Robotic Adversarial Coverage (ICAPS-15 Doctoral Consortium)

Roi Yehoshua
Supervisors: Dr. Noa Agmon and Prof. Gal A. Kaminka
Computer Science Department
Bar Ilan University
Ramat-Gan 52900, Israel
roiyeho@gmail.com

Abstract

Coverage is a fundamental problem in robotics, where one or more robots are required to visit each point in a target area at least once. While all previous work concentrated on finding a solution that completes the coverage as quickly as possible, in this thesis I consider a new and more general version of the problem: *adversarial coverage*. Here, the robot operates in an environment that contains threats that might stop the robot. The objective is to cover the target area as quickly as possible, while minimizing the probability that the robot will be stopped before completing the coverage. The adversarial coverage problem has many real-world applications, from performing coverage missions in hazardous environments such as nuclear power plants or the surface of Mars, to surveillance of enemy forces in the battle field and field demining. In my thesis I intend to formally define the adversarial coverage problem, analyze its complexity, suggest different algorithms for solving it and evaluate their effectiveness both in simulation and on real robots.

Introduction

Area coverage is an important task for mobile robots, with many real-world applications in various domains, from automatic floor cleaning (Colegrave and Branch 1994) and coating in supermarkets (Endres, Feiten, and Lawitzky 1998) and train stations (Yaguchi 1996), to humanitarian missions such as search and rescue and field demining (Nicoud and Habib 1995). In these, a robot is given a bounded work-area, possibly containing obstacles, and is required to visit every part of it as efficiently as possible.

While all previous studies of the coverage problem concentrated on finding a solution that completes the coverage as quickly as possible, in this thesis I consider a new version of the problem: *adversarial coverage*. Here, the robot operates in an environment that contains threats that might stop the robot. Each point in the area is associated with a probability of the robot being stopped at that point and the probabilities can vary from one point to another. The objective of the robot is to complete the given mission—to cover the *entire* target area—as quickly as possible while minimizing the probability that the robot will be stopped before completing the coverage.

The adversarial coverage problem has an intrinsic complexity that is not present in the general coverage problem, since it presents a delicate tradeoff between minimizing the accumulated risk and minimizing the total coverage time. Trying to minimize the risk involved in the coverage path could mean making some redundant steps, which in turn can make the coverage path longer, and thus increase the risk involved, as well as increase the coverage time.

The adversarial coverage problem has many different variants:

- Offline vs. Online. In the offline version of the problem, the map of threats is given in advance, therefore the coverage path of the robot can be determined prior to its movement. Conversely, in the online version of the problem, the robot has no map or a priori information about the environment.
- Approximate vs. Exact cellular decomposition. In exact decomposition, the free space is decomposed into a set of regions, whose union fills the entire area exactly, while in approximate cellular decomposition the free space is approximately covered by a grid of equally-shaped cells.
- Single-Robot vs. Multi-Robot. In the single-robot version of the problem, only one robot is used to cover the entire target area. If it is stopped by a threat, the coverage mission cannot be completed. In the multi-robot version of the problem, a team of robots is used to cover the target area. Even if one robot is totally damaged, others may take over its coverage subtask.
- Static vs. Dynamic environments. In dynamic environments, the environment itself can change during the coverage process. In our case, this means that the locations of the threats and/or the obstacles can change over time. In static environments only the states of the robots can change with the passage of time.
- Delaying vs. Stopping threats. The threats may inflict different effects on the robot covering the area. For example, they may delay its movement for a certain amount of time or they can stop it completely.
- Arbitrary vs. Planned threats. We can assume that the threats are arbitrarily placed on the map, or that they are placed by an adversary who strives to inflict the maximum damage on the robot and ultimately prevent it from

Copyright © 2015, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

completing its coverage mission. This adversary can have different levels of knowledge on the robot's coverage plan (from zero-knowledge to full-knowledge).

Related Work

The problem of robot coverage has been extensively discussed in the literature (see (Galceran and Carreras 2013) for a recent exhaustive survey). Grid-based coverage methods, such as we utilize here, use a representation of the environment decomposed into a collection of uniform grid cells, e.g., (Gabriely and Rimon 2003), (Luo et al. 2002).

The coverage problem is analogous to the traveling salesman problem, which is $\mathcal{NP}$-complete even on simple graphs such as grid graphs (Papadimitriou 1977). However, it is possible to find solutions to the coverage problem that are close to optimal in polynomial or even linear time through heuristics and abstractions (e.g., (Arkin, Fekete, and Mitchell 2000), (Gabriely and Rimon 2003), (Grigni, Koutsoupias, and Papadimitriou 1995), (Xu, Viriyasuthee, and Rekleitis 2011)).

Papers in the robotic literature that take into account the presence of an adversary such as (Bortoff 2000), (Likhachev and Stentz 2007), (Zabarankin, Uryasev, and Pardalos 2002), present algorithms and methods for risk avoidance. These works examine the path planning problem of a single robot, in order to bypass and avoid the adversary's threats. In the patrol problem (Elmaliach, Agmon, and Kaminka 2009), a multi-robot team needs to patrol around a closed area with the existence of an adversary attempting to penetrate into the area. The patrol problem resembles the coverage problem 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 often seeks to maximize it (while still visiting all points).

Other optimization problems related to adversarial coverage include the Canadian Traveller Problem (CTP) (Papadimitriou and Yannakakis 1989), in which the objective is to find the expected shortest path between two nodes in a partially-observable graph, where some edges may be non-traversable. In contrast, here the graph is fully-observable and the agent must visit every node in the graph (some of them may stop the robot).

Adversarial Coverage Problem Definition

We are given a map of a target area T, which contains obstacles and also points with threats, which may stop the robot. We assume that T can be decomposed into a regular square grid with n cells, whose size equals the size of the robot. Some cells in T contain threat points. Each threat point i is associated with a threat probability p_i, which measures the likelihood that the threat will stop the robot. The robot's task is to plan a path through T such that every accessible free cell in T is visited by the robot at least once.

Figure 1 shows an example map of the world. Obstacles are represented by black cells, safe cells are colored white and dangerous cells are represented by 5 different shades of purple. Darker shades represent higher values of p_i (more dangerous areas).

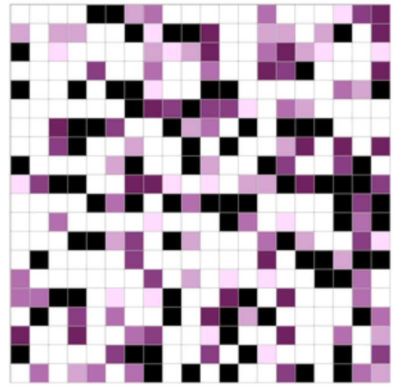

Figure 1: An example map of the world. Obstacles are colored black and dangerous cells are colored gray. Darker purple cells represent more dangerous areas.

We consider two different objectives in regard to the robot's survivability:

1. Minimize the total accumulated risk along the coverage path (i.e., maximize the probability of covering the whole target area).
2. Maximize the coverage percentage of the target area before the robot is first hit (i.e., maximize the expected coverage percentage).

Let us now formally define these objective functions. First, we denote the coverage path followed by the robot by $A = (a_1, a_2, ..., a_m)$. Note that $m \geq n$, i.e., the number of cells in the coverage path might be greater than the number of cells in the target area, since the robot is allowed to repeat its steps. We define the event S_A as the event that the robot is not stopped when it follows the path A. The probability that the robot is able to complete this path is:

$$P(S_A) = \prod_{i \in (a_1, \ldots, a_m)} (1 - p_i) \tag{1}$$

Thus, the first objective is to find a coverage path A that maximizes the probability $P(S_A)$. Note that in this objective, the order of visits of the cells is not important, as long as the number of visits of threat points along the coverage path is minimized (ideally, visiting each threat point only once).

For the second objective, we denote the sequence of newly discovered cells along the coverage path A by $(b_1, ..., b_n)$. Note that $b_i \neq b_j$ for each $i \neq j$, and the number of cells in this sequence is exactly the number of cells in the grid (n). For each cell in the sequence b_i, we will denote the sub-path in A that leads from the origin cell a_1 to it by g_i. Let the number of the new cells discovered by the robot until it is stopped be C_A. Then, under the threat probability function p, the expected number of new cells that the robot visits until it is stopped can be expressed as:

$$E(C_A) = \sum_{i \in (b_1, \ldots, b_n)} \prod_{j \in g_i} (1 - p_j) \tag{2}$$

i.e., $E(C_A)$ is the sum of the probabilities to reach all the newly discovered cells along the coverage path.

Thus, the second objective is to find a coverage path A that maximizes the expected coverage $E(C_A)$. Note that in this objective, the visit order of the cells is crucial, since the robot is trying to cover as much as possible before getting hit by a threat (ideally, covering all the safe cells before visiting a single threat point).

To illustrate these definitions, let us consider the following simple grid, which is composed of 4 cells: a_{11}, a_{12}, a_{21} and a_{22}, with the probabilities for danger p_{ij} specified in each cell.

0	0.1
0.2	0.5

Assume that the initial location of the robot is in cell a_{11}. Since there are no obstacles in this grid, there are coverage paths that visit each cell exactly once. These paths have both minimum length and maximum probability to complete. In our example there are two such coverage paths: $A_1 = (a_{11}, a_{12}, a_{22}, a_{21})$ and $A_2 = (a_{11}, a_{21}, a_{22}, a_{12})$. Their probability to complete is the same and equals to:

$$P(S_{A_1}) = P(S_{A_2}) = 1 \cdot 0.9 \cdot 0.8 \cdot 0.5 = 0.36$$

However, these paths don't have the maximum possible expected coverage. The expected coverage of A_1 is:

$$\begin{aligned} E(C_{A_1}) &= 1 + 1 \cdot 0.9 + 1 \cdot 0.9 \cdot 0.5 + 1 \cdot 0.9 \cdot 0.5 \cdot 0.8 \\ &= 1 + 0.9 + 0.45 + 0.36 = 2.71 \end{aligned}$$

A similar computation shows that the expected coverage of A_2 is: $E(C_{A_1}) = 2.56$.

However, the path with the maximum expected coverage is $A_3 = (a_{11}, a_{12}, a_{11}, a_{21}, a_{22})$. To compute its expected coverage, we first note that the sequence of new cells discovered along this path is $(a_{11}, a_{12}, a_{21}, a_{22})$. Thus, the expected coverage of A_3 is:

$$\begin{aligned} E(C_{A_3}) &= 1 + 1 \cdot 0.9 + 1 \cdot 0.9 \cdot 1 \cdot 0.8 \\ &\quad + 1 \cdot 0.9 \cdot 1 \cdot 0.8 \cdot 0.5 \\ &= 1 + 0.9 + 0.72 + 0.36 = 2.98 \end{aligned}$$

Therefore, by making one additional step, the robot is able to raise its expected number of covered cells from 2.71 to 2.98.

The robot's task is to plan a path through T such that every accessible free cell in T (including the threat points) is visited by the robot at least once. In particular, given T, three questions may be asked:

1. What is the minimum coverage time for T, and at what survivability?
2. What is the maximum survivability for T, and at what coverage time?
3. Given required levels of survivability and coverage time, what is the optimal coverage path?

In order to help us answer these questions, we will define the following weighted cost function that takes both the survivability and the coverage time factors into consideration. For a given coverage path A, define

$$f(A) = -\alpha \cdot \mathcal{S}(A) + \beta \cdot |A| \quad (3)$$

where $\alpha, \beta \geq 0$ are given up front, according to the required balance between the risk and the time factors. $\mathcal{S}(A)$ is the survivability of the robot, which can be measured in two different ways as explained earlier, and $|A|$ is the number of the steps the robot needs to take in order to complete the coverage path. Thus, the problem is to find a coverage path A that minimizes the cost function $f(A)$, i.e., $f(A) \leq f(B)$ for all possible coverage paths B.

When $\alpha = 0$, objective (3) translates to finding a minimum time coverage path, regardless of the risk involved. Achieving this objective will provide an answer to our first question, which is equivalent to finding a solution to the general coverage problem. This means that the coverage problem becomes a special case of the adversarial coverage problem. When $\beta = 0$, the objective translates to finding a coverage path with a minimal risk, without a limit on the path length. Achieving this objective will provide an answer to our second question. Lastly, setting fixed levels for α and β will provide an answer to our third question. In the last case, the ratio α/β will determine how strongly the objective prefers safer coverage paths over shorter ones.

Main Results

In (Yehoshua, Agmon, and Kaminka 2013) we have formally defined the offline adversarial coverage problem for a single robot. We have proposed an initial heuristic algorithm that generates a coverage path which tries to minimize a cost function, that takes into account both the survivability of the robot and the coverage path length. However, the heuristic algorithm worked only for obstacle-free areas, and without any guarantees.

In (Yehoshua, Agmon, and Kaminka 2014) we have addressed a specific version of the adversarial coverage problem, namely, finding the safest coverage path. We have shown that the problem is $\mathcal{NP}$-Complete, and thus we have suggested two heuristic algorithms for solving the safest path problem: STAC and GSAC. STAC (Spanning-Tree Adversarial Coverage) splits the target area into connected areas of safe and dangerous cells, and then it covers the safe areas before moving to the dangerous ones. On the other hand, GSAC (Greedy Safest Adversarial Coverage) follows a greedy approach, which leads the robot from its current location to the nearest safest location which has not been covered yet. We have provided optimality bounds on both algorithms, and proven that these algorithms produce close to optimal solutions in polynomial time. Experimental results have shown that while STAC tends to achieve higher expected coverage, GSAC produces shorter coverage paths with lower accumulated risk (see Figure 2).

In (Yehoshua, Agmon, and Kaminka 2015 to appear) we have shown how to model the adversarial coverage problem as a Markov Decision Process (MDP), and proven that finding an optimal policy of the MDP also provides an optimal

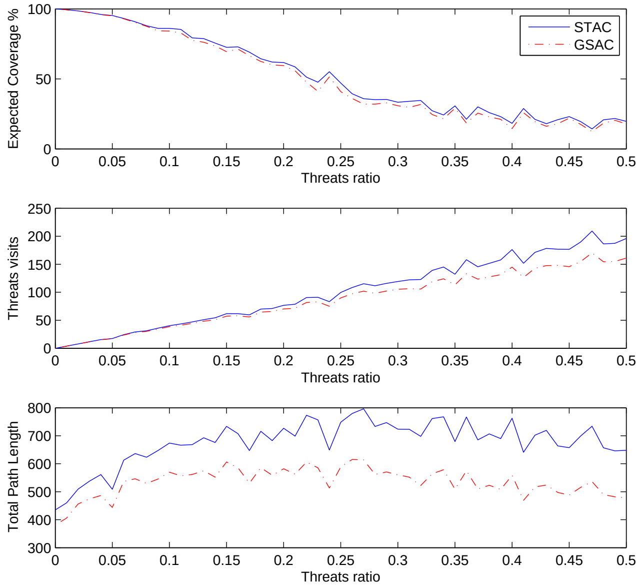

Figure 2: Expected coverage percentage, number of threats visits and coverage path length for different threat ratios in environments with randomly scattered threat points and obstacles.

solution to this problem. The states in the MDP represent all the possible configurations of the environment's coverage status and the robot's location. A coverage status of the environment is represented by a boolean matrix that indicates for each cell in the grid if it has already been visited by the robot or not. The state captures all relevant information from the history of the robot's movements, thus it satisfies the Markovian property. The actions in the model are the four actions the robot can perform - go up, down, left or right. The transition function describes the probability that the robot will be able to move from its current location to the next location on its coverage path.

To demonstrate the model, let us consider the following simple grid (cells are numbered 1 to 2 from top to bottom and left to right, the numbers in the cells indicate the threat probabilities p_i):

0	0
0.4	0.2

Assume that the robot starts the coverage at cell $(1, 1)$ and then moves right to cell $(1, 2)$. Let us denote the current state of the environment and the robot by s_1. Figure 3 shows the graph describing the possible transitions from s_1. Circular nodes of the graph represent states of the MDP and the rectangular nodes represent actions. Inside each state node there is a description of the coverage status of the environment and the robot's position (marked by 'R'). Edges from actions to states are annotated with transition probabilities and costs. See (Yehoshua, Agmon, and Kaminka 2015 to appear) for more details on how the transition probabilities and costs are defined.

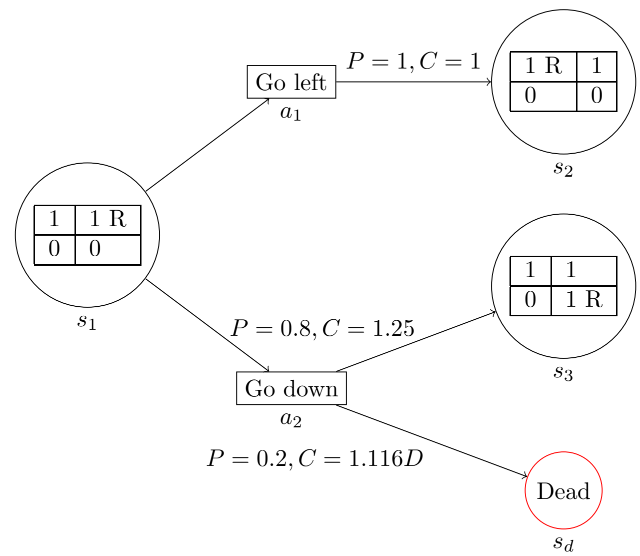

Figure 3: An example for a state in the MDP and its outgoing transitions. Edges from actions to states are annotated with transition probabilities and costs.

Since the state space of the MDP is exponential in the size of the target area's map, we have used real-time dynamic programming (RTDP), a well-known heuristic search algorithm for solving MDPs with large state spaces. Although RTDP achieves faster convergence than value iteration on this problem, practically it cannot handle maps with sizes larger than 7×7. Hence, we have introduced the use of frontiers, states that separate the covered regions in the search space from those uncovered, into RTDP. Frontier-Based RTDP (FBRTDP) avoids fruitless cyclic returns in the search graph, by maintaining a list of *frontier* states. Each time a new state is encountered by an FBRTDP trial, it goes over all its possible successors, and adds to the frontier list all the unvisited successors that are not already in this list. At each step of the trial, FBRTDP examines all the possible paths from the current state to one of the frontier states, and chooses the path to a frontier with the minimum expected cost according to the current value function.

We have shown that Frontier-Based RTDP (FBRTDP) converges orders of magnitude faster than RTDP, and obtains significant improvement over the greedy algorithm (GAC). Figure 4 displays the evolution of the expected cost to the goal as a function of time for the different algorithms. FBRTDP shows the best profile, converging to the optimal policy in only 2.83 seconds, while RTDP, LRTDP (Labeled RTDP) (Bonet and Geffner 2003) and VI (Value Iteration) converge to the optimal policy in 541, 530, and 803 seconds, respectively.

In (Yehoshua and Agmon 2015 to appear) we have built a more sophisticated model of the adversary, in which it can choose the best locations of the threat points, such that the probability of stopping the covering robot is maximized. In

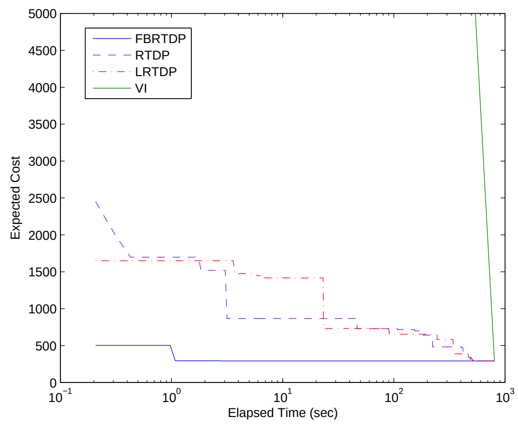

Figure 4: Expected cost to the goal vs. time for VI, RTDP, LRTDP and FBRTDP. The time axis is plotted on a logarithmic scale.

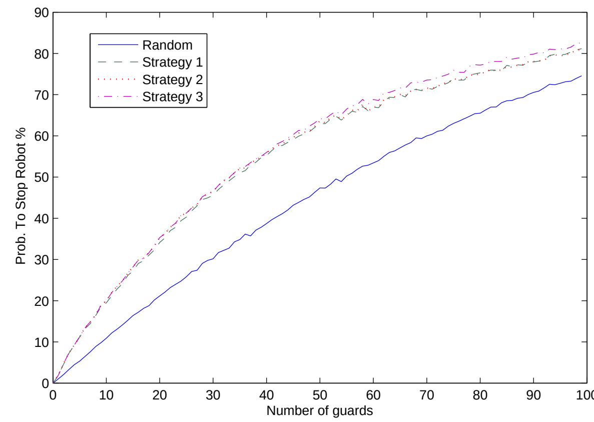

Figure 5: The probability of stopping the covering robot for different numbers of guards.

other words, we have examined the problem of finding the best strategy to defend a given area from being covered by an agent, using k given guards. We have examined the impact of the adversarial knowledge of the coverage path on the choice of the guards' locations, and provided solutions for adversaries having no knowledge and full knowledge of the coverage path. We have shown that for a full-knowledge adversary there is a simple algorithm that provides the optimal strategy, whereas finding an optimal strategy for a zero-knowledge adversary is, in general, $\mathcal{NP}$-Hard. However, for some values of k such an optimal strategy can be found in polynomial time, and for others we have suggested heuristics that can significantly improve the random baseline strategy. We have also discussed some cases in which the adversary has partial knowledge of the coverage path (for example, when it only knows where the coverage begins). Figure 5 shows the probability that the robot will be stopped along its coverage path for different numbers of guards. It compares between different levels of adversarial strategies, where strategy level 0 is the random baseline strategy. See (Yehoshua and Agmon 2015 to appear) for more details about the definition of the different strategies.

Future Work

There are several areas we plan to pursue in future work. First, we are interested in finding algorithms for the online version of the adversarial coverage problem, in which the coverage has to be completed without the use of a map or any a-priori knowledge of the target area. Second, we would like to consider non-stationary environments, where the locations of the threat points can change over time. Finally, we would like to extend the suggested algorithms for multi-robot systems. Using multiple robots for coverage has the potential for more efficient coverage and greater robustness; even if one robot is totally damaged, others may take over its coverage subtask.

References

Arkin, E. M.; Fekete, S. P.; and Mitchell, J. S. 2000. Approximation algorithms for lawn mowing and milling. *Computational Geometry* 17(1):25–50.

Bonet, B., and Geffner, H. 2003. Labeled RTDP: Improving the convergence of real-time dynamic programming. In *ICAPS*, volume 3, 12–21.

Bortoff, S. A. 2000. Path planning for UAVs. In *Proceedings of the American Control Conference*, volume 1, 364–368.

Colegrave, J., and Branch, A. 1994. A case study of autonomous household vacuum cleaner. *AIAA/NASA CIRFFSS* 107.

Elmaliach, Y.; Agmon, N.; and Kaminka, G. A. 2009. Multi-robot area patrol under frequency constraints. *Annals of Mathematics and Artificial Intelligence* 57(3-4):293–320.

Endres, H.; Feiten, W.; and Lawitzky, G. 1998. Field test of a navigation system: Autonomous cleaning in supermarkets. In *IEEE International Conference on Robotics and Automation (ICRA-98)*, volume 2, 1779–1781.

Gabriely, Y., and Rimon, E. 2003. Competitive on-line coverage of grid environments by a mobile robot. *Computational Geometry* 24(3):197–224.

Galceran, E., and Carreras, M. 2013. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems* 61(12):1258–1276.

Grigni, M.; Koutsoupias, E.; and Papadimitriou, C. 1995. An approximation scheme for planar graph TSP. In *36th IEEE Annual Symposium on Foundations of Computer Science*, 640–645.

Likhachev, M., and Stentz, A. 2007. Goal directed navigation with uncertainty in adversary locations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-07)*, 4127–4134.

Luo, C.; Yang, S. X.; Stacey, D. A.; and Jofriet, J. C. 2002. A solution to vicinity problem of obstacles in complete coverage path planning. In *IEEE International Conference on Robotics and Automation (ICRA-02)*, volume 1, 612–617.

Nicoud, J. D., and Habib, M. K. 1995. The Pemex-B autonomous demining robot: perception and navigation strategies. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 'Human Robot Interaction and Cooperative Robots'*, volume 1, 419–424.

Papadimitriou, C. H., and Yannakakis, M. 1989. Shortest paths without a map. In *Automata, Languages and Programming*. Springer. 610–620.

Papadimitriou, C. H. 1977. The euclidean travelling salesman problem is np-complete. *Theoretical Computer Science* 4(3):237–244.

Xu, A.; Viriyasuthee, C.; and Rekleitis, I. 2011. Optimal complete terrain coverage using an unmanned aerial vehicle. In *IEEE International Conference on Robotics and Automation (ICRA-11)*, 2513–2519.

Yaguchi, H. 1996. Robot introduction to cleaning work in the east japan railway company. *Advanced Robotics* 10(4):403–414.

Yehoshua, R.; Agmon, N.; and Kaminka, G. A. 2013. Robotic adversarial coverage: Introduction and preliminary results. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-13)*, 6000–6005.

Yehoshua, R.; Agmon, N.; and Kaminka, G. A. 2014. Safest path adversarial coverage. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-14)*, 3027–3032.

Yehoshua, R.; Agmon, N.; and Kaminka, G. A. 2015, to appear. Frontier-based RTDP: A new approach to solving the robotic adversarial coverage problem. In *International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-15)*.

Yehoshua, R., and Agmon, N. 2015, to appear. Adversarial modeling in the robotic coverage problem. In *International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-15)*.

Zabarankin, M.; Uryasev, S.; and Pardalos, P. 2002. *Optimal risk path algorithms*. Springer.