Mobile Agents and Exploration (1952; Claude Shannon)

Evangelos Kranakis, http://www.scs.carleton.ca/∼kranakis
Danny Krizanc, http://dkrizanc.web.wesleyan.edu/
entry editor:

SYNONYMS and INDEX TERMS: Distributed Algorithms, Graph Exploration, Mobile Agent, Navigation, Rendezvous, Routing, Time/Memory Tradeoffs.

1 PROBLEM DEFINITION

How can a network be explored efficiently with the help of mobile agents? This is a very broad question and to answer it adequately it will be necessary to understand more precisely what mobile agents are, what kind of networked environment they need to probe, and what complexity measures are interesting to analyze.

Mobile agents. Mobile agents are autonomous, intelligent computer software that can move within a network. They are modeled as automata with limited memory and computation capability and are usually employed by another entity (to which they must report their findings) for the purpose of collecting information. The actions executed by the mobile agents can be discrete or continuous and transitions from one state to the next can be either deterministic or non-deterministic, thus giving rise to various natural complexity measures depending on the assumptions being considered.

Network model. The network model is inherited directly from the theory of distributed computing. It is a connected graph whose vertices comprise the computing nodes and edges correspond to communication links. It may be static or dynamic and its resources may have various levels of accessibility. Depending on the model being considered, nodes and links of the network may have distinct labels. A particularly useful abstraction is an anonymous network whereby the nodes have no identities, which means that an agent cannot distinguish two nodes except perhaps by their degree. The outgoing edges of a node are usually thought of as distinguishable but an important distinction can be made between a globally consistent edge-labeling versus a locally independent edge-labeling.

Efficiency measures for exploration. Efficiency measures being adopted involve the time required for completing the exploration task, usually measured either by the number of edge traversals or nodes visited by the mobile agent. The interplay between time required for exploration and memory used by the mobile agent (*time/memory tradeoffs*) are key parameters considered for evaluating algorithms. Several researchers impose no restrictions on the memory but rather seek algorithms minimizing exploration time. Others, investigate the minimum size of memory which allows for exploration of a given type of network (e.g., tree) of given (known or unknown) size, regardless of the exploration time. Finally, several researchers consider time/memory tradeoffs.

Main Problems. Given a model for both the agents and the network, the graph exploration problem is that of designing an algorithm for the agent that allows it to visit all of the nodes and/or edges of the network. A closely related problem is where the domain to be explored is presented as a region of the plane with obstacles and exploration becomes visiting all unobstructed portions of the region in the sense of visibility. Another related problem is that of rendezvous where two or more agents are required to gather at a single node of a network.

2 KEY RESULTS

Claude Shannon [1] is credited with the first finite automaton algorithm capable of exploring an arbitrary maze (which has a range of 5×5 squares) by trial and error means. Exploration problems for mobile agents have been extensively studied in the scientific literature and the reader will find a useful historical introduction in Fraigniaud et al.[2].

Exploration in general graphs. The network is modeled as a graph and the agent can move from node to node only along the edges. The graph setting can be further specified in two different ways. In Deng and Papadimitriou [7] the agent explores strongly connected directed graphs and it can move only in the direction from head to tail of an edge, but not vice-versa. At each point, the agent has a map of all nodes and edges visited and can recognize if it sees them again. They minimize the ratio of the total number of edges traversed divided by the optimum number of traversals, had the agent known the graph. In Panaite and Pelc [8] the explored graph is undirected and the agent can traverse edges in both directions. In the graph setting it is often required that apart from completing exploration the agent has to draw a map of the graph, i.e., output an isomorphic copy of it. Exploration of directed graphs assuming the existence of labels is investigated in Albers and Henzinger [9] and Deng and Papadimitriou [7]. Also in Panaite and Pelc [8], an exploration algorithm is proposed working in time $e + O(n)$, where is n the number of nodes and e the number of links. Fraigniaud et al. [2] investigate memory requirements for exploring unknown graphs (of unknown size) with unlabeled nodes and locally labeled edges at each node. In order to explore all graphs of diameter D and max degree d a mobile agent needs $\Omega(D \log d)$ memory bits even when exploration is restricted to planar graphs. Several researchers also investigate exploration of anonymous graphs in which agents are allowed to drop and remove pebbles. For example in Bender et al. [10] it is shown that one pebble is enough for exploration, if the agent knows an upper bound on the size of the graph, and $\Theta(\log \log n)$ pebbles are necessary and sufficient otherwise.

Exploration in trees. In this setting it is assumed the agent can distinguish ports at a node (locally), but there is no global orientation of the edges and no markers available. *Exploration with stop* is when the mobile agent has to traverse all edges and stop at some node. For *exploration with return* the mobile agent has to traverse all edges and stop at the starting node. In *perpetual exploration* the mobile agent has to traverse all edges of the tree but is not required to stop. The upper and lower bounds on memory for the exploration algorithms analyzed in Diks et al. [11] are summarized in the table, depending

Exploration	Knowledge	Lower Bounds	Upper Bounds
Perpetual	$\emptyset$	None	$O(\log d)$
w/Stop	$n \leq N$	$\Omega(\log \log \log n)$	$O(\log N)$
w/Return	$\emptyset$	$\Omega(\log n)$	$O(\log^2 n)$

on the knowledge that the mobile agent has. Here, n is the number of nodes of the tree, $N \geq n$ is an upper bound known to the mobile agent, and d is the maximum degree of a node of the tree.

Exploration in a geometric setting. Exploration in a geometric setting with unknown terrain and convex obstacles is considered by Blum et al. [3]. They compare the distance walked by the agent (or robot) to the length of the shortest (obstacle-free) path in the scene and describe and analyze robot strategies that minimize this ratio for different kinds of scenes. There is also related literature for exploration in more general settings with polygonal and rectangular obstacles by Deng et al. [4] and Bar-Eli et al. [5], respectively. A setting that is important in wireless networking is when nodes are aware of their location. In this case, Kranakis et al. [6] give efficient algorithms for navigation, namely compass routing and face routing that guarantee delivery in Delaunay and arbitrary planar geometric graphs, respectively, using only local information.

Rendezvous. The rendezvous search problem differs from the exploration problem in that it concerns two searchers placed at different nodes of a graph that want to minimize the time required to rendezvous (usually) at the same node. At any given time the mobile agents may occupy a vertex of the graph and can either stay still or move from vertex to vertex. It is of interest to minimize the time required to rendezvous. A natural extension of this problem is to study multi-agent mobile systems. More generally, given a particular agent model and network model, a set of agents distributed arbitrarily over the nodes of the network are said to rendezvous if executing their programs after some finite time they all occupy the same node of the network at the same time. Of special interest is the highly symmetric case of anonymous agents on an anonymous network and the simplest interesting case is that of two agents attempting to rendezvous on a ring network. In particular, in the model studied by Sawchuk [12] the agents cannot distinguish between the nodes, the computation proceeds in synchronous steps, and the edges of each node are oriented consistently. The table summarizes time/memory tradeoffs known for six algorithms (see Kranakis et al. [13] and Flocchini et al. [14]) when the k mobile agents use indistinguishable pebbles (one per mobile agent) to mark their position in an n node ring.

Memory	Time	Memory	Time
$O(k \log n)$	$O(n)$	$O(\log n)$	$O(n)$
$O(\log n)$	$O(kn)$	$O(\log k)$	$O(n)$
$O(k \log\log n)$	$O\left(\frac{n \log n}{\log\log n}\right)$	$O(\log k)$	$O(n \log k)$

Kranakis et al.[15] show a striking computational difference for rendezvous in an oriented, synchronous, $n \times n$ torus when the mobile agents may have more indistinguishable tokens. It is shown that two agents with a constant number of unmovable tokens, or with one movable token each cannot rendezvous if they have $o(\log n)$ memory, while they can perform rendezvous with detection as long as they have one unmovable token and $O(\log n)$ memory. In contrast, when two agents have two movable tokens each then rendezvous (respectively, rendezvous with detection) is possible with constant memory in a torus. Finally, two agents with three movable tokens each and constant memory can perform rendezvous with detection in a torus. If the condition on synchrony is dropped the rendezvous problem becomes very challenging. For a given initial location of agents in a graph, De Marco et al [16] measure the performance of a rendezvous algorithm as the number of edge traversals of both agents until rendezvous is achieved. If the agents are initially situated at a distance D in an infinite line, they give a rendezvous algorithm with cost $O(D|L_{\min}|^2)$ when D is known and $O((D+|L_{\max}|)^3)$ if D is unknown, where $|L_{\min}|$ and $|L_{\max}|$ are the lengths of the shorter and longer label of the agents, respectively. These results still hold for the case of the ring of unknown size but then they also give an optimal algorithm of cost $O(n|L_{\min}|)$, if the size n of the ring is known, and of cost $O(n|L_{\max}|)$, if it is unknown. For arbitrary graphs, they show that rendezvous is feasible if an upper bound on the size of the graph is known and they give an optimal algorithm of cost $O(D|L_{\min}|)$ if the topology of the graph and the initial positions are known to the agents.

3 APPLICATIONS

Interest in mobile agents has been fueled by two overriding concerns. First, to simplify the complexities of distributed computing, and second to overcome the limitations of user interface approaches. Today they find numerous applications in diverse fields such as distributed problem solving and planning (e.g., task sharing and coordination), network maintenance (e.g., daemons in networking systems for carrying out tasks like monitoring and surveillance), electronic commerce and intelligence search (e.g., data mining and surfing crawlers to find products and services from multiple sources), robotic exploration (e.g., rovers, and other mobile platforms that can explore potentially dangerous environments or even enhance planetary extravehicular activity), and distributed rational decision making (e.g., auction protocols, bargaining, decision making). The interested reader can find useful information in several articles in the volume edited by Weiss [17].

4 OPEN PROBLEMS

Specific directions for further research would include the study of time/memory tradeoffs in search game models (see Alpern and Gaal [18]). Multi-agent systems are particularly useful for content-based searches and exploration, and further investigations in this area would be fruitful. Memory restricted mobile agents provide a rich model with applications in sensor systems. In the geometric setting, navigation and routing in a three dimensional environment using only local information is an area with many open problems.

5 EXPERIMENTAL RESULTS

None is reported.

6 DATA SETS

None is reported.

7 URL to CODE

None is reported.

8 CROSS REFERENCES

Deterministic Search, Online Robotics, Routing, Search.

RECOMMENDED READING

[1] C. Shannon, Presentation of a Maze Solving Machine, in Cybernetics, Circular, Causal and Feedback Machines in Biological and Social Systems. Trans. 8th Conf, March 15-16, 1951, New York, N.Y., H. von Feerster, M. Mead and H. L. Teuber, eds., Josiah Mary Jr. Foundation, New York, 1952, pp. 169-181.

[2] P. Fraigniaud, D. Ilcinkas, G. Peer, A. Pelc and D. Peleg Graph exploration by a finite automaton, Theoretical Computer Science, Vol. 345, pp. 331–344, 2005.

[3] A. Blum, P. Raghavan and B. Schieber, Navigating in unfamiliar geometric terrain, SIAM Journal on Computing 26 (1997), 110-137.

[4] X. Deng, T. Kameda and C. H. Papadimitriou, How to learn an unknown environment I: the rectilinear case, Journal of the ACM 45 (1998), 215-245.

[5] E. Bar-Eli, P. Berman, A. Fiat and R. Yan, On-line navigation in a room, Journal of Algorithms 17 (1994), 319-341.

[6] E. Kranakis, H. Singh, and J. Urrutia, Compass Routing in Geometric Graphs, in proceedings of 11th Canadian Conference on Computational Geometry, CCCG-99, 51–54, Vancouver Aug. 15-18, 1999.

[7] X. Deng and C. H. Papadimitriou, Exploring an unknown graph, Journal of Graph Theory 32 (1999), 265-297.

[8] P. Panaite and A. Pelc, Exploring unknown undirected graphs, Journal of Algorithms 33 (1999), 281-295.

[9] S. Albers and M. R. Henzinger, Exploring unknown environments, SIAM Journal on Computing 29 (2000), 1164-1188.

[10] M.A. Bender, A. Fernandez, D. Ron, A. Sahai and S. Vadhan, The power of a pebble: Exploring and mapping directed graphs, Proc. 30th Ann. Symp. on Theory of Computing (1998), 269–278.

[11] K. Diks, P. Fraigniaud, E. Kranakis, A. Pelc, Tree exploration with little memory, Journal of Algorithms 51 (2004), 38–63.

[12] C. Sawchuk, *Mobile Agent Rendezvous in the Ring*, PhD thesis, Carleton University, School of Computer Science, Ottawa, Canada, 2004.

[13] E. Kranakis, D. Krizanc, N. Santoro, and C. Sawchuk, Mobile Agent Rendezvous Search Problem in the Ring, *Proc. International Conference on Distributed Computing Systems (ICDCS)*, pp. 592-599, 2003.

[14] P. Flocchini, E. Kranakis, D. Krizanc, N. Santoro, and C. Sawchuk, Multiple Mobile Agent Rendezvous in the Ring, *Proc. LATIN 2004*, LNCS 2976, pp. 599-608, 2004.

[15] E. Kranakis, D. Krizanc, E. Markou, Mobile Agent Rendezvous in a Synchronous Torus. In proceedings of LATIN 2006, 7th Latin American Symposium, Valdivia, Chile, March 20-24, 2006. J. Correa, A. Hevia, M. Kiwi. SVLNCS, Vol. 3887, pp. 653-664.

[16] G. De Marco, L. Gargano, E. Kranakis, D. Krizanc, A. Pelc, U. Vaccaro, Asynchronous Deterministic Rendezvous in Graphs. Theoretical Computer Science, 355 (2006), 315-326.

[17] G. Weiss, ed., Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, MIT Press, 1999.

[18] S. Alpern and S. Gal, *The Theory of Search Games and Rendezvous*, Kluwer Academic Publishers, Norwell, Massachusetts, 2003.