University of Liverpool

Department of Computer Science

Second Year Report

Control Problems for Mobile Agents

Author:
Thomas Gorry

Supervisors:
Russell Martin
Leszek Gąsieniec

May 31, 2013

Contents

1 Introduction

1.1 Area of Work

The broad area that this PhD project is aimed at is Algorithms for Distributed Computing. More specifically however the project aims to carry out research on the performance analysis of mobile agents in a variety of settings. These settings could range from different types of networks to different geometric settings. Furthermore in each of these settings different models could be adopted that open this research up to an unlimited number of possibilities regarding both the differences in the agents themselves and the environment in which they are deployed. Examples of some variations regarding the agents themselves could be things like using different maximum speeds, limited memory, ability to leave markers or messages, variations to communication limitations or methods, etc... Whereas some examples of the variations regarding the environment would be more like diversification of topologies in network settings as well as differences in the geometric domains used. These are just a few examples and as mentioned before there are an endless number of combinations that could be studied. Furthermore, the algorithms themselves that we aim to develop could make use of both deterministic and randomised approaches to these types of problems

1.2 Background and Contributions

There has been a lot of interest in the area of Mobile Agents over the years, with work being done on a variety of control problems for groups of robots [1, 2, 3, 4]. Many have focused on tasks such as gathering and rendezvous [2, 3, 4] as well as topology discovery in geometric settings [7, 8, 9, 10] and networks [11, 12, 13, 14].

The desired outcome of this PhD is to augment this field by studying similar situations but with the added challenges of limited communication and access to a global picture of the environment as well as the, previously rarely studied, aspect of multiple different top speeds of the agents in a multitude of environments and models. Already our work had gained interest from the academic community with one of our papers published last year [15] being cited in a recent publication [16].

We also aim to see if any contributions can be made to this area by integrating knowledge from graph and network theory in some way.

2 Second Year

2.1 Location Discovery

2.1.1 Description of the work

Throughout the year we have continued work from our paper on mobile agents located on the circumference of a circle and with very limited communication [15] with the hope of completing a journal version of the paper to submit before the end of the year.

2.1.2 Improved Result

For the journal version of our paper we made improvements to the time complexity of the algorithm by utilizing the unlimited memory that the model allowed. We reduced the time complexity from $O(n \log^2 n)$ rounds *w.h.p.* to $n+O(\log^2 n)$ rounds *w.h.p.* , clearly a large saving.

To obtain this improvement we altered the strategy slightly and also made a simple adjustment to the model. Where as in the past an agent would be able to chose their initial starting direction from a choice of *Clockwise* and $Anti-Clockwise$ they now have the added option of *Stationary* this choice would mean that an agent chooses to remain still initially rather than immediately move off in one of the two directions. This adjustment helped us with the cases when n is even as it allowed us to employ our new strategy, effectively acting in every case as if n was odd, regardless of the parity of the system.

The new strategy involved reducing the number of expended *Stages* and subsequently *Rounds* that the agent moved that did not yield some benefit. The method of doing this is quite simple in that an agent would simply remember the moves it had previously made, i.e the starting direction it had adopted during each round and where it ended up at the end of that round. By doing this an agent is able to build up a mapping it could refer to every time it ended a round on a previously undiscovered *HomeBase* of another agent. Using this mapping an agent can follow a recursive algorithm to quickly discover all of the *HomeBases* in the system while minimizing the number of low yielding rounds. A small example of this, and the pseudo-code for the algorithm can be found in Appendix A at the back of this document.

2.2 Evacuation Problem

2.2.1 Background

The evacuation problem is something that we have been looking at over the past year with many different variations of this problem under consideration. We are close to finishing a paper [17] about this problem and hope to have it submitted to a conference by the end of the year.

The evacuation problem is that of a group search problem where a group of k agents that are located on a line perform a search for a specific destination. These agents initially share a common starting point(*origin*) on the line and the target destination is located at an unknown distance d either to the left or right of the *origin*. The task of the agents is for all of them to locate and travel to this point in the shortest time possible.

The problem with k agents is known as the Cow-Path Problem and is known to have the complexity of $9d$ in the worst case [18]. It is also known that this is the case for $k > 1$ agents. This problem has since been considered in the same form and different variations by [18, 19, 20, 21, 22, 23, 24]

2.2.2 Improvements

Our contribution to the field is building upon the previous work of the papers mentioned above is to present a clearer way of proving the bound of $9d$. We concentrate in [17] on the model where communication is limited to collisions between agents only. We also to dispute claims made by [18] that if the number of agents is greater than two then the goal can be reached in a smaller time. We have included a proof showing that $9d$ is also optimal for these cases.

2.2.3 Multiple-Speeds

A further addition to our contribution was for us to study the cases where there are two agents but they have different maximum speeds. We wanted to see what effect this would have on the worst case and the magnitude of such an effect given the slowest maximum speed.

We found that when the maximum speed of the slower agent is $\geq \frac{1}{3}$ of the maximum speed of the faster agent then a bound of $9d$ can still be achieved.

A brief sketch of the proof for this can be found in Appendix B at the back of this document.

2.3 Social Network Analysis

Over the last year we have been working with an industrial partner on different social network analysis problems. Two of our main activities with this partner are listed below.

2.3.1 Reach and Frequency

This project is an ongoing project to produce reach and frequency reports based on an initial set of users on Twitter. Once we have this data we use the Twitter API to obtain data of the users' followers. After this we then produce reports showing how much exposure that a Tweet from these people has received. This information is processed and presented as a detailed breakdown of how many people the Tweet was seen by and also how many times they saw this Tweet, as each Tweet can be seen multiple times in a users feed if more than one person they are following Tweets the message.

2.3.2 Time-lapse Videos

From time to time our industrial partner also tasks us with producing a time-lapse video of different topics topics, usually add campaigns. This involves taking keywords about an event or topic and mining Tweets over a given period of time. Once we have obtained enough data we can use software that we have produced to visualise the underlying network of ReTweets, where a Tweet is a vertex and a ReTweet forms an edge between the two. We can then display this network by either using a force directed algorithm we have implemented or by using the geometry data we get from the Twitter API to map each vertex to their respective geographical locations. The time-lapse video is produced using this same software, where each frame is a still image of the network at different time intervals. We then stitch these frames together to produce the videos.

3 Work to be Completed

3.1 Evacuation Problem

We will be completing and submitting our paper on this problem in the coming year. I will also be carrying out further study on the Evacuation Problem described in Section 2.2. One of the avenues of research in this area include extending the environment from a line to a ring. In this scenario the agents would initially be located at a point of origin on the circumference of the ring and must locate a destination at some unknown distance away that is also located on the circumference of the ring.

Another avenue that this work could take is using the environment of the disk. In this environment we could look at the cases when agents initial starting location is at the centre of the disk and the destination they are searching for in located somewhere along the circumference of the disk. Another variation on this could be that instead of starting at the centre of the disk the agents in fact all start at the same random location within the disk and are unaware of how far away from the edge of this disk they are. This variation would mean that not only do the agents need to coordinate to locate and reach the destination point but they must also coordinate to locate the circumference of the disk as well.

There are many other variations on these two avenues that could be employed as well as a variety of other avenues that could be investigated.

3.2 Biological Network Analysis

As of 1st May 2013 I have been involved in a joint six month project with the Bioscience department. This project is a NERC grant for Efficient Biological Network Discovery and Analysis. The grant covers the analysis of functional genomics datasets of environmental influences. This will be accomplished by adding extra functionality to our current network analysis software. The extra functionality that will be developed over the coming months will include the following:

1. Adapt current software to enable loading/saving of some of the more common file formats used by the biological community.

2. Providing the functionality of importing data direct from several on-line databases to augment the biological network that has already been loaded into the software.

3. Research different types of modularity and community clustering techniques. Then implement the technique that fits best with our situation. The aim here is to produce code that will be able to create clusters iteratively until a user defined threshold is met, thereby producing accurate clusters for the type of data being processed.

4. Improve the speed of our current implementation of a force positioning layout.

Following the six month placement there will be a workshop in Cardiff that is also funded by this grant. The workshop will include a formal demonstration of the completed software that will include an analysis of complex networks and the prediction of biological outcomes from network models.

3.3 Social Network Analysis

Our industrial partnership looks set to continue in the coming year. The partner has expressed interest in implementing our current software as a service on the Internet. Also we will be developing a crawler that will mine websites that we obtain through our Twitter collection service.

We will also be devoting more of our time on new research areas in social network analysis. One of these areas is bot detection in networks. While analysing twitter networks we have noticed that visually it is easy to detect the pretence of bots in these networks. We believe that there should be an easy way to computationally locate these bot sub networks, therefore this will be one area we will be looking into over the next year.

4 Thesis Time Table

I will be using the three paper model for the completion of my thesis. This model is shown below. The exact structure and timings of this plan may change slightly over the next year, for example it may be the case that there will be a fourth paper to include as well. Also I wIll be handing draft sections of my thesis to my supervisors as they are completed for review, therefore it may be the case that certain items below will end up taking longer than specified here.

1. Introduction and outline of the problem. - 3 weeks
2. Literature review of the subject area. - 3 weeks
3. First Paper [15] - 4 weeks
4. Second paper [17] - 4 weeks
5. Third paper [unwritten] - 5 weeks
6. Discussion of work on network analysis - 2 weeks
7. Conclusion and implications for further research. - 2 weeks

A Location Discovery

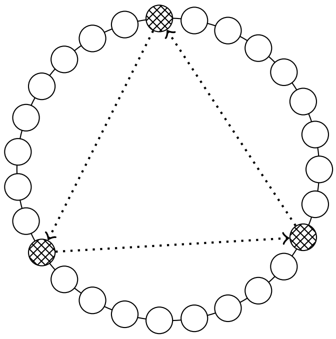
procedure FasterDiscovery(n: integer)
Integer i;
List $Points$, $Directions$, $History$, $VectorSet$;
$\{C, A\}$ dir;
$\mathbb{R}$ $Location$, loc, $NewLocation$;
start
(1) $Points \leftarrow \emptyset$
(2) $Directions \leftarrow \{\}$
(3) $History \leftarrow \{\}$
(5) pick direction dir from $\{C, A\}$ uniformly in random and append dir to $Directions$;
(2) **repeat**
(3)   append $location$ to $Points$;
(4) $move$ for 1 round to $newLocation$;
(5) $location \leftarrow newLocation$;
(6)   append $vector$ to $VectorSet$;
(7) **until** ($location = 0$)
(7) remove last element from VectorSet
(8) append $VectorSet$ to $History$;
(9) append dir to $Directions$;
(4) **while**($|points| < n$)**do**
(5)   pick direction dir from $\{C, A\}$ uniformly in random
(6) $location \leftarrow 0$;
(7) $VectorSet \leftarrow$ IncreaseGranularity(location, dir, Directions, History);
(7)   append $VectorSet$ to $History$;
(8) **end while**
(9) **return** ($Points$);
end.

Figure 1: The improved location-discovery procedure of an agent.

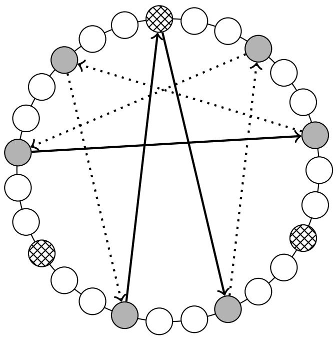
procedure INCREASEGRANULARITY($location$: $\mathbb{R}$, dir: $\{C, A\}$, $Directions$: list, $History$: list)
List $VectorSet$
start
repeat

    $move$ for 1 round to $newLocation$ and set $location \leftarrow newLocation$;
    if ($location \notin Points$) **then**
        append $vector$ to $VectorSet$;
        append dir to $Directions$;
        append $location$ to $Points$;
        $i \leftarrow 0$;
        repeat
            $dir \leftarrow Directions[i]$;
            IncreaseGranularity(location, dir, Directions, History);
            $i{+}{+}$;
        until ($i > |Directions| \; || \; |points| = n$)
    else continue
until ($location \in Points$)
return $VectorSet$
end.

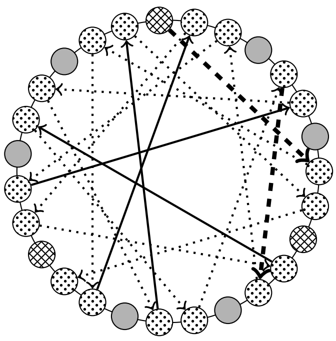
Figure 2: The improved location-discovery procedure of an agent.



Stage 1
$n_c - n_a = 18$
$gcd(27, 18) = 9$
Vector $= V_1$
Vector Set $= V_1,\ V_1,\ V_1$
History saved $= V_1,\ V_1$



Stage 2
$n_c - n_a = 12$
$gcd(27, 18, 12) = 3$
Vector $= V_2$
Vector Set $= (V_2,\ V_1,\ V_1,\ V_2,\ V_1,\ V_1,\ V_2)$
History saved $= (V_2,\ (V_1,\ V_1)\ V_2,\ (V_1,\ V_1))$

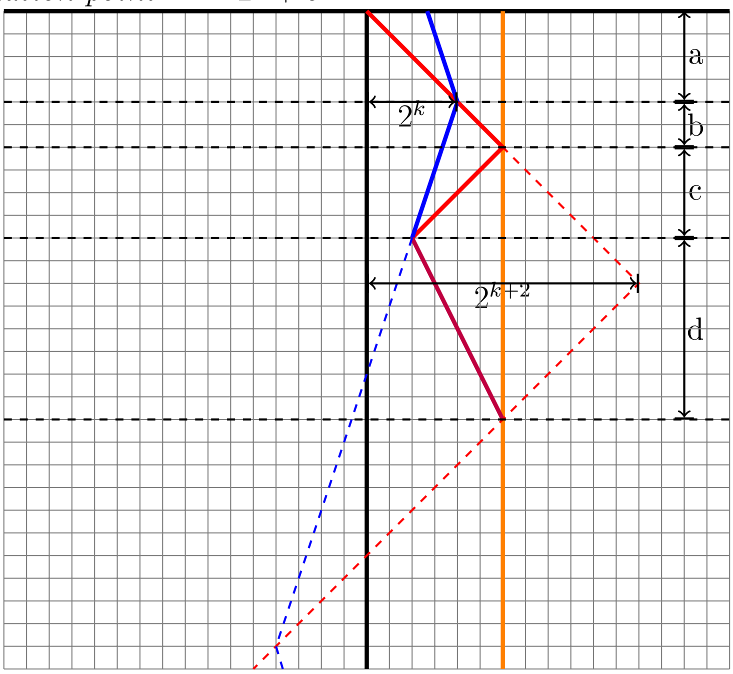


Stage 3
$n_c - n_a = 7$
$gcd(27, 18, 12, 7) = 1$
Vector $= V_3$
Vector Set $= (V_3,\ V_1,\ V_1,\ V_2,\ V_1,\ V_1,\ V_2,\ V_1,\ V_1, V_3,\ V_1,\ V_1,\ V_2,\ V_1,\ V_1,\ V_2,\ V_1,\ V_1)$
History saved $= (V_3,\ ((V_1,\ V_1),\ (V_2,\ (V_1,\ V_1),\ V_2,\ (V_1,\ V_1)), V_3,\ ((V_1,\ V_1),\ (V_2,\ (V_1,\ V_1),\ V_2,\ (V_1,\ V_1))))$

Figure 3: A simple example of the algorithm at work for when $n = 27$.

B Evacuation Problem

Figure 4: Example of $9d$ for *evacuation problem*, where slower agent has $s \geq \frac{1}{3}$ and the *evacuation point* is at $2^k + \epsilon$



Mathematical proof for time of $9d$ for the faster agent and the slower agent with speeds of 1 and $\frac{1}{3}$ respectivly:

Definitions:

$D =$ distacnce from origin to evacuation point

$S =$ Speed of slow agent

$2^{k-2} < D \leq 2^k$

$D = 2^{k-2} + \epsilon$

The time taken for faster agent to find D ($a + b$ in Figure 4):

$$(2 \times \sum_{i=0}^{k-1} 2^i) + D \tag{1}$$

The time taken for the faster agent to inform the slower (c in Figure 4):

$2e$

The time taken for faster agent and slower agent to get back to the *evacuation point* (d in Figure 4):

$\frac{2e}{S}$

We assume that we use the speed of $\frac{1}{3}$ for the slower agent, in this case:

$$\begin{aligned}
&2(2^k - 1) + D + 2\epsilon + 6\epsilon \\
&= 2(4 \times 2^{k-2} - 1) + D + 2\epsilon + 6\epsilon \\
&= 2\left(4(2^{k-2} + \epsilon) - 4\epsilon - 1\right) + D + 2\epsilon + 6\epsilon \\
&= 8D - 8\epsilon - 2 + D + 8\epsilon \\
&= 9D - 2
\end{aligned} \tag{2}$$

$$\frac{9D - 2}{D} = 9$$

References

[1] H. Ando, I. Suzuki, and M. Yamashita, Formation and agreement problems for synchronous mobile robots with limited visibility, *Proc. IEEE Symposium on Intelligent Control* 1995, pp. 453–460.

[2] M. Cieliebak, P. Flocchini, G. Prencipe, and N. Santoro, Solving the robots gathering problem, *ICALP 2003*, pp. 1181–1196.

[3] R. Klasing, E. Markou, and A. Pelc, Gathering asynchronous oblivious mobile robots in a ring, *Theoretical Computer Science* **390**, pp. 27–39, 2008.

[4] K. Sugihara and I. Suzuki, Distributed algorithms for formation of geometric patterns with many mobile robots, *J. Robotic Systems* **13(3)**, pp. 127–139, 1996.

[5] P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer, Hard tasks for weak robots: The role of common knowledge in pattern formation by autonomous mobile robots, *ISAAC 1999*, pp. 93–102.

[6] P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer, Pattern formation by autonomous robots without chirality, *SIROCCO 2001*, pp. 147–162.

[7] X. Deng, T. Kameda, and C. Papadimitriou, How to learn an unknown environment I: the rectilinear case, *Journal of ACM*, **45(2)**, pp. 215–245, 1998.

[8] J. Czyzowicz, A. Labourel, and A. Pelc, Optimality and competitiveness of exploring polygons by mobile robots, *Information and Computation*, **209(1)**, pp. 74–88, 2011.

[9] F. Hoffmann, C. Icking, R. Klein, and K. Kriegel, The polygon exploration problem, *SIAM J. Computing* **31(2)**, pp. 577–600, 2001.

[10] C.S. Kong, N.A. Peng, and I. Rekleitis, Distributed coverage with multi-robot systems, *Proc. Robotics and Automation 2006*, pp. 2423–2429.

[11] P. Fraigniaud, L. Gąsieniec, D.R. Kowalski, and A. Pelc, Collective tree exploration, *Networks* **48(3)**, pp. 166–177, 2006.

[12] C. Cooper, A.M. Frieze, T. Radzik, Multiple Random Walks and Interacting Particle Systems, *ICALP 2009*, pp. 399–410.

[13] J. Chalopin, P. Flocchini, B. Mans, and N. Santoro, Network exploration by silent and oblivious robots, *WG 2010*, pp. 208–219.

[14] M.A. Bender and D. Slonim, The power of team exploration: Two robots can learn unlabeled directed graphs, *FOCS 1994*, pp. 75–85.

[15] T. Friedetzky, L. Gąsieniec, T. Gorry and R .Martin, Observe and remain silent (communication-less agent location discovery), *MFCS 2012*, pp. 407–418.

[16] J. Czyzowicz, L. Gąsieniec, A. Kosowski, E. Kranakis, O. Ponce and E. Pacheco, Position discovery for a system of bouncing robots, *DISC 2012*, pp. 314–355.

[17] S. Das, L. Gąsieniec, T. Gorry, R. Martin, M. Shalom, S. Zaks, Group Search on the Line - A tale of 9d, *Unpublished, In progress*

[18] R A. Baeza-Yates and R. Schott, Parallel searching in the plane, *Computational Geometric Theory and Applications*, vol. 5, no. 3, pp. 143–154, 1995.

[19] R A. Baeza-Yates , J C. Culberson , and G J.E. Rawlins, Searching in the plane, *Information and Computation*, vol. 106, no. 2, pp. 234–252, 1991.

[20] S K. Ghosh and R. Klein, Online algorithms for searching and exploration in the plane, *Computer Science Review*, vol. 4 no. 4 pp. 189–201, 2010.

[21] A. Jeż and J. Łopuzański, On the two-dimensional cow search problem, *Information Processing Letters*, vol 131, no. 11, pp. 543–547, 2009.

[22] M Y. Kao, J H Reif, and S R. Tate, Searching in an Unknown Environment: An Optimal Randomized Algorithm for the Cow-Path Problem, *Information and Computation*, vol 109, no. 1, pp. 63–79, 1996.

[23] H. Li and K P. Chong, Search on lines and graphs, *Proc. 48th IEEE Conference on Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009.*, vol 109, no. 11, pp. 5780–5785, 2009.

[24] T. Temple and E. Frazzoli, Whittle-indexability of the Cow Path Problem, *American Control Conference (ACC), 2010*, pp. 4152–4158, 2010.