Terrain Coverage with Ant Robots: A Simulation Study

Sven Koenig
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280
skoenig@cc.gatech.edu

Yaxin Liu
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280
yxliu@cc.gatech.edu

ABSTRACT

In this paper, we study a simple means for coordinating teams of simple agents. In particular, we study ant robots and how they can cover terrain once or repeatedly by leaving markings in the terrain, similar to what ants do. These markings can be sensed by all robots and allow them to cover terrain even if they do not communicate with each other except via the markings, do not have any kind of memory, do not know the terrain, cannot maintain maps of the terrain, nor plan complete paths. The robots do not even need to be localized, which completely eliminates solving difficult and time-consuming localization problems. In this paper, we use real-time heuristic search methods to implement ant robots and present a simulation study with several real-time heuristic search methods to study their properties for terrain coverage. Our experiments show that all of the real-time heuristic search methods robustly cover terrain even if the robots are moved without realizing this, some robots fail, and some markings get destroyed. These results demonstrate that terrain coverage with real-time heuristic search methods is an interesting alternative to more conventional terrain coverage methods.

Keywords

Multi-Agent Coordination, Multi-Agent Simulation

1. INTRODUCTION

When one builds physical agents (such as robots) one can either build complex ones or simple ones. In this paper, we study a simple means for coordinating teams of simple agents, in the context of terrain coverage, and demonstrate its advantages. We study both one-time coverage, where the robots visit every location at least once, and continuous coverage, where the robots continuously cover a large terrain without getting switched off, so that every part of the terrain gets visited once every while. Applications of terrain-coverage methods include vacuum cleaning, lawn mowing, crop plowing, contamination cleanup, mine sweeping, surveillance, and surface inspection, yet terrain coverage has been studied far less in the robotics literature than other navigation tasks [7, 9, 8, 13, 22, 37].

Ant robots are simple robots with limited sensing and computational capabilities. They have the advantage that they are simple to design, easy to program, and cheap to build. However, they cannot use conventional planning or coordination methods due to their limited sensing and computational capabilities which limit their planning capabilities even for simple planning tasks such as path planning or the coverage of terrain. Thus, they might not be able to cover terrain as efficiently as robots with more powerful sensing and computational capabilities. On the other hand, groups of ant robots can take advantage of both their fault tolerance (since they fail gracefully even if some ant robots malfunction) and their parallelism (since groups of ant robots can cover terrain faster than a single ant robot) [6]. Small ant-like robots that have the capability to vacuum clean or mow the lawn are already on the consumer market and others are expected to be on the consumer market soon, including the Koala robot, DC06 robot, Electrolux robot, Cye robot, Eureka robot, Robomow robot, Solar Mower robot, and Dophin robot. The question then arises how single ant robots can cover terrain once or repeatedly, and how teams of ant robots can coordinate their activities.

Theoretical work (including some of our own work) suggests that real-time (heuristic) search methods [21] can be used to cover graphs once or repeatedly [16, 34, 15, 35]. In this paper, we present a simulation study with several real-time heuristic search methods to study their properties for terrain coverage and demonstrate their advantages. Real-time search methods have been developed in artificial intelligence as an alternative to more traditional search methods, such as the A* search method [25]. We show that they can be used by single ant robots as well as groups of ant robots to cover terrain even if the ant robots have only very limited computational capabilities and look-aheads. The ant robots only have to leave markings in the terrain, sense the markings at their neighboring locations, and change the marking at their current location. This is what real ants do [1]. Our experiments show that real-time search methods robustly cover terrain even if the ant robots are moved without realizing this (say, by people running into them), some ant robots fail, and some markings get destroyed. These results demonstrate that terrain coverage with real-time search methods is an interesting alternative to more conventional terrain coverage methods.

To summarize, we present robust terrain coverage meth-

AGENTS'01, May 28-June 1, 2001, Montréal, Quebec, Canada.

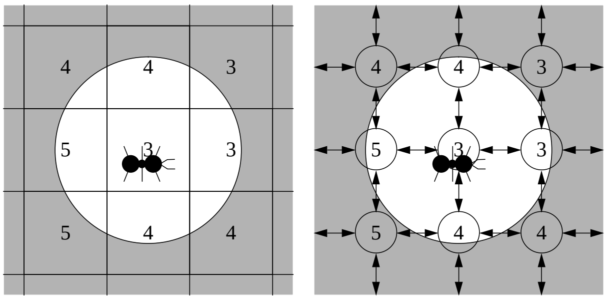

Figure 1: Ant Robots and Real-Time Search

ods that can be used by groups of robots with limited sensing and computational capabilities. Other researchers have also been inspired by ants but study very different methods in the context of "ant colony optimization" to solve discrete optimization problems [11], including finding good routing policies in computer networks [10]. While we use ideas from biology, we use them only at a very high level. We do not strive to imitate real ants faithfully, as some robotics researchers do [23]. However, our ant robots do leave long-lived markings in the terrain (just like real ants) while other robots use only virtual traces [32]. Our ant robots are therefore similar to those robot teams where one robot leaves a short-lived trace in the terrain that another robot then follows [30, 27, 28]. However, different from our approach, those approaches have already been demonstrated on real robots. The purpose of this paper then is to inspire robotics researchers that are interested in the coordination of teams of agents to explore the concept of terrain coverage with real-time search on mobile robots. This includes developing robots that leave long-lived markings in the terrain. The long-lived markings need to have the property that they can be of different intensity (strength) and survive activities such as robots moving over them. Yet, the robots should be able to remove them on purpose. We believe that terrain coverage with real-time search will be a reality some day because robotics researchers have already made progress in this direction. For example, as mentioned above, mobile robots have been built that leave short-lived markings in the terrain such as odor traces [28], heat traces [27], or alcohol traces [30].

2. TASK

We model terrains as directed graphs, for instance by imposing regular grids on them. Figure 1 (left), for example, shows a regular four-connected grid, and Figure 1 (right) shows the corresponding graph. However, we could also derive Voronoi diagrams or similar graph representations of the terrain [24]. The task of the ant robots then is to cover the graph (visit all vertices) once or repeatedly.

3. REAL-TIME SEARCH

Real-time (heuristic) search methods [21] interleave planning and plan execution, and allow for fine-grained control over how much planning to perform between plan executions. Planning is done via local searches, that is, searches that are restricted to the part of the graph around the current vertex of an agent. The white area of Figure 1 illustrates the limited look-ahead of real-time search methods.

S denotes the finite set of vertices (states) of the graph, and $s_{start} \in S$ denotes the start vertex. The number of vertices is $n = |S|$. $A(s) \neq \emptyset$ is the finite, nonempty set of (directed) edges leaving vertex $s \in S$ (actions that can be executed in state s). $succ(s, a)$ denotes the successor vertex that results from the traversal of edge $a \in A(s)$ in vertex $s \in S$. We also use two operators with the following semantics: Given a finite set X, the expression "one-of$X$" returns an element of X according to an arbitrary rule. A subsequent invocation of "one-of$X$" can return the same or a different element. The expression "$\arg\min_{x \in X} f(x)$" returns the elements $x \in X$ that minimize $f(x)$, that is, the set $\{x \in X | f(x) = \min_{x' \in X} f(x')\}$, where f is a function from X to the non-negative integers.

Initially, the u-values $u(s)$ are zero for all $s \in S$.

1. $s := s_{start}$.
2. $a :=$ one-of $\arg\min_{a \in A(s)} u(succ(s, a))$.
3. Update $u(s)$ using the value-update rule.
4. Traverse edge a.
5. $s := succ(s, a)$.
6. Go to 2.

Figure 2: Real-Time Search with Look-Ahead One

ods. A good overview of real-time search methods is given in [14]. Real-time search methods have been developed in artificial intelligence as an alternative to more traditional search methods in deterministic and nondeterministic domains. For example, variants of LRTA* have successfully been used for traditional search [21], STRIPS-type planning [5], and planning with totally observable Markov decision processes models [4] and partially observable Markov decision processes models [12]. In this paper, we apply real-time search methods in a very different way, namely to terrain coverage.

We study several real-time search methods that have a look-ahead of only one edge traversal and fit the algorithmic skeleton shown in Figure 2. All of them associate a u-value $u(s)$ with each vertex $s \in S$ and initialize them with zeroes. The u-values correspond to markings that the ant robots leave at the vertices of the graphs, see Figure 1. This is what some insects do. Real ants, for example use chemical (pheromone) traces to guide their navigation [1]. The ant robots always decide which neighboring vertex to move to based only on the u-values of the neighboring vertices. Before moving to that vertex, the ant robots can change the u-value of their current vertex. The real-time search methods first decide which edge the ant robots should traverse in their current vertex (Line 2). They look one edge traversal ahead (larger look-aheads are possible) and always greedily let the ant robots traverse the edge that leads to a neighboring vertex with a smallest u-value (ties can be broken arbitrarily). Then, they update the u-value of their current vertex using a value-update rule that depends on the semantics of the u-values and thus the real-time search method (Line 3). We assume that the ant robots execute Lines 2 and 3 together and atomically. Finally, all real-time search methods let the ant robots traverse the selected edge (Line 4), update the current vertex (Line 5), and iterate the procedure (Line 6). All real-time search methods can be executed by ant robots without memory, require only a very limited look-ahead and computational capabilities, and can

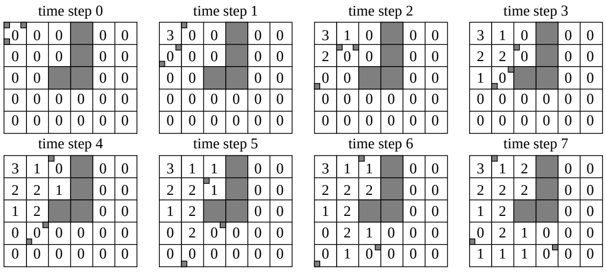

Figure 3: Example: Node Counting

be used by single ant robots as well as groups of ant robots that share the u-values but do not directly communicate with each other. These properties match the limited sensing and computational capabilities of ant robots.

We study four real-time search methods that differ only in their value-update rules, see Table 1. (For the notation, see Figure 2.) For example, Node Counting always moves its ant robot to the adjacent vertex that has been visited the least number of times by all ant robots. Figure 3 demonstrates how three ant robots that each use Node Counting cover a regular four-connected grid. The ant robots can move to the four neighboring cells of their current cell provided that the destination cell is traversable (white). The ant robots share the u-values but do not directly communicate with each other. The cells are marked with their u-values, and the markings play the role of a communication channel. An overview of this kind of indirect communication is given in [2]. For simplicity, we assume in the figure that the ant robots move in a given sequential order and that several ant robots can be in the same location at the same time. If a cell contains an ant robot, one of its corners is marked. Different corners represent different ant robots.

4. TERRAIN COVERAGE

A strongly connected graph is a directed graph which has a path from each vertex to every other vertex. In this section, we show that ant robots that use one of the four real-time search methods cover strongly connected graphs repeatedly and thus avoid cycling forever in parts of the graphs, despite their limited sensing and computational capabilities. We prove this first for Node Counting, by contradiction, using arguments similar to those used in [26, 29]. Assume that, from some point in time on, the ant robots do not cover the graphs. Then, there is some (possibly later) point in time when they only visit those vertices again that they visit infinitely often; they cycle on part of the graphs. The u-values of all vertices in the cycle then increase beyond every bound since Node Counting increases the smallest u-value of the vertices in the cycle by at least one every time an ant robot leaves the vertex. But then the u-values of all vertices in the cycle increase above the u-values of all vertices that border the cycle. Such vertices exist according to our assumptions that the graph is strongly connected but will not be covered again. Then, however, at least one ant robot is forced to leave the cycle, which is a contradiction. The same argument also applies unchanged to ant robots that use the other real-time search methods. This argument holds no matter in which order the ant robots move even if some ant robots move less often than others. Thus, the real-time search methods cover graphs repeatedly. Furthermore, they have advantages over other search methods that also cover graphs repeatedly. For example, they are far more systematic than random walks and, different from chronological backtracking (depth-first search), can be suspended and restarted elsewhere, without even knowing where they get restarted. This is important for terrain coverage with ant robots because sometimes the ant robots might get pushed accidentally to a different location. Most of the time they will not even realize this. Ant robots that use real-time search methods handle these situations automatically. Different from most coverage path planning approaches, they do not need to know (or learn) a map of the terrain.

5. THEORETICAL PROPERTIES

Several research groups (including ours) have studied theoretical properties of real-time search methods, most notably how quickly they reach a goal vertex. If a real-time search method needs x steps in the worst case to reach a goal vertex on a given graph (where an adversary can determine which vertex is the goal vertex), then its cover time is also x steps in the worst case, and vice versa. Thus, the theoretical results transfer to the cover time of real-time search methods and thus the application studied in this paper. We are able to unify the previous theoretical results with the following theorem.

THEOREM 1. *Let s_t denote the current vertex of a single ant robot at point in time t, and s_{t+1} the vertex of the ant robot after it has traversed the chosen edge. Let $u_t(s)$ denote the u-value of vertex s immediately before the ant robot updates the u-value of s_t, and $u_{t+1}(s)$ the same u-value immediately after the update. The time it takes the single ant robot to cover strongly connected graphs with n vertices is at most $(f(n)+1)f(n)n^2$ if all of the following conditions hold.*

1. *At every point in time t, the u-values $u_t(s)$ are integers.*
2. *At every point in time t, it holds that $u_t(s_t) \leq u_{t+1}(s_t)$.*
3. *At every point in time t where $u_t(s_t) \leq \min_{a \in A(s_t)} u_t(succ(s_t, a))$, it holds that $u_t(s_t) + 1 \leq u_{t+1}(s_t)$.*
4. *At every point in time t, it holds that $|u_t(s) - u_t(succ(s,a))| \leq f(n)$ for all vertices $s \in S$ and $a \in A(s)$, where $f(n) \geq 1$ is a function of the number of vertices n.*

We prove this theorem in [20]. It is then easy to show that the theorem holds for LRTA* on undirected graphs with $f(n) = 1$ and directed graphs with $f(n) = n - 1$ using previous results in [17, 34], for Wagner's Value-Update Rule on undirected graphs with $f(n) = 1$ using previous results in [33], and for Thrun's Value-Update Rule with $f(n) = n$ using previous results in [31]. Consequently, the cover time of these three real-time search methods is polynomial in the number of vertices in the worst case. It is known that the cover time of Node Counting can be exponential in the number of vertices in the worst case [18] even on undirected graphs [19]. We are also able to generalize these results to groups of ant robots.

Table 1: Different Real-Time Search Methods

Value-Update Rules (Line 3)	
$u(s) := 1 + u(s)$.	Node Counting [26, 3]
$u(s) := 1 + u(succ(s,a))$.	Learning Real-Time A* (LRTA*) [21, 17]
if $u(s) \leq u(succ(s,a))$ then $u(s) := 1 + u(s)$.	Wagner's Value-Update Rule [33]
$u(s) := \max(1 + u(s), 1 + u(succ(s,a)))$.	Thrun's Value-Update Rule [31]

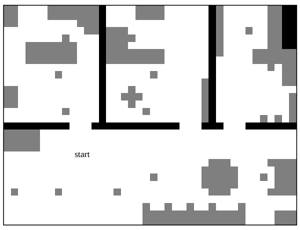

Figure 4: Terrain

	individual markings		shared markings	
	μ	σ	μ	σ
Wagner's Rule	884.75	230.09	481.00	86.479
LRTA*	853.09	212.10	458.09	81.456
Node Counting	868.92	219.07	458.96	80.474
Thrun's Rule	839.96	216.64	439.06	73.950

Table 2: Cover Time for Eight Ant Robots with Individual and Shared Markings

While a polynomial cover time is certainly desirable, these theoretical properties are not sufficient to suggest that terrain coverage with ant robots is a good idea. For example, vacuum cleaning ant robots should visit each cell approximately equally often in the long run and the visits should be spaced approximately equally far apart. Furthermore, they have to be robust in the presence of failures, such as when ant robots are moved without realizing this (say, by people running into them) or markings get destroyed. In the following we therefore present a simulation study with the four real-time search methods that investigates these properties.

6. SIMULATION STUDY

In this section, we report simulation results for the cover times of vacuum-cleaning ant robots in part of an office building that contains three offices and a small waiting area, see Figure 4 (the results for different terrains were similar). We imposed a regular four-connected grid over the terrain, resulting in 40×30 cells. The ant robots could move to each of the four neighboring cells of their current cell provided that the destination cell was traversable (white). Cells were untraversable if they contained either walls (black) or furniture (grey). All ant robots started near the cell marked "start" and used the same real-time search method, breaking ties randomly. The ant robots executed Lines 2 and 3 together and atomically, and changed the markings of their current cells just before they moved. Each ant moved once during each time step in a given sequential order.[1]

[1]We also performed experiments where the ants moved asynchronously or in random order but this led to results similar to the ones reported here.

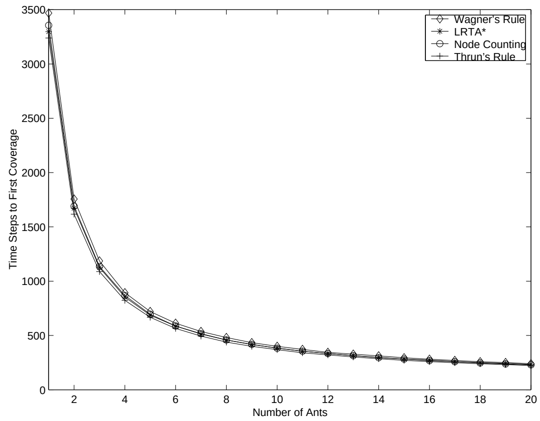

Figure 5: Cover Time

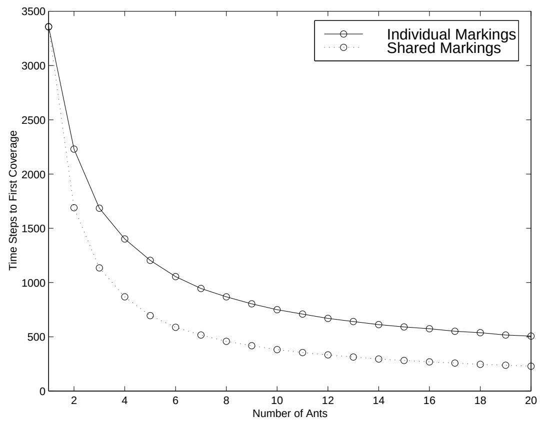

Figure 6: Cover Time for Node Counting with Individual and Shared Markings

In our first three experiments, we studied the cover times of the four real-time search methods for one-time vacuum cleaning and their visit frequencies for continuous vacuum cleaning. In our first experiment, we measured the number of time steps until the ant robots covered the terrain for the first time (cover time), averaged over 2,000 runs. This is important for one-time vacuum cleaning since each cell has to be vacuumed at least once. We assume that each move of an ant robot takes one time step and that each ant robot moves once during each time step. Figure 5 shows the results. The trend was the same for all real-time search methods. The cover times improved as more ant robots were added although the rate of improvement decreased. The question arises whether sharing the markings actually helps the ant robots to cover terrain faster and thus how effective a means of communication the markings are. We therefore compared

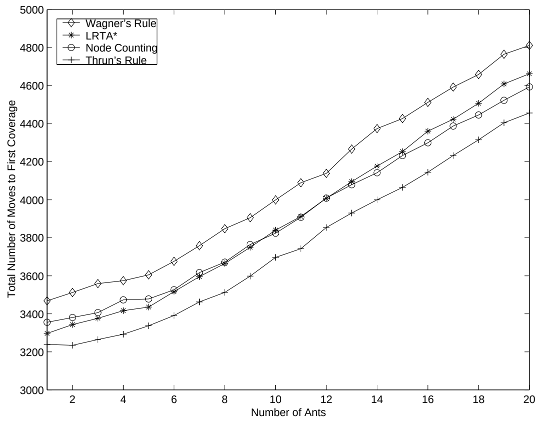

Figure 7: Total Number of Moves to First Coverage

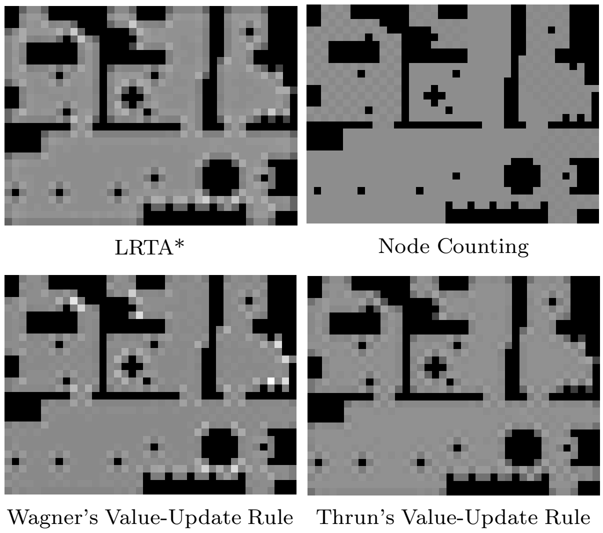

Figure 8: Visit Frequencies for Repeated Coverage

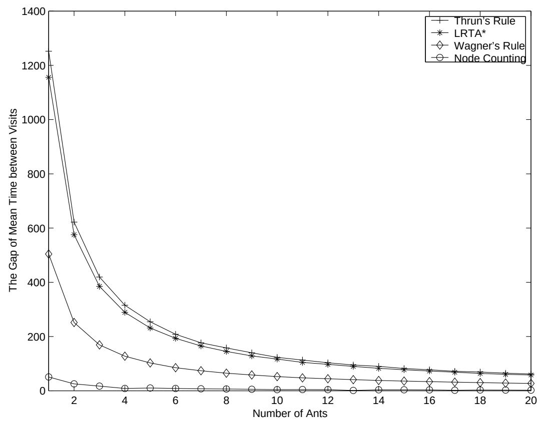

Figure 9: Difference of Largest and Smallest Mean Time between Visits for Repeated Coverage

the cover times of ant robots that share their markings, the case discussed so far, and ant robots that each use their individual markings. Figure 6 shows the results for Node Counting. Sharing the markings cuts the cover time approximately in half. The result is similar for the other real-time search methods. Table 2, for example, contains the means and standard deviations of the cover times for eight ant robots. From now on, we consider only shared markings. Figure 5 showed that the cover times of all real-time search methods were similar. To differentiate better between the real-time search methods, Figure 7 shows the total number of moves made by all ant robots, that is, (roughly) the number of ant robots times the cover time. It appears that the difference in the total number of moves of the different real-time search methods was independent of the number of ant robots. Thrun's Value-Update Rule was best, followed by LRTA* and Node Counting (that were almost indistinguishable), and finally followed by Wagner's Value-Update Rule. The cover time of single ant robots was between 3,200 and 3,500 with a standard deviation between 600 and 700. Thus, the difference in performance was dominated by the standard deviation. This experimental result is interesting because of the theoretical results that show that the cover time of Node Counting can be exponential while the cover time of LRTA* is guaranteed to be polynomial in the number of vertices. We also experimented with random walks but their cover time was much larger than those of the real-time search methods above. For example, the cover time of single ant robots that use random walks was approximately 49,000 with a standard deviation of about 20,000.

In our second experiment, we measured the frequency with which the ant robots visited each cell of the terrain when they covered the terrain repeatedly. This is important for continuous vacuum cleaning because one probably wants to vacuum each cell equally often in the long run (although one could argue that most dirt accumulates where people walk and thus that one wants to vacuum cells close to obstacles less often). It is also of theoretical interest [36]. We stopped the ant robots after 2,000,000 time steps. The visit frequencies did not change significantly as the number of ant robots increased, no matter which real-time search methods were used. We therefore report the visit frequencies for single ant robots only. Figure 8 shows the results. Darker cells in the figure were visited less frequently. As the figure shows, some real-time search methods visited the cells more uniformly than others. For example, Node Counting visited the cells much more uniformly than LRTA*. This is interesting because the total number of movements of LRTA* and Node Counting was similar in our first experiment. One measure for the uniformity of the visit frequencies is their entropy. The larger the entropy $-\sum_s P(s) \log_2 P(s)$, the closer to uniform the visit frequencies $P(s)$ are. The entropy of uniform visit frequencies is 9.7830 and the entropies of the visit frequencies of the real-time search methods were: Node Counting (9.7829), Wagner's Value-Update Rule (9.7779), Thrun's Value-Update Rule (9.7772), and LRTA* (9.7727). Another measure for the uniformity of the visit frequencies is the difference between the mean times between visits to the cells with the largest and smallest mean time between visits. The smaller this value, the closer to uniform the visit frequencies. Figure 9 shows the results, that were similar to those for the entropies. Only the positions of Thrun's Value-Update Rule and LRTA* were switched.

In our third experiment, we measured whether the times

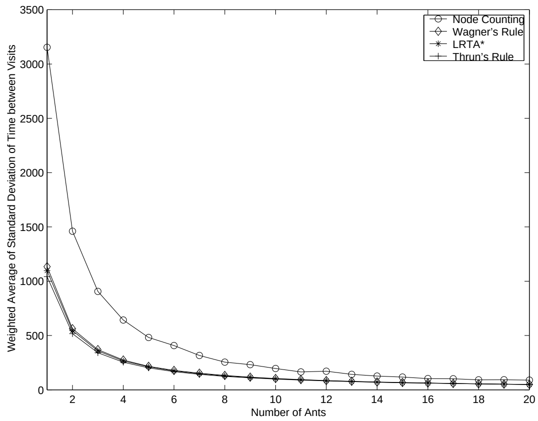

Figure 10: Standard Deviation of the Time between Visits for Repeated Coverage

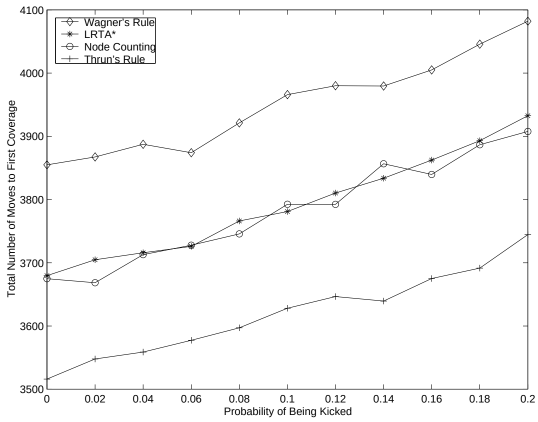

Figure 11: Total Number of Moves for First Coverage if the Ant Robots Were Kicked

between visits to the cells were spread out evenly. This is important for continuous vacuum cleaning because it seems better if a cell gets visited every 100 time steps than if it gets visited every 10 time steps for 9 times in a row and then again only after 910 time steps. One measure of how evenly the times between visits are spread out is the average of the standard deviations of the times between visits over all cells, weighted with their visit frequencies. The closer to zero this value, the more evenly spread out the times between visits. We stopped the ant robots after 2,000,000 time steps. Figure 10 shows the results. The times between visits were very unevenly spread out for Node Counting and more evenly spread out for all other real-time search methods, with almost no difference among them. This is interesting because Node Counting had very uniform visit frequencies in our second experiment.

In the next three experiments, we studied the cover times of the real-time search methods for one-time vacuum cleaning under various failure conditions. (We also performed experiments where we studied the visit frequencies of the real-time search methods for continuous vacuum cleaning but the effect of the failure conditions on the visit frequencies was small.) In our fourth experiment, we measured the robustness of the real-time search methods when the ant

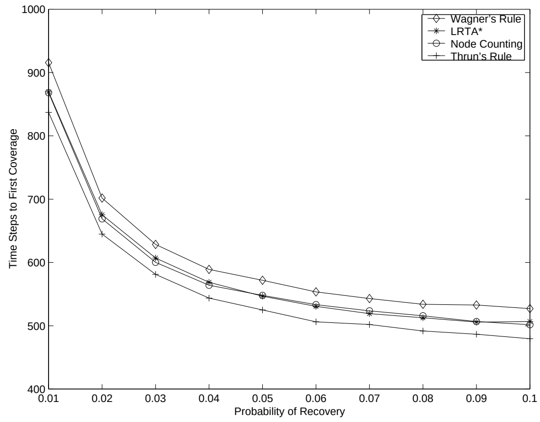

Figure 12: Cover Time for First Coverage if Ant Robots Malfunction (recover probability varies)

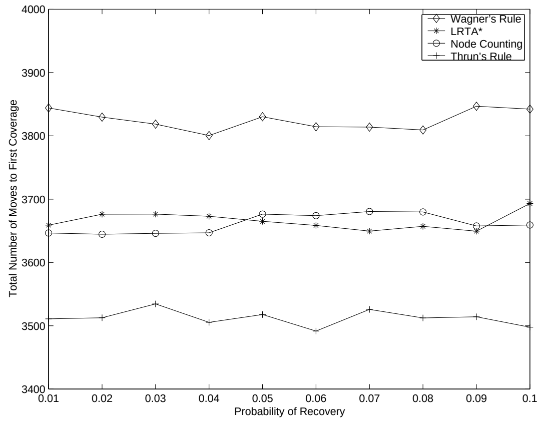

Figure 13: Total Number of Moves for First Coverage if Ant Robots Malfunction (recover probability varies)

robots were moved away from their current cell without realizing this. This is important because people can easily run into small vacuum-cleaning ant robots and accidentally push them to a different location. During each time step, with a given probability exactly one ant robot was moved (otherwise no ant robot was moved). If an ant robot was moved, exactly one ant robot and its new cell were chosen with uniform probability, with the restriction that the ant robot was moved by at most two cells. Figure 11 shows the total number of moves until eight ant robots covered the terrain for the first time, averaged over 2,000 runs. All real-time search methods continued to cover the terrain, and the number of movements increased gracefully as the probability of being moved increased. The order of the real-time search methods remained unaffected by this failure condition.

In our fifth experiment, we measured the robustness of the real-time search methods when the ant robots failed. This is important because robots can malfunction. During each time step, each functional ant robot failed with a given failure probability and each failed ant robot can recovered with a given recovery probability. Failed robots did not move. Figures 12 shows the cover time and Figure 13 shows the total number of moves until eight ant robots covered the ter-

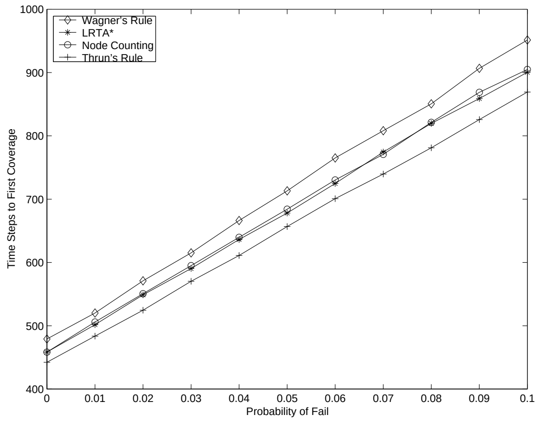

Figure 14: Cover Time for First Coverage if Ant Robots Malfunction (failure probability varies)

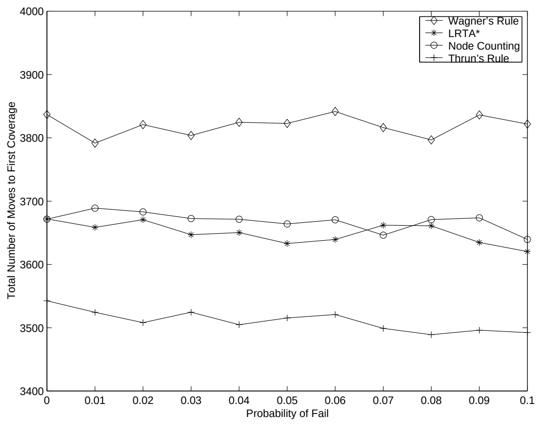

Figure 15: Total Number of Moves for First Coverage if Ant Robots Malfunction (failure probability varies)

rain for the first time, averaged over 2,000 runs. We kept the failure probability constant at 0.01 and varied the recovery probability from 0.01 to 0.10 (implying that it took a failed ant robot an average of 10 to 100 time steps to recover). Figures 14 and 15 show the results of a similar experiment where we kept the recovery probability constant at 0.10 and varied the failure probability from 0.00 to 0.10. All real-time search methods continued to cover the terrain as long as the failure probability was no larger than the recovery probability. The number of movements remained roughly the same and the cover time increased gracefully as the failure probability increased or the recovery probability decreased. The order of the real-time search methods remained unaffected by this failure condition.

In our sixth and last experiment, we measured the robustness of the real-time search methods when the markings were erased. This is important because physical markings can get destroyed. During each time step, with a given probability exactly one marking was erased (otherwise no marking was erased). If a marking was erased, exactly one cell was chosen with uniform probability and its u-value was set to zero (because ant robots cannot distinguish between a cell without a marking and a cell whose marking was destroyed). Figure 16

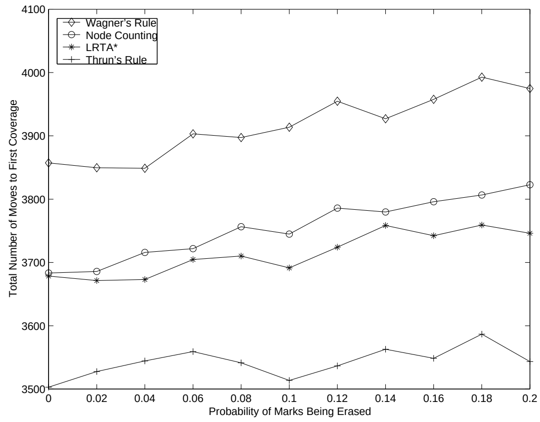

Figure 16: Total Number of Moves for First Coverage if Markings Were Erased

shows the total number of moves until eight ant robots covered the terrain for the first time, averaged over 2,000 runs. All real-time search methods continued to cover the terrain, and the number of movements increased gracefully as the probability with which markings were erased increased. The order of the real-time search methods remained unaffected by this failure condition.

7. CONCLUSIONS

In this paper, we studied a simple means for coordinating teams of simple agents. In particular, we studied ant robots and how they can cover terrain once or repeatedly (as required for vacuum cleaning, lawn mowing, crop plowing, contamination cleanup, mine sweeping, surveillance, and surface inspection) by leaving markings in the terrain, similar to what ants do. These markings can be sensed by all robots and allow them to cover terrain even if they do not communicate with each other except via the markings, do not have any kind of memory, do not know the terrain, cannot maintain maps of the terrain, nor plan complete paths. The robots do not even need to be localized, which completely eliminates solving difficult and time-consuming localization problems. We presented a simulation study that demonstrates that terrain coverage with real-time search methods is an interesting alternative to more conventional terrain coverage methods. All real-time search methods robustly cover terrain even if the robots are moved without realizing this (say, by people running into them), some robots fail, and some markings get destroyed. Our results also reveal some trade-offs between the different real-time search methods. For example, Thrun's Value-Update Rule minimized the total number of moves to first coverage. Node Counting had the most uniform visit frequencies but, unfortunately, spread out the times between visits to a location more unevenly than the other real-time search methods. These results provide a first step towards an implementation of terrain coverage with ant robots on real robots since they supplement existing theoretical results about the cover time of real-time search methods with data for more realistic scenarios than those that can be analyzed formally.

8. ACKNOWLEDGMENTS

The Intelligent Decision-Making Group is partly supported by an NSF Award to Sven Koenig under contract IIS-9984827. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. All experiments were performed by Yaxin Liu.

9. REFERENCES

[1] F. Adler and D. Gordon. Information collection and spread by networks of patrolling ants. *The American Naturalist*, 140(3):373–400, 1992.

[2] R. Arkin. *Behavior-Based Robotics*. MIT Press, 1998.

[3] T. Balch and R. Arkin. Avoiding the past: A simple, but effective strategy for reactive navigation. In *International Conference on Robotics and Automation*, pages 678–685, 1993.

[4] A. Barto, S. Bradtke, and S. Singh. Learning to act using real-time dynamic programming. *Artificial Intelligence*, 73(1):81–138, 1995.

[5] B. Bonet, G. Loerincs, and H. Geffner. A robust and fast action selection mechanism. In *Proceedings of the National Conference on Artificial Intelligence*, pages 714–719, 1997.

[6] R. Brooks and A. Flynn. Fast, cheap, and out of control: A robot invasion of the solar system. *Journal of the British Interplanetary Society*, pages 478–485, 1989.

[7] H. Choset, E. Acar, A. Rizzi, and J. Luntz. Exact cellular decompositions in terms of critical points of morse functions. In *International Conference on Robotics and Automation*, 2000.

[8] H. Choset and P. Pignon. Coverage path planning: The boustrophedon decomposition. In *International Conference on Field and Service Robotics*, 1987.

[9] R. N. de Carvalho, H. Vidal, P. Vieira, and M. Ribeiro. Complete coverage path planning and guidance for cleaning robots. In *Proceedings of the International Symposium on Industrial Electronics*, pages 677–682, 1997.

[10] G. Di Caro and M. Dorigo. Antnet: Distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research*, 9:317–365, 1998.

[11] M. Dorigo, V. Maniezzo, and A. Colorni. The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics; Part B*, 26(1):29–41, 1996.

[12] H. Geffner and B. Bonet. Solving large POMDPs by real-time dynamic programming. Technical report, Departamento de Computación, Universidad Simón Bolivar, Caracas (Venezuela), 1998.

[13] S. Hert, S. Tiwari, and V. Lumelsky. A terrain-covering algorithm for an AUV. *Autonomous Robots*, 3:91–119, 1996.

[14] T. Ishida. *Real-Time Search for Learning Autonomous Agents*. Kluwer Academic Publishers, 1997.

[15] S. Koenig. Exploring unknown environments with real-time search or reinforcement learning. In *Proceedings of the Neural Information Processing Systems*, pages 1003–1009, 1999.

[16] S. Koenig and R. Simmons. Exploration with and without a map. In *Proceedings of the AAAI Spring Symposium on Learning Action Models*, pages 28–32, 1993. Available as AAAI Technical Report WS-93-06.

[17] S. Koenig and R. Simmons. Real-time search in non-deterministic domains. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1660–1667, 1995.

[18] S. Koenig and R. Simmons. Easy and hard testbeds for real-time search algorithms. In *Proceedings of the National Conference on Artificial Intelligence*, pages 279–285, 1996.

[19] S. Koenig and B. Szymanski. Value-update rules for real-time search. In *Proceedings of the National Conference on Artificial Intelligence*, pages 718–724, 1999.

[20] S. Koenig, B. Szymanski, and Y. Liu. Efficient and inefficient ant coverage methods. *Annals of Mathematics and Artificial Intelligence – Special Issue on Ant Robotics*, 2001. (In Print).

[21] R. Korf. Real-time heuristic search. *Artificial Intelligence*, 42(2-3):189–211, 1990.

[22] D. Kurabayashi, J. Ota, T. Arai, and E. Yoshida. Cooperative sweeping by multiple mobile robots. In *Proceedings of the International Conference on Robotics and Automation*, 1996.

[23] D. Lambrinos, R. Möller, R. Labhart, R. Pfeifer, and R. Wehner. A mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems*, 30:39–64, 2000.

[24] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[25] N. Nilsson. *Problem-Solving Methods in Artificial Intelligence*. McGraw-Hill, 1971.

[26] A. Pirzadeh and W. Snyder. A unified solution to coverage and search in explored and unexplored terrains using indirect control. In *Proceedings of the International Conference on Robotics and Automation*, pages 2113–2119, 1990.

[27] R. Russell. Heat trails as short-lived navigational markers for mobile robots. In *Proceedings of the International Conference on Robotics and Automation*, pages 3534–3539, 1997.

[28] R. Russell, D. Thiel, and A. Mackay-Sim. Sensing odour trails for mobile robot navigation. In *Proceedings of the International Conference on Robotics and Automation*, pages 2672–2677, 1994.

[29] S. Russell and E. Wefald. *Do the Right Thing – Studies in Limited Rationality*. MIT Press, 1991.

[30] R. Sharpe and B. Webb. Simulated and situated models of chemical trail following in ants. In *Proceedings of the International Conference on Simulation of Adaptive Behavior*, pages 195–204, 1998.

[31] S. Thrun. Efficient exploration in reinforcement learning. Technical Report CMU-CS-92-102, School of Computer Science, Carnegie Mellon University, Pittsburgh (Pennsylvania), 1992.

[32] R. Vaughan, K. Stoey, G. Sukhatme, and M. Mataric. Whistling in the dark: Cooperative trail following in uncertainty localization space. In *Proceedings of the International Conference on Autonomous Agents*, 2000.

[33] I. Wagner, M. Lindenbaum, and A. Bruckstein. On-line graph searching by a smell-oriented vertex process. In S. Koenig, A. Blum, T. Ishida, and R. Korf, editors, *Proceedings of the AAAI Workshop on On-Line Search*, pages 122–125, 1997. Available as AAAI Technical Report WS-97-10.

[34] I. Wagner, M. Lindenbaum, and A. Bruckstein. Efficiently searching a dynamic graph by a smell-oriented vertex process. *Annals of Mathematics and Artificial Intelligence*, 24:211–223, 1998.

[35] I. Wagner, M. Lindenbaum, and A. Bruckstein. Distributed covering by ant-robots using evaporating traces. *IEEE Transactions on Robotics and Automation*, 15(5):918–933, 1999.

[36] V. Yanovski, I. Wagner, and A. Bruckstein. Edge ant walk for patrolling networks. In *Proceedings of the International Workshop on Ant Algorithms*, 2000.

[37] A. Zelinsky, R. Jarvis, J. Byrne, and S. Yuta. Planning paths of complete coverage of an unstructured environment by a mobile robots. In *Proceedings of the International Conference on Advanced Robotics*, 1993.