Minimalist Coherent Swarming of Wireless Networked Autonomous Mobile Robots

Julien Nembrini   Alan Winfield   Chris Melhuish

Intelligent Autonomous Systems Laboratory
University of the West of England
Coldharbour Lane
Bristol, United Kingdom

Abstract

This paper presents the results of simulations investigating a decentralised control algorithm able to maintain the coherence of a swarm of radio-connected robots. In this work the radius of communication is considerably less than the global diameter of the swarm. The study explores coherent movement towards a beacon while avoiding obstacles and maintaining global shape. All behaviours are emergent as the study constrains itself to using only a restricted range omnidirectional radio, a beacon sensor and avoiding sensors. In spite of such restrictions we achieve the proposed aims. Moreover, the fact that the algorithm relies only on local information makes it highly scalable with an increase in the number of robots. The proposed approach also shows graceful degradation in the presence of noise.

1. Introduction

Bacteria and Amoebae are impressively efficient in surviving in their respective niche and such an accomplishment has never been reached by robotic artifacts. Symbolic A.I. has shown its limitations in real robot experiments (Pfeifer and Scheier, 1999), and in contrast behaviour-based robotics has put the emphasis on the relationships of an agent with its environment. It is refered to as *situatedness.* Moreover some researchers argue that the physical implementation of the robot, its *embodiment,* is essential not to bypass assumptions about the real world (Brooks, 1991). This study is conducted within this framework.

Situatedness is now a widely used concept in the robotics field. It is commonly agreed that an intelligent agent cannot be considered apart from the environment with which it is supposed to interact. This has focused research on the morphology of the agent as crucial for its interaction with the environment. It has been shown that a specific morphology adapted to a specific environment can greatly enhance the robot's abilities without an increase in the agent's computational complexity (Pfeifer and Scheier, 1999, Melhuish et al., 1998). Artificial evolution of morphology has also been investigated, mainly in simulation, and demonstrated some impressive results (Lipson and Pollack, 2000, Sims, 1994).

In these examples the change in morphology, either man-made or automatic, made the robot more efficient in a defined task and hence more adapted to its environment. This success raised the idea of a robot able to modify its own shape while executing a task. Such a skill would increase the adaptability of the robot to changing environments.

For a robot to be able to conduct this metamorphosis, it could be composed of smaller components that can rearrange into different shapes. A good deal of research is now investigating reconfigurable robots that consist of basic modules able to perform simple tasks such as attaching with, detaching from or moving relative to a neighbour. Tasks such as locomotion of the whole group is the result of a combination of these basic actions (Kotay and Rus, 1999).

This research extends the work of Winfield (Winfield, 2000) related to sensor networks, as well as the work of Melhuish (Melhuish, 1999) that focused on secondary swarming. This study looks at the employment of mobile robots to combine sensing, locomotion and morphological adaptivity. Following this preliminary direction of research, we propose here to study a swarm of autonomous mobile robots communicating with limited-range radios. Instead of a physically connected system, this approach investigates virtually (wireless) connected robots.We believe that this approach brings greater versatility in morphology and robustness to failures and noise. Also the hardware involved is readily available, allowing us to concentrate on the development of the algorithms. Of course the drawback lies in the fact that much effort has to be made to keep the robots together, *i.e.* to maintain the *coherence* of the swarm.

Working initially in simulation, we show that using only an omni-directional radio and a collision avoidance

device is enough to achieve this coherence in an unbounded environment. The swarm then forms a one-connected component communicating network. Adding a sensing ability, we achieve emergent directed swarming towards a light beacon, and present the potential to control the swarm global shape .

In this research we are primarily concerned with developing algorithms that are:

scalable: An increase in the number of agents in the swarm should not lead to complete failure. Indeed the efficiency should show graceful degradation to such an increase. The study is therefore relying on distributed solutions that make use of strictly local information. Our robots are all identical and exchange information only about their neighbourhoods.

robust: The algorithm should provide the swarm with the ability to cope with the unreliability of the real world and show graceful degradation when confronted with noise and component failures. The last requirement is again a reason to seek distributed algorithms controlling homogeneous robots.

2. Related Work

As stated above, the main area of research concerned with online morphology change is the field of reconfigurable robots. The motivation is to achieve function from shape, allowing individual modules to connect and reconnect in various ways in order to achieve the required function.

Because of the technical challenge it represents, recent work has emphasized the design of physical robots. But the control algorithms have as yet mainly been investigated in simulation and the focus appears to be on state representation and planning (Kotay and Rus, 1999, Yoshida et al., 2000) instead of distributed algorithms. An interesting example of distributed control is the recent work of Støy that achieves caterpillar locomotion and multi-legged walking with the use of a fully distributed algorithm that is able to deal with online reconfiguration (Støy et al., 2002).

Our work is concerned with groups of robots that are more loosely connected. While in the field of reconfigurable robots, individuals are normally physically linked together, our approach considers limited-range radio connections that build a dynamic wireless network. Connections are unreliable and often lost. An interesting example of group pattern formation with the use of wireless connection can be found in (Wessnitzer et al., 2001)

Another field related to the work presented here is the study of formation control. The aim is to move a group of robots while maintaining relative position, hence forming a global shape such as, for instance, migrating birds. The work of Balch and Hybinette (Balch and Hybinette,) for instance relies on distributed solutions and shows desirable properties such as scalability and locality. But trying to maintain accurate relative positions involves high-level sensing abilities. On the other hand the control algorithm described in this paper allows some freedom of movement for an individual while maintaining the global shape of the swarm.

Although biomimetics was not the aim at the beginning of our project, it appears that the moving swarm shows comparable properties with amoeboid plasmodium such as *Physarum* or *Dictyostelium*. Interesting work in the field of robotics that follows this idea can be found in (Takahashi et al., 2001).

In the work most closely related to this research, Støy (Støy, 2001) studies real robots trying to stay together by sending messages to others. The approach uses the limited range as a physical advantage: it enables a robot to count how many others are in range and hence get an idea of its neighbourhood density. The control algorithm then makes the robot turn and come back if the number of neighbours is decreasing. The aim is to find whether it can help robots keep together within a bounded arena. The result is positive, but the robots sometimes lose the group and are able to rejoin only by chance thanks to the bounded arena.

Our aim was to remove the constraint of a bounded arena. We have therefore looked for algorithms that conserve the integriy of the group, since the lack of absolute or relative position information in case of deconnection would make the loss of a robot critical. However, our concern for the scalability of the process leads us to use a similar idea for sharing neighbour information.

3. Methods

The research follows a method of investigation that consist of, firstly, designing and implementing the algorithm in simulation to gain insights into the dynamics involved and the problems that may arise. It must be borne in mind that the results depend highly on the degree of accuracy of the simulation. This is the reason that the resulting algorithms must, secondly, be implemented on real robots, while taking full advantage of the simulation work. This paper presents the results of the first stage only, as we are yet to verify our algorithm on real robots.

Following the focus on minimalist design of Melhuish (Melhuish et al., 1998) the aim is to keep the robots as simple as possible. It is believed that coherence is achievable only with a radio device with limited range for communication and proximity sensors for avoidance. The key idea is that the limited range is giving sufficient information on relative position. Such severely constrained conditions oblige us to tie together the act of communicating with other behaviours of the robot. It is referred in (Støy, 2001) as *situated communication*.

These assumptions on hardware make the require-

ment on coherence even more critical, as a disconnected robot will not be able to return to the swarm, as it lacks any information about relative or absolute position.

Due to the underlying motivation on sensory network applications we also seek to minimise the communication overhead that is necessary to achieve coherence. This is to retain maximum bandwidth for data gathering.

At the same time, our requirement on robustness in the presence of noise is the reason for the solution presented to the problems of directed swarming and global shape control, that both emerge from spatial differentiation.

3.1 Simulation Details

As developing a simulation that tries to take every interaction that contributes to the dynamics into account would be impractical, a simple approach is implemented, bearing in mind the aim of physical realisation to avoid unfeasible solutions. As a consequence noise is not simulated following real models, but instead introduced as false sensor readings or loss of messages in a random manner; the purpose being to test the robustness of the algorithm.

We assume the robots are able to move forward and turn on-the-spot with reasonable precision, that they have infra-red avoidance sensors, are equipped with limited-range radio devices and that they carry an omnidirectional light sensor able to detect whether a robot is illuminated or not.

Communication is implemented as follow: two robots in range are considered to communicate perfectly. Noise is introduced using a constant probability to lose the entire message. This probability can range from zero to 0.1 and we do not attempt to model buffer overflow or any other real phenomena occuring such as signal decay. The motivation for such assumptions can be found in (Winfield, 2000).

In addition to wireless communication an avoidance behaviour is implemented that makes use of three infrared sensors, one on each side of the front of the robot, and the third at the back. All sensors have the same range, smaller than the range of the communication device. The avoidance behaviour causes the robot to turn in the opposite direction of a single activated front sensor, move backwards if both front sensors are activated, or stop if all three sensors are activated. There is also an option to introduce random noise as a probability of false sensor readings to this feature.

In order to study taxis and adaptable morphology, a beacon is introduced as well as its decay and occlusion through line-of-sight obstruction. Again the design was dictated by its possible realisation, the metaphor being a bright light beacon. Noise is also introduced with probabilistic false sensor readings. The simulation also includes the option to introduce into the arena some
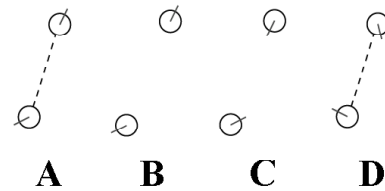

Figure 1: Basic algorithm

occlusive obstacles in the path of the swarm.

3.2 Algorithms

Basic Algorithm

This study restricts itself to use only the information on connections between robots. In other words whether a particular robot is receiving a signal from another or not. The omnidirectionality of the radio device implies that there is no positional indication about where to go in case of deconnection. To transform this limited information into good use we exploit the ability of the robot to turn on the spot with good precision: in the default state the robot moves forward. As soon as the control algorithm detects a deconnection, the robot assumes it is going in the wrong direction and turns back.

For simplicity let us restrict ourselves to the case of two robots (see figure 1):

Assume that the robots are initially in communication range, moving forward with random headings (A). Unless they have parallel or crossing trajectories[1], they will eventually lose contact(B). In order to check whether this is the case or not, the algorithm uses a call-answer mechanism: with a certain periodicity each robot sends a message to the other ("are you there") and waits for a reply ("yes I'm here"). If the reply is not received within a certain time the robot assumes it is out of range (B) and reacts immediately by turning 180 degrees in order to reconnect (C). Then as soon as it receives a reply to its calling messages it chooses a new random heading (D).

As no global time is implemented the robots should react asynchronously. However each robot has the same range of communication so both reactions should occur within a short time, depending on the periodicity of the calling messages.

This behaviour leads the two robots to maintain themselves in range as if they were attached with an elastic band. The choice of a random heading when reconnection occurs makes the pair as a whole follow a random walk. It is important to observe that the reciprocity of reaction even though not simultaneous is crucial to retaining the connection. Homogeneous robots

---

[1]the former situation wouldn't harm the connection and the latter is dealt by the avoidance behaviour.
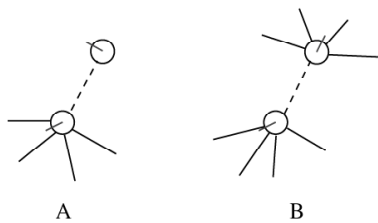
Figure 2: Extreme states

Figure 3: Shared neighbour

have equal velocities and the reaction of only one robot could lead to an endless pursuit.

Following Støy, it is worth emphasizing here the fundamental characteristic of this paradigm to achieve swarming. It is not the semantic content of the message alone that matters but it being tied with an environmental cue, that is the presence -or absence- of its response. This algorithm stands in the framework of situated communication where the semantic content of a message is closely linked to an environmental meaning.

Applying this basic algorithm to a greater number of robots by making a robot react to every loss of connection leads to an over-reactive swarm which clumps together. To react to every connection is equivalent to aiming towards a complete graph where each vertex is connected to every other. This is not our aim.

Trying to make the robots less reactive has demonstrated extreme situations that must be avoided in order to assure the coherence of the swarm (see figure 2). When a robot (or a group) is linked to the rest of the swarm by a single communication link, the danger lies in the possibility of a robot not reacting to the loss of such a connection essential to global connectivity. In graph theory a vertex representing such an important connection is known as a *bridge*.

Coherence

To avoid these situations we make use of the graph theory concept of *clustering*. Instead of considering only its own degree of connection to trigger a reaction, the robot will receive from its neighbours their adjacency table - their neighbours' list - in order to check whether a particular neighbour is *shared* by other ones, that is whether a particular neighbour is the neighbour of other robots' neighbours.

The algorithm works as follows: for each lost connection the robot checks how many of its remaining neighbours still have the lost neighbour in their neighbourhood. If this number is less than or equal to the fixed threshold β the robot turns back. In parallel if its degree of connections is rising the robot chooses a random heading.

For instance in the situation on figure 3, robot A, when losing the connection with robot B, will check its
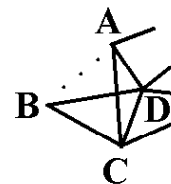
other neighbours and find that robots C and D share B as neighbour. Hence A will react and turn back only if β is equal or greater than two. Our algorithm makes the robot try to maintain the triangulation observable on the picture, therfore avoiding critical states.

The pseudo-code of the algorithm for one robot is set out below.

```
Create list of neighbours for robot, Nlist
k = number of neighbours in Nlist

loop forever {
   Save copy of Nlist in Oldlist
   Save copy of k in LastK
   Set reaction indicator Back to FALSE

   Send radio 'ping' to neighbourhood
       every 100 time steps
   Listen for return calls from robots
       in range that received the 'ping'
   Create Nlist from all returns
   k = number of neighbours in Nlist

   Create LostList, list of robots which have
       lost contact since previous 'ping'

   for (each robot in LostList) {
     Find nShared, number of shared neighbours
     if nShared <= beta (threshold value) {
       Set reaction indicator Back to TRUE
     }
   }

   if Back=TRUE {
     turn robot through 180 degrees
   }
   else if k > LastK {
     make random turn
   }
}
```

It is interesting to note that the robot tries to maintain β shared connections with each neighbour and one might think that this would lead to over-connectivity, as described earlier. But in fact each connection can contribute to different sharings and such a condensed clus-

tering is never reached. On the other hand if the robot for instance establishes a new link with a robot that does not have other connections with the robot's surrounding neighbourhood, it will react to the loss of this neighbour until the shared connections are also established. This is precisely the behaviour we are aiming for.

Running the simulation confirms that such an algorithm increases swarm coherence, as the triangulation is perfectly observable and therefore critical states avoided. A value of β equal to one is enough to achieve coherent spread.

Of course the communication bandwidth of the whole process is somewhat increased compared to the basic algorithm, as well as the processing power needed for the robot. More sensitivity to the message content (semantics) is also introduced. However, the communication is still situated, and hence message loss or misinterpretation only leads to over-reactivity and an increase in connectivity without loss of robots, as introduction of noise in the simulation confirms. Also this increase in bandwidth does not affect the scalability of the algorithm as it concerns only exchanges between neighbouring robots and will therefore not be propagated more than a single hop in the network.

Directed Swarming Algorithm

When following a beacon gradient (chemical, sonic or light...) a possible solution consists of placing two differents sensors on both sides of the robot and then making the robot turn towards the sensor indicating the highest value. This implementation is highly dependent on signal-to-noise ratio as the robot's sensors are not situated far from each other which makes the two different sensing values very similar.

We raise the hypothesis that it could be possible to use different robots to take a sample, share it with their neighbours to generate an approximation of the position of the beacon and then make the whole swarm move towards it.

Although the problem of localization of the beacon might seem the most difficult to answer, the movement also represents a real challenge in our case. Indeed, how is it possible to make a group of robots that do not have any precise idea of relative directions or positions follow a particular path ?

The answer is in binding these two problems together using the light occlusion implementation described earlier. The idea is to make an illuminated robot become special to its neighbours by entering a special state - let us call it 'red'. Then a layer is added on top of the shared neighbour behaviour that always triggers a robot's reaction if the lost connection involved a red robot. This holds whatever the number of shared neighbours.

In figure 4, depicted in grey are the robots illuminated by a beacon situated north of the picture.
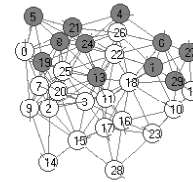

Figure 4: Illuminated robots

The following line is added in the *for* loop of the pseudo-code:

```
if (color of lost robot is red) then
        set Back to TRUE
```

This is in effect a spatial differentiation in the β threshold value, setting it to infinity for the red robots. As a result, these robots try to build complete graphs among themselves, reacting to each loss and therefore not moving much while the others always stick to the red ones. But as the red ones build their complete graph they occlude the previous red ones that stand inside and a slow translation of the swarm starts towards the beacon.

It is important to note that the basic sensing of 'being illuminated' or not does not make a single robot able to reach the beacon. The taxis behaviour only results from the interaction of the red robots and the rest. This is a truly emergent behaviour and as such it is highly dependent on the various parameters in action: communication range, rate of occlusion, avoiding range, etc. These dependencies remain to be investigated in real robot experiments. But it is much less sensitive to noise than a classic two-sensor beacon taxis as the sensing does not need high sensitivity: the robot is either illuminated or not.

With occluding obstacles in the path between the swarm and the beacon, the swarm is able to find its way through. After an initial random spread, the beacon is sensed by a few robots that attract the swarm through the obstacles. If the space between the obstacles is restricted, the swarm will adapt its shape to be able to pass through the gap (see figure 5).

To control the shape of the swarm more differentiation is introduced: either the red robots are made much less rapid compared to the others or the contrary. The result is a swarm that forms a line oriented perpendicularly to the direction of the beacon or towards it, respectively.
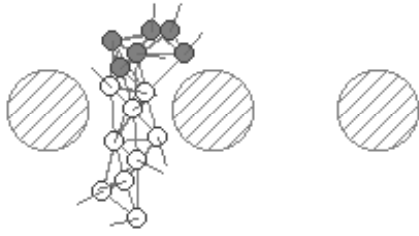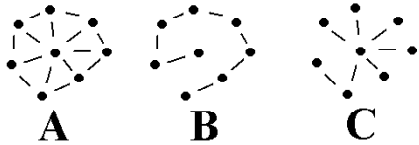
Figure 5: A swarm moving through obstacles

Figure 6: A graph(A) & two disjoint spanning trees(B,C)

3.3 Measures

In this study, the first feature that has to be measured is *swarm coherence*. But as the loss of a connection is a discrete event, the quantity that needs to be measured is not obvious. Should we restrict ourselves to detect when the swarm becomes disconnected, or should we look for finer criteria?

To answer this question we introduce here the concept of *disjoint spanning trees* (see figure 6). The aim being to have a gradation in levels of coherence. We stated earlier the importance of bridges as extreme states to be avoided. This raised the idea that in the presence of a bridge there is no possibility to find a spanning tree disjoint to the one that goes through the bridge. Extending this idea, we measure the number of edge-disjoint spanning trees that can be constructed on the graph of the robots' network.

This is an integer value that gives information about the global connectivity of the graph. It is related to the n-connectivity (Hobbs, 1989). Also unlike the n-connectivity which is computationally NP-hard, an approximation algorithm to compute such a value is readily available. As heuristic to search the graph, we use the degree of connections and follow the highest value downwards.

Following our underlying concern for sensor networks, the relevance of such a quantity to the ability of the network to propagate messages is another motivation for measuring it.

Another quantity that is important to measure is the

area coverage of the whole swarm. This can be done by triangulation of the bounding polygon of the graph. It can also be approximated through the calculation of the mean degree of connections over the whole graph. This number depends on the radius of communication and is therefore highly correlated with area coverage.

To measure success of taxis with or without obstacles, we compute the remaining distance of the centroid of the swarm to the beacon. This is to be able to study the scalability of the algorithm as we increase the number of robots, as well as the influence of noise on the speed of the swarm.

In the case of reactive morphology change, we measure the ratio between the added square distance of each robot to the line starting from the beacon passing through the centroid and the added square distance to its perpendicular. This gives an idea of the shape of the swarm as would the ratio of the two radii of an ellipse.

$$ratio = \frac{\sum_{i=1}^{N} \left(d_{beacon}(R_{x_i}) - R_{y_i} \right)^2}{\sum_{i=1}^{N} \left(d_{perpendicular}(R_{x_i}) - R_{y_i} \right)^2}$$

4. Results

In this section we present the results of simulation runs to test coherence, scalability and robustness to noise of our algorithms in different situations.

4.1 Swarm Coherence

In this series of runs, the environment contains no beacon and is free of obstacles. The aim is to measure the influence of the parameter β on the area coverage and the coherence of the swarm as defined in section 3.3. As such an influence is dependant on the number of robots involved in the swarm, we plot these results as surfaces. The results are averaged over sampling time on each run and over ten different runs.

We can see in figure 7 that the single parameter β appears to be sufficient to control the area coverage of the swarm. Taking a close look at the plot of the area coverage reveals that this ability is best observed with a higher number of robots. Indeed the possibility for expansion is multiplied by the number of robots.
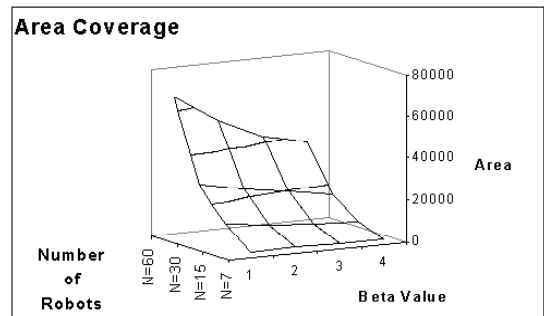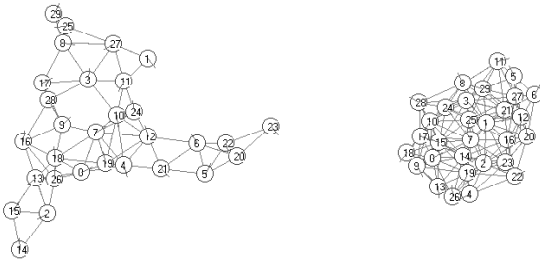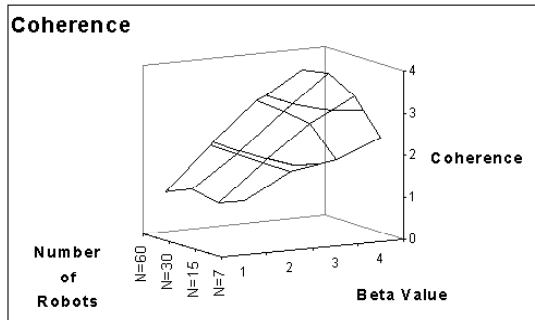
Figure 7: Typical disposition for $\beta = 1$ and $\beta = 4$

We observe that the coherence increases almost linearly with the parameter β for larger swarms. We expect of course a levelling for larger values as the avoiding behaviour starts to play its role, but in this range only the smaller swarm shows such a feature.

When we take a look at the occurrence of breaking the swarm, we note that the value of $\beta = 1$ is not a safe choice, especially for a small number of robots. In fact small swarms experience high fluctuations that are highly unstable. For the following experiments we will therefore set the value of β to two.

Below is plotted the influence of noise on the area coverage and coherence for 30 robots with $\beta = 4$. We can see that the area coverage is not particularly affected by the introduction of noise, but of course the coherence shows some degradation, as expected.
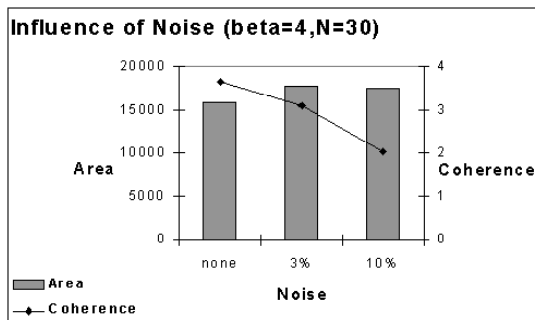

4.2 Taxis

In the experiments of this section we introduce a beacon in the environment to attract the swarm. We measure the progression of the centroid of the swarm towards the beacon every 20,000 steps.

The graph below shows such a progression for a single run of two swarms, one composed of 7 robots and the other of 30 robots. We observe higher fluctuations in the progression of the smaller swarm while the larger one moves almost linearly towards the beacon. This is due to inertia resulting from individual moves to maintain coherence in the swarm.
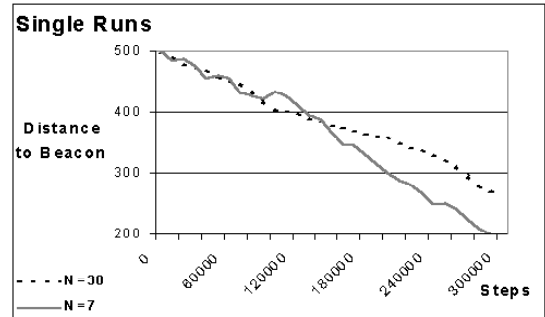
Averaging over ten different runs gives the graph below, where the slope of the lines represents different average progression speed for different sizes of swarms. The loss of speed due to an increase in the number is noticeable. Of interest is the case N=7 that shows that the fluctuations mentioned earlier actually lead to an average progression speed smaller than for a 15 robot swarm.
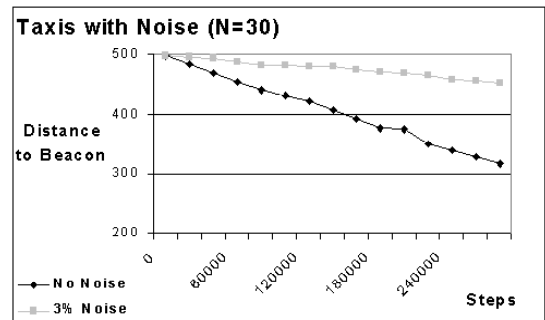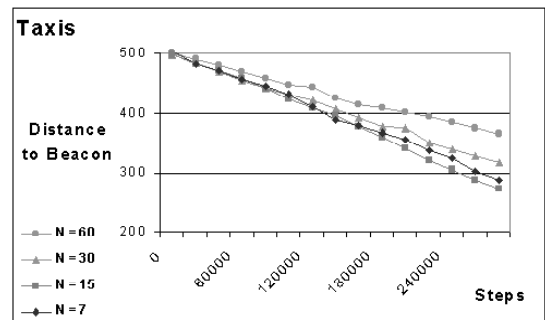
As shown in the graph above, introducing noise dramatically reduces the speed of the swarm. In fact introducing 10% of probability to lose a message does not allow the swarm to move at all. This is due to the fact that a single loss of message triggers a turn back reaction. With such a level of noise, the robots find themselves reacting too often. As a consequence the benefit of

differentiation between the illuminated robots and others tends to be cancelled, and the resulting movement no longer occurs. This shows the influence of the accuracy of the radio device (it has to be noted that the illumination sensor does not show such a sensitivity as the swarm moves towards the swarm, even with 10% of noise). A solution to deal with such a problem, following (Støy, 2001), is to make the robot wait two or three listening loops before acting on the loss of the missing robot. This would circumvent message loss due to noise and reduce the sensitivity of the whole algorithm.

Obstacle Avoidance

Adding occlusive obstacles renders the progression towards the beacon more difficult for the swarm. We observe a degradation in speed as the swarm, firstly, has to detect the beacon hidden behind an obstacle and, secondly, has to find its way through.

In figure 8 is a sequence of screenshots showing a swarm of 30 robots moving through the obstacles while aiming towards the beacon due north. Smaller swarms would go through only one aperture as in figure 5.

For larger swarms ($N = 60$ in this case), the attraction tears the swarm apart, as can be seen in figure 9. This is because the robots that have already crossed the obstacle mask the beacon to the remaining ones that are therefore no longer attracted. The remaining block is too big to be attracted through the obstacles only by the coherence requirement and the tearing occurs. Increasing the β value could be a solution, but will also decrease the differentiation, with the already described undesirable effects.

It has to be noted that the critical size of 60 robots for a swarm not to be able to cross the obstacles depends on the width of the gap between them, as well as on the level of noise.
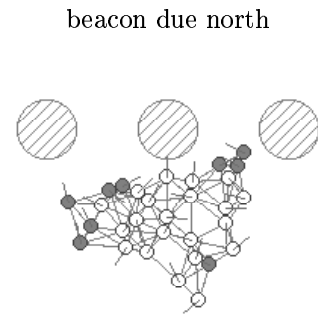
Encapsulation

An interesting side-effect of our algorithm is the behaviour of the swarm when it reaches the beacon. The interaction of the avoidance behaviour, the taxis and the reconnection strategy makes the swarm wrap itself around the beacon and encapsulate it as shown in figure 10. It is an emergent enclosure behaviour.

In a three-dimensional version of the algorithm, such a behaviour could lead to isolation of rogue material from an environment. This is somewhat analogous to Amoeba or blood macrophage phagocyte.
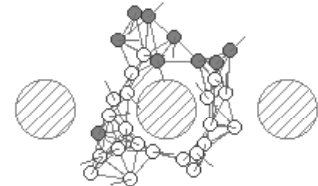
4.3 Reactive Change in Morphology

In this section, we illustrate the potential of our algorithm to make the swarm adopt a specific overall shape reacting to the beacon.
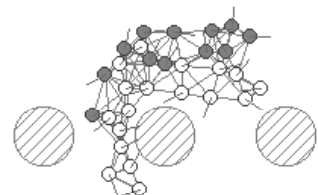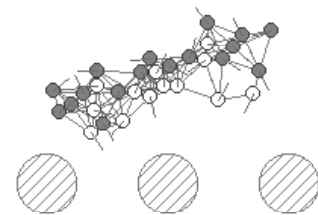
beacon due north

A
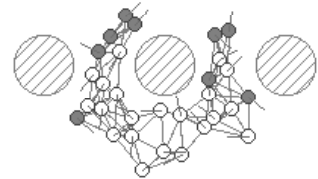
B

C

D

E

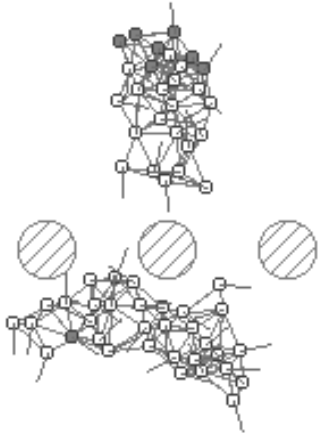Figure 8: Typical run of a swarm of 30 robots
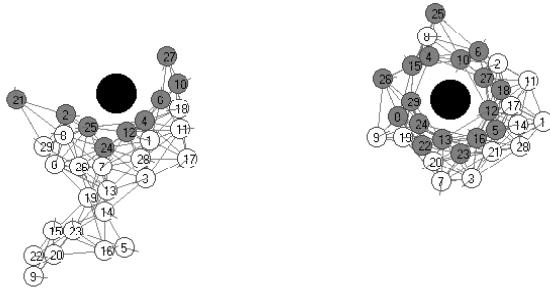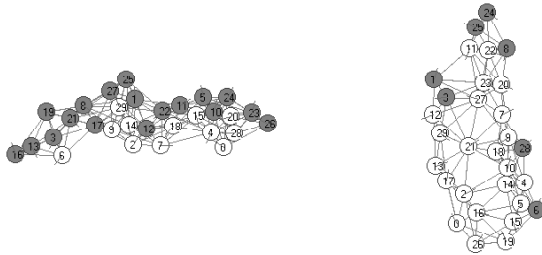
Figure 9: A broken swarm of 60 robots

Figure 10: Encapsulation of the beacon

$$v_{red} = \frac{v}{10} \qquad v_{red} = 10\,v$$

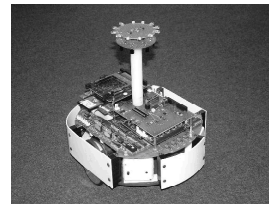Figure 11: Change in Global Shape

Figure 12: Linuxbot

We introduce a difference between the speed of the red robots and the remaining ones. Again this single parameter is able to control a dramatic change in the shape. When the red ones moves ten times faster, the swarm forms a line perpendicular to the direction of the beacon due north. On the other hand, if the red robots move 10 time slower, the swarm forms a line towards the beacon. Figure 11 shows typical examples of the two cases.

This behaviour allows us to control the exposure of the swarm to the beacon. The perpendicular shape being the most exposed. This relates to column or line formations in rigid formation studies, only tuning a single parameter and without involving high-level sensing capabilities.

In the context of sensor networks, such a behaviour could make use of a natural beacon such as the sun, in order to dispose itself along isoclines to undertake measurements. For instance it could be of interest to dispose underwater sensor robots either at the same depth or vertically. Moreover, the use of judicious switching between speeds according to environmental cues could make the morphology of the swarm adaptive.

The graph presented below shows that the presence of noise slows down the process but does not affect the qualitative result.
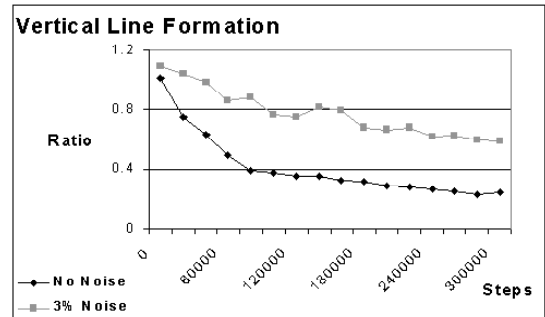
5. Future Work

The next step in our research is to implement these algorithms on real robots. For this purpose, we shall use a fleet of Linuxbots (see figure 12) developed in the IASLab. A description of this platform is to be found in (Winfield and Holland, 2000). We will equip them with an infra-red sensing tower to simulate the local-

ity of the radio-connection (the restricted dimensions of the arena preventing us from using the normal range of our WLAN).

To implement the behaviours presented in this paper we need an omnidirectional light sensor that is sufficiently occluded by the body of other robots to show the same behaviours. A solution could be the integration of different sensors disposed around the body of the linuxbot.

While conducting this hardware research, we shall continue to study the potential of the chosen direction to develop truly adaptive distributed algorithms. We are specially keen to be able to control the morphology without the help of a beacon.

Simultaneously we will investigate the relationship of our work with exemples in biology, principally amoeba and bacteria (Shapiro, 1988, Takahashi et al., 2001)

6. Conclusion

In this paper we have presented a fully distributed algorithm that is able to ensure the global coherence of a swarm of mobile robots, despite their limited-range communication abilities. We achieve also emergent coherent movement of the swarm towards a beacon, making it able to avoid obstacles and adapt its shape to move through them. Moreover the approach allows us to control the swarm global shape by tuning a single parameter (the β threshold in the case of area coverage, and speed in 'linear' formation). The algorithm has been implemented in simulation and real experiments are currently in progress.

As a consequence of the algorithm, global connectivity of the communication network formed by the swarm of robots is also achieved. This could lead to applications in large mobile sensor arrays which require them to adapt their shape to provide appropriate sensing. Indeed we believe this approach has so far revealed only little of its potential for adaptive behaviours and related applications.

References

Balch, T. and Hybinette, M. Social potentials for scalable multi-robot formations. In *Proc.Int.Conf. on Robotics and Automation ICRA '00*, volume 1.

Brooks, R. (1991). Intelligence without reason. In *Proc. Int. Joint Conf.on Artificial Intelligence*, pages 569–595.

Hobbs, A. (1989). Computing edge-toughness and fractional arboricity. In *Contemporary Mathematics*, volume 4, pages 89–106.

Kotay, K. and Rus, D. (1999). Locomotion versatility through self-reconfiguration. *Jour. of Robotics & Autonomous Systems*, 26:217–232.

Lipson, H. and Pollack, J. (2000). Automatic design and manufacture of robotic lifeforms. *Nature*, 406:974–978.

Melhuish, C. (1999). Employing secondary swarming with small scale robots: a biologically inspired collective approach. In *Proc. of the 2nd Int.Conf. on Climbing & Walking Robots CLAWAR*.

Melhuish, C., Holland, O., and Hoddell, S. (1998). Collective sorting and segregation in robots with minimal sensing. In *From Animals to Animat*, volume 5, pages 465–470. MIT Press.

Pfeifer, R. and Scheier, C. (1999). *Understanding Intelligence*. MIT Press.

Shapiro, J. (1988). Bacteria as a multicellular organism. *Scientific American*, June 1988:62–69.

Sims, K. (1994). Evolving 3d morphology and behavior by competition. *Artificial Life VI*, pages 28–39.

Støy, K. (2001). Using situated communication in distributed autonomous mobile robotics. In *7th Scandinavian Conf. on AI*.

Støy, K., Shen, W.-M., and Will, P. (2002). Global locomotion from local interaction in self-reconfigurable robots. In *7th Int.Conf. on Intelligent & Autonomous Systems (IAS7)*.

Takahashi, N., Nagai, T., Yokoi, H., and Kakazu, Y. (2001). Control system of flexible structure multicell robot using amoeboid self-organisation mode. In *Proc.6th Europ.Conf. on Artificial Life, ECAL 2001*, pages 563–572. Springer.

Wessnitzer, J., Adamatzky, A., and Melhuish, C. (2001). Towards self-organising structure formations: A decentralised approach. In *Proceedings of ECAL 2001*, pages 573–581. Springer.

Winfield, A. (2000). Distributed sensing and data collection via broken ad hoc wireless connected networks of mobile robots. In *Distributed Autonomous Robotic Systems*, volume IV, pages 273–282.

Winfield, A. and Holland, O. (2000). The application of wireless local area network technology to the control of mobile robots. *Microprocessors and Microsystems*, 23:597–607.

Yoshida, E., Murata, S., Kokaji, S., Tomita, K., and Kurokawa, H. (2000). Micro self-reconfigurable robotic systems using shape memory alloy. In *Distributed Autonomous Robotic System*, volume IV, pages 145–154.