2009 International Symposium on Nonlinear Theory and its Applications
NOLTA'09, Sapporo, Japan, October 18-21, 2009

A Method for Navigating Mobile Agent toward Target Location without Gradients in Sensor Field

Keisuke Sei, Akira Takahashi and Takaya Miyano

Graduate School of Science and Engineering, Ritsumeikan University
1-1-1 Noji-Higashi, Kusatsu, Shiga, 525-8577 Japan
Email: tmiyano@se.ritsumei.ac.jp

Abstract– We have devised algorithms for navigating a mobile agent whose mission is to find the location where the physical status of the environment takes the maximum using sensor data from wireless sensor systems sparsely deployed in the environment. Our algorithms allow determining the goal location without gradients in the sensor field. Their performance is assessed in terms of the probability and time for finding the goal by numerical experiments.

1. Introduction

Sensor networks are becoming important information infrastructures of the society in the post PC era. This technology enables us staying at home or in the office to monitor the physical status of the environment where wireless sensor systems are deployed and form a network for communications [1]-[5]. Sensor data are transmitted from one sensor node to another in a multipoint relaying manner toward a host computer as a sink node in the network. However, sensor networks have a critical bottleneck in that they consume much energy to establish and maintain communication routes between sensor nodes. This often prohibits real-world applications of sensor networks, because the electric power assigned to each sensor node is limited.

Motivated by recent epoch-making findings in complex networks science [6], we have proposed a hierarchical sensor network model toward fixing the problem of energy consumption due to frequent re-establishments of communication routes [7]. In our model, wireless sensors of multiple classes each of which is differentiated on the basis of hardware specifications such as power source and communication capability form a network with a particular topological structure reminiscent of scale-free networks. The resulting network has robustness to external random attacks in that faults of sensor nodes are the most likely to occur in terminal nodes having a single edge, achieving saving energy for re-establishing communication routes.

Although our hierarchical model is worth being further investigated, we have also studied an alternative approach to fixing the energy-consumption problem. In this approach, we use a mobile agent, for instance, a small flying object with a micro computer, a global positioning system and a wireless communication system, which aggregates sensor data including the location of each sensor node and brings them back to the sink node. Instead, the sensor nodes do not form a network, sending their data to the mobile agent coming closer to them. Mobile agent methods for sensor systems have been investigated in previous literature [3, 8, 9]. In [8] and [9], for instance, Dijkstra's algorithm [10] was applied to determining the optimal paths for a mobile agent to aggregate sensor data and bring them back to the sink, and for mobile sensor nodes that can move to new locations to improve the coverage of the sensor network, respectively.

In this paper, we show algorithms for navigating a mobile agent whose mission is to find using sensor data the location where the physical status of the environment takes the maximum as a sign of anomaly that indicates a disastrous event happening in the environment. Our algorithms aim at achieving the minimal time for finding the target location, because the amount of energy that the agent can use is limited. In this sense, the purpose of this study is different from those of the previous studies [8, 9], rather is closer to that of the previous work by Vergassola et al [11]. They investigated a mathematical model concerning how animals such as moths can search their partners with sporadically sensed odours emanated from the partners and carried by an erratic flow of air. In fact, our mobile agent tracks sporadic data from nearby sensor nodes to the target location, but it cannot rely on gradients in the sensor field, since the data available at a time can provide gradients of zero.

This paper is organized as follows. In Section 2, we describe our search algorithms without gradients. In Section 3, the performance of the algorithms is assessed with numerical experiments. Discussion and conclusion are given in Section 4.

2. Algorithms

The sensor field is defined as the environment mapped into a Euclidean space in which sensed information is put on each location of the sensor nodes randomly deployed in the environment. The sensor information represents the physical status within the detectable area of each sensor node. The detectable area is assumed to be a circle d_s in radius centered at each sensor node. The union of all

detectable areas should cover the whole environment. However, there may be cases in which uncovered areas are interspersed in the environment. In our model, an uncovered area is termed a hole.

The mobile agent aggregates sensor data from the sensor nodes within its communicable distance d_a. The mean over the acquired data is supposed to represent the status of the environment at the location of the agent. Under these assumptions, we design two different algorithms, as shown in the following subsections.

2.1. Moving Starting Location Method

An algorithm referred to as the moving starting location (MS) algorithm is constructed as follows (Fig.1).

(1) Set an initial starting location. At this location, the mobile agent aggregates sensor data to estimate the mean.
(2) Move the mobile agent by a distance of d_1 in sequence from the starting location to search locations in the north, northwest, west, southwest, south, southeast, east and northeast. At each search location, aggregate sensor data and estimate its status.
(3) Compare the status of the starting location to those of the search locations and find the best search location whose status takes the maximum. If there is no maximum, go to step (6). Otherwise, go to step (4).
(4) Move the agent randomly to a location on the quarter arc in radius d_2 centered at the starting location.
(5) Set the current location to a new starting location and do steps (1) to (3). If the current maximum is greater than the previous maximum, double d_2. Otherwise, reduce d_2 by half. Go to step (4).
(6) Move the agent randomly to a location on the full circle in radius d_2 centered at the starting location. Set the location to be a new starting location and do steps (1) to (3).
(7) The MS algorithm is terminated when d_2 becomes less that a critical distance d_c.

The final location is taken as the goal. As a variant of the MS algorithm, we may make the transition from the starting location to the best search location in step (4) a stochastic process. This can be performed with simulated annealing [12]. In this algorithm, the transition probability is defined as $p = \exp[-(e-e')/T]$ if $e > e'$ and as $p = 1$ otherwise, where the status at a starting point and a search location are e and e', respectively, and T is a temperature parameter.

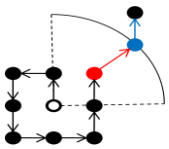

Figure 1: Schematic diagram of the MS model. The initial starting location is indicated by a white dot, its search locations by black dots and the best location by a red dot. The subsequent starting location is marked by a blue dot.

2.2. Contracting Search-Area Method

Another algorithm referred to as the contracting search-area (CA) algorithm is constructed as follows (Fig.2).

(1) Set an initial starting location, where sensor data are not to be aggregated by the mobile agent.
(2) Move the mobile agent by a large distance of d_3 in sequence from the starting location to search locations in the north, northwest, west, southwest, south, southeast, east and northeast. At each search location, aggregate sensor data and estimate its status.
(3) Compare the status of the search locations and find the best location whose status takes the maximum.
(4) Set the best location to a new starting location and reduce d_3 by half. Do steps (2) to (3).
(5) Iterate steps (2) to (4) by I_1 times. Then the algorithm is terminated.

The final location is taken as the goal. To circumvent the trapping of the agent into a local maximum, the CA algorithm may be iterated by I_2 times.

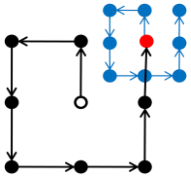

Figure 2: Schematic diagram of the CA model. The initial starting location is indicated by a white dot, its search locations by black dots and the best location by a red dot. The subsequent search locations are marked by blue dots.

3. Numerical Experiments

Numerical experiments were conducted to assess the performance of the algorithms in terms of the probability and dimensionless time for a mobile agent to reach the goal location whose status takes the global maximum. We first defined a dimensionless square area in which the physical status of the environment was assigned on each grid as numbers and N sensor nodes were randomly deployed. The size of the area was given as $X \times X$ with $X = 48$, 93, 138 and 183. The detectable area of each sensor node was set to $d_s = 1.2$. The (dimensionless) velocity of the mobile agent was assumed to be 1.2 in both x and y directions.

In the following, we show results only for $X = 183$, because other settings of X provided essentially similar results. The parameter settings were $d_a = 1.2$, $d_1 = 2$ and $d_1/10 \le d_2 \le X/4$ ($d_c = d_1/10$ and initially $d_2 = d_1$) in the MS algorithm, and $d_3 = X/4$ in the CA algorithm.

3.1. Optimal Number of Sensor Nodes

To determine an appropriate number of sensor nodes, the following experiments were conducted. We assigned a single extremum, i.e., a global maximum to the area, and conducted 200 trials of searching the maximum using the

MS without simulated annealing and the CA algorithm with $I_1 = 4$ to 6 and $I_2 = 0$ as a function of the number of sensors increasing by 50. The density of sensor nodes per unit area increased from 0 to 1.5. Typical trajectories of the agent during searching are shown in Fig.3, where the global maximum is near the left-bottom corner.

Although the CA algorithm required more sensor nodes than the MA algorithm, a density of sensor nodes per unit area of 1.2 was found to be sufficient for the agent to almost surely reach the goal regardless of the search algorithms.

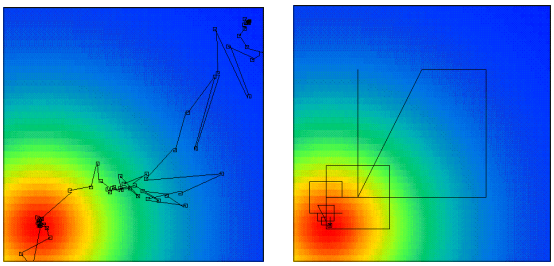

Figure 3: Trajectories of a mobile agent ($X = 183$). Left panel: MS; Right panel: CA. Color spectrum represents the magnitude of the physical status of the environment in increasing order from blue to red.

3.2. Probability and Time for Reaching the Goal

We next conducted 200 trials of searching to assess the probability and time for the agent to reach the goal. In these experiments, the number of sensor nodes was set to a node density of 1.2. The performance of the algorithms was assessed for three cases of extrema assigned to the area: a single extremum and multiple extrema. As a benchmark, we used the entire search over the whole area to aggregate all sensor data. Our algorithms should achieve reaching the goal in a shorter time with a practically high probability than the benchmark. Otherwise, they are not useful.

Figures 4 to 7 show typical trajectories of the agent during searching. The performance of the algorithms is summarized in Tables 1 to 3.

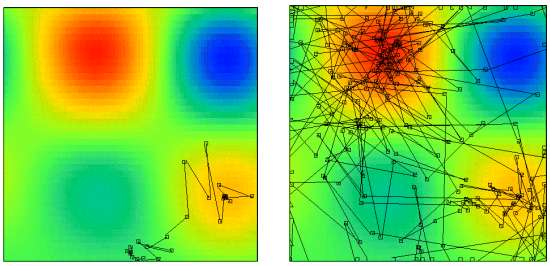

Figure 4: Trajectories of a mobile agent ($X = 183$). Left panel: MS; Right panel: MS with simulated annealing. The global maximum is near the left-top corner.

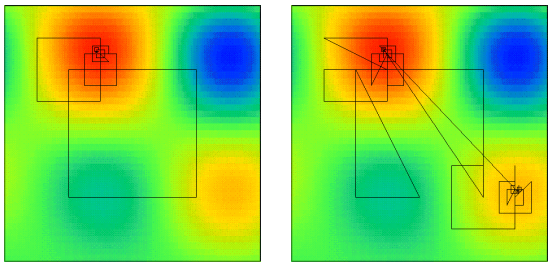

Figure 5: Trajectories of a mobile agent ($X = 183$). Left panel: CA with $I_1 = 6$ and $I_2 = 0$; Right panel: CA with $I_1 = 6$ and $I_2 = 2$. The global maximum is near the left-top corner.

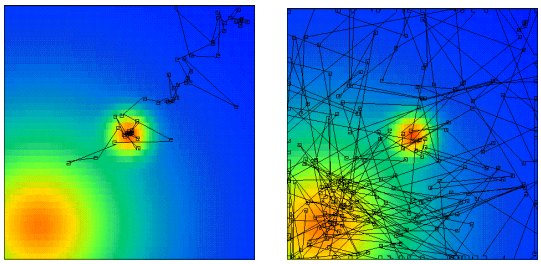

Figure 6: Trajectories of a mobile agent ($X = 183$). Left panel: MS; Right panel: MS with simulated annealing. The global maximum is in the center.

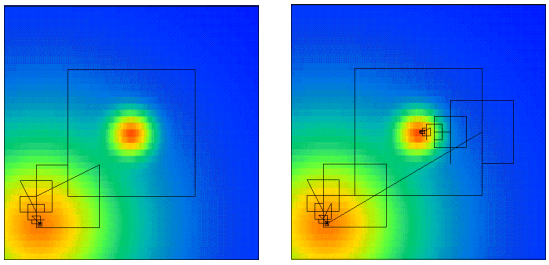

Figure 7: Trajectories of a mobile agent ($X = 183$). Left panel: CA with $I_1 = 6$ and $I_2 = 0$; Right panel: CA with $I_1 = 6$ and $I_2 = 2$. The global maximum is in the center.

Table 1: Average performance of the MS and CA algorithms over 200 trials for a single maximum in the area (Fig.3). CA(0) expresses $I_2 = 0$ in the CA algorithm.

		48×48	93×93	138×138	183×183
Probability	Benchmark	1	1	1	1
	MS	1	1	1	0.9250
	CA(0)	1	0.9950	0.9100	0.9750
Time	Benchmark	525.33	2013.33	4444.67	7826.00
	MS	215.99	371.58	450.01	585.13
	CA(0)	117.02	231.05	345.88	458.19

Table 2: Average performance of the MS and CA(I_2) algorithms over 200 trials for multiple extrema in the area (Figs.4 and 5). MS(SA) stands for MS associated with simulated annealing.

		48×48	93×93	138×138	183×183
Probability	Bechmark	1	1	1	1
	MS	0.4700	0.3800	0.3750	0.2650
	MS(SA)	0.8200	0.8000	0.8300	0.7800
	CA(0)	1	1	0.9850	1
	CA(1)	1	1	1	1
Time	Benchmark	525.33	2013.33	4444.67	7826.00
	MS	142.18	232.44	304.38	328.20
	MS(SA)	501.96	1364.25	4062.97	8314.16
	CA(0)	117.79	230.78	346.13	459.71
	CA(1)	194.38	386.25	575.09	769.42

Table 3: Average performance of the MS and CA(I_2) algorithms over 200 trials for multiple extrema in the area (Figs.6 and 7). MS(SA) stands for MS associated with simulated annealing.

		48×48	93×93	138×138	183×183
Probability	Benchmark	1	1	1	1
	MS	0.2900	0.2850	0.3000	0.2600
	MS(SA)	0.4650	0.4600	0.7600	0.6900
	CA(0)	0.0050	0	0	0
	CA(1)	0.7350	0.9900	0.8750	0.9700
Time	Benchmark	525.33	2013.33	4444.67	7826.00
	MS	159.72	294.72	437.28	544.96
	MS(SA)	558.11	1630.64	4453.05	10180.75
	CA(0)	122.50	none	none	none
	CA(1)	195.08	387.84	578.67	768.97

4. Discussion and Conclusion

The numerical results suggest that a node density of 1.2 per unit (dimensionless) area is sufficient to circumvent the influence of holes, which should be improved with increasing detectable area of a sensor node. The MS algorithm seems to need less sensor nodes than the CA algorithm. A mobile agent subject to the MA method tends to read more sensor nodes than the CA method because of the initial increment of d_2 followed by its reduction toward the goal location. This makes the search process less sensitive to holes in that there are more opportunities for the agent to complement the deficiencies of sensor data.

When the physical status of the environment has a single maximum, the MS method can find the goal location more surely than the CA method, despite taking a longer time, which may be expected from the fact that the agent subject to the MS method is likely to read more sensor nodes. However, when there are multiple local extrema in the environment, the MS method is more likely to incur the trapping of a mobile agent into local maxima. Simulated annealing does not improve the MS method, as shown in Tables 2 and 3. In fact, simulated annealing brings an outrageous increase in time to find the goal. It takes longer time than the benchmark. This could be due to idiosyncrasy in the distribution of the sensor data. For the present, it is unclear why simulated annealing did not work in our experiments.

In actual applications, there are often cases where the physical status of the environment has multiple extrema. Accordingly, the CA method with $I_2 \geq 1$ is better than the CS method for operating a mobile agent. It achieves a practically high probability and shorter time, as small by 1/10 as the time required by the benchmark method.

In conclusion, we have devised two algorithms without gradients for navigating a mobile agent toward the goal location whose physical status takes the maximum. The CA algorithm seems to be better for practical use. Other methods without gradients, for instance, the association of Bayesian inference with the MS and CA methods or using a class of zigzag movement of the agent, may be possible. These will be deferred to future studies.

References

[1] J. M. Kahn, R. H. Katz and K. S. J. Pister, "Next Century Challenges: Mobile Networking for Smart Dust," *Proc. 5th Ann. ACM/IEEE Int. Conf. on Mobile Computing and Networking*, pp.271–278, 1999.

[2] J. Elson and D. Estrin, "Time Synchronization for Wireless Sensor Networks," *Proc. 2001 Int. Parallel and Distributed Processing Symp*., pp.1965–1970, 2001.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "A Survey on Sensor Networks," *IEEE Commun. Mag*., vol.40, issue 8, pp.102–114, 2002.

[4] B. Thuraisingham, "Secure Sensor Information Management and Mining," *IEEE Signal Processing Mag*., vol.21, no.3, pp.14–19, 2004.

[5] D. Butler, "Everything, Everywhere," *Nature*, vol.440, pp.402–405, 2006.

[6] R. Albert and A. -L. Barabasi, "Statistical mechanics of complex networks," *Rev. Mod. Phys*., vol.74, pp.47–97, 2002.

[7] K. Nemoto, K. Sei, T. Toriyama and T. Miyano, "A Method for Networking Hierarchical Sensor Systems for Energy-Saving Robust Sensor Networks," *IEEJ Trans. Sensors and Micromachines*, to appear in vol.129, no.11, 2009.

[8] X. Wu, Y. Niu, L. Shu and J. Cho, "Relay Shift Based Self-Deployment for Mobility Limited Sensor Networks," *Lect. Notes in Comp. Sci*., vol.4159, pp.556–564, 2006.

[9] E. Shakshuki, X. Xing and H. Malik, "Mobile Agent for Efficient Routing Among Source Nodes in Wireless Sensor Networks," *Proc. 3rd Int. Conf. Autonomic and Autonomous Syst*., Issue 19-25, p.39, 2007.

[10] E. W. Dijkstra, W. H. J. Feijen and J. Sterringa, *A Method of Programming*, Addison-Wesley, 1988.

[11] M. Vergassola, E. Villermaux and B. I. Shraiman, "'Infotaxis' as a strategy for searching without gradients," *Nature*, vol.445, pp. 406–409, 2007.

[12] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol.220, pp.671-680, 1983.