Proceedings of the 2009 Winter Simulation Conference
M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, eds.

INTERPRETING DIGITAL PHEROMONES AS PROBABILITY FIELDS

H. Van Dyke Parunak

Vector Research Center, TTGSI
3520 Green Court, Suite 250
Ann Arbor, MI 48105-1579

ABSTRACT

Stigmergic agent-based systems often use digital pheromones, scalar variables indexed to the agents' environment, as a means of coordination among agents. These fields are potentially valuable in another way, as a source of information to programs external to the MAS. In many cases, and in particular in the polyagent modeling construct, pheromones can be interpreted as probability fields, allowing inferences about the range of behaviors accessible to the system from a single execution. We motivate this interpretation of digital pheromone fields, develop the mathematics that supports their analysis, and illustrate their use in a battlefield simulation.

1 INTRODUCTION

An important technique for multi-agent coordination is stigmergy, in which the agents interact, not by directly sensing one another, but by modifying a shared environment that guides their behavior. The concept originated to explain the complex group behavior of social insects (Grassé 1959), but is also evident in human interactions (Parunak 2006). The blackboard systems popular in early distributed AI (Nii 1986a; Nii 1986b) are an example of stigmergic coordination.

In one stigmergic technique, modeled on ant path planning, agents augment scalar variables, or "digital pheromones," at the locations they visit in the environment. These variables aggregate deposits from different agents, propagate them to smooth the variation due to discrete agents, and evaporate over time to discard obsolete information. They enable individual agents to assess where their peers have been, and to modulate their behavior accordingly.

In most stigmergic systems, digital pheromones serve only *internally*, to coordinate the actions of the agents. However, the pheromone field that the agents construct contains information that may be useful *externally*, to other, non-stigmergic systems. In many cases, the pheromone field can be interpreted (up to a straightforward normalization) as a probability field defining the likelihood of finding an agent at a given location. Building on this interpretation, a stigmergic multi-agent simulation can yield distributions not only over the location of a single agent, but also over more complex conditions. In their native form these distributions marginalize over many other variables, but in some cases it is possible to derive bounds on the joint conditions that underlie them.

The results of an agent model can be reported in many ways to other systems. Using a probability distribution is natural because it captures the uncertainty of the simulation in a natural way that is independent of the modeling technology. Pheromone-based models have a particular advantage in this respect. If the external application requires distributions over alternative states of the world, doing enough replications of a conventional model to generate those distributions can be computationally prohibitive. Stigmergic agents, and in particular polyagents, can generate a complete distribution with a single run.

Section 2 reviews the use of digital pheromones in multiagent systems, and explains how the polyagent modeling construct explores multiple possible futures of the world concurrently. Section 3 develops the basic mathematics of deriving probability distributions over conditions of interest from digital pheromone fields. Section 4 shows how this analysis can be extended to joint and conditional distributions. Section 5 illustrates the approach with an example. Section 6 concludes.

2 DIGITAL PHEROMONES AND POLYAGENTS

Stigmergic multi-agent systems using digital pheromones have proven useful in a wide array of applications, including ant-colony optimization (Dorigo and Stuetzle 2004), shop-floor control (Brueckner 2000), air-traffic control (Sauter, Matthews et

978-1-4244-5771-7/09/$26.00 ©2009 IEEE

al. 2005), telecommunications (Schoonderwoerd, Holland et al. 1997), and battle prediction (Parunak 2007). All of these applications generate pheromone fields for their *internal* use. In each case, the techniques in this paper can be used to exploit those fields for *external* applications, providing a powerful means for hybridizing swarm and conventional intelligence (Parunak, Nielsen et al. 2006).

For concreteness, we focus on the polyagent modeling construct (Parunak and Brueckner 2006), as applied to combat modeling of the interaction of Red (hostile) and Blue (friendly) forces. Polyagents represent each entity in the domain by a set of agents (thus, a polyagent). A persistent *avatar* manages a stream of transient *ghosts*, each of which explores an alternative future for the entity in a simulated world. As the ghosts of different avatars interact, they explore alternative futures for their individual entities, complete with the full range of possible interactions that might result from the alternative futures of other entities. These futures are executed in one or more virtual *environments*, such as a book of temporally successive geospatial maps or a task network, whose topology reflects that of the problem domain.

Ghosts are tropistic. Their behavior is determined by a set of digital pheromone fields in their environment. Each pheromone flavor associates a scalar value with each cell of the environment. Some of these fields are emitted by objects of interest (such as roads or buildings). Others are deposited by the ghosts as they move about. A ghost's behavior is determined by a weighted sum of the pheromones it senses in its vicinity, where the weights define the ghost's personality and can be either manually coded or learned by observation of the entity that the ghost represents. In the real world, one entity's behavior can be modulated by the presence or absence of another entity. A ghost's behavior is modulated by other entities through the probability field encoded in their pheromones, and thus reflects an average response across all of the locations of the other entities that their ghosts have explored. Clearly, the accuracy of the predictions yielded by the set of ghosts depends on the correctness of their behavioral models. We evolve these models against the observed behavior of the entity represented by the polyagent in the recent past, avoiding the expense of manual knowledge engineering.

Each ghost has a strength, which changes to reflect its interaction with the environment (such as combat in a battlefield model). The ghost's strength can be interpreted as its degree of health, or more abstractly as the probability that the entity that it represents would be at full combat strength at the ghost's time and place. A log of each ghost's strength as a function of time is an additional resource, alongside the pheromone fields deposited by the ghost, for deriving distributions. For example, an entity's avatar can estimate the strength of its entity by taking the average of the strengths of its ghosts.

Each ghost deposits a pheromone corresponding to the entity it represents as it moves (Figure 1). The strength of this particular field at a location represents how frequently ghosts of that entity visit that location. The amount of the deposit depends on the ghost's strength, so its field takes into account the effects of attrition. A ghost can deposit pheromones of several different flavors, for example, one corresponding to its own avatar, one to the entire force (Red or Blue) to which its avatar belongs, and one to a unit within the force (company or battalion). A pheromone field modulated by strength yields an estimate of the probability of encountering a unit of force at each location. Pheromones can also be modulated by a ghost characteristic than strength, such as degree of preference at the moment for a given course of action, yielding a field with different semantics that may be useful in some applications.

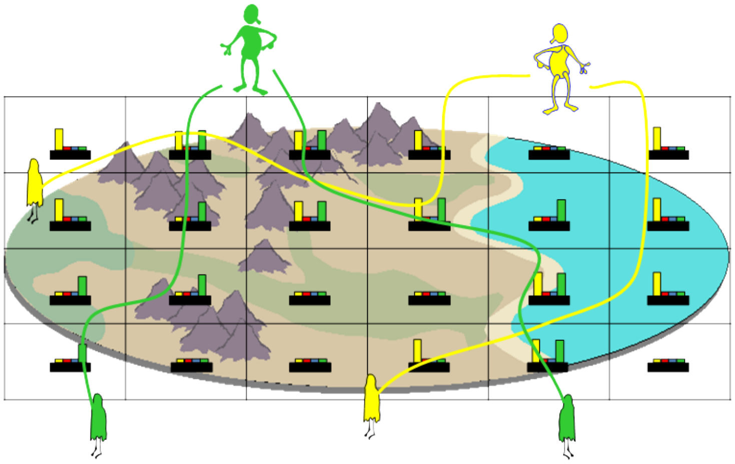

Figure 1: Ghosts deposit pheromones reflecting their presence, and sense the presence of ghosts of other entities through their pheromones.

Pheromone strength depends not only on entity type and location, but also on time. In predictive applications, we maintain a set of pheromone maps, one for each successive time step from a specified time in the past (the "insertion horizon") to a specified time in the future (the "prediction horizon"). Each page covers the entire area of interest (AOI). This set of maps is called the "book of pheromones." The number of pages is fixed, and as real time advances, we drop the oldest page and add a new page one time step further into the future. Pages are indexed by τ. $\tau = 0$ corresponds to "now." $\tau < 0$ indexes pages in the past (used to train the ghosts by evolution against observations (Parunak 2007)), and $\tau > 0$ indexes pages over which behavior is to be predicted.

Up to a normalizing constant, the pheromone field deposited by the ghosts of a given entity on a given page of the book of pheromones is a probability field estimating the entity's location at the time represented by that page over all possible futures explored by that entity's ghosts. We can use the field to estimate the probability that the entity is in a given region of the page. Let A be the total amount of the entity's pheromone deposited over the entire page, and B the amount deposited in a region of interest (Figure 2). Then B/A (in which the normalizing constant cancels itself) estimates the probability that the entity is in the limited region. The entity's most likely location is given by the center of mass of the probability field.

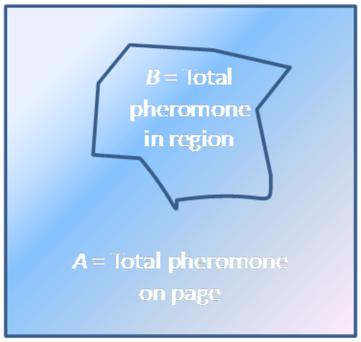
B = Total pheromone in region
A = Total pheromone on page

Figure 2: Computing probability that an entity is in a region

Often we are interested not in the location of an individual entity, but in the distribution of a group of entities (for example, all Red entities). In this case, all ghosts of Red entities deposit the same flavor of pheromone, which now reflects the probability of encountering any Red entity over the area of interest.

In spite of the obvious abstractions in this formulation, its consideration of multiple alternative futures in sophisticated war games has yielded predictions that were superior not only to other technical approaches (game-theoretic and statistical), but also to experienced military officers (Parunak 2007). The polyagent construct has also proven successful in other domains, itemized in the first paragraph of this section, including telecommunications, manufacturing, logistics, and air traffic control. It is applicable wherever there is inherent uncertainty, which is to say in most real-world problem domains.

3 DISTRIBUTIONS OVER CONDITIONS

Often we want to know more than just the probability of finding an entity at a location. In this section, we show how to compute the distribution of various conditions. These conditions might be used by an external reasoner to predict when a course of action is likely to terminate and in what state. We consider three classes of conditions: those that depend on the number of entities in a defined subregion of the AOI, those that depend on the number of forces in the entire AOI, and those that depend on distances between forces and designated locations or other forces. Table 1 summarizes our notation.

3.1 Number of Entities in a Region

Figure 2 shows a page from the book of pheromones, with a gradient representing the pheromone field over the entire AOI, and a subregion of particular interest. A common class of condition depends on the strength of a given force within the region. For example, a Blue attack might terminate when the Red strength within a defended region γ falls below 25%.

A naïve approach would be to compute $\Gamma(R, \tau)/\Phi(R, \tau)$, which estimates the probable percentage of Red in the region, and flag the condition if this ratio falls below 0.25. This approach is unsatisfactory for two reasons.

1. It allows only a Boolean condition. For some purposes we would like to know a probability distribution over the proposition that Red is below 25% in the region.

2. The simple ratio estimates the percentage of Red's current strength in the region. Often we are more interested in the percentage of Red's strength at some previous time.

Table 1: Notation for Examples

Notation	Meaning
E	A single domain entity (with subscripts if needed), represented by a single avatar (for example, an Army company)
R, B	A set of entities (Red or Blue), represented by multiple avatars (with subscripts if there are different groups of the same force)
$S(E, \tau)$	Strength of entity E on page τ (= average strength of E's ghosts on that page)
$\Phi(E, \tau)$	Total E pheromone on page τ. We use the same notation with R or B to indicate the total amount of pheromone from all the avatars belonging to the designated group.
$\Gamma(E, \tau)$, $\Delta(E, \tau)$, $\Lambda(E, \tau)$, …	Total E pheromone in named region γ, δ, λ, … Again, we overload the function to handle groups of entities.
(x_γ, y_γ)	Coordinates of named point γ in a specified pheromone map (or "page" in a diachronic book of pheromones)
$\phi(E, x, y, \tau)$	E pheromone strength in cell (x, y) on page τ (with usual overloading for R and B)
$p(E, x, y, \tau)$	$\phi(E, x, y, \tau)/\Phi(E, \tau)$ = probability that E (R, B) is at (x, y, τ)

To address the first issue, note that the actual number of Red entities in the region follows a binomial distribution. Whether or not each Red entity is in the region is a Bernoulli trial whose probability of success is $\Gamma(R, \tau)/\Phi(R, \tau)$. So the probability that exactly $k \leq n \equiv |R|$ Red entities are in γ is just

$$\binom{n}{k}\left(\frac{\Gamma(R,\tau)}{\Phi(R,\tau)}\right)^{k}\left(1-\frac{\Gamma(R,\tau)}{\Phi(R,\tau)}\right)^{n-k} \tag{1}$$

To address the second condition, $S(R, 0)$ gives us the effective number of Red entities on page 0. So we can express k as a fraction of this original strength. The probability $p(C_1, \tau)$ that the strength of Red in region γ is below 25% of Red's original strength is then:

$$p(C_1,\tau)=\sum_{i=0}^{\lfloor .25*S(R,0)\rfloor-1}\binom{n=S(R,\tau)}{i}\left(\frac{\Gamma(R,\tau)}{\Phi(R,\tau)}\right)^{i}\left(1-\frac{\Gamma(R,\tau)}{\Phi(R,\tau)}\right)^{n-i} \tag{2}$$

Sometimes we may want to detect a condition like C_2, the population in region γ exceeds 60% of the original strength. The computation is almost the same, yielding the probability:

$$p(C_2,\tau)=\sum_{i=\lfloor .6*S(A,0)\rfloor+1}^{S(A,\tau)}\binom{n=S(A,\tau)}{i}\left(\frac{\Gamma(A,\tau)}{\Phi(A,\tau)}\right)^{i}\left(1-\frac{\Gamma(A,\tau)}{\Phi(A,\tau)}\right)^{n-i} \tag{3}$$

3.2 Number of Entities in the entire AOI

Another class of condition concerns the proportion of a force's original strength remaining in the entire AOI. We might want to know when that proportion drops below (say) 25%. A naïve Boolean evaluation is simply $S(A,\tau)/S(A,0) < 0.25$. Computing a distribution is a bit trickier. We cannot use the binomial distribution because we are interested in the overall strength of A, not just the strength in some region. The per-trial probability becomes 1, $1 - p$ becomes 0, and the probability vanishes.

Polyagents give us the raw materials we need to estimate such a distribution, because each entity in the set A is represented by a swarm of ghosts, each exploring a different future for its entity and recording its attrition as a reduction in its strength. In Section 3.1, an entity's avatar estimated its own strength (the probability that it would be fully capable) by averaging its ghosts' strengths, but in the process we discarded information about the variation in those strengths. We can retain that information to estimate the distribution over total force strengths. For example, if we have three entities, with ghost strengths (.1, .2, .5), (.3, .4, .6), (.7, .8, .9), the possible strengths for the overall force are .1+.3+.7, .1+.3+.8, .1+.3+.9, .1+.4+.7, .1+.4+.8, etc.

Actually forming these tuples for all ghosts of all avatars in a force is prohibitive. We use a sampling approach, repeatedly forming tuples by randomly drawing one ghost from each avatar and adding their strengths.

3.3 Distances between Forces

Another condition of interest concerns the distance between a force and a region, say, the condition C_3 that Red is more than 1 km from γ. A naïve solution would be to measure the distance between the center of mass of Red's pheromone distribution over the page, and the point γ (or the nearest edge of the region γ). This would yield a Boolean condition, but not an estimate of $p(C_3, \tau)$. To obtain such an estimate, we use $p(R, x, y, \tau)$, the probability of Red's presence in each cell of the pheromone map. We count a cell's contribution only when it is more than 1km from γ, using the Heaviside function $H(x)$, which returns 1 when $x > 0$ and 0 otherwise.

$$p(C_3,\tau) = \sum_{(x,y)\in Page} p(R,x,y,\tau) * H\left(\sqrt{(x_\gamma - x)^2 + (y_\gamma - y)^2} - 1km\right) \tag{4}$$

Sometimes we need to estimate the distribution over the separation between two forces. In principle, we can generalize the approach in Eq. 4. The probability of an entity from *B1* being at (x_{B1}, y_{B1}) and of an entity from *B3* being at (x_{B3}, y_{B3}) at time τ is just

$$p(B1, x_{B1}, y_{B1}, \tau) * p(B3, x_{B3}, y_{B3}, \tau) \tag{5}$$

We can use the Heaviside function to select only those combinations with the appropriate range of separations. If we assume that the acceptable region around the 5km separation is ±0.5km, we have Eq. 6.

$$p(C_4,\tau) = \sum_{(x,y)\in Page} p(B1,x,y,\tau) \sum_{(\xi,\upsilon)\in Page} p(B3,\xi,\upsilon,\tau) * H\left(\sqrt{(\xi - x)^2 + (\upsilon - y)^2} - 5.5km\right) \tag{6}$$

While theoretically straightforward, this construction is computationally prohibitive. A brute force implementation requires comparison of every pair of cells in the pheromone map, and a map can contain as many as 10^4 cells. Several optimizations suggest themselves.

1. Where there is a hard distance constraint (as here), clever evaluation can avoid referencing those cells that obviously violate it.
2. We may use a coarser grid for evaluating this class of condition, reducing the number of cells that need to be considered.
3. Operations of this sort are common in vision processing. We are exploring the use of a highly parallel subprocessor such as a GPU to support this computation.
4. In the worst case, we may need to settle for a Boolean condition based on the distance between the centers of mass of two forces.

3.4 Probabilities over Time

In any diachronic system, one must distinguish the immediate probability $p(C, \tau)$ of an event occurring in a time interval from the cumulative probability $p_{cum}(C, \tau)$ that an event obtains at or before a given time.

The probabilities computed from a pheromone field laid down by ghosts on one page of the book of pheromones describe the distribution over the position of the entities represented by those ghosts at the period of time represented by that page. The cumulative probability over the condition *C* that those entities have been at a given location on or before time τ is then derived recursively,

$$p_{cum}(C, 0) = p(C, 0); p_{cum}(C, \tau) = p_{cum}(C, \tau\text{-}1) + p(C, \tau)(1 - p_{cum}(C, \tau\text{-}1) \tag{7}$$

Each ghost's strength reflects its experience moving through time. Thus probabilities computed from logs of ghost strengths are already cumulative. Immediate probabilities can be derived as differences between the cumulative probabilities at successive time steps,

$$p_{cum}(C, \tau) = p_{cum}(C, \tau) - p_{cum}(C, \tau\text{-}1) \tag{8}$$

$p_{first}(C, \tau)$, the probability that a condition obtains first at time τ and not before, is then

$$p_{first}(C, 0) = p(C, 0); p_{first}(C, \tau) = p(C, \tau)(1 - p_{cum}(C, \tau\text{-}1) \tag{9}$$

While $p(C, \tau) + p(\neg C, \tau) = 1$, the same complementarity does not hold for p_{cum} or p_{first}.

4 CONDITIONAL DISTRIBUTIONS

Let $\{C_i\}$ be propositions over state variables (e.g., $C_7 \equiv$ *unitAStrength* > 10). The previous section showed how to derive probability density functions $p(C, \tau)$ for these over time. One common application of these time-varying distributions is for some other application to assess that at some point in the future being modeled by the polyagent system, a condition C_i is in fact satisfied. Further reasoning will then require the distributions of other state assertions *conditioned on* $p(C_i, \tau) = 1$, e.g, $p(C_j, \tau|C_i)$. This section discusses ways to do this.

If the propositions are independent, conditioning doesn't change anything. But how can we know that they are dependent? Omitting τ for clarity,

1. We have $p(a)$ and $p(C)$.
1. We want $p(a|C)$.
2. One way to get this would be to compute $p(a, C)$ and then compute $p(a|C) = p(a, C)/p(C)$.

So the real issue is how to get the joint probabilities for any pair of propositions. The Red and Blue pheromone fields from which we evaluate the margins are aggregated over all futures. How can we tease apart those futures in which both conditions are satisfied from those in which only one is satisfied?

We consider two different approaches. First, we summarize theoretical ways to derive a joint distribution from margins. Then we describe the sampling approach that we have found more useful in practice.

4.1 Theoretical Approaches

There are at least two theoretical approaches to estimating a joint distribution from margins. One, expounded by Li, Zhu, Ogihara, and Cheng (Li, Zhu et al. 2002) ("LZOC"), takes repeated samples of the marginal distributions. By looking for correlations in the variation of the margins, we can estimate the joint distribution. The other (Cheeseman and Stutz 2004) relies on the maximum entropy principle to construct a joint distribution that is maximally uncertain given the known margins.

LZOC offer two methods for estimating a joint distribution from margins. These methods are motivated by data mining, and are designed to deal with discrete counts of attributes over data records. The basic idea of both LZOC methods is to divide the N records into k non-overlapping partitions of n points each, and compute the marginal distribution of the attributes of interest over each partition. In their development,

1. A_i is the number of records in partition i that contain attribute a, and $A = \Sigma A_i$.
2. B_i is the number of records in partition i that contain attribute b, and $B = \Sigma B_i$.
3. The problem is to estimate $q = p(a, b)$, the probability that a record contains both a and b, from the cell margins A_i and B_i.

We want to think in terms of probabilities rather than counts. To be specific, consider a probability (pheromone) field indicating the location of a single unit. Then the LZOC partitions correspond (for example) to the cells of a pheromone map.

1. p_{ai} is the probability of finding unit a in cell i. $p_a = \Sigma p_{ai}$ is the probability that the unit is anywhere in the AOI.
2. p_{bi} is the probability of finding unit b in cell i. $p_b = \Sigma p_{bi}$ is the probability that the unit is anywhere in the AOI.
3. The problem is to estimate $p(a, b)$, the probability that unit a and unit b are in the same cell.
4. In LZOC's terms, $p_a = A/N$, and $p_b = B/N$. But neither A nor N has any independent meaning in our framework. (The number of partitions k *is* meaningful in our framework; it is the number of cells over which we are reasoning.)

Terminologically, we describe the underlying distribution of attributes in the original LZOC case as an array of N *elements*, records in a data mining application. These are grouped into k *partitions* of n elements (or records) each. In the case of our running pheromone example, *cells* are analogous to the LZOC *partitions*, and there is no natural counterpart to the *elements*.

The first LZOC method, minAB, is based on the observation (in the discrete domain) that the number of records in a partition containing multiple attributes is bounded above by the number of records containing the least frequent attribute in the partition. Thus we can compute an upper bound on the number of records in the total data set with all attributes by summing the minimums over each partition. In the two-attribute case,

$$p(a,b) \le \frac{1}{N}\sum_{i=1}^{k} Min(A_i, B_i) \tag{10}$$

The method transfers naturally to our continuous domain, where the partitions are represented by the cells of the pheromone map. We transform the data set of counts to one of probabilities by dividing all entries by N, and then compute

$$p(a,b) \le \frac{1}{N}\sum_{i=1}^{k} Min(p_{ai}, p_{bi}) \tag{11}$$

Clearly, this is the same as Eq. 10, since the constant divisor distributes over summation and minimization.

A closer estimate is provided by another derivation that LZOC call the *prodAB* method. Their basic approach is

1. Derive expressions for the expected values EA_i, EB_i, EA_iB_i, and $E(\Sigma A_i \Sigma B_i)$ in terms of n, k, p_a, p_b, and q,
2. Solve for q in terms of these expected values,
3. Then estimate EA_i with $\Sigma A_i/k$, EB_i with $\Sigma B_i/k$, EA_iB_i with $\Sigma A_iB_i/k$, and $E(\Sigma A_i \Sigma B_i)$ with $\Sigma A_i \Sigma B_i$ to estimate $\hat{q}$.

Their result is

$$\hat{q} = \frac{(nk-1)\sum_i A_i B_i - (n-1)\sum_i A_i \sum_i B_i}{n^2 k(k-1)} \tag{12}$$

As with *minAB*, we can express A_i and B_i in terms of p_{ai} and p_{bi}. The result (in the limit of large n and k for clarity) is

$$\hat{q} = nk\sum_i p_{ai} p_{bi} - \sum_i p_{ai} \sum_i p_{bi} \tag{13}$$

Unfortunately, the pheromone domain, unlike the data mining domain, has nothing corresponding to n, the number of records in a partition. Mappings such as $p_a = A/N$ discard the independent values of both A and N (and thus n). Efforts to recover this information (for example, from the variances over the p_{ai}, assuming an underlying binomial distribution) have been so far unsuccessful.

The maximum entropy approach allows derivation of a joint distribution from low-order marginals. It requires enough second-order marginals to link all the marginal variables with one another, a condition that unfortunately is not generally satisfied in our data.

4.2 Sampling Approaches

In practice, we have found it more satisfactory to estimate joint distributions by sampling. Consider the joint probability $p(Red, Blue)$ of finding Red and Blue at the same location. As a thought experiment, imagine that each ghost deposits an interaction pheromone proportional to the number of ghosts of the other side in its cell. The resulting field is just what we want. But the process violates the stigmergic principle of handling all interchanges through the environment, and is likely to incur computational penalties, of two sorts. First, ghosts have to sense one another directly. Second, storage of the joint pheromone will be costly.

A more promising approach is to have each ghost deposit an echo pheromone proportional to the field it senses from its counterpart. For example, each Red ghost would deposit a Red-Echoing-Blue (REB) pheromone. The resulting field differs from the original Blue field because it is deposited only where Red actually visits. Thus the REB field is $p(B|R)$. If this is the conditional distribution we want, we're done. If we want the joint (say, to compute $p(R|B)$, we can multiply by $p(R)$, which we already have. The additional computation involved to do the echo deposit is fairly trivial, but we do have to store an additional time-varying pheromone field for each interaction pair, which will consume space.

Since joint distributions are in general of interest to other algorithms and not to our ghosts themselves, there is no need to store the time-varying field. Instead, we can log relevant information (e.g., unit ID, strength, location, shift #) from each ghost as it passes each page. In a typical application, we have on the order of 4E10 cells per page and on the order of 60 pages, so each pheromone flavor requires on the order of 2.4E6 scalars. Each avatar typically generates 200 ghosts in all over 100 shifts, and we can limit our attention to the last 40 or so ghosts per avatar, representing the converged state of the system. Thus we have to log 40 ghosts x 4 information items x 60 pages = 9600 scalars per avatar. Logs for 100 avatars consume only on the order of E6 scalars, from which we can generate many different combinations of conditions. We select from the logged ghosts those that satisfy conditions of interest to them, have them deposit pheromone on a single static map, and analyze it with the methods above. For example, to explore the joint distribution of strength and location, we can sample ghosts in different bins of strengths and generate their spatial pheromone distribution.

5 AN EXAMPLE

In this section, we illustrate our methods by analyzing data from a simulated wargame. Specifically, we are modeling a Red force that is defending a region γ, and a Blue force that is moving toward γ with the objective of occupying it. Blue's goal G is to get at least 50% of its initial strength into γ. We will derive:

1. The probability that Blue achieves its goal, as a function of future time τ;
2. The distributions of Blue location and Red strength, in the case that Blue does reach its goal;
3. The distributions of Blue location and Red strength, in the case that Blue does not reach its goal.

The methods used in 2 and 3 for Blue location and Red strength can as easily be applied to Red location and Blue strength.

For this example, Table 2 gives unit strengths on each page of the book of pheromones, computed as the average of the logged strengths of each unit's ghosts.

Table 2: Mean Strength Data for Example

Page τ	$S(R, \tau)$	$S(B, \tau)$
0	100	100
1	100	100
2	100	100
3	90	100
4	80	100
5	70	98
6	60	96
7	50	94
8	40	92
9	30	90

5.1 Probability that Blue reaches its Goal

Let G be the condition that Blue gets at least 50% of its strength in γ. We compute this distribution using Eq. 3, with the lower limit of summation $i = \lfloor 0.5S(B,0) \rfloor$. Figure 3 shows the probability of the goal as a function of page index, both absolute and as a log.

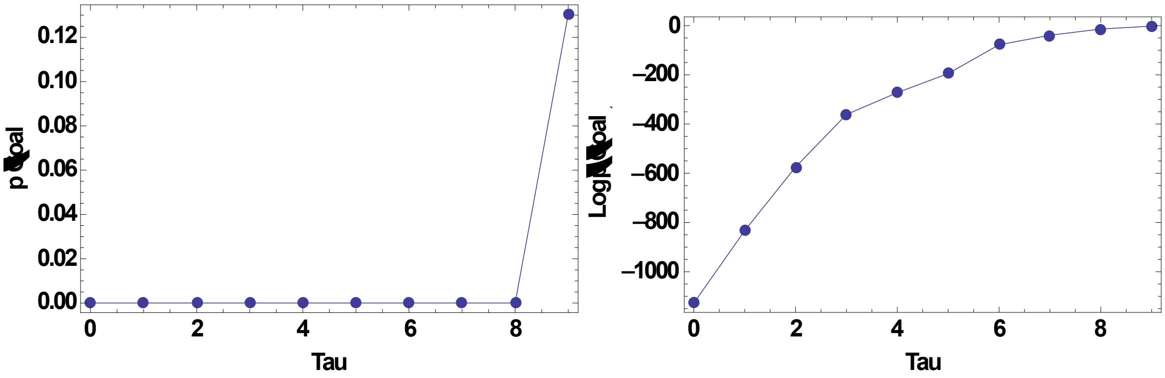

Figure 3: Example of Goal Distribution

5.2 Joint Location and Strength

Let's focus on Blue location as a function of time, represented by the Blue pheromone field. A page in the book of pheromones actually represents a set of (in this case) 100 marginal probabilities: the probability that the Blue unit is in cell 1, the probability that it is in cell 2, and so forth. Let L be the condition "Blue is in cell 70," where cell 70 is within γ. For this computation, we use the logs of ghost location and strength. Each ghost samples a possible future of Blue. We count the ghosts on page τ whose strength is at least 50% of their original strength and whose location is cell 70, and report the ratio of this number to the total number of ghosts as the desired joint probability $p(G,L)$.

6 CONCLUSIONS

Digital pheromones have proven themselves an effective means of coordination among multiple agents in a wide array of domains. In many systems, they summarize the behaviors of many similar agents, and so it makes sense to interpret them as a probability field over that space of behavior. From this perspective, they are potentially valuable not only *internally* to the stigmergic MAS that generates them, but also *externally*. In particular, when polyagents are used to explore multiple alternative futures of a system, the resulting pheromone fields are a useful way to communicate the resulting predictions to other applications. By expressing the results in the form of probability distributions, we enable the consumers of these predictions to reason about the uncertainty inherent in statements about the future in a commonly understood language. Conversely, when an application requires probabilistic estimates, generating these estimates by conventional single-trajectory simulation can be prohibitively expensive, while this technique can generate them with a single run of a polyagent model.

The derivation of distributions over simple conditions is straightforward. The major area for improvement is in obtaining joint distributions. The theoretical methods of (Li, Zhu et al. 2002) and (Cheeseman and Stutz 2004) are promising, but their preconditions are difficult to realize, and we currently rely on sampling ghosts to estimate these distributions.

ACKNOWLEDGMENTS

This work would not be possible without experimental support provided by Bob Matthews, Jorge Goic, and Andrew Yinger, and has benefited from comments by John Sauter and Jacob Crossman.

REFERENCES

Brueckner, S. 2000. Return from the Ant: Synthetic Ecosystems for Manufacturing Control. *Computer Science*. Berlin, Germany, Humboldt University Berlin.

Cheeseman, P. and J. Stutz. 2004. On The Relationship between Bayesian and Maximum Entropy Inference. In Proceedings of 24th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering, AIP Conference Proceedings.

Dorigo, M. and T. Stuetzle. 2004. *Ant Colony Optimization*. Cambridge, MA, MIT Press.

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.

Li, T., S. Zhu, M. Ogihara and Y. Cheng. 2002. Estimating joint probabilities from marginal ones. In Proceedings of 4th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2002), Springer.

Nii, H. P. 1986a. Blackboard Systems. *AI Magazine* 7(3): 40-53.

Nii, H. P. 1986b. Blackboard Systems. *AI Magazine* 7(4): 82-107.

Parunak, H. V. D. 2006. A Survey of Environments and Mechanisms for Human-Human Stigmergy. *Proceedings of E4MAS 2005*. D. Weyns, F. Michel and H. V. D. Parunak, Springer. LNAI 3830**:** 163-186.

Parunak, H. V. D. 2007. Real-Time Agent Characterization and Prediction. In Proceedings of International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'07), Industrial Track, Honolulu, Hawaii, ACM.

Parunak, H. V. D. and S. Brueckner. 2006. Concurrent Modeling of Alternative Worlds with Polyagents. In Proceedings of the Seventh International Workshop on Multi-Agent-Based Simulation (MABS06, at AAMAS06), Hakodate, Japan, Springer.

Parunak, H. V. D., P. E. Nielsen, S. Brueckner and R. Alonso. 2006. Hybrid Multi-Agent Systems. In Proceedings of the Fourth International Workshop on Engineering Self-Organizing Systems (ESOA'06), Hakodate, Japan, Springer.

Sauter, J. A., R. Matthews, H. V. D. Parunak and S. A. Brueckner. 2005. Demonstration of Digital Pheromone Swarming Control of Multiple Unmanned Air Vehicles. In Proceedings of AAAI Infotech@Aerospace, Arlington, VA, AIAA.

Schoonderwoerd, R., O. Holland and J. Bruten. 1997. Ant-like agents for load balancing in telecommunications networks. In Proceedings of Autonomous Agents 97, Marina del Rey, CA, Association for Computing Machinery.

AUTHOR BIOGRAPHY

H. VAN DYKE PARUNAK is Chief Scientist of the Vector Research Center for Enterprise Performance, TTGSI, in Ann Arbor, MI. He received his Ph.D. in Near Eastern Languages and Civilizations from Harvard University, and his MA in Computer and Communications Sciences from the University of Michigan. He serves as an Associate Editor for the Journal of Autonomous Agents and Multi-Agent Systems, the ACM Transactions on Autonomous and Adaptive systems, the Journal of Intelligent Manufacturing, AI EDAM (Artificial Intelligence for Engineering Design, Analysis and Manufacturing), and the International Journal of Computer Integrated Manufacturing. His email is <van.parunak@newvectors.net>.