Swarming Methods for Geospatial Reasoning

H. Van Dyke Parunak, Sven A. Brueckner, Robert Matthews, John Sauter

Altarum Institute, 3520 Green Court, Suite 300,
Ann Arbor, MI 48105-1579
Tel: +1 734-302-4600
FAX: +1 734-302-4991
Email: {van.parunak, sven.brueckner, robert.matthews, john.sauter}@altarum.org

Abstract

Geospatial data is often used to predict or recommend movements of robots, people, or animals ("walkers"). Analysis of such systems can be combinatorially explosive. Each decision that a walker makes generates a new set of possible future decisions, and the tree of possible futures grows exponentially. Complete enumeration of alternatives is out of the question. One approach that we have found promising is to instantiate a large population of simple computer agents that explore possible paths through the landscape. The aggregate behavior of this swarm of agents is a useful estimator for the likely behavior of the real-world system. This paper will discuss techniques that we have found useful in swarming geospatial reasoning, illustrate their behavior in specific cases, and discuss the convergence and application of such systems.

1. Introduction

Geospatial data is often used to predict or recommend movements of robots, people, or animals ("walkers"). Ecologists may be interested in how changes to the landscape will affect migratory patterns. Emergency preparedness workers may need to evaluate the flow of evacuees through a road network. Security personnel may want to track the likely location of suspected terrorists. Roboticists may want autonomous vehicles to find their way through a landscape populated by threats and targets. In each case, one or more walkers move in response to stimuli, constrained by the topology and topography of their environment.

Analysis of such systems faces the challenge of combinatorial explosion. Each decision that a walker makes generates a new set of possible future decisions, and the tree of possible futures explodes exponentially rapidly. Complete enumeration of alternatives is out of the question.

One approach that we have found promising is to instantiate a large population of simple computer agents that explore possible paths through the landscape. The aggregate behavior of this swarm of agents is a useful estimator for the likely behavior of the real-world system. Such a system can *predict* the behavior of walkers that move autonomously (e.g., for studies of animal migration or the movement of military troops), or *plan* movements that walkers execute under external control (e.g., for path planning for robots or search teams). Predictive and planning can be combined to deal with systems in which one population of walkers must anticipate and respond to the movements of another.

This paper introduces these swarming methods for geospatial reasoning. Section 2 outlines several techniques that we have found useful in such systems. Section 3 gives concrete examples of three such systems, two for planning and one for prediction, and shows how the two could be combined. Section 4 discusses some practical issues in deploying such systems, including speed of convergence, support for the computational environment on which our methods rely in distributed robotics, and ways to exploit the information generated by these methods in predictive systems. Section 5 concludes.

2. Modeling Techniques

Swarming geospatial reasoning is a variety of agent-based modeling. We characterize agent-based models in contrast with other computational models, and then distinguish swarming models from other agent-based models.

2.1. Agent-Based vs. Equation-Based Modeling

At some risk of oversimplification, computational models can be divided into two broad classes. In agent-based modeling (ABM), the model consists of a set of agents that encapsulate the behaviors of the various individuals that make up the system, and execution consists of emulating these behaviors. In equation-based modeling (EBM), the model is a set of equations, and execution consists of evaluating them.

Equation-based models, such as systems of differential equations, were the only practical form of mathematical model in the days before computers. They are quite mature (Sterman, 2000), and can be executed extremely rapidly using numerical integration. Agent-based models have become popular only with the advent of inexpensive computers. They offer significant benefits over equation-based models in respect to the underlying structure of a model, the naturalness of its representation of a system, and the verisimilitude of a straightforward representation (Parunak et al., 1998), and have rapidly grown in popularity in many fields, including geographical information systems (Gimblett, 2002).

2.2. Distinctives of Swarming Agent-Based Modeling

The most straightforward application of agent-based modeling to a geographical scenario would be to assign one agent to each entity, model the agent's behavior on the entity's as closely as possible, then execute the set of agents and observe their behavior.

In a swarming approach, the agents that explore the landscape differ from real agents (and from a naïve agent representation) in several ways. Swarming methods are inspired by mechanisms exhibited by social animals, notably insects (Parunak, 1997), and these mechanisms often rely on characteristics that differ from those associated with real-world people or robots. These differences include the number of walkers, their internal logic, stochasticity, and stigmergic information exchange.

Number of Walkers.—In a conventional multi-agent model, agents are in one-to-one correspondence with physical entities in the real world. Swarming systems achieve self-organization through the repeated interactions of many agents, and if the population of real agents is too small, a one-to-one correspondence will not yield the required dynamics. Thus each physical agent may correspond to many computational agents. In some cases, we may even instantiate computational agents that do not correspond to any specific physical agent.

Using a many-to-one representation has another benefit in addition to enabling the dynamics of self-organization. Representing a single entity by a population of agents is analogous to representing a single particle by a wave function. We are shifting our focus from the unique behaviour of one individual to a collection of behaviours that sample the space of possible actions. It is sometimes helpful to interpret their movements as concurrent Monte Carlo. The resulting distribution of swarming agents can then serve as an estimate of the probability function of real walker distribution over the landscape.

Walkers' Internal Logic.—The gold standard in conventional ABM is accurately modeling the internal logic of each real-world entity. When those entities are humans, each agent becomes an independent artificial intelligence, most commonly modeled in terms of its beliefs, desires, and intentions (Haddadi and Sundermeyer, 1996; Müller, 1996; Rao and Georgeff, 1991). The development of the complex symbolic knowledge bases needed to support such "BDI" agents is an instance of the knowledge acquisition problem, which has long plagued the development of realistic AI applications.

The constraints imposed on the agents by the environment (including one another) often outweigh the effect of different decision algorithms. This phenomenon is analogous to the emergence of identical critical exponents in widely different physical substances at their critical points in statistical physics, and borrowing from the physics vocabulary, we term it "universality" (Parunak et al., 2004b). As in physics, so in multi-agent systems we do not fully understand what makes universality happen (or even how to predict when it will or will not apply). But knowing that it can happen encourages us to begin modeling with very simple rules rather than with the complete decision logic of a real-world agent. We focus on simple environmental clues ("prefer the path with the lowest gradient") and tropisms ("head in a general southerly direction"), and then enhance the agent sophistication only as long as it improves the performance of the system. The "brain" of an agent is not a knowledge base, but a simpler (usually quantitative) structure such as a neural network or a polynomial. These structures can be tuned using synthetic evolution (Parunak, 2005; Sauter et al., 2002), which is a much more efficient process than the knowledge acquisition required for BDI agents.

Stochasticity.—Rational agents are typically deterministic, driven by computations that are modeled on theorem proving or optimization theory. For example, an agent will typically have an objective function that it seeks to maximize. One consequence of this approach is that agents with identical state will make identical decisions, and the system will quickly fall into a stagnant state. To overcome such symmetries, one must invest in detailed knowledge engineering to capture the distinctions that always exist among real-world agents.

A simpler way to break symmetries is to have agents chose stochastically from among alternative behaviors. Typically, we use a using a Boltzmann-Gibbs function. For example, at a point in its evolution, an agent might have n possible choices, each with perceived value v_i. Instead of making the choice with the highest value, the agent chooses among them, assigning each the probability

$$p_i = \frac{e^{v_i / T}}{\sum_j e^{v_j / T}} \tag{1}$$

In this equation, T is a temperature parameter that determines the degree of stochasticity in the decision. When T is small, the agent chooses the highest-valued option almost deterministically. When T is large, the choice becomes almost equal among the alternatives. This approach, inspired by simulated annealing (Kirkpatrick et al., 1983), breaks symmetries among the agents without expensive knowledge engineering and avoids local minima. We have developed local measures that agents can use to estimate the degree of convergence of the system and thus adjust T dynamically as the model runs (Brueckner and Parunak, 2003).

Stigmergic Information Exchange.—Conventional agents interact primarily by sending messages to one another. These messages are typically symbolic, with a grammar based on standards such as KQML/KIF (Finin, 1997) or FIPA (FIPA, 2000). While the messages necessarily pass through a communication infrastructure such as the internet, the agents are not aware of this infrastructure, and behave as though they had direct telepathic capabilities.

Studies of insect behavior have revealed the importance of indirect communication, mediated by a shared environment. The French entomologist Grassé (Grassé, 1959) coined the term "stigmergy" from the Greek words *stigma* "sign" and *ergon* "action," to capture the notion that an agent's actions leave signs in the environment, signs that it and other agents sense and that determine their subsequent actions. A common form of stigmergy, and one that we exploit heavily, is the use by ants of chemical markers (pheromones) that they deposit and sense (Brueckner, 2000). Stigmergic interaction has several benefits over message-based interaction, including simplicity, scalability, robustness, and the ability to take advantage of environmental noise to support the need for stochasticity in decision making (Parunak, 2003; Parunak and Brueckner, 2004).

The use of a distinct environment as the primary medium of interaction avoids computational paradoxes that can arise from direct agent-to-agent communication (Michel, 2004). It is one thing for an agent to intend to change the world, but quite another for the change to succeed. Classical agent models often assume unrealistically that an agent's actions achieve their intended purpose (Ferber and Müller, 1996). The environment provides a computational locus where the actions of different agents can be integrated and arbitrated, just as the laws of physics do in the real world.

3. Experiments

We describe three different systems that use swarming agents to do geospatial reasoning. The first two, planning applications intended for robotic units, illustrate how swarming agents can develop paths that balance the influence of environmental threats and targets. The third, a predictive application, illustrates reasoning about more detailed topographical information.

3.1. Path Planning with Threats and Targets

Consider an unmanned air vehicle (UAV) that must find its way around a network of surface-to-air missiles in order to reach a target. Figure 1 shows one possible configuration, in which a gauntlet of threats guards access to the target.

A common mechanism for path planning in this kind of problem is to define a loss function at each point in space on the basis of proximity to threats and targets, integrate it to generate a potential field, and then climb the gradient of the potential field (Rimon and Kodischek, 1992). In addition to requiring centralized computation, such methods have difficulty solving configurations such as Figure 1. The field can easily achieve a local maximum outside of the gauntlet, trapping the hill-climbing search prematurely. In one experiment, researchers could only solve this configuration with standard potential methods by first manually defining a waypoint at the entrance to the gauntlet, and then planning separately the segments of the path from the base to the waypoint and then from the waypoint to the target.

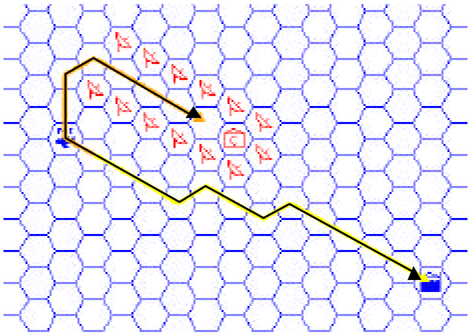

Figure 1: Path planning with threats and targets

Our architecture for solving this problem uses digital pheromones, and has four components.

- A distributed network of *place agents* maintains the pheromone field and performs aggregation, evaporation, and diffusion. Each place agent is responsible for a region of the physical space. In our simulation, we tile the physical space with hexagons, each represented by a place agent with six neighbors but in principal both regular and irregular tiling schemes can be employed. Place agents ideally are situated physically in the environment using unattended ground sensors distributed over an area and connected to nearest neighbors through a wireless network. They may also be located in a distributed network of command and control nodes.
- *Avatars* represent physical entities. Red avatars represent enemy targets and threats, while blue avatars represent friendly UAVs. Blue avatars are normally located on the robot. The name "Avatar" refers to the incarnation of a Hindu deity, and by extension describes a temporary manifestation (a software agent) of a persistent entity.
- Blue avatars continuously emit *Ghost agents* that wander over the place agents looking for targets and then continually building a path from the avatar to the target. The avatars and ghosts all deposit pheromones at their current location.
- Different classes of agents deposit distinct *pheromone flavors*. Agents can sense pheromones in the place agent in whose sector they reside as well as the neighboring place agents. (Brueckner, 2000) develops the underlying mathematics of the pheromone field, including critical stability theorems.

Battlefield intelligence from sensors and reconnaissance activities causes the instantiation of Red avatars representing known targets and threats. These agents deposit pheromones on the places representing their location in the battlespace. The field they generate is dynamic since targets and threats can move, new ones can be identified, or old ones can disappear or be destroyed. A blue avatar representing a UAV is associated with one place agent at any given time. It follows the pheromone path created by its ghost agents.

Ghosts initially wander through the network of place agents, attracted to pheromones deposited by targets and repelled by threat pheromones. Once they find a target, they return over the network of place agents to the walker depositing pheromones that contribute to building the shortest, safest path to the target. The basic pheromone flavors are *RTarget* (deposited by a Red target avatar, such as the Red headquarters), *RThreat* (deposited by a Red threat avatar, such as an air defense installation), *GTarget* (deposited by a ghost that has encountered a target and is returning to its blue avatar, forming the path to the target), and *GNest* (deposited by a gho st that has left the blue avatar and is seeking a target).

A ghost agent chooses its next sector stochastically by spinning a roulette wheel with six weighted segments (one for each of its six neighbors). The size of each segment is a function of the strength of the pheromones and is designed to guide the ghost according to the algorithm above. We experimented with several different forms of the function that generates the segment sizes. Manual manipulation yielded the current form (for outbound ghosts):

$$F_n = \frac{q \cdot RT\,\mathrm{arg}\,et_n + g \cdot GT\,\mathrm{arg}\,et_n + b}{(r \cdot GNest_n + b)(Dist_n + j)^{d + a(RThreat_n + 1)} + b} \qquad (2)$$

F_n is the resultant attractive force exerted by neighbor n and *Dist* is the distance to the target if it is known. Table 1 lists the tunable parameters in the equation and the effect that increasing the parameter has on the ghost's behavior.

Table 1. Tunable Parameters and their Effects on Ghosts

Parameter	Effect on Ghost
α	Increases threat avoidance farther from the target
δ	Increases probability of ghosts moving towards a known target in the absence of *RTarget* pheromone
φ	Increases threat avoidance near target
ρ	Increases ghost exploration (by avoiding *GhostNest* pheromone)
θ	Increases attraction to *RTarget* pheromone
β	Avoids divison by zero

Though this table provides general guidance to the practitioner, in practice, the emergent dynamics of the interaction of ghost agents with their environment makes it impossible to predict the behavior of the ghosts. Thus tuning the parameters of this or any pheromone equation becomes a daunting task. We use synthetic evolution to adjust these parameters in real time, as the system is operating (Parunak, 2005; Sauter et al., 2002). As the avatar emits new ghosts, it breeds them from the fittest ghosts that have already returned. Fitness takes into account three characteristics of those ghosts:

1. Ghosts have a fixed lifetime. Ghosts that complete their search faster have longer to breed, and generate more offspring. Thus we favor ghosts that found shorter paths.
2. Ghosts encounter threats during their search. We favor ghosts that found safer paths.
3. Targets differ in value. We favor ghosts that found more valuable targets.

This system is extremely robust and adaptable (Parunak et al., 2004a), and has been deployed on physical robots (Sauter et al., 2005). It can solve the scenario of Figure 1 (among many others). It outperforms the classical potential field algorithm due to the large number of ghosts and the stochastic element of their movements. Some of them stumble into the gauntlet purely by chance. Once they find the target, the trail pheromones they deposit raise the probability that other ghosts will follow them, leading to emergent paths.

3.2. Area Surveillance

A common task for uninhabited robotic vehicles is surveillance of a region of territory. Such surveillance must satisfy several characteristics. In this example we focus on one: the vehicles should spread out over the area to avoid double coverage and reduce the time needed to cover the entire area. A convenient metric for such a system is how rapidly the agents initially cover the territory that they must monitor, tracking the fraction of the area that has been seen as a function of time.

A simple algorithm for this problem (Sauter et al., 2005) uses digital pheromones (Parunak et al., 2004a; Parunak et al., 2002a; Parunak et al., 2002b). Unlike our other examples, each physical entity corresponds to only a single agent. The pheromone infrastructure represents the environment as a square grid, each cell of which has a place agent.

1. Once a second, each place agent deposits twenty units of attractive pheromone in its cell, propagates pheromone to the eight neighboring cells, and evaporates the pheromone by a fixed proportion.
2. Every time a vehicle enters a new cell (on average, once every 4.8 seconds), it deposits two units of repulsive pheromone and zeros out the attractive pheromone in its current cell.
3. Once every twelve seconds, each place evaporates (but does not propagate) its repulsive pheromone by a fixed proportion.
4. Each vehicle moves to the neighboring cell for which difference (attractive pheromone – repulsive pheromone) is greatest.

Agents' decisions use only the information available in their immediate vicinity, and thus are local (though the propagation of attractive pheromone in step 1 provides some spread of information over time). In the absence of a vehicle, step 1 leads to an asymptotically constant level of attractive pheromone in each cell, drawing in vehicles. Step 2 causes the vehicles to spread out from one another, and avoids repeat visits that are close to one another. Because step 1 repeats after step 2, and because the repulsive pheromone from step 2 evaporates over time, eventually each site will be revisited.

3.3. Topographical Reasoning

Our third application illustrates the use of swarming geospatial reasoning to *predict* the movements of entities that we do not control In this example, insurgents are fleeing southward from friendly forces through a complex mountainous terrain. The task is to identify where in this terrain they are most likely to pass, so that surveillance assets can be deployed to detect and intercept them. Unlike the previous two examples, this example does not use digital pheromones, but relies entirely on exogenous environmental variables (direction and gradient).

The sequence in Figure 2 illustrates how swarming mechanisms can address this problem. The shading of the terrain indicates the steepness of the terrain. White terrain is level, while the steeper regions are successively darker shades of green, and the patch of red near the center bottom is the steepest area.

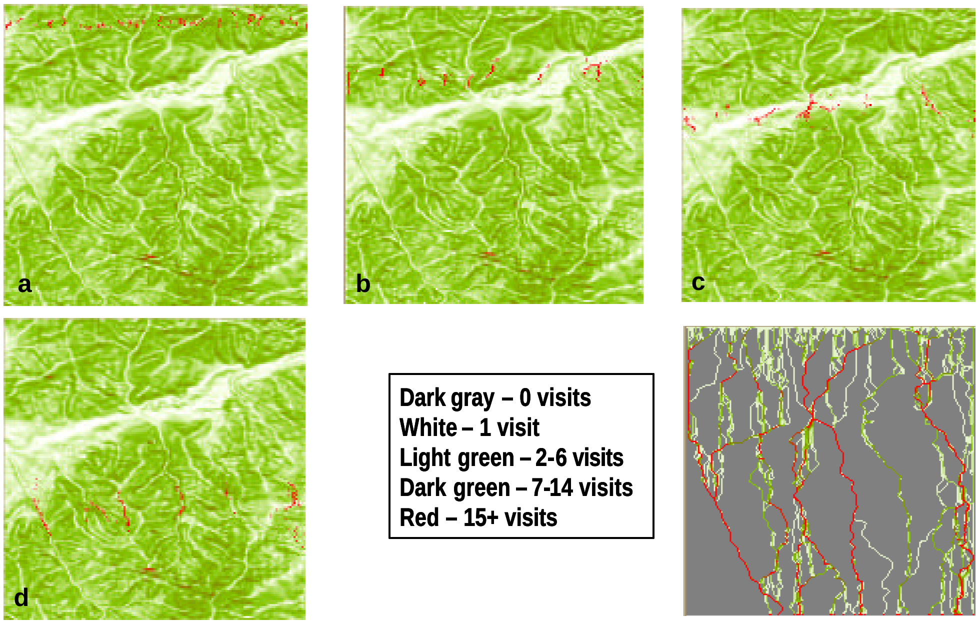

Figure 2: Predicting Movements in a Complex Topography

The area under study is divided into a square lattice (200 x 200), and the gradient in each cell is computed on the basis of the elevations in its Moore neighborhood (the eight adjacent cells). In addition, we compute a spatially smoothed gradient for each cell based on 5 x 5 Moore neighborhoods.

We begin with a uniform distribution of 200 Red agents across the width of the battlespace along the northern edge, one per cell. At each time step, an agent assigns a weight to each of the cells in its 3 x 3 Moore neighborhood, as the product of three values:

1. the inverse of the cell's gradient, normalized to sum to 1 over the neighborhood;
2. the inverse of the cell's smoothed gradient, again normalized to 1;
3. a directional weight to encourage southward movement: from NW to W, clockwise, these weights are (0, 0, 0, 0.13, 0.23, 0.28, 0.23, 0.13).

The agent's current cell and the northern cell are omitted, the products renormalized, and then the agent selects from among the five eligible directions using the Boltzmann-Gibbs distribution. In these experiments, we used a temperature of 0.01.

After only a short time (Figure 2a), the result of the gradient information and the southward pressure guides the agents into clusters. By the time they reach the southern third of the territory (Figure 2d), all 200 agents have merged into six groups. The final (lower right) plot shows the history of their distribution, emphasizing how the initial broad distribution rapidly narrows into only a few likely tracks.

A swarming model such as this can define areas of interest to focus a surveillance activity. These swarming Red agents could deposit the attractive pheromones used to route surveillance assets to the regions through which Red forces are most likely to pass.

A slight modification of this algorithm can be used to model an intelligent Red adversary that is familiar with the terrain. The evolutionary mechanisms used in UAV path planning could also be usd to evolve individuals that are able to make it from the North to the South in the shortest time with the least likelihood of being detected by blue. The movement of these evolved red units would be tracked by pheromones that would indicate the likely paths that a more knowledgeable adversary might take through this same region. These same pheromone paths can then be used to attract blue surveillance units, with the stronger, more probable paths attracting greater surveillance. Thus a swarm could be automatically perform highly directed surveillance activities against a knowledgeable enemy anywhere in the world without having to explicitly direct the units where to survey and how often they should monitor the different possible paths or choke points.

4. Discussion

In this section we discuss how rapidly these mechanisms converge, and how the results of such analysis can be used.

4.1. Convergence Speed

A major justification of the swarming approach is its potential for overcoming the combinatorial complexity of classical methods. However, the use of stochastic decisions raises the question of how rapidly a swarming system can itself converge.

We have developed a general model for the convergence speed of swarming systems (Parunak et al., 2005) based on an extension of the adaptive walk model (Kauffman and Levin., 1987). Consider a binary vector $S \in \{0,1\}^N$ of length N. Initially, all elements of S are 0. The objective of this system is to maximize $N_1 = \Sigma S$, the number of elements of S that are set to 1.At each time step, the adaptive walk takes the following actions:

- Select an element of S at random.
- If the element is currently 0, set it to 1 with probability p_{01}. If it is currently 1, set it to 0 with probability p_{10}. Note that p_{01} and p_{10} are independent. In particular, there is no requirement that they sum to 0.

Analysis of the master equation for this system shows that

$$N_1 = p_{01}\left(1 - e^{-lt}\right)/\,l \qquad (3)$$

where

$$l \equiv \left(p_{10} + p_{01}\right)/N \tag{4}$$

Though simple, this model has the essential features shared by many more realistic systems.

- Each element of S is an agent, and the array corresponds to the entire system of agents.
- The system objective is global over the entire system.
- The agents do not have access to this global measure in making their decisions. In fact, in this simple model, they do not consider the state of any other agent in making their decisions, but choose probabilistically. p_{01} reflects the probability that their local decision will advance the global goal, while p_{10} reflects the likelihood that their local decision will oppose the overall system objectives.

In spite of its simplicity, this model can be used to analyze the convergence of real systems. To illustrate, we apply it to the system described in Section 3.2, in a case where 15 vehicles are responsible for maintaining surveillance of an area 200 cells square. Figure 3 shows three plots of coverage as a function of time derived from this system: an upper bound, the actual observed performance, and the estimate given by our theory. The time step in all cases is 4.8 seconds (the average time it takes a vehicle to move from one cell to another).

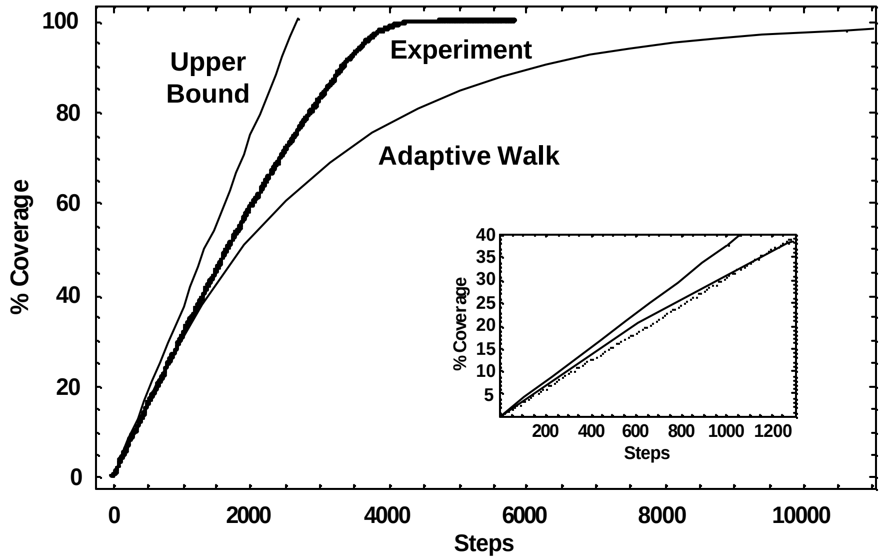

Figure 3: Comparison of Adaptive Walk with Pheromone-Guided Area Surveillance

The upper bound is given by the observation that for the fastest possible coverage, at each time step, each vehicle should move immediately to a cell that has not yet been visited. Such a strategy is physically impossible, because it would sometimes require vehicles to move directly between noncontiguous cells. But it provides an upper limit, visiting 15 new cells or 15/40000 = 0.0375% on each time step.

The experimentally observed convergence is nearly linear until the coverage is almost saturated, well over 95%. A linear fit to the region before the elb ow yields a slope of 0.030.

To derive the plot representing our model, observe that 15 vehicles are sampling cells of a 40000-cell area. In terms of the model, this is comparable to |*S*| = 40000/15 ~ 2667. For $p_{01} = 1$, $p_{10} = 0$, we have λ = 3.75E-4 (the same as the slope of the upper bound, and $N_1 = 2667\left(1 - e^{-3.75E-4t}\right)$.

In spite of its simplicity, the model provides an excellent fit to the experimental data up to about 40% coverage. The inset shows how the model rises slightly faster than the experiment, then begins to fall below it. We attribute the rapid rise to the assumptions of the model: while the air vehicles are constrained by the need to move between contiguous cells, and so must often repeat coverage of cells that have already been seen, the model can go directly to any cell. Of more interest is the shortfall above 40% coverage. The adaptive walk slows as more and more of the area is covered, so that the random selection of the next site to visit frequently selects a site that has already been visited.

The continued straight-line progress of the experiment shows the effectiveness of the pheromone mechanism in improving over the random selection of the next site to visit. This improvement arises because pheromones reduce the locality of the decision process, in two ways. First, the propagation of attractive pheromone makes information from one cell available in a neighboring cell, reducing spatial locality and generating a gradient that guides the movement of vehicles. Second, the persistence of pheromone deposited by one vehicle for sensing by another reduces the temporal locality of decisions, enabling decisions at one point in time to take into account the results of previous decisions.

From an engineering point of view, this example illustrates how the adaptive walk model can provide a lower bound for estimating the achievable performance of a real system, and for measuring the effectiveness of mechanisms for overcoming locality.

4.2. Use of Results

The examples we have presented illustrate two different ways in which swarming geocomputation can be deployed. In this section we discuss some of the issues involved in deploying these results.

The path planning and surveillance examples show how swarming can directly *control* physical hardware. Avatars controlling physical vehicles continuously manage a population of ghosts whose interactions emergently yield a solution to the path planning problem. In this case, although there are many more ghosts than vehicles, the output of the system is a single recommended path for each vehicle.

An important consideration in control applications concerns the implementation of the environment that maintains the digital pheromones. The pheromone variables need to be maintained on processors that can update them (to provide evaporation and propagation), while remaining accessible to the agents. There are at least three workable alternatives.

1. A single central computer can maintain the pheromone infrastructure. While this approach limits the distribution and scaling of the system, it is the simplest. Because pheromone

computations are so simple, in practice we can handle a system of 24,000 ghost agents on a grid of 40,000 cells on a single off-the-shelf laptop computer in real time.

2. At the opposite extreme, each region of space can be assigned its own processor. The most elegant approach is to embed the processors in the space, as unattended ground sensors that have on-board storage, processing, and communications. Only nearest neighbors need to communicate with one another, so power requirements can be limited.
3. Each avatar can maintain a pheromone map for regions it has recently visited, and exchange maps with other avatars when it comes near them. Agents move continuously through space, so the areas of most interest are those close to an agent's current location, which is the region for which an agent-based pheromone map will be most accurate.

The topographical reasoning example shows how swarming can *predict* behavior by generating a probability distribution over possible futures. The frequency with which the agents visit different regions of the landscape is proportional to the probability that a single agent would traverse that region. Such an interpretation is useful in guiding the search for walkers of interest, or in planning traffic networks, among other applications.

We have described the utility of evolutionary methods in tuning control applications of swarming geospatial reasoning. Synthetic evolution is also useful in predictive applications. We are currently applying swarming geospatial prediction to the movement of soldiers in urban combat (Parunak, 2005). We begin the swarming simulation in the past, and adjust the individual parameters of each ghost to fit the observed recent behavior of the corresponding real-world entity. Then we allow the fittest ghosts to run into the future to form our prediction. This mechanism allows us to base our predictions on a much richer model of the individual agent's behavior than would otherwise be possible, without the need for time-consuming knowledge acquisition.

5. Conclusion

Swarming methods are a fruitful resource for reasoning about the movements of entities constrained by topological or topographical features of the environment. By making disciplined use of enlarged populations of agents with greatly simplified internal logic, appropriate application of stochastic decisions, and stigmergic information exchange, we can solve problems that would be intractable by classical enumerative techniques and prohibitively expensive to implement with more sophisticated agent-based simulation. These methods converge with reasonable speed, and can support both robotic control and prediction of natural systems.

References

Brueckner, S., 2000. Return from the Ant: Synthetic Ecosystems for Manufacturing Control. Dr.rer.nat. Thesis, Humboldt University Berlin, Berlin, Germany. http://dochost.rz.hu-berlin.de/dissertationen/brueckner-sven-2000-06-21/PDF/Brueckner.pdf.

Brueckner, S. and Parunak, H.V.D., 2003. *Resource-Aware Exploration of Emergent Dynamics of Simulated Systems, Autonomous Agents and Multi-Agent Systems (AAMAS 2003),* Melbourne, Australia, pp. 781-788. http://www.altarum.net/~vparunak/AAMAS03APSE.pdf.

Ferber, J. and Müller, J.-P., 1996. *Influences and Reactions: a Model of Situated Multiagent Systems*, *Second International Conference on Multi-Agent Systems (ICMAS-96)*, pp. 72-79.

Finin, T., 1997. *UMBC KQML Web*. In: T. Finin (Editor). University of Maryland, Baltimore Campus. http://www.cs.umbc.edu/kqml/.

FIPA, 2000. *FIPA Agent Communication Language Specifications*. FIPA. http://www.fipa.org/repository/aclspecs.html.

Gimblett, H.R.(Editor), 2002. *Integrating Geographic Information Systems and Agent-based Modeling Techniques for Simulating Social and Ecological Processes*. Santa Fe Institute Studies in the Sciences of Complexity. Oxford University Press, New York, NY.

Grassé, P.-P., 1959. La Reconstruction du nid et les Coordinations Inter-Individuelles chez *Bellicositermes Natalensis et Cubitermes sp.* La théorie de la Stigmergie: Essai d'interprétation du Comportement des Termites Constructeurs. *Insectes Sociaux*, **6**: 41-84.

Haddadi, A. and Sundermeyer, K., 1996. Belief-Desire-Intention Agent Architectures. In: G.M.P. O'Hare and N.R. Jennings (Editors), *Foundations of Distributed Artificial Intelligence*. John Wiley, New York, NY, pp. 169-185.

Kauffman, S.A. and Levin., S., 1987. Toward a general theory of adaptive walks on rugged landscapes. *J. Theoret. Biol.*, **128**: 11-45.

Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P., 1983. Optimization by Simulated Annealing. *Science*, **220**: 671-80.

Michel, F., 2004. Formalisme, méthodologie et outils pour la modélisation et la simulation de systèmes multi-agents. Doctorat Thesis, Université des Sciences et Techniques du Languedoc, Montpellier, France. http://www.lirmm.fr/~fmichel/these/index.html.

Müller, J.P., 1996. *The Design of Intelligent Agents.* Lecture Notes in Artificial Intelligence 1177. Springer, Berlin, xiv + 227 pp.

Parunak, H.V.D., 1997. 'Go to the Ant': Engineering Principles from Natural Agent Systems. *Annals of Operations Research*, **75**: 69-101. http://www.altarum.net/~vparunak/gotoant.pdf.

Parunak, H.V.D., 2003. *Making Swarming Happen, Swarming and Network-Enabled C4ISR.* ASD C3I, Tysons Corner, VA. http://www.altarum.net/~vparunak/MSH03.pdf.

Parunak, H.V.D., 2005. *Evolving Swarming Agents in Real Time, Genetic Programming Theory and Practice (GPTP05)*. Springer, Ann Arbor, MI, pp. (forthcoming).

Parunak, H.V.D., Brueckner, S. and Sauter, J., 2004a. *Digital Pheromones for Coordination of Unmanned Vehicles*. In: F.M. Danny Weyns, and H. Van Dyke Parunak (Editor), *Workshop on Environments for Multi-Agent Systems (E4MAS 2004)*. LNAI. Springer, New York, NY, pp. (forthcoming). http://www.altarum.net/~vparunak/E4MAS04_UAVCoordination.pdf.

Parunak, H.V.D., Brueckner, S. and Savit, R., 2004b. *Universality in Multi-Agent Systems, Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2004)*. IEEE, New York, NY, pp. 930-937. http://www.altarum.net/~vparunak/AAMAS04Universality.pdf.

Parunak, H.V.D. and Brueckner, S.A., 2004. Engineering Swarming Systems. In: F. Bergenti, M.-P. Gleizes and F. Zambonelli (Editors), *Methodologies and Software Engineering for Agent Systems*. Kluwer, pp. 341-376. http://www.altarum.net/~vparunak/MSEAS03.pdf.

Parunak, H.V.D., Brueckner, S.A. and Sauter, J., 2002a. *Digital Pheromone Mechanisms for Coordination of Unmanned Vehicles, First International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2002)*, Bologna, Italy, pp. 449-450. www.altarum.net/~vparunak/AAMAS02ADAPTIV.pdf.

Parunak, H.V.D., Brueckner, S.A., Sauter, J.A. and Matthews, R., 2005. *Global Convergence of*

Local Agent Behaviors, *Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS05)*, Utrecht, The Netherlands, pp. (forthcoming). http://www.altarum.net/~vparunak/AAMAS05Converge.pdf.

Parunak, H.V.D., Purcell, M. and O'Connell, R., 2002b. *Digital Pheromones for Autonomous Coordination of Swarming UAV's*, *First AIAA Unmanned Aerospace Vehicles, Systems,Technologies, and Operations Conference*. AIAA, Norfolk, VA. www.altarum.net/~vparunak/AIAA02.pdf.

Parunak, H.V.D., Savit, R. and Riolo, R.L., 1998. *Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide*. In: N. Gilbert, R. Conte and J.S. Sichman (Editors), *Multi-agent systems and Agent-based Simulation (MABS'98)*. Lecture Notes on Artificial Intelligence 1534. Springer, Paris, FR, pp. 10-25. http://www.altarum.net/~vparunak/mabs98.pdf.

Rao, A.S. and Georgeff, M.P., 1991. *Modeling Rational Agents within a BDI Architecture*. In: J. Allen, R. Fikes and E. Sandwall (Editors), *International Conference on Principles of Knowledge Representation and Reasoning (KR-91)*. Morgan Kaufman, pp. 473-484.

Rimon, E. and Kodischek, D.E., 1992. Exact Robot Navigation Using Artificial Potential Functions. *IEEE Transactions on Robotics and Automation*, **8**(5 (October)): 501-518.

Sauter, J.A., Matthews, R., Parunak, H.V.D. and Brueckner, S., 2002. *Evolving Adaptive Pheromone Path Planning Mechanisms*, *Autonomous Agents and Multi-Agent Systems (AAMAS02)*, Bologna, Italy, pp. 434-440. www.altarum.net/~vparunak/AAMAS02Evolution.pdf.

Sauter, J.A., Matthews, R., Parunak, H.V.D. and Brueckner, S.A., 2005. *Performance of Digital Pheromones for Swarming Vehicle Control*, *Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Utrecht, Netherlands, pp. (forthcoming). http://www.altarum.org/~vparunak/AAMAS05SwarmingDemo.pdf.

Sterman, J., 2000. *Business Dynamics*. McGraw-Hill, New York, NY.