

# Collective Intelligence and Bush Fire Spotting

David Howden  
davidhowden@hotmail.com

Tim Hendtlass  
thendtlass@swin.edu.au

Centre for Information Technology Research  
Swinburne University of Technology  
Melbourne, Australia 3122

## ABSTRACT

Bush fires cause major damage each year in many areas of the world and the earlier that they can be detected the easier it is to minimize this damage. This paper describes a collective intelligence algorithm that performs localized rather than centralized control of a number of unmanned aerial vehicles (UAV) that can survey complex areas for fires, devoting attention in proportion to the user specified importance of each area. Simulation shows that not only is the algorithm able to perform this action successfully, it is also able to automatically adapt to a simulated malfunction in one of the UAVs.

## Categories and Subject Descriptors

I.2.8 [Problem Solving, Control Methods and Search]:  
collective intelligence

## General Terms

Algorithms, Experimentation

## Keywords

Algorithms, scheduling, collective intelligence, collaborative search, unmanned autonomous vehicles

## 1. INTRODUCTION

Bush fires are a persistent problem in Australia (as well as other countries) where hot dry summers leave vegetation tinder dry. Lightning strikes can then set fire to this vegetation with the resultant fires spreading and advancing at tens of kilometers an hour, destroying all in their path. The earlier a fire is spotted the smaller it tends to be and the easier and more effective fire fighting efforts are. Fires that are in remote areas and inaccessible gullies pose special problems as, by the time they are first noticed, they may already be large. While manned fixed wing aircraft have been used

as spotters for many years, this situation is obviously suitable for surveillance by a number of small unmanned aerial vehicles (UAV).

Previous research conducted into the problem of multiple UAV agents searching a finite problem space varies from this paper in two important regards [1, 2, 3, 4]. Firstly a high degree of direct communication is assumed in other papers, with agents discussing and agreeing upon search plans and also usually the assignment of duties to individuals. This differs greatly from the approach discussed in this paper in that agents have no direct communication with each other, and essentially act individually based on the most recent information received. This method, utilising only stigmergy, is much more robust in regards to communication failures and loss of individual agents. Secondly, previous work has focused on locating and destroying discreet individual targets. This is very different from the problem of monitoring an area to survey the spread of a bush fire. With the surveying problem, the entire map is a potential 'target', and areas need to be re-searched at intervals. With seek and destroy problems a map has a set number of targets, and once they have been found the problem is considered solved. In this work the UAVs, while they may be in regular contact with a base in order to report any fire that is found, are not under central control but are autonomous in so far as determining their path over the area to be surveyed. They do, however, periodically share information so that collective intelligence results in an overall performance that is better than the sum of the individual performances without information exchange.

The algorithm takes inspiration from both particle swarm optimisation and ant colony optimisation. In particle swarm optimisation, originally proposed by [5], a particle moves through problem space under the influence of several factors, commonly an attraction towards the best location yet found by the swarm and also an attraction to the best location yet found by this particle. In addition they are presumed to have momentum so that they cannot change direction instantaneously. In this work the UAVs are attracted in the general direction of the cell closest to the current position most due for survey (so far as this UAV is concerned) together with an attraction to a second point found by a local search as described below. Like conventional particle swarm they also have momentum and additionally have a repulsion effect on each other if they get too close [7]. In ant colony optimisation [6] a solution is constructed incrementally through the making of a series of choices, with the intensity of artificial pheromone being used to indicate the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '08 July12–16, 2008, Atlanta, Georgia, USA.

Copyright 2008 ACM 978-1-60558-130-9/08/07...5.00.

perceived merit of a particular choice. After the completion of a solution more pheromone is laid on the choices made, the actual amount added being proportional to the quality of the solution. Pheromone decays over time so that old information that is no longer beneficial is slowly lost (pheromone corresponding to information that is still useful will tend to be preserved by having more pheromone laid down on it than is lost to evaporation). In this algorithm, pheromone is used to mark the need for a survey of a particular point, the higher the pheromone level the greater the need. Pheromone is increased automatically each time interval (by an amount that is proportional to the required survey frequency) and is reset to zero when a survey is made.

Each UAV keeps its own internal map, resetting the pheromone on each point it visits. When two UAVs are within a certain distance (called the communication range), they each receive information broadcast by the other which allows each to set the pheromone level of all cells on their pheromone map to the lowest of the values on their individual maps. In practice this would be achieved by each UAV continually broadcasting its map values. UAVs that come too close are repulsed from each other which, together with the shared information, prevents more than one UAV surveying the same point at the same time (or very shortly after each other).

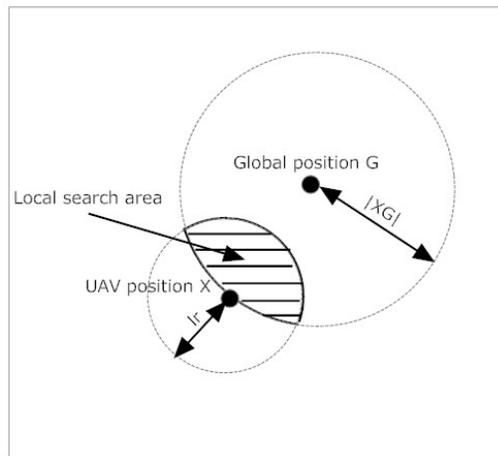
The net effect of this is that the UAVs spread out rather than converge, surveying locations with the highest pheromone levels preferentially so that all areas on the map are surveyed as frequently as possible and in the required frequency ratio.

## 2. THE ALGORITHM IN DETAIL

Each UAV maintains its own map that shows the region to be surveyed divided into cells ( $C_{ij}$ ), each cell having its own user supplied constant pheromone update factor  $\Delta P_{ij}$ . Ideally each cell will be surveyed exactly when the pheromone level reaches one, so that the required survey frequency is  $P_{ij}^{-1}$ . Each UAV also knows its current velocity vector  $\overline{V}_{current}$  and its net repulsion vector  $\overline{R}$ . This vector  $\overline{R}$  results from the interaction between two (or more) UAVs that come too close to each other. If two UAVs come closer to each other than a user chosen small number of cells, each experiences a repulsion force directed away from the other UAV and with a magnitude of  $r = \frac{rnd*40}{d}$  where  $d$  is the UAV separation in cells and  $rnd$  is a random number [0..1]. Each iteration  $\overline{R}$  is attenuated so that any repulsion effect decreases with time before the new repulsion forces  $\overline{r}$  (if any) are added to  $\overline{R}$  so that the effect lasts for more than one iteration.

At each iteration the following actions take place for each UAV.

- The internal pheromone map values are increased, each cell being increased by the  $\Delta P_{ij}$  corresponding to that position.
- Each UAVs repulsion vector  $\overline{R}$  is decreased (in this work by 20%). The repulsion vector  $\overline{r}$  from any UAVs within 5 cells are calculated and added to  $\overline{R}$ .
- The internal maps of all UAVs that are with communication range are compared with the internal map of this UAV, with the pheromone value at some point  $C_{ij}$



**Figure 1: The local search area for a UAV at position  $X$  when the cell with the highest value of  $T_{ij}$  is at  $G$**

being set to the lowest of the pheromone values for this cell on any of the UAV maps.

- The quantity  $T_{ij} = \frac{P_{ij}^2}{1+D_{ij}}$  in which  $D_{ij}$  is the distance of cell  $C_{ij}$  from the current position  $X$  of the UAV and  $P_{ij}$  is the pheromone of that cell) is calculated for each cell. The cell with the highest value of  $T_{ij}$  is identified ( $G$ ) and the distance from the current position to that cell ( $XG$ ) noted.
- $\overline{XG}$  sets the general direction in which the UAV seeks to go, however a local search is then performed of all cells closer to  $X$  than  $lr$  (the local search radius) and closer to  $G$  than  $XG$  for the cell  $L_{ij}$  with the highest value of  $T_{ij}$  that meets these conditions. Effectively this can only come from the shaded area in figure 1. The vector component  $\overline{S}$  (in equation 1) is set to  $\overline{XL}$ . The net effect of this local search is that the UAV, while still heading in the general direction of  $G$  does so along a path that as best as possible addresses the survey needs of points between these end points.
- The new movement vector of the UAV is calculated from:

$$\overline{v}_{new} = rnd * \overline{v}_{current} + \frac{\overline{S} + \overline{R}}{t} \quad (1)$$

where:

- $rnd$  is a uniform random number in the range from zero to one, and
- $t$  is the time between updates (required for dimensional integrity but taken to be unity as the time scale is arbitrary).
- Finally the pheromone values for all cells currently being surveyed by the UAV are set to zero. For the work reported here each UAV was assumed to survey a 3 by 3 grid of cells at any one time.

## 3. METHODOLOGY

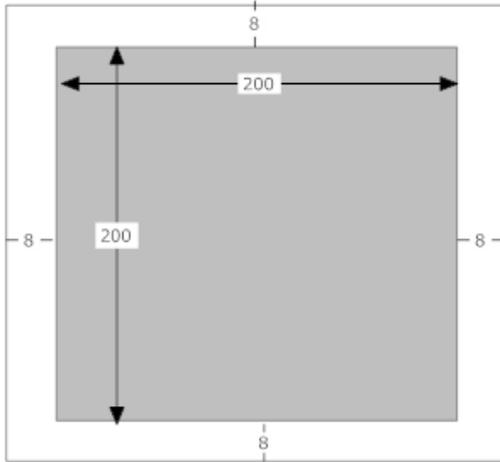
Experiments have been conducted on a range of surveillance maps; results presented in this paper are from the

simplest and the most complex of these maps. Each map has a number of cells at the boundary that do not need to be surveyed; some have similar regions inside the map. In all cases UAVs are allowed to cross these cells if they need to while turning (boundary cells) or while on their way to other reportable cells (internal regions). Surveying earlier than required is considered acceptable but failure to survey in time is considered an error.

Initially the UAVs were assigned random locations and zero velocities with each cell of the pheromone map being set to zero and  $R$  also being zero. Experimentation showed that the magnitude of the repulsion force was not critical for any of these maps and so the formula described in section 2 was used when generating all the results presented here. UAVs also have a fixed speed that was set to 3 cells per iteration (consistent with the  $3 \times 3$  grid a UAV surveys simultaneously) for all the work described in this paper.

## 4. RESULTS

### 4.1 A Simple Search Area



**Figure 2: The simple search map. All cells in the uniform grey region are to be surveyed ideally once every 2000 iterations.**

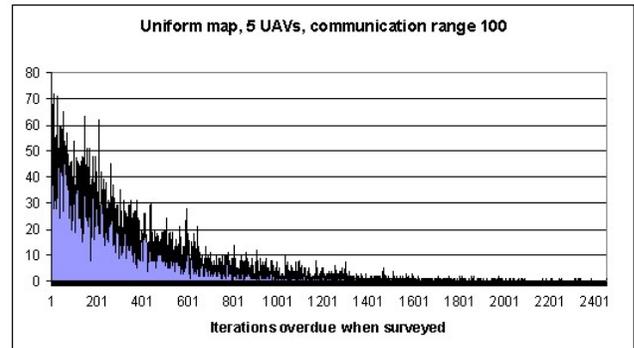
Consider the search area shown in figure 2. This consists of a  $200 \times 200$  cell region that requires uniform surveillance with a boundary of 8 cells that require no surveillance. The pheromone update factor is set to 0.0005 corresponding to a requirement that each cell that is to be surveyed should be surveyed every 2000 time intervals.

There are  $200 \times 200$  cells to be surveyed once every 2000 time intervals and since each UAV is, for the purposes of the work reported here, assumed to be able to survey 9 cells (a  $3 \times 3$  survey pattern) each time interval, the lower limit to the number of UAVs required is given by the next integer above  $\frac{200 \times 200}{2000 \times 9}$  which is 3. This figure is obviously impossible to achieve in practice as it would require the UAVs to be able to move to any cell on the map in a single iteration. Since the UAVs modeled here must travel without teleporting, it is to be expected that the number required will be (much) higher than this figure. Table 1 shows the performance of various numbers of UAVs as a function of their communication range. For each combination of number and

communication range two figures are presented: the first is the percentage of cells that are overdue on average for the iterations from 2000-10000 (no cell can be overdue in the first 2000 iterations obviously), the second is how many iterations on average these cells are overdue. As can be seen from table 1, 15 UAVs that do not communicate can not survey all cells often enough while with a communication range of 100 they can.

Although the percentage that is inadequately surveyed is small, those that are not adequately surveyed are on average left for an extra over 260 iterations (just over 13% of the required survey period). As the communication range rises, resulting in more and more collective intelligence, the performance rises. With a communication range of 100 cells it only requires about nine UAVs to get a similar performance. Fifteen UAVs, each with a communication range of 100 cells, are able to meet all the survey requirements with no cell being surveyed less frequently than desired (many, of course, are being surveyed far more frequently than strictly required).

Averages, while informative, do not tell the whole story and figures 3 and 4 use histograms to show the distribution of lateness for 5 UAVs and for 10 UAVs respectively. In each case the cell communication range is 100 cells. Note that the majority of cells that are overdue for surveying are only slightly overdue and, unsurprisingly, that there is a significant improvement from a doubling of the number of UAVs.



**Figure 3: A histogram of cell lateness for the simple map when surveyed using 5 UAVs and a communication range of 100 cells.**

### 4.2 A More Complex Area

In practice it is improbable that all areas will need to be surveyed equally frequently: a lake need never be surveyed and bare rock escarpments infrequently surveyed while bush land, especially in dry gullies, needs to be surveyed often. The map shown in figure 5 has been developed to simulate this. This map has a  $300 \times 300$  grid of cells that, in general, need surveying. This map has three types of area, one requiring high, one requiring medium and one requiring low frequency surveillance (in the ratio of 4:2:1). In addition there are regions that need not be surveyed in the map. A similar calculation to that made for the simple map shows that 6 UAVs could collectively survey enough cells per iteration to allow these requirements to be met, assuming that the UAVs could move from any cell to any other cell on

UAV count	Communication range					
	0 cells		30 cells		100 cells	
	% late	Iterations late	% late	Iterations late	% late	Iterations late
1	53.48	1825.54				
2	31.53	1163.62	31.32	1064.09	27.99	965.19
3	19.85	886.88	16.24	662.80	13.20	596.27
4	10.72	681.23	7.59	457.39	5.64	419.39
5	6.62	586.29	3.54	357.90	1.86	325.66
6	4.26	513.63	1.13	296.41	0.66	270.72
7	2.59	441.98	0.41	247.04	0.23	225.94
8	1.60	406.68	0.17	242.65	0.08	186.47
9	0.89	359.05	0.06	196.24	0.02	149.77
10	0.73	349.36	0.02	168.66	0.01	126.60
11	0.44	314.41	0.01	147.48	0.00	39.66
12	0.26	283.64	0.00	86.84	0.00	7.69
13	0.20	277.83	0.00	51.17	0.00	10.42
14	0.17	272.95	0.00	2.16	0.00	0.75
15	0.05	262.43	0.00	7.92	0.00	0.00
No collective intelligence			With collective intelligence			

**Table 1: The performance of varied number of UAVs on the simple problem. The figures quoted are the average values over iterations 200-10,000 of a run**

Communication Range = 0	Survey Frequency Required					
	High		Medium		Low	
UAV count	% late	Iterations late	% late	Iterations late	% late	Iterations late
5	11.33	350.87	56.67	2768.34	68.58	2782.79
10	2.10	197.65	41.76	2469.14	56.83	2784.00
15	0.47	104.43	34.54	2455.12	50.37	2784.99
20	0.09	58.71	30.60	2256.29	47.24	2782.98
25	0.02	46.72	29.26	2487.53	43.38	2799.78
30	0.02	50.06	27.04	2126.20	40.20	2829.03
35	0.00	38.67	26.16	2328.72	38.71	2812.43

**Table 2: The performance of each of the regions without collective intelligence (communication range = 0)**

Communication Range = 100	Survey Frequency Required					
	High		Medium		Low	
UAV count	% late	Iterations late	% late	Iterations late	% late	Iterations late
5	23.55	424.46	28.21	895.01	27.78	1796.70
10	5.58	205.54	4.80	412.74	3.03	802.88
15	1.04	152.78	0.68	293.61	0.16	531.51
20	0.19	112.71	0.10	221.62	0.02	418.88
25	0.02	60.06	0.01	143.36	0.00	255.71
30	0.00	9.36	0.00	46.97	0.00	0.00
35	0.00	1.30	0.00	4.12	0.00	0.00

**Table 3: The performance of each of the regions with collective intelligence (communication range = 100)**

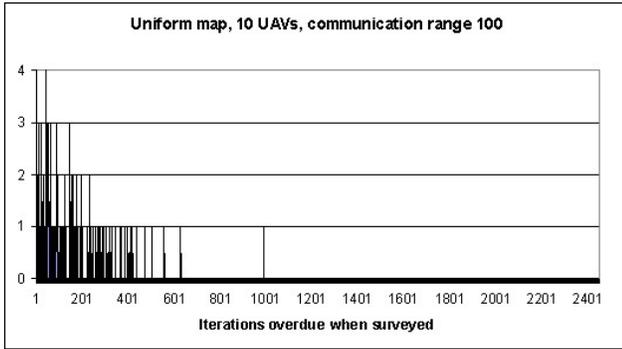


Figure 4: A histogram of cell lateness for the simple map when surveyed using 10 UAVs and a communication range of 100 cells.

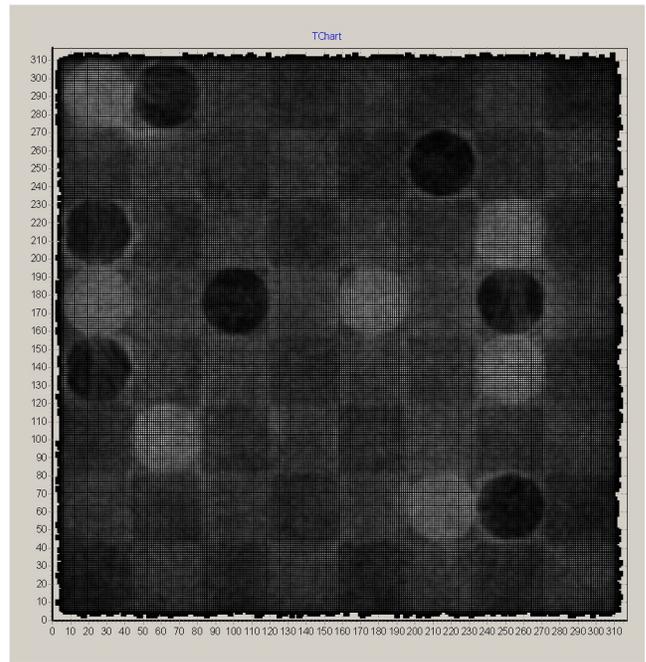


Figure 6: The relative actual survey frequencies for the complex map after 10,000 iterations. The grey scale used is the same as in figure 5

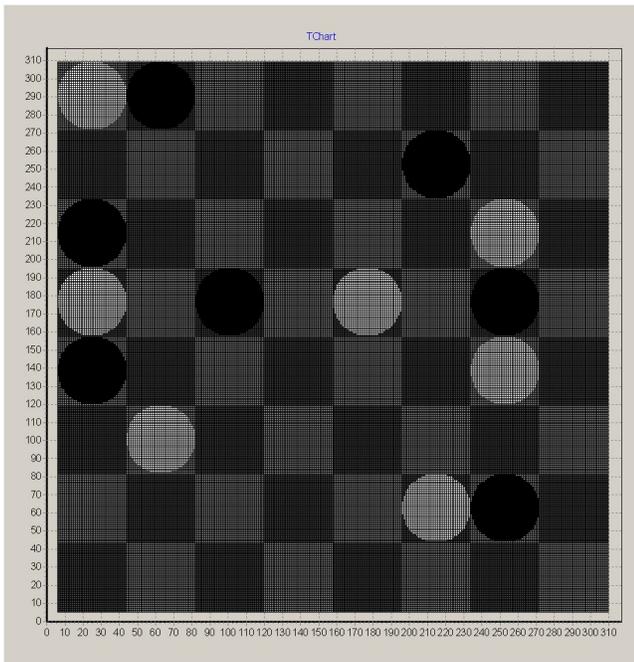


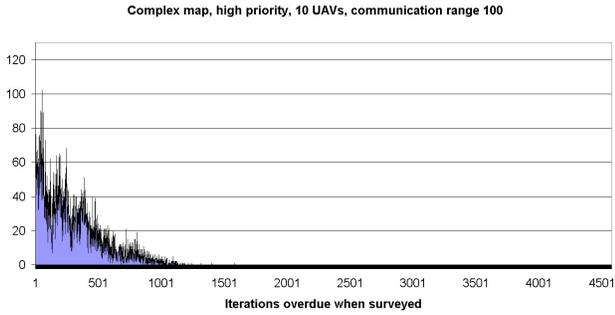
Figure 5: The complex search map. The grey scale shows the relative survey frequency required. The lighter the area the more often it is to be surveyed. The relative survey frequencies are 4:2:1:0

the map in one iteration. Clearly this is impractical and the number that might be required would be (significantly) higher. Figure 6 shows, using the same gray scale, the frequency of actual visits made to each cell on the map (visits to cells that need not be surveyed are not shown) by 30 UAVs running for 10,000 iterations.

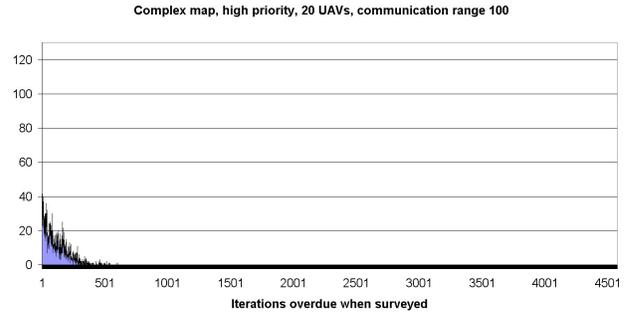
Inspection shows that the ratio of the actual survey frequencies is remarkably close to that required, although there are some edge effects. There were some surveys done in the border region as UAVs approaching the boundary of the map swung back towards the regions to be surveyed (these are the jagged edges on figure 6). However, a grey scale representation of survey frequency hides much detail and table 2 shows the average performance for each of the reportable regions in this complex map, when the communication range is set to zero (this is no collective intelligence). Similarly, table 3 shows the average performance for each of the reportable regions in this complex map, when the communication range is set to 100 (that is, with collective intelligence).

Comparing the two tables we can see that, when the number of UAVs is far too small for the task, effort is concentrated on the high priority cells in the absence of collective intelligence. With collective intelligence, the effort is more evenly distributed across the three priorities. As the number of UAVs increases, the performance with collective intelligence rapidly outstrips the performance without. Effort is still more evenly distributed across the priorities with collective intelligence. With 25 UAVs the performance of both on the high priority is very good, but the performance on the other two priorities is poor without collective intelligence.

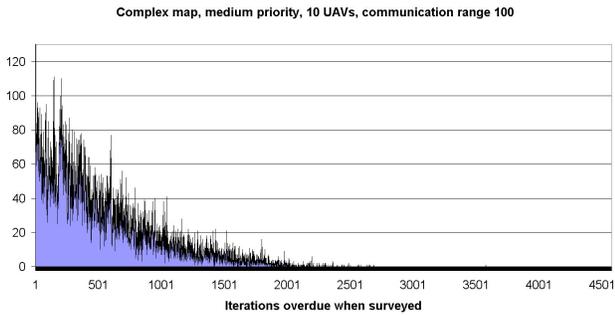
Finally figures 7(a), 7(c) and 7(e) respectively show histograms of the high priority, medium priority and low priority cells surveyed late for 10 UAVs, while figures 7(b), 7(d)



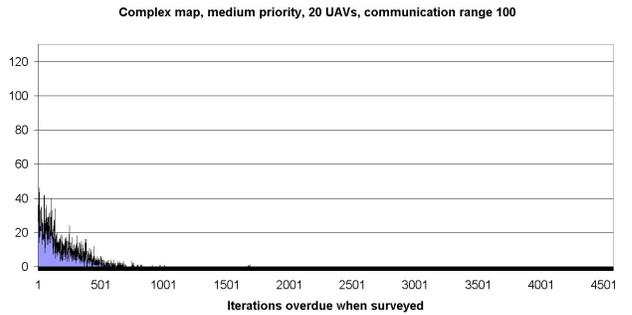
(a) Histogram for the high priority cells using 10 UAVs.



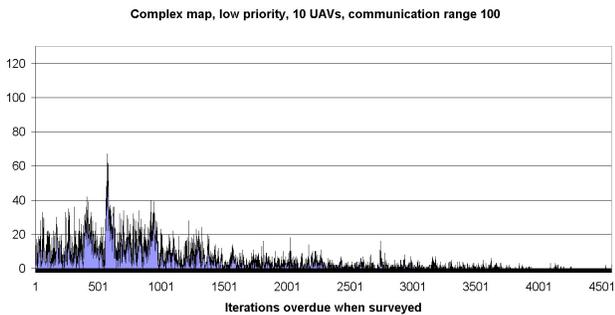
(b) Histogram for the high priority cells using 20 UAVs.



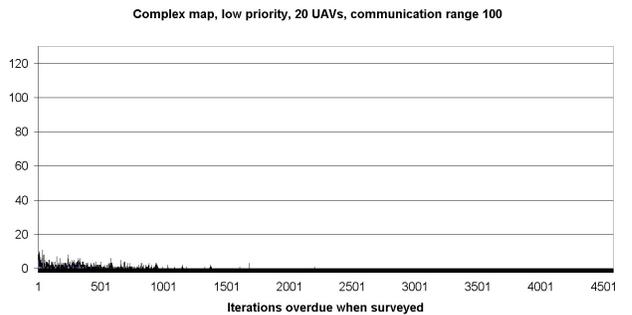
(c) Histogram for the medium priority cells using 10 UAVs.



(d) Histogram for the medium priority cells using 20 UAVs.



(e) Histogram for the low priority cells using 10 UAVs.

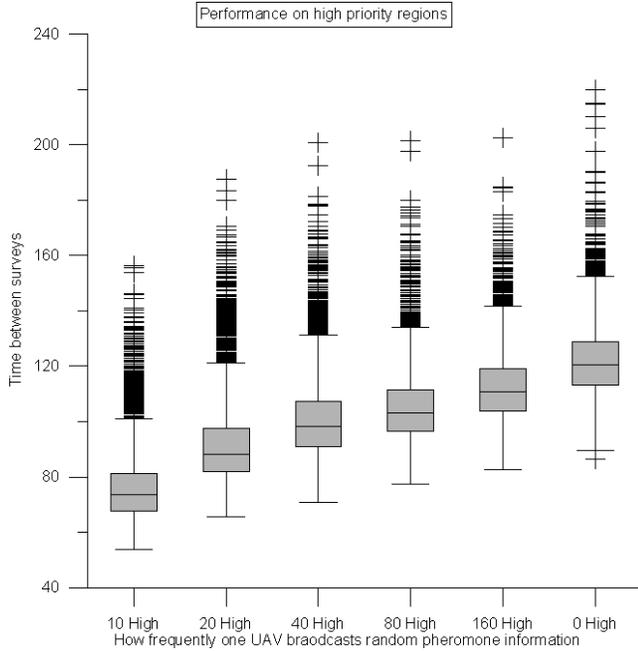


(f) Histogram for the low priority cells using 20 UAVs.

**Figure 7: Histograms of cell lateness for different priority cells when using the complex map and a communication range of 100 cells. Left hand histograms involve 10 UAVs, right hand histograms involve 20 UAVs.**

and 7(f) respectively show histograms of the high priority, medium priority and low priority cells surveyed late for 20 UAVs. These show that in all cases most of the overdue cells are surveyed when only slightly overdue, especially for the case when 20 UAVs are being used.

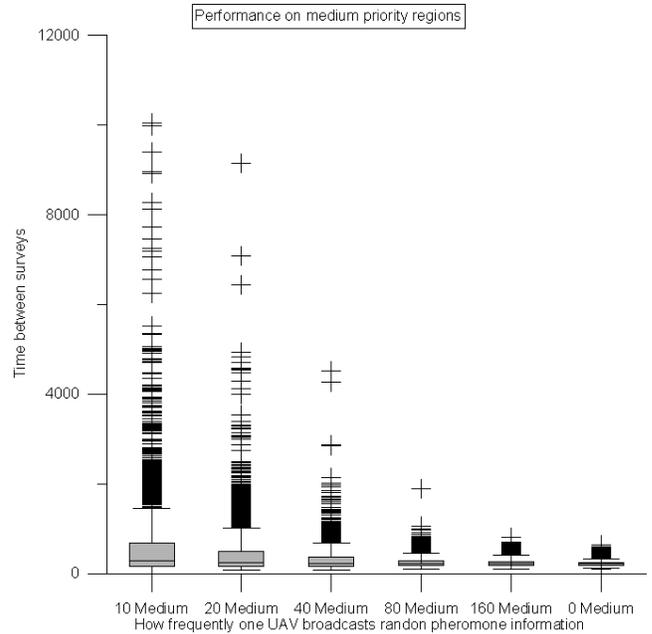
### 4.3 A More Complex Area With Communication Failures



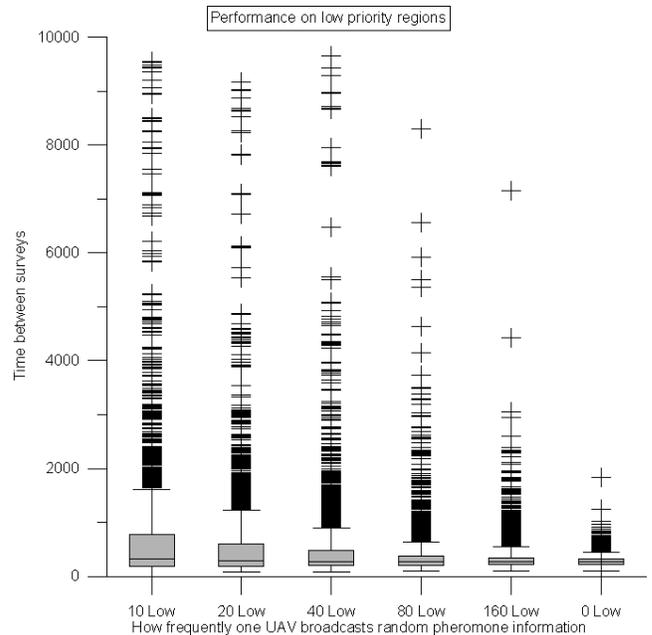
**Figure 8:** The time between surveys of high priority cells as a function of how frequently one UAV broadcasts random pheromone information

In the real world it cannot be assumed that all UAVs will operate ideally at all times, for example it is possible that one might crash. In this case the performance rapidly degrades to the normal performance that would be expected from the new number of UAVs. A more serious failure would be one that resulted in a UAV broadcasting incorrect pheromone information to the other UAVs in reception range. This has been simulated by making one UAV broadcast random pheromone values every  $N$  iterations. Figures 8 to 10 show the performance for the high, medium and low priority regions respectively under these conditions when 40 UAVs, each with a communication range of 100 cells, are used. Note the very different vertical scales for these three figures. The horizontal axis shows how frequently one UAV broadcasts random pheromone information, the last plot (iterations between random communication events = 0) shows the performance when no random pheromone values are ever communicated. For each of these box and whisker plots the caps at the end of each box indicate the extreme values (minimum and maximum), the box is defined by the lower and upper quartiles, and the line in the center of the box is the median. All points above the upper quartile are marked by a cross so that the distribution of these worst case outliers can be seen.

Inspection of these three figures shows that the net effect of this miscommunication is for performance on the local



**Figure 9:** The time between surveys of medium priority cells as a function of how frequently one UAV broadcasts random pheromone information



**Figure 10:** The time between surveys of low priority cells as a function of how frequently one UAV broadcasts random pheromone information

priority areas to decrease, particularly in respect of the outliers. This decrease is caused by the high priority cells being over surveyed (slightly) as a result of the misinformation. The overall effect on the majority of points (of any priority) is small and the algorithm seems able to adapt well to a certain amount of noise being injected into the pheromone maps of the UAVs.

## 5. CONCLUSION AND FUTURE WORK

The algorithm clearly shows that it is able to survey an area containing regions with differing survey requirements. It is able to adapt to the loss or addition of one (or more) UAVs, with the performance changing gracefully. Communication, and hence collective intelligence, is mandatory to get this level of performance and occasional communication errors appear able to be readily absorbed without a major performance downgrade. Some parameters, most noticeably those involved in repulsion between UAVs that come too close to each other, have not yet been investigated in detail. The values do not appear to have a significant effect on the overall performance but it is not claimed that the figures used in this work are optimal or that they would necessarily be effective for some other problem.

In a real life situation UAVs would initially leave from one (or a few) locations, rather than be randomly distributed across the map. Also they would need to return from time to time for fuel. While these conditions remain to be explicitly simulated, observing the behavior of the UAVs in the many simulations already done leads the authors to suggest that neither of these changes will significantly change the performance of the algorithm.

## 6. ACKNOWLEDGMENTS

Our thanks to other members of the Intelligent Systems Group of the Centre for Information Technology Research for valuable discussions.

## 7. REFERENCES

- [1] P. Scerri, Y. Xu, E. Liao, J. Lai, M. Lewis, K. Sycara, 'Coordinating very large groups of wide area search munitions', Recent Developments in Cooperative Control and Optimization, Dordrecht, NL: Kluwer Academic Publishers
- [2] Enns, D.; Bugajski, D.; Pratt, S., 'Guidance and control for cooperative search', American Control Conference, 2002. Proceedings of the 2002 , vol.3, no., pp. 1923-1929 vol.3, 2002
- [3] Polycarpou, M.M.; Yanli Yang; Passino, K.M., 'A cooperative search framework for distributed agents', Intelligent Control, 2001. (ISIC '01). Proceedings of the 2001 IEEE International Symposium on , vol., no., pp.1-6, 2001
- [4] M. L. Baum and K. M. Passino. 'A search-theoretic approach to cooperative control for uninhabited air vehicles'. In AIAA Guidance, Navigation, and Control Conference and Exhibit, August 2002
- [5] Kennedy, J and Eberhart R.C. 'Particle Swarm Optimization' Proc. IEEE International Conference on Neural Networks, Perth Australia, IEEE Service Centre, Piscataway NJ USA Vol IV pp. 1942-1948. (1995)
- [6] Dorigo, M., & Stützle, T. (2002). 'The Ant Colony Optimisation Metaheuristic: Algorithms, Applications and Advances'. In F. Glover & G. Kochenberger (Eds.), Handbook of Metaheuristics (Vol. 57, pp. 251-285). Boston, MA: Kluwer Academic Publishers.
- [7] Urfalioglu, O. 'Robust Estimation of Camera Rotation, Translation and Focal Length at High Outlier Rates', Proceedings of the 1st Canadian Conference on Computer and Robot Vision (CRV'04), IEEE Computer Society Washington, DC, USA. Pages: 464 - 471. 2004