Open Problem: Analyzing Ant Robot Coverage*

Sven Koenig
Computer Science Department
University of Southern California
941 W 37th Street
Los Angeles, CA 90089-0781, USA
skoenig@usc.edu

Abstract

Ant robots can repeatedly and robustly cover terrain by always moving away from the trails that they leave in the terrain. This coverage strategy can be modeled with graph traversal strategies similar to real-time search methods (such as Learning Real-Time A*) and reinforcement learning methods (such as Real-Time Dynamic Programming). The resulting worst-case cover times are known to be exponential in the number of vertices on both directed and undirected graphs in general. The known undirected graphs with large worst-case cover times have unbounded degree vertices. However, existing ant robots navigate on grids, a special case of undirected planar graphs with bounded degree vertices. Their experimental cover times appear to scale almost identically to those of coverage strategies with polynomial worst-case cover times. However, it is an open problem to prove whether the resulting worst-case cover times on grids are indeed polynomial in the number of vertices.

Ant robots are robots that either 1) leave trails in the terrain and use them for navigation, similar to learning graphs by dropping indistinguishable pebbles (Bender et al., 2002), and/or 2) use greedy navigation strategies that depend only on local observations of the terrain and thus require only limited sensing, processing and communication capabilities (Wagner & Bruckstein, 2001). Researchers have built actual ant robots that fit both definitions and cover terrain repeatedly by always moving away from the trails that they leave in the terrain, see Figure 1. Single ant robots (individually) and groups of ant robots (cooperatively) cover terrain robustly even if they do not have any memory, do not know the terrain, cannot maintain maps of the terrain nor plan complete paths. They cover terrain even if some ant robots fail, they are moved without realizing this (say, by people running into them and pushing them accidentally to a different location), the trails are of uneven quality or some trails are destroyed. Their coverage strategy can be modeled with Node Counting (Koenig et al., 2001; Wagner et al., 1999), a graph traversal strategy similar to real-time search methods (such as Learning Real-Time A* (Korf, 1990)) and reinforcement learning methods (such as Real-Time Dynamic Programming (Barto et al., 1995)). Node Counting assigns an integer counter $u(s)$ to every vertex (= node) s of the graph, that represents the amount of trail in that location. All counters are initially zero. Every ant robot always increments the counter of a vertex by one when it enters the vertex and then moves to a successor vertex with the smallest counter (using an arbitrary tie breaking rule), see Figure 3. Thus, it moves to a successor vertex that has been visited the least number of times by ant robots, with the idea to quickly get to a vertex that has not yet been visited. Note that Step 3 of Node Counting is: $u(s) := 1 + u(s)$. For simplicity, we consider only a single ant robot in the following since it is easy to generalize the results to groups of ant robots. Node Counting covers strongly connected graphs repeatedly, which is why we assume in the following that the graphs are strongly connected. The worst-case cover times of Node Counting are known to be exponential in the number of vertices on both directed graphs (trivial proof for the graph topology shown in Figure 4 left) and undirected graphs (longer proof for the graph topology shown in Figure 4 right) in general (Koenig et al., 2001). The known undirected graphs with large worst-case cover times are thus (planar) trees with unbounded degree vertices. However, existing ant robots navigate on grids with blocked and unblocked cells, which are special cases of undirected planar graphs with bounded degree vertices, see Figure 2. The experimental cover times of Node Counting on grids appear to scale almost identically to those of known coverage strategies with polynomial worst-case cover times on all strongly connected graphs. These coverage strategies are similar to Node Counting but more difficult to implement on actual ant robots (Koenig & Simmons, 1992), including Learning Real-Time A*. Step 3 of Learning Real-Time A* is: $u(s) := 1 + \min_{a \in A(s)} u(succ(s,a))$. We now list interesting open problems for

*This overview of open problems is based on (Svennebring & Koenig, 2003) and (Koenig et al., 2001) and the figures contained therein. It was supported by, or in part by, NSF under contract/grant number 0413196, ARL/ARO under contract/grant number W911NF-08-1-0468 and ONR under contract/grant number N00014-09-1-1031.

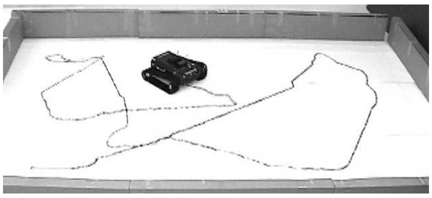

Figure 1: Actual Ant Robot

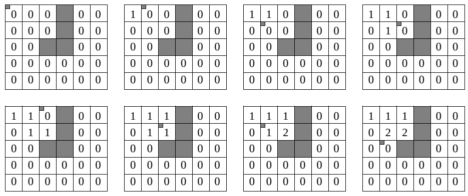

Figure 2: Coverage of Four-Neighbor Grid

We use the following notation: S denotes the finite set of vertices of the graph, and $s_{start} \in S$ denotes the start vertex of an ant robot. $A(s) \neq \emptyset$ is the finite, nonempty set of directed edges that leave vertex $s \in S$. $succ(s, a)$ 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 values $u(s)$ are zero for all $s \in S$.

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

Figure 3: Node Counting

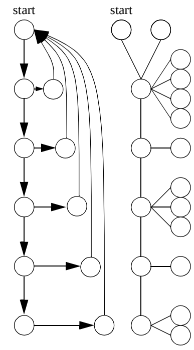

Figure 4: Graphs

Node Counting, the solutions of which would help to lay a solid theoretical foundation for ant robotics and perhaps other kinds of simple agents (such as mobile code that has to explore computer networks): Prove whether the cover times of Node Counting are polynomial in the number of vertices a) for undirected graphs with bounded degree vertices or, if not, b) for grids (a subset of these graphs) if the worst case in both cases is taken over all graphs with a given number of vertices, start vertices and equally good successor vertices (= that is, successor vertices with the smallest counter) and thus tie breaking rules. If not, assume that the ant robot uses the tie breaking rule to select randomly among all equally good successor vertices. Prove whether the resulting cover times are polynomial if the worst case is taken over all graphs with a given number of vertices and start vertices but the average case is taken over all equally good successor vertices. Of course, it is also important to analyze more complex and thus more realistic versions of Node Counting, such as versions that model the saturation of the terrain with trails or the clean-up of trails by the ant robot to avoid such a saturation. For example, Step 3 of a version of Node Counting that models the saturation of the terrain with trails is: with probability $(k - u(s))/k$ execute $u(s) := 1 + u(s)$ for a given positive integer k. Additional information and related work are presented in (Svennebring & Koenig, 2003), in (Koenig et al., 2001) and on the ant robotics web pages at idm-lab.org/antrobots.

References

Barto, A., Bradtke, S., & Singh, S. (1995). Learning to act using real-time dynamic programming. *Artificial Intelligence*, *73*, 81–138.

Bender, M., Fernandez, A., Ron, D., Sahai, A., & Vadhan, S. (2002). The power of a pebble: Exploring and mapping directed graphs. *Information and Computation*, *176*, 1–21.

Koenig, S., & Simmons, R. (1992). *Complexity analysis of real-time reinforcement learning applied to finding shortest paths in deterministic domains* (Technical Report CMU–CS–93–106). School of Computer Science, Carnegie Mellon University, Pittsburgh (Pennsylvania).

Koenig, S., Szymanski, B., & Liu, Y. (2001). Efficient and inefficient ant coverage methods. *Annals of Mathematics and Artificial Intelligence*, *31*, 41–76. The proofs can be found in the technical report 'The Complexity of Node Counting on Undirected Graphs' at idm-lab.org/bib/abstracts/Koen99g.html.

Korf, R. (1990). Real-time heuristic search. *Artificial Intelligence*, *42*, 189–211.

Svennebring, J., & Koenig, S. (2003). Building terrain-covering ant robots. *Autonomous Robots*, *16*, 313–332.

Wagner, I., & Bruckstein, A. (2001). Special issue on ant robotics. *Annals of Mathematics and Artificial Intelligence*, *31*.

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