Flexible Multi-Robot Formation Control: Partial Formations as Physical Data Structures

Michael de Denus and John Anderson and Jacky Baltes

Department of Computer Science, University of Manitoba, Winnipeg, Manitoba, R3T 2N2 Canada

Email: mdedenus,andersj,jacky@cs.umanitoba.ca

Abstract

Formations are often seen in nature, and bring many benefits for the group as a whole. They can allow a group to explore a large area more effectively, can ease movement of the group through the environment, and can increase group perceptual coverage and increase defensive capabilities, for example. The benefits of any particular formation vary and are obtained from the structure the formation provides. Robotic formations can have similar applications. To date, the techniques used and formations employed in robotic applications are significantly simpler than those seen in nature. Current techniques often require some level of global knowledge, central processing or other unrealistic assumptions. We seek to develop a formation control technique that has as few of these limitations as possible. Each agent under our approach has only local knowledge of the environment, uses no broadcast communication, and can communicate only over a limited range. Formations are achieved by organizing agents into a graph structure, where agents occupying the vertices take on the role of maintaining an appropriate number of agents on each edge, thus preserving the formation's shape and scale. We do not assume a known or static population: the evolving formation acts as a physical data structure to assist in placing and rearranging agents as the population changes. This approach does not require a global coordinate system, fixed positions within the formation, or any single lead agent. All agents within our approach are peers, and any can adopt any role within the formation.

Introduction

There are a number of varied ways in which agents and robots can take advantage of knowledge or data structured deliberately in the world as opposed to an internal representation. In modern AI research, the concept was brought into focus by Agre, Chapman, and others (Agre 1988; Chapman 1990) in the late 1980s and early 1990s, through their emphasis on agents designed to respond to and structure the world in lieu of employing the elaborate internal world models emphasized by earlier approaches. The deictic representations they advocated, in conjunction with work by roboticists such as Brooks, who equally adamantly emphasized perception over the need for elaborate world models (Brooks 1990), led to perceptually-oriented, behaviour-based approaches that are now common to modern AI systems.

Modern multi-agent systems research has seen a number of approaches that use knowledge that is either deliberately planted in the world or results from a side-effect of agent activity in order to avoid communication and increase efficiency in groups. The most obvious of these are stigmergic approaches, where perceivable elements are deliberately left in the physical world (e.g. (Wurr and Anderson 2006)) or in a shared virtual space (Vaughan et al. 2000) as knowledge for other agents. More subtly, similar knowledge can be left (possibly deliberately) through the performance of actions not themselves intended as communication. In early multi-robot experiments, for example, Balch notes that explicit communication does not speed up performance in a grazing task, because the act of grazing leaves a perceivable trail that should cause others performing the same task to turn their attention elsewhere and thus avoid redundant work (Balch and Arkin 1994). Similar effects can also be seen in multi-robot exploration and other area coverage tasks: the fact that another robot is in view can be used to repel other agents, producing dispersion and hopefully less redundant work (Pearce et al. 2003). Positive results from simply observing the actions of other agents can also be seen in search and gathering applications (Rybski et al. 2004)

There are also a number of coordination schemes that involve using robots themselves to directly act as structured knowledge in the world. A robot can be left as a physically immobile landmark, at one extreme, in much the same way that one would drop a marker of any sort, but with the added benefit that the landmark has the ability to do perceive and communicate. In the distributed sensor network of (Howard, Matarić, and Sukhatme 2002; Parker and Howard 2006), for example, robots are marked with fiducials, allowing their presence to act as an identifiable landmark for localization or other purposes. Other approaches further use such agents to convey information and perform active computation and communications relay as well (e.g. (Sauter et al. 2002)).

In terms of multi-robot systems, one of the most important situations where robots themselves can form an active data structure in the environment is formation-controlled movement. There is a broad range of advantages to using formations, from protecting the bulk of a body of agents from outside harm through a limited perimeter, to increasing the perceptual coverage of a collective, to being able to make fast movement decisions with less worry about collisions between agents. These benefits come from the structure provided by the formation, and thus vary depending on the formation chosen. In creating and maintaining the formation, the partly-structured formation itself should be able to serve as a physical data structure, simplifying the choices of agents joining the formation and increasing the efficiency and effectiveness of the formation process.

Maintenance is an important part of employing a formation, especially in a multi-robot environment. Breakdowns are far more common in robots than in natural creatures, for example, so it should be expected that robots will naturally leave a formation in this manner (thereby decreasing the amount of data in the physical data structure), in addition to doing so voluntarily. Moreover, it is common to lose robots from the group as navigation progresses, especially in situations where some robots are sensor-poor (Howard, Matarić, and Sukhatme 2002; Parker and Howard 2006), or where the environment is particularly challenging. Because of these losses, formations are required to be adaptable to new agents as well: valuable domains where robot losses would be likely (e.g. search and rescue) would require replacements to be released into the environment, hopefully to join existing teams. Such incremental deployment is also a natural consequence in some multi-robot problems, in that units may not all be available at the same time (e.g. belonging to different organizations responding to a disaster). Any formation management technique should allow for a freely-changing population, with minimal disruption to the formation itself.

There are a number of other important criteria for an approach to creating and managing formations in multi-robot systems. Current techniques often require some level of global knowledge that may not be possible given a changing population, or rely on central processing unique to one member of the formation (making the approach vulnerable to failure because of reliance on one agent) or continual offline processing. Approaches may also require broadcast communication, which may not be desirable or even possible in some domains. Such assumptions may work in isolated situations, but prevent such approaches from being broadly applicable in complex, challenging domains.

There are also restrictions in some approaches based on assumed roles for agents. Given that a formation is a physical arrangement of agents, it requires some sort of reference point by which to define the arrangement. This reference point could be a lead agent (Dierks and Jagannathan 2009; Shin et al. 2007), a neighbouring agent (Fredslund and Mataric 2002; Lee and Chong 2007), or an external reference point (Balch and Arkin 1998). Requiring a lead agent in particular introduces a strong point of vulnerability and require additional processing to decide which of many possible agents becomes the leader, how to recognize when a leader is failing, and replacing a leader when this happens. In being able to build a formation quickly and with as little communication as possible, it is desirable to keep the number of required references to a minimum, while still reducing the amount of global knowledge required.

The work described in this paper involves the development of a formation control technique which avoids many of the limiting assumptions used in other research. In our approach, each agent has only local knowledge of the environment. Agents can also communicate only directly to another agent - there is no broadcast communication - and communication itself is assumed to have a limited distance. Formations are achieved by organizing agents into a graph structure, where agents occupying the vertices take on the role of maintaining an appropriate number of the agents on each edge, thus preserving the formation's shape and scale. We do not assume a known or static population: the evolving formation acts as a physical data structure to assist in placing and rearranging agents as the population changes. References in the definition of a formation are always defined in a relative manner, and never tied to an individual agent or fixed point. This means that our approach does not require a global coordinate system, fixed positions within the formation, or any single lead agent. All agents within our approach are peers, and any can adopt any role within the formation.

The remainder of this paper describes the approach, its implementation, and illustrations of the implemented approach in operation. Before this, we relate the approach to additional prior work.

Related Work

In previous work, (Balch and Arkin 1998) describe a system for controlling a group of robots in formation. This technique relies on knowing the number of other robots, and their positions. Our technique will be more appropriate for use from a local perspective.

(Fredslund and Mataric 2002) describe a method for maintaining formation using only a neighbour as a reference point. The limitation of this approach is that the neighbour in question is fixed beforehand. Our method allows agents to vary the neighbour that they use as a reference point.

(Lee and Chong 2007) provide a method for applying simple local rules to create a formation. The limitation of this technique is that it has little variety in the number of formations that it can create. Our technique will allow for more flexibility in the types of possible formations.

(Dierks and Jagannathan 2009; Shin et al. 2007) discuss methods for formation control using a leader and multiple followers. These approaches are limited by the fact that the leader is a single point of failure. Our approach has no such single point of failure.

Formation Control Approach

In our approach, a formation is described using a graph data structure. Creating a formation definition assigns a physical meaning to the edges, vertices and weights in a graph.

A formation consists of a series of interconnected line segments, which will correspond to edges in the formation graph. Each point where two or more line segments meet will be represented by a vertex in the graph. In addition to attributes of edges (direction, cost) normally associated with a graph, our approach requires that two additional pieces of information be associated with each edge. The first is a distance function. This defines how far apart agents in this edge should be. The second is an angle function, specifying the desired relative angle between any two agents on the edge. The weight of an edge determines its length relative to other edges. Since we have no way of knowing how many agents will make up the formation, we use the ratio between the weights to determine how many agents should be on each edge (this is discussed in further detail below). The direction of the edge has a meaning as well. If an edge originates at a vertex, the agent occupying the vertex position will be behind the first agent in the edge. If an edge terminates at a vertex, the agent occupying the vertex position will be in front of the agents on the edge. This will be necessary later, when we discuss the creation of formations.

Our approach employs direct agent to agent communication. Communication between two agents is possible only if they are within a limited distance. This distance represents the transmission power of a communication device carried by the agent, given the conditions of the domain. It thus allows us to experiment using the approach on domains with highly restrictive communication (e.g. the interference in a military setting or rescue scenario). Each agent has a unique ID, enabling others to communicate with it. Our approach does not assume that we know each agent's ID ahead of time. Instead, we require only that the agent's ID can be sensed visually. This can be done by uniquely marking a robot (e.g. through barcodes or colored patterns). Less structured arrangements are also possible, where a robot could be communicated to through a description of its appearance, and when this matches, responding with an ID that could be recorded for further communication purposes. We are only working with the former to date.

Roles within the formation

There are two possible roles an agent can take on in the formation. The first role is that of *edge agent*. An edge agent is responsible for maintaining the proper relative distance and angle from its neighbour. It uses the desired distance and angle functions of its edge to determine how it should position relative to its neighbour. The more interesting role is the *vertex agent*. A vertex agent has several additional responsibilities. It is responsible for negotiating entry into the formation, counting members as they join and maintaining the current lengths of the different edges. Vertex agents are also responsible for passing on the details of the formation to any agents that join. Each agent has a library of known formations, but these can be expanded through communication.

Each vertex agent periodically checks the number of agents in each attached edge. It does this by sending out messages, which are passed along by each of the edge agents, incrementing a count with each hop. When one of these messages reaches a vertex agent, it sends the message back along the edge.

There are two situations which require an agent to assume the role of vertex agent. The first is when an agent has one or fewer neighbours. This indicates that it is on the end of an edge, and therefore must act as a vertex. The second situation is when an agent has neighbours in multiple edges. This can only happen if it is at the point where two edges join. This again must be a vertex.

Joining a Formation

The most obvious way an agent can join a formation is to discover it: that is, the formation is already in existence. In this case, the partial formation serves as data to allow an agent to intelligently join it. The agent locates the nearest agent in the formation, moves within communication range, and sends a *joining* request. The response to this message indicates the direction to the nearest vertex, given the local knowledge of the contacted agent. The joining agent then moves towards the vertex. When arriving at the vertex, the joining agent contacts the vertex agent with another joining request. The vertex agent examines the current lengths of each edge connected to it. It uses the edge weights in the graph to determine which edge is farthest below its desired length. It then selects this edge as the destination for the joining agent, and sends a message indicating this. The joining agent then moves into the specified edge.

If there is no established formation, an agent can create one as soon as it meets another agent. Upon receiving a join request message, an agent not in a formation will respond with a *not-in-formation* message. In this case, both agents join together to create a new formation. For the purposes of this research, they will simply select the first formation that is known to both of them. They locate the edge in the graph with the highest weight, and each acts as a vertex agent connected to that edge. The agent with the lower ID will begin the edge, the agent with the higher ID will terminate the edge.

Balancing a formation

As a formation evolves, it will at times become unbalanced. A formation may need a given number of agents to be balanced for example (e.g. a diamond), and will be unbalanced if two agents begin such a formation, by definition. There is also no guarantee that agents joining a formation will approach all vertices equally: by the nature of the environment, all agents may be joining a formation from one side, for example. For this reason, we have a balancing operation, used to redistribute agents across edges as needed.

The vertex agent first sums its stored counts of agents in each edge. It then uses the ratios between edge lengths to compute the desired number of agents per edge. Next, it looks at the difference between these desired values and the actual numbers of agents in each edge. The vertex agent then expresses the actual number of agents on the edge as a percentage of the desired number of agents for that edge. The edge with the highest percentage is chosen as the source, and the edge with the lowest percentage is chosen as the destination. If there is a tie for highest or lowest percentage, the tie is broken randomly. Finally, the vertex agent sends a message to the nearest agent in the source edge, instructing it to move to the destination edge.

Merging Two Formations

If a number of agents are released into an environment with the desire to create a formation and without the concept of a predefined leader, they will begin by starting formations with agents they encounter (described above). Single agents will then later join these formations (also described above), but these partially formed formations will also encounter one another in a similar way, requiring formations to be merged. When two existing formations meet, the two closest vertex agents in each formation communicate. Each vertex agent uses the lengths of all edges connected to them to estimate the size of their formation. They exchange these estimated formation sizes, and the smaller formation then joins the larger one. It is possible that multiple vertex agents within each formation begin this negotiation. In the case of a poorly balanced formation, this could cause both to disband. Our merging technique can recover from this situation.

When merging formations, the vertex agent in the joining formation sends out a message to its neighbours. This message is then propagated across the entire formation. This message tells agents to form a line behind their nearest neighbour. The agents then join the new formation at the specified destination vertex. Because a destination vertex agent is chosen, even if both formations collapse, each will have a destination, and they will eventually join together.

Implementation

Our agents are behaviour-based (Arkin 1998), and have four behaviours. The first of these is the goal seeking behaviour. In a real environment, a formation would need some way of agreeing on a direction in which to move. Our system simulates this by having a pre-defined global goal for agents.

The second behaviour is formation keeping. This behaviour ensures that agents maintain the proper distance and angle from their neighbours.

Next, we have obstacle avoidance. This generates a vector away from each obstacle and other agent. This behaviour only has an impact if an agent gets to within a certain distance of an obstacle.

Finally, we have direction hint following. Several interactions in the system have one agent telling another agent roughly where to go. This behaviour allows agents to act on these instructions.

Our intended implementation platform for this work is a mixed-reality approach using Citizen microrobots running on a 42” flat panel LCD display. This allows a sizeable population to be examined, and the mixed reality environment allows us to project obstacles and have virtual terrain (some areas slowing robots down) that is perceivable as real by the robots involved. This allows for greater consistency than would normally be possible in the physical world with small, light robots. At the time of this writing, the implementation was only complete for a Player/Stage (Gerkey et al. 2001) simulation of this final system. Within the simulation, we use simulated Pioneer 2DX robots. For sensing, we use a laser scanner with fiducial tracking to allow robots to be identifiable. Such sensors would be replaced with an overhead camera for global vision, and the use of coloured markers to perceive identity in the microrobot environment.

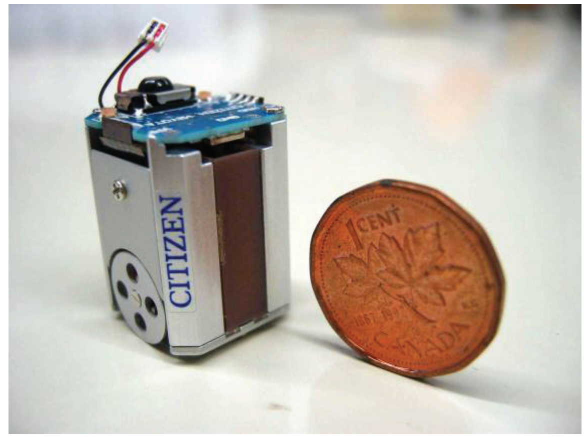

Figure 1: Citizen Eco-Be Microrobot (v.1).

Demonstration

In order to demonstrate our system at work, we use a V formation as an example. This is a very common formation, and helps to clearly demonstrate the operation of the system. Any figure that can be described as a collection of related segments (possibly direct graph edges, or curves approximated using multiple graph edges) is possible.

Figure demonstrates the process for a single agent joining an established formation, as described in the *Approach* section above. First, the agent senses the formation. It then sends its join request. Finally, it receives a reply, telling it which edge to join. It then acts on that information and joins the formation.

Figure demonstrates how the balancing operation works within the formation, as described in the *Approach* section above. We start with a group of agents arranged in such a way that they would be very likely to all join on a single edge. As expected, the agents do initially form a single edge. Then, the leading vertex agent performs a series of balancing operations. These balancing operations cause two of the edge agents to move to the neighboring edge. The final result is a correctly balanced formation.

While this demonstrates the functionality, we have not yet begun to examine its performance over time (e.g. rebalancing when a formation is disturbed by encountered obstacles or terrain that slows some but not all of the agents). We intend to show such performance evaluations at the workshop.

Discussion

We expect that this approach will scale well to large numbers of agents in the physical world. This will be demonstrated once the implementation is ported to the Citizen mixed-reality environment. However, the fact that the approach relies only on local perspectives and limited scope of communication suggests that our approach will perform well with large numbers or agents.

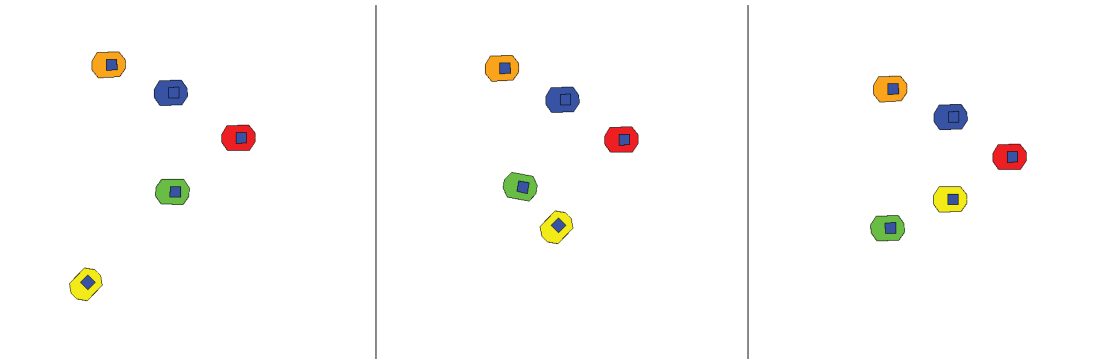

Figure 2: The joining procedure. In the first frame, we see an agent approaching an established formation. It communicates its intent to join the formation, waits for a response, then joins in the appropriate place. Finally, we see the formation with the new agent integrated.

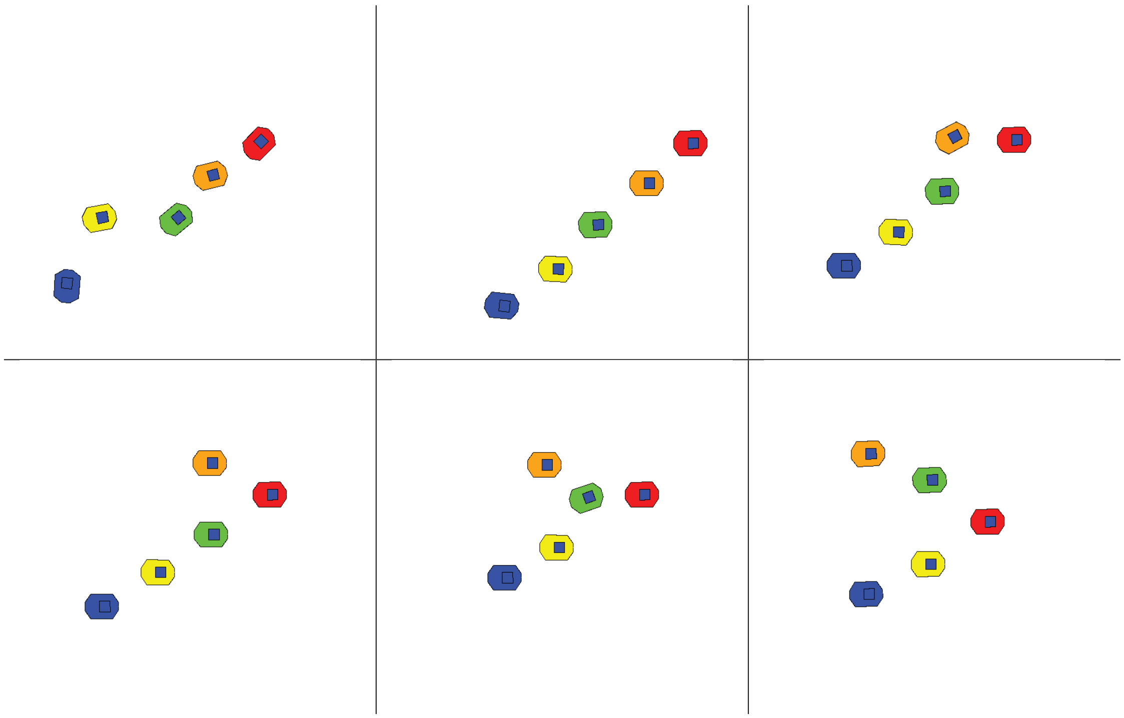

Figure 3: The balancing procedure. In the upper left panel, we see the starting configuration of agents. When the simulation starts, they initially assemble into the formation seen in the upper middle panel. Two balancing operations eventually lead to the formation in the lower right panel.

In the future, we plan to examine an additional method for merging existing formations, and compare it to that described in this paper. This method would have the smaller formation split off into groups, each consisting of an edge. These groups would then attempt to locate a position in the target formation corresponding to their position in the old formation. This would be more complex to implement, but could allow for faster merging of formations with a smaller number of balancing operations required afterwards.

References

Agre, P. E. 1988. *The Dynamic Structure of Everyday Life*. Ph.D. Dissertation, MIT, Cambridge, MA.

Arkin, R. C. 1998. *Behavior-based Robotics*. MIT Press, Cambridge.

Balch, T., and Arkin, R. C. 1994. Communication in reactive multiagent robotic systems. *Autonomous Robots* 1:27–52.

Balch, T., and Arkin, R. 1998. Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation* 14:926–939.

Brooks, R. A. 1990. Elephants don't play chess. *Robotics and Autonomous Systems* 6:3–15.

Chapman, D. 1990. *Vision, Instruction and Action*. Ph.D. Dissertation, MIT, Cambridge, MA.

Dierks, T., and Jagannathan, S. 2009. Neural network control of mobile robot formations using rise feedback. *IEEE Transactions on Systems, Man, and Cybernetics* 39(2):332–347.

Fredslund, J., and Mataric, M. 2002. A general algorithm for robot formations using local sensing and minimal communication. *Robotics and Automation, IEEE Transactions on* 18(5):837–846.

Gerkey, B.; Vaughan, R.; Stoy, K.; Howard, A.; Sukhatme, G.; and Mataric, M. 2001. Most valuable player: a robot device server for distributed control. In *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, 1226–1231.

Howard, A.; Matarić, M. J.; and Sukhatme, G. S. 2002. An incremental self-deployment algorithm for mobile sensor networks. *Auton. Robots* 13(2):113–126.

Lee, G., and Chong, N. Y. 2007. Adaptive self-configurable robot swarms based on local interactions. In *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4182–4187.

Parker, L. E., and Howard, A. 2006. Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection. *International Journal of Robotics Research* 25:431–447.

Pearce, J. L.; Rybski, P. E.; Stoeter, S. A.; and Papanikolopoulos, N. 2003. Dispersion behaviors for a team of multiple miniature robots. In *In IEEE International Conference on Robotics and Automation*, 1158–1163.

Rybski, P. E.; Larson, A.; Veeraraghavan, H.; Lapoint, M.; and Gini, M. 2004. Communication strategies in multi-robot search and retrieval: Experiences with mindart. In *Proc. 7th Int. Symp. Distributed Autonomous Robotic Systems*, 301–310.

Sauter, J. A.; Matthews, R.; Van Dyke Parunak, H.; and Brueckner, S. 2002. Evolving adaptive pheromone path planning mechanisms. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, 434–440. New York, NY, USA: ACM.

Shin, J.; Kim, H. J.; Kim, S.; and Yoon, Y. 2007. Formation flight control under communication failure. In *RoboComm '07: Proceedings of the 1st international conference on Robot communication and coordination*, 1–4. Piscataway, NJ, USA: IEEE Press.

Vaughan, R. T.; Støy, K.; Sukhatme, G. S.; and Matarić, M. J. 2000. Whistling in the dark: cooperative trail following in uncertain localization space. In *AGENTS '00: Proceedings of the fourth international conference on Autonomous agents*, 187–194. New York, NY, USA: ACM.

Wurr, A., and Anderson, J. 2006. Stigmergic navigation for multi-agent teams in complex environments. In Ajith Abraham, C. G., and Ramos, V., eds., *Stigmergic Optimization*. Berlin: Springer-Verlag. chapter 4, 85–116.