) are tour search problems aiming at minimizing total travel cost [3]. These problems are indeed optimization problems with respect to travel costs associated with a known graph, initial positions, and a pre-specified number of agents. The k-TSP deals with multiple agents, but is limited to a known number of agents as in typical coverage problems. Frederickson *et al.* have studied an approximation algorithm for the k-agents TSP problem [9]. Ny *et al.* have studied the TSP for a Dubins vehicle [10]. An approximation algorithm for k-server routing problems is examined in [11].

III. PROBLEM DESCRIPTION

This section describes the terms in order to introduce our coverage problem and details our objectives. We define two terms, the *key positions of interest* and the *parallel multi-agent coverage problem*, as described in Definitions 1 and 2, respectively. Prior to describing our version of the coverage problem, we discuss the shape of the area covered by a single sensor at any given time step. Most coverage problems have traditionally dealt with a circular zone as the area covered by a sensor (i.e., an obstacle avoidance sensor or an omnidirectional camera). However, many sensors such

as visual and proximity sensors have more limited fields of view. For a camera-based search mission (especially when a monocular camera is used), a disk shaped coverage model for the sensor is not appropriate. Furthermore, we may not need to cover the entire area in an environment, but visit and search specific locations of interest. The issues lead us to define a new concept in a coverage problem. We first define the key positions of interest.

Definition 1: A Key Position of Interest (KPOI) is the position which a robot in a group must visit. This is to say, a robot is required to visit a location where the respective KPOI is within the active range of the robot's sensor.

This concept of KPOI helps in establishing the first requirements of the coverage problem: (i) a predefined set of KPOIs in the environment is known *a priori*. The robots must visit the entire set of KPOIs, and (ii) the sensing zones of the robots' sensors may not always be circular. We also assume that: (iii) there is an unknown number of robots to be deployed, (iv) the robots are organized into groups, and (v) robots in a group are placed at the same initial location. Depending upon an active sensing range of a sensor, we can specify uniformly distributed KPOIs. When KPOIs are uniformly distributed, our coverage problem introduced in Definition 2 is the *dual* of a traditional coverage problem dealing with a known number of uniformly distributed robots. In this work the KPOIs may not be uniformly distributed spatially. For our coverage problem, the travel cost between two locations is a function of their Euclidean distance. Additionally, KPOIs should be visited only once by either of the deployed robots. However, multiple visits to the same KPOI are allowed only if including them in a path reduces the respective path cost.

Definition 2: The Parallel Coverage Problem limits the space of acceptable solutions as follows. The paths traversed by each robot agent should be characterized by the following: (i) every robot path includes a subset of KPOIs visited only once by the respective robot, and (ii) the robots' KPOIs are complementary and disjoint.

Our coverage problem differentiates finding a path and finding a tour. The problem of finding a path does not pay much attention to the paths in order to return to the start node. On the other hand, finding a tour considers both the time and the path when returning to the start node. We therefore treat the Parallel Path Coverage Problem (PPCP) and the Parallel Tour Coverage Problem (PTCP) differently. Next, we will illustrate our strategies and algorithms to: (i) find the optimum (minimum) travel cost and the corresponding paths in a given environment, (ii) find the optimal number of robots to keep the total coverage cost less than or equal to the minimum travel cost, and also (iii) find the optimal robots for a given but limited time.

For the PPCP multiple robots simultaneously move towards their goal positions, while the PTCP is to find tours with the minimum cost for an unknown number of robots. We need to find the optimal number of robots as well as the paths (and tours) with the minimal costs for both parallel path/tour coverage problems. Let $V = \{v_1, \dots, v_n\}$ be a set

of KPOIs. We minimize the maximum path cost from a start position for the PPCP. The objective is

$$c_p = \min\{\max_i \text{cost}(\pi_i)\}, \forall i = 1, \dots, r. \quad (1)$$

Here, π_i and P_i are the i^{th} path and KPOI as shown in Table I. Let $\{G_1, \dots, G_r\}$ be a set of subsets of V , and ϕ_i be the i^{th} tour, a sequence of positions in G_i . Similarly, the objective of the PTCP is to find a set of subsets, $\{G_i | i = 1, \dots, l\}$, with

$$c_t = \min\{\max_i \text{cost}(\phi_i)\}, \forall i = 1, \dots, l. \quad (2)$$

Note that the variables r and l are the numbers of the robots which are initially unknown.

We propose the following strategies for our problem, which find the minimum path cost and number of robots. Finding a path with the minimum cost is achieved by finding the furthest position (v_{max}) from the start. Let σ_i be paths from the start to v_{max} . Then the optimal path cost is $c_{max} = \min_i \text{cost}(\sigma_i)$, where i indicates each path from the start to v_{max} . Note that we define c_{max} as the minimal path cost. Now, we want to find the optimal number of robots when the cost (c_p) is given. It is represented by $\min r$, where r is the number of robots corresponding to Table I. It needs to satisfy the following conditions: (i) $1 < r \leq n$, (ii) $\bigcup_i P_i = V$, (iii) $\bigcap_i P_i = \emptyset$, and (iv) $\text{cost}(\pi_i) \leq c_p, \forall i$. Here, n is the maximum number of robots. The conditions of Definition 2 are mathematically modified as follows:

(Def. 2-1) $|P_i| = |\pi_i|$,

(Def. 2-2) $P_i \subset V$,

(Def. 2-3) $\bigcap_i P_i = \emptyset$, and

(Def. 2-4) $\bigcup_i P_i = V$.

TABLE I

THE CORRESPONDING RELATIONS BETWEEN ROBOTS, PATHS, PATH COSTS, AND THE SETS OF KPOIS THAT MAKE UP THE PATHS.

robots	paths	costs	KPOIs
1	π_1	$\text{cost}(\pi_1)$	P_1
\vdots	\vdots	\vdots	\vdots
r	π_r	$\text{cost}(\pi_r)$	P_r

IV. COVERAGE OPTIMIZATION

We have discussed the parallel coverage problem, the objectives, and the concept of the coverage strategies in Section III. This section outlines the optimization associated with the coverage problem. The coverage problem has properties which depend on the available resources (time and robots). Finding a path or a tour demands a strategy that is based on the selection of a path with the minimal cost when we have enough (\leq the number of nodes in the worst case) robots. This strategy is to quickly disperse robots in the environment for a coverage completion. Because of the assumptions associated with the initial location, we can find the minimal cost (c_{max}) in a deterministic fashion. We also take into account the cost (c_p) which is initially given. The properties in Prop. 1 characterizes the minimization of

the coverage cost, and Prop. 2 describes the optimization associated with the number of robots at a given time (c_p). For proofs we select the start position (s) from V , since every robot shares the same s at the beginning.

Proposition 1: (Minimum Path Cost). The minimum path cost for the PPCP is the shortest path cost from the start position to the position in V , which has the maximum cost from the start.

Proof: Let $w_{i,i+1}$ be the cost from the position v_i to the position v_{i+1} . Then $\text{cost}(\pi_{ij}) = \sum_{t=v_i, \dots, v_{j-1}} w_{t,t+1}$, where π_{ij} is the shortest path from v_i to v_j . Let $s \notin V$ be the start position. We choose an end position $v_d (= v_{max})$ as follows:

$$d = \arg \max_i \text{cost}(\pi_{si}), \forall v_i \in V.$$

Let $P_1 = \{v_1, \dots, v_d\}$ be a subset of V included in the selected shortest path (π_{sd}) from s to v_d , and $c_{max} = \text{cost}(\pi_{sd})$ be the minimum path cost. Then

$$\exists v_i \in V \setminus P_1 \cdot \text{cost}(\pi_{si}) \leq c_{max} \text{ if } V \setminus P_1 \neq \emptyset.$$

Let $P_2 = P_1 \cup$ a subset of $V \setminus P_1$ containing the path π_{si} . Similarly, we can find all paths containing every position in V until $V \setminus P_i = \emptyset$. Since $\text{cost}(\pi_{sj}) \leq c_{max}, \forall v_j \in V$, the minimum cost is $\text{cost}(\pi_{sd})$, which is the path from the start to the v_{max} in V . ■

Proposition 2: (Optimal Number of Robots). Let c_p be the desired time of completion for the parallel coverage problem. Then the minimal number of robots is an integer number in the range of 1 to $|V| - |P_{max}| + 1$. Specifically, it is the number of disjoint subsets of KPOIs consisting of paths with the condition of $\text{cost}(\pi_i) \leq c_p, \forall i = 1, \dots, r$.

Proof: Let c_{max} be the path (π_{max}) cost from the start to $v_{max} \in V$, and P_{max} be a subset of V corresponding to π_{max} .

i) We first show the minimum and the maximum numbers of robots. To show the maximum number of robots, let $V' = V \setminus P_{max}$. Then $|V'| \leq n - 1, (\because P_{max} \neq \emptyset)$ (includes at least v_{max}). Since $\bigcap_i P_i = \emptyset$ (by Def. 2-3) and $\bigcup_i P_i = V, \forall i$ (by Def. 2-4),

$$\exists v_j \in V' \cdot \exists \cdot |P_j| = 1, \text{cost}(\pi_j) \leq c_{max} \text{ for } \forall j = 1, \dots, |V'|.$$

Therefore, there are at most $|V| - |P_{max}| + 1$ paths. Now, when $V = V \setminus P_{max} = \emptyset, |V| = 1$ and the minimum number of paths is only one. These extreme cases are shown in Figs. 2(c) and 2(d), respectively. Hence,

$$1 \leq k \leq |V| - |P_{max}| + 1.$$

ii) Since the coverage problem cannot have a lower cost than c_{max} , we only consider the condition of $c_p \geq c_{max}$. Let $\Phi = \{\pi_1, \dots, \pi_k\}$ be a set of paths. Then

$$\text{cost}(\pi_i) \leq c_{max} \leq c_p,$$

for $i = 1, \dots, k$. Since $\bigcap_i P_i = \emptyset$ (by Def. 2-3) and $\bigcup_i P_i = V$ (by Def. 2-4), k is the number of disjoint subsets of V . The examples showing the selection of the paths are shown in Figs. 2(a) and 2(b). Therefore, let $\pi_1 = \{s, \dots, v_j, \dots, v_{max}\}$ for $v_j \in V$, where v_{max} is an element

in V with the maximum cost from s . Then $\exists P_i \subset V \cdot \exists \cdot v_j \in \pi_i \Rightarrow v_j \in P_i$ (satisfying $\bigcap_i P_i = \emptyset$ and $\bigcup_i P_i = V$), and the minimum number of robots is the number of P_i , disjoint subsets of V . ■

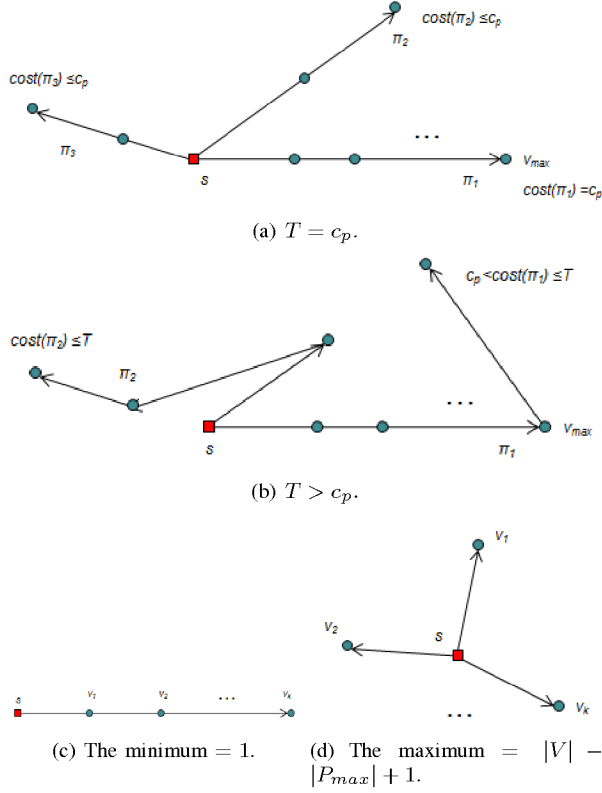


Fig. 2. Examples showing the relationship between paths, path costs, and the number of robots.

Proposition 3: (k -TSP). Given a number (k) of robots, the PTCP in a graph representation (a dense graph) is considered as the k -agents Traveling Salesman Problem (k -TSP). Any algorithms such as k NN (k -Nearest Neighbor) or k NI (k -Nearest Insertion) could afford minimal costs.

Prop. 3 describes how the parallel coverage problem could be connected with well-known problems such as the k -TSP one. Fig. 3 shows an example that every node is located at the same distance from the start. The tours (Fig. 3(b)) are produced by connecting the shortest paths (Fig. 3(a)).

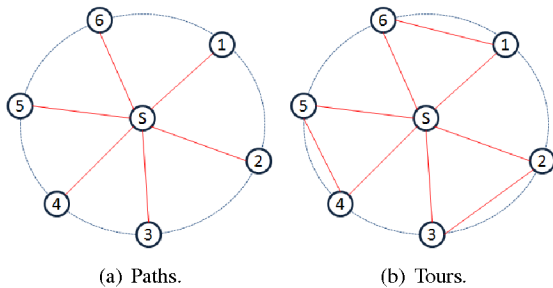


Fig. 3. The choice of the shortest paths and tours in the worst case. The position s is the start, and every other position is located at the same distance from s . The required number of robots is $|V| = 6$ and $\frac{|V|}{2} = 3$ for the paths and the tours, respectively.

V. COVERAGE ALGORITHM

We have discussed the parallel coverage problem in Section III and some optimization properties in Section IV. This section takes into account the representation with the generation of the virtual paths and the path/tour coverage algorithms. Any representations such as a grid map (a cell representation) or a graph map can be used, and we here choose a graph representation.

For a certain number of locations of interest on a map, we need to find a path to connect them. Since an environment may not be an open space (obstacle free), a straight line between KPOIs may not be possible as a path. We adopt intermediate points to connect the KPOIs. Let us call the resulting path as a *virtual path* as described in Definition 3. The construction of a map and the decision of the positions of interests are out of the scope of this paper. We delegate these issues to the exploration and mapping research fields. We apply a geometric algorithm [12] to find a path between positions in the obstacle-free space in order to produce a graph. A graph representation with the given KPOIs on a map as nodes and the selected connections as edges is used in our coverage problem.

Definition 3: (The virtual path and the virtual node). A path $\pi_{i,i+1} = \{v_i, n_1, \dots, n_j, v_{i+1}\}$ is a path from the node v_i to the node v_{i+1} through the virtual nodes n_1, \dots, n_j where $v_i \in V, \forall i$ and $n_1, \dots, n_j \notin V$. The virtual nodes are selected from the open space such that $\min \text{cost}(\pi_{i,i+1})$.

The proposed coverage algorithms find the shortest paths/tours from the start node for an unknown number of robots. Once the virtual paths are achieved, a graph is generated based on the KPOIs as the nodes and the constructed paths as the edges. We first pick a destination node, which is defined as the node with the highest path cost from the beginning. It has the minimum cost for the path coverage problem as we mentioned in Proposition 1. Algorithm I describes the PPCP algorithm as follows:

Algorithm 1: Parallel Path Coverage Algorithm (Optimizes the path cost and finds the optimal number of robots and the corresponding paths.)

Input: the graph $G = (V, E)$ with a given start node $s (\notin V)$.

Output: the optimal number of robots (k) and the optimal path (π_l), where $l = 1, \dots, k$.

- 1) Find shortest paths (π_{si}) from s to every node in V , where $i = 1, \dots, |V|$.
- 2) Choose the node (v_d) and set it to the destination node, where $d = \arg \max_i \text{cost}(\pi_{si})$. Here, $v_d = v_{max}$.
- 3) Set the path (π_{s1}) to the sequence of the shortest path from s to v_d , and set $P_1 = \{v_j | v_j \in \pi_{s1} \cap V\}$. Note that $\pi_{s1} = \pi_{max}$.
- 4) Eliminate the nodes in π_{s1} from V , and $V = V \setminus P_1$.
- 5) Repeat the following until $V = \emptyset$. Set $i = 2$.
 - a) Choose a path (π_{sl}) to connect $v_m (\in V)$ satisfying $m = \arg \min_l \text{cost}(\pi_{sl}), l = 1, \dots, |V|$.

- b) Set the selected node to the end node (v_m) for the path (π_{si}).
 - c) Set $P_i = \{v_j | v_j \in \pi_{si} \cap V\}$, and $V = V \setminus P_i$.
 - d) For $j = 1, \dots, |V|$, repeat the following:
 - Find a node (v_j) in V with the minimum cost from the node (v_m).
 - If $cost(\pi_{si}) + w_{v_m, v_j} \leq cost(\pi_{max})$, insert $\pi_i = \pi_i \cup \{v_j\}$. Otherwise, break.
 - e) Increase i by 1.
- 6) Set k to i .

In the algorithm π_1 is the path having the minimum cost for the PPCP. The steps in 6) find and connect the nodes, available for a path satisfying the condition that the path cost is $\leq c_{max}$. When $c_p \neq c_{max}$, we have the same algorithm. The only difference is that there are supplementary steps between 4) and 5) as follows:

Algorithm 1-1: The supplementary steps for the path coverage algorithm.

For $v_l \in V, \forall l = 1, \dots, |V|$,

- 1) Set $m = \arg \min_l \pi_{sl}$.
- 2) If $cost(\pi_{s1}) + w_{v_d, v_m} \leq c_p$, add $v_m (\in V)$ in π_{s1} and set $v_d = v_m$. Otherwise, break.
- 3) Set $P_1 = P_1 \cup \{v_m\}$.
- 4) Set $V = V \setminus \{v_m\}$.

The tour algorithm combines the shortest paths. The combination is based on the cost between the end KPOIs in each path. The following algorithm shows the PTCP algorithm.

Algorithm 2: Parallel Tour Coverage Algorithm (Optimizes the path cost and finds the optimal number of robots and the corresponding paths.)

Input: the graph $G = (V, E)$ and a set of paths $\Phi = \{\pi_1, \dots, \pi_k\}$.

Output: a set of tours $\{\phi_1, \dots, \phi_{k/2}\}$.

Set $i = 1$ and repeat the followings:

- 1) Pick $a = \arg \max_l \{cost(\pi_{sl}), \forall l = 1, \dots, |\Phi|\}$.
- 2) Set v_d to the final node in the path π_a .
- 3) Pick $b = \arg \min_l \{cost(\pi_{sd}) + w_{v_d, v_l} + cost(\pi_{sl}), \forall l = 1, \dots, |\Phi|\}$.
- 4) Connect $\phi_i = \pi_a + \pi_b$.
- 5) Set $\Phi = \Phi \setminus \{\pi_a, \pi_b\}$.
- 6) If $\Phi = \emptyset$, break. Otherwise increase i by 1.

VI. EXPERIMENTAL RESULTS

For the experimental validation, we show finding the minimal number of robots and the corresponding paths/tours by using the proposed algorithms. For PPCP we compare the results of the proposed algorithm with a Minimal Spanning Tree (MST) and a Nearest Neighbor (NN). For PTCP we compare the proposed tour algorithm with the solutions from the k-TSP (kNN and kNI).

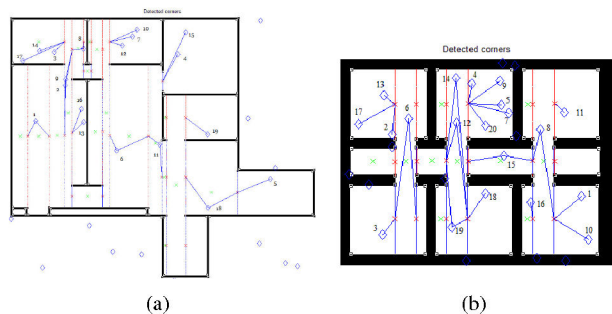


Fig. 4. The randomly generated KPOIs (the blue diamonds) where the robots must go through, a vertical cell decomposition (the dashed lines), and the connections between each position and the corresponding first virtual nodes (the red crosses).

We generate a graph using the image map as in Fig. 4 showing an office and a simple room layout (maps from the Player/Stage, a robot simulator [13]). The blue diamonds are the given KPOIs, randomly generated in this case. The edges have the weights represented by the Euclidean distance. The generated graphs are almost complete graphs, and the path length between nodes is often greater than a straight line due to routes through virtual nodes (the red crosses). We discard the generated positions if they are located outside of the layout. We apply a corner detection algorithm on the maps, and the small squares located at each corner correspond to the detected corners. The dashed lines are the selected vertical lines generated by applying a vertical cell decomposition algorithm for each corner point.

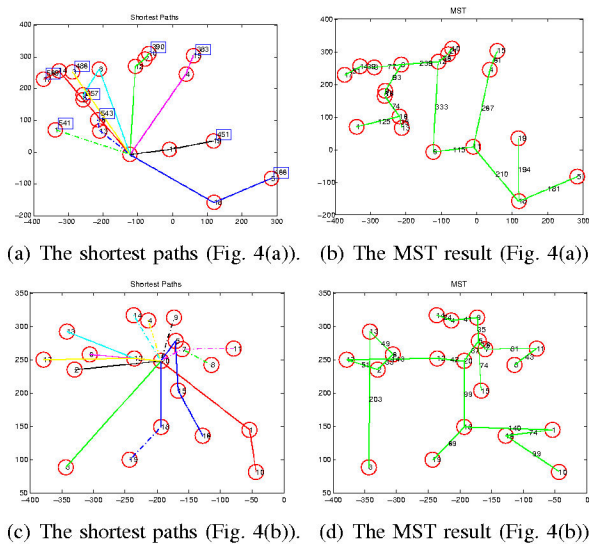


Fig. 5. The results showing the shortest paths and the result of the MST, starting with start node the node 6 (up) and 20 (down). The minimum costs in (a) and (b) are 569 and 739, respectively.

We first generate a graph from each map, and shortest paths are then computed by using the parallel path algorithm. Figs. 5(a) and 5(c) show the results of shortest paths corresponding to Figs. 4(a) and 4(b), respectively. The start node can be chosen as any node, but each graph shows node 6 and node 20 as the start. As a

result, the minimum cost to the destination (node 17) is 569 in Fig. 5(a). In this case the minimal number of robots is nine, and the paths for nine robots are $\{6; 18; 5\}; \{6; 11; 19\}; \{6; 4; 15\}; \{6; 12; 7; 10\}; \{6; 8; 7; 2\}; \{6; 3\}; \{6; 13; 16\}; \{6; 14; 17\}$ and $\{6; 1\}$, respectively. We compare the result of the proposed algorithm with the MST as in Fig. 5. Figs. 5(b) and 5(d) show each result of the MST. From the node 6, the minimum cost to the node 17 (the furthest) is 739 with the path $\{6; 12; 3; 14; 17\}$. Fig. 6(a) shows the comparison of the minimum costs of the proposed algorithm with the MST and NN according to the choices of the start node (the x-axis), and Fig. 6(b) shows the minimal number of robots generated by the proposed algorithm. The proposed algorithm gives the minimum cost when compared with the NN or MST results.

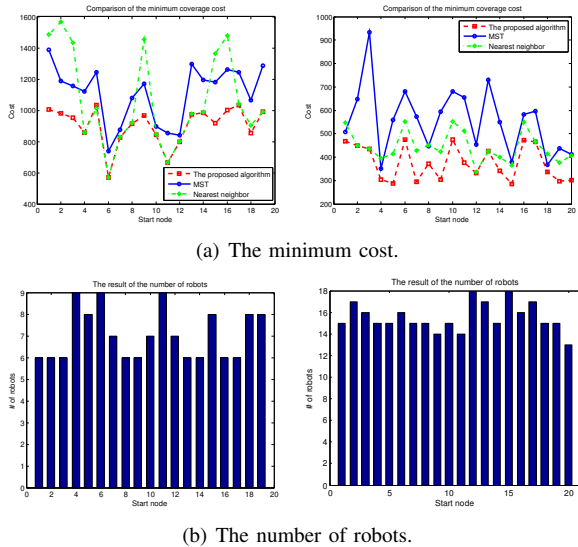


Fig. 6. The results comparing the minimum coverage costs of the proposed algorithm with the MST and the NN (up) and the selected number of robots in each start node (down) for each map in Fig. 4.

Fig. 7 shows the result of the shortest tours of the map in Fig. 4(a) for the minimal number of robots. Fig. 7(a) is the result of four tours requiring four robots (the minimal number of robots for the PTCP) and the minimum tour cost is 1500. Fig. 7(b) shows the result of 4-NN tours, and the minimum cost is 2440. We have also applied the 4-NI tour algorithm, and the minimum tour cost is 2531.

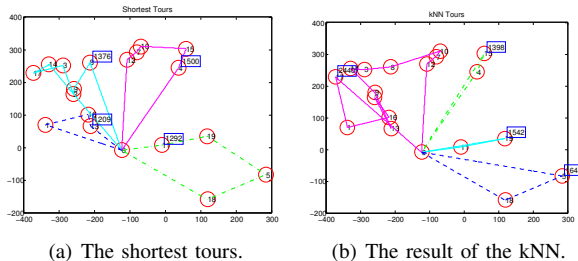


Fig. 7. The result showing the shortest tours from node 6 as the start node (a) and the result of the kNN algorithm (b). The minimum costs among tours are 1500 and 2440 in (a) and (b), respectively.

VII. CONCLUSIONS AND FUTURE WORK

This work presented a new approach regarding a coverage problem that involves an unknown number of multiple robots. It also showed the optimal number of robots required for a given time frame, and the speed by which a search mission could be accomplished. We also highlighted that the performance is dependent on the available resources (time and number of robots). For search missions, particularly when it is required to complete the task within a given time frame, we proposed the parallel path/tour algorithms. They provide the minimal number of robots, which is less or equal to the number of nodes in a graph in the worst case, and the corresponding paths. We also presented the parallel tour algorithm as a combination of optimal paths. Not only we introduced a modified coverage problem, but we also proved the optimality regarding costs and the number of robots. Our work could be applicable to many real-world robotic applications. For future work we will try implementation of our algorithms on the Explorer robots, and investigate alternative strategies to improve the results.

VIII. ACKNOWLEDGEMENTS

This material is based upon work supported in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under contract #911NF-08-1-0463 (Proposal 55111-CI) and the National Science Foundation through grants #CNS-0324864, #IIP-0443945, #IIP-0726109, #CNS-0708344, #IIP-0934327, #IIS-1017344, #IIP-1032018, #CNS-1039741, and #CNS-1061489.

REFERENCES

- [1] K. Konolige, D. Fox, C. Ortiz, A. Agno, M. Eriksen, B. Limketkai, J. Ko, B. Morisset, D. Schulz, B. Stewart, and R. Vincent, *Centibots: Very Large Scale Distributed Robotic Teams*, experimental robotics ix ed. Springer-Verlag Berlin Heidelberg, 2006, pp. 131–140.
- [2] N. Agmon, N. Hazon, and G. A. Kaminka, “The giving tree: constructing trees for efficient offline and online multi-robot coverage,” *Annals of Mathematics and Artificial Intelligence*, vol. 52, pp. 143–168, 2008.
- [3] R. Sedgewick, *Algorithms*. Addison-Wesley, 1983.
- [4] S. Koenig, “Agent-centered search,” *AI Magazine*, vol. 22, pp. 109–131, 2001.
- [5] A. Datta and C. Icking, “Competitive searching in a generalized street,” in *Prof. 10th Annual ACM Symp. on Computational Geometry*, 1994, pp. 175–182.
- [6] A. Weitzenfeld, “A prey catching and predator avoidance neural-schema architecture for single and multiple robots,” *Journal of Intelligent and Robotic Systems*, vol. 51, pp. 203–233, 2008.
- [7] N. Agmon, S. Kraus, and G. A. Kaminka, “Multi-robot perimeter patrol in adversarial settings,” in *IEEE Int. Conf. on Robotics and Automation*, 2008, pp. 2339–2345.
- [8] M. Moors, T. Röhling, and D. Schulz, “A probabilistic approach to coordinated multi-robot indoor surveillance,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2005, pp. 3447–3452.
- [9] G. N. Frederickson, M. S. Hecht, and C. E. Kim, “Approximation algorithms for some routing problems,” *SIAM Journal on Computing*, vol. 7, pp. 178–193, 1978.
- [10] J. L. Ny, E. Feron, and E. Frazzoli, “On the curvature-constrained traveling salesman problem,” *IEEE Transactions of Automatic Control*, 2008.
- [11] V. Bonifaci and L. Stougie, “Online k -server routing problems,” *Theory of Computing Systems*, vol. 45, pp. 470–485, 2009.
- [12] S. M. LaValle, *Motion planning*, 1st ed. NY, USA: Cambridge University Press, 2006.
- [13] (2010) The Player project. [Online]. Available: <http://playerstage.sourceforge.net>