Submitted to the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS05), Utrecht, Netherlands, July 2005

Performance of Digital Pheromones for Swarming Vehicle Control

John A. Sauter, Robert Matthews, H. Van Dyke Parunak, Sven A. Brueckner,
Altarum Institute
3520 Green Court, Suite 300
Ann Arbor, MI 48105 USA
(john.sauter, robert.matthews, van.parunak, sven.brueckner)@altarum.org

Abstract

The use of digital pheromones for controlling and coordinating swarms of unmanned vehicles is studied under various conditions to determine their effectiveness in multiple military scenarios. The study demonstrates the effectiveness of these pheromone algorithms for surveillance, target acquisition, and tracking. The algorithms were demonstrated on hardware platforms and the results from the demonstration are reported.

1. Introduction

The word "swarming" is currently in vogue to describe two widely different types of systems. Students of biological systems use it to describe decentralized self-organizing behavior in populations of (usually simple) animals [2, 3, 5, 11]. Examples include path formation, nest sorting, food source selection, thermoregulation, task allocation, flocking, nest construction, and hunting behaviors in many species. Military historians use it to describe a battlefield tactic that involves decentralized, pulsed attacks [1, 7-9].

The link between these two uses of the word is not coincidental. Insect self-organization is robust, adaptive, and persistent, as anyone can attest who has tried to keep ants out of the kitchen or defeat a termite infestation, and military commanders understand the advantage of being able to inflict the confusion, frustration, discomfort, and demoralization that a swarm of bees can visit on their victims.

In spite of the military promise of swarming, little attention has been given to how to implement the mechanisms observed in biological communities into military systems. This paper describes the use of digital pheromones to produce swarming behavior in military systems and studies their effectiveness in performing various functions.

The sequence of sections reflects the engineering process of moving from requirements, through tool selection, implementation, testing, and deployment. Section 2 summarizes three applications for swarming derived from specific miliary scenarios. Section 3 outlines the particular swarming mechanism that we applied to these applications, a computational analog of insect pheromones. Section 4 details how we applied this mechanism to each of the applications. Section 5 outlines specific experiments that were conducted to test the algorithms, and Section 6 describes a physical demonstration of the capabilities. Section 7 concludes.

2. Identification of Required Functions

The focus of this study was the analysis of swarming algorithms to support a range of swarming scenarios. A suite of realistic scenarios were developed by military experts addressing three capability areas: (1) Intelligence, Surveillance and Reconnaissance (ISR), (2) Communications and (3) Battle Damage Assessment (BDA).

Analysis of these swarming scenarios identified three swarm functions that were targeted for this study:

1. **Surveillance and patrol** – a single or continuous sweep of an area by the swarming platforms to look for entities (possibly mobile) of interest.
2. **Target acquisition** – configuring and coordinating the alignment of the right sensors to determine the location, class, and identification of an entity in an area.
3. **Target tracking** – continuously or intermittently maintaining sensor contact with a moving entity to determine its location and heading.

In addition to these three functions, three additional functions were identified but were not included in the study due to time and cost constraints. These were responding to human commands, maintaining line of sight communications among the swarm entities, and plume monitoring.

Parunak [10] reviews the major classes of swarming algorithms that have been applied to the Command and Control (C2) of multiple robotic entities. In this paper we report on the results of experiments with one of those swarming techniques: digital pheromones.

3. Digital Pheromones

Digital pheromones are modeled on the pheromone fields that many social insects use to coordinate their behavior. A formal model of the essentials of these fields has been developed and applied to a variety of problems.

Digital pheromones support three primary operations, inspired by the dynamics of chemical pheromones.

1. They can be deposited and withdrawn from an area. Deposits of a certain flavor are added to the current amount of that flavor of pheromone located at that place. (*Information fusion and aggregation*).
2. They are evaporated over time. This serves to forget old information that is not refreshed. (*Truth maintenance*).
3. They propagate from a place to its neighboring places. The act of propagation causes pheromone gradients to be formed. (*Information diffusion and dissemination*).

These dynamics can be modeled in a system of difference equations across a network of "places" at which agents can reside and in which they deposit and sense increments to scalar variables that serve as "digital pheromones". These equations are provably stable and convergent [4]. They form the basis for a "pheromone infrastructure" that can support swarming for various functions, including path planning [14] and coordination for unpiloted vehicles [6, 15], positioning multi-sensor configurations [12], and maintaining line of sight communications in mobile ad hoc networks [13], several of the functions required by the swarming scenarios

This section describes the basic pheromone framework that underlies all the pheromone algorithms described later.

3.1 Digital Pheromones and Place Agents

A digital pheromone represents information about the system. Different "flavors" of pheromones convey different kinds of information. Digital pheromones exist within in an artificial space called a *pheromone map*. The map is composed of an arbitrary graph of *place* agents. In principle, there are no restrictions on the graph of place agents. In swarming robotics where movement decisions are an important function, it is convenient to have the place agents represent regions of the geographical space. In this study, we tile the physical space with squares, each representing a place agent with eight neighbors.

3.2 Walkers and Avatars

Software agents wander through the pheromone map. Two classes of agents are used, called *walkers* and *avatars*.

A walker agent controls a single platform in the swarm. A walker deposits, withdraws, and reads pheromones in the map and uses that information to make movement and action decisions. The walker can read sensory and other telemetry from the platform and issue commands to control its actions.

Avatars are used to represent the other entities in the environment. These can include friendly (blue), enemy (red), and neutral (green) entities. The avatar receives intelligence about the type, location, heading, speed, and possibly other identifying information that it uses to estimate the location of the unit between contacts. The avatar can also deposit, withdraw, and read pheromones in the map, which it uses to make estimate about where the unit may move next.

Agents can start and stop what is called a pheromone pump. A pheromone pump resides in a place agent and continuously deposits a pheromone of a particular flavor.

An agent uses an *interpreting equation* to weight the pheromones that it senses and decide which place agent to move to next. Some pheromones may attract the agent, while other pheromones may repel it. The interpreting equation is used to assign a scalar value ($V(p)$) to the current place agent and each of its neighbor place agents. The agent then makes either a deterministic move (to the place agent with the largest $V(p)$), or a probabilistic move (using a roulette wheel weighted by each $V(p)$).

3.3 Pheromone Equations

Each place agent maintains a scalar variable corresponding to each pheromone flavor. It performs the basic functions of aggregation, evaporation, and propagation. The underlying mathematics of the field developed by such a network of places, including critical stability theorems [4], rest on two fundamental equations. The parameters governing the pheromone field are:

- $P = [p_i]$ = set of place agents
- $N: P \to P$ = neighbor relation between place agents. Thus the place agents form an asymmetric multigraph.
- $s(\Phi_f, p, t)$ = strength of pheromone flavor f at place agent p and time t.
- $d(\Phi_f, p, t)$ = external deposits of pheromone flavor f within the interval $(t$-1, $t]$ at place agent p.
- $g(\Phi_f, p, t)$ = propagated input of pheromone flavor f at time t to place agent p.
- $E_f \in (0,1)$ = evaporation factor for flavor f.
- $G_f \in [0,1)$ = propagation factor for flavor f.
- T_f = threshold below which $s(\Phi_f, p, t)$ is set to zero.

The first equation describes the evolution of the strength of a single pheromone flavor at a given place agent.

$$s(\Phi_f, p, t+1) = E_f * \left[(1 - G_f) * \left(s(\Phi_f, p, t) + d(\Phi_f, p, t)\right) + g(\Phi_f, p, t)\right]$$

E_f reflects evaporation of pheromone, the 1 - G_f factor calculates the amount remaining after propagation to its neighbors, $s(\Phi_f, p, t)$ represents the amount of pheromone from the previous cycle, $d(\Phi_f, p, t)$ represents the total deposits made since the last update cycle (including pump auto-deposits) and $g(\Phi_f, p, t)$ represents the total pheromone propagated in from all the neighbors of p. Each place agent applies this equation to each pheromone flavor once during every update cycle.

The second fundamental equation describes the propagation received from the neighboring place agents:

$$g(\Phi_f, p, t) = \sum_{p' \in N(p)} \frac{G_f}{|N(p')|} \left(s(\Phi_f, p', t-1) + d(\Phi_f, p', t-1)\right)$$

This equation states that each neighbor place agent p' propagates a portion of its pheromone to p each update cycle, the proportion depending on the parameter G_f and the total number of its neighbors.

4. Algorithm Descriptions

This section describes in detail the pheromone algorithms that perform the functions identified earlier.

Each pheromone flavor has a number of parameters, or "settings", that tune the pheromones to the task. Four different settings are available for use with each pheromone flavor.

1. Update cycle time – the regular time at which propagation, pump auto-deposits, and evaporation occur for this flavor.
2. Propagation factor – The fraction of the pheromone in a place agent that is distributed equally among all the place agent's neighbors.
3. Evaporation factor – the fraction of the pheromone that remains after the evaporation cycle.
4. Minimum place agent pheromone level (threshold) – If after propagation and evaporation, the amount of pheromone in a place falls below this level then the pheromone level is set to zero.

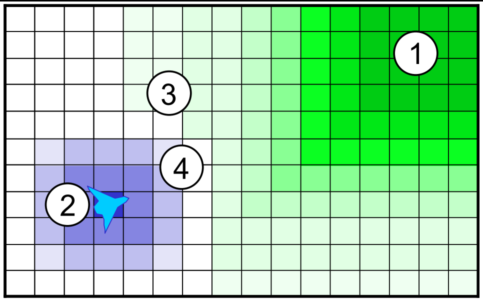

Figure 1: Attractive And Repulsive Pheromones For Surveillance, 1. Surveillance area deposits attractive pheromone, 2. ASV deposits repulsive pheromone, 3. Pheromone infrastructure propagates both attractive and repulsive pheromone to form gradient, 4. ASV climbs net gradient, withdrawing attractive pheromone.

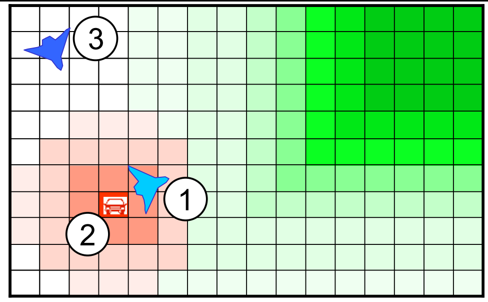

Figure 2: Pheromones Attracting Confirming Sensors – 1. ASVdet detects target and Red avatar is created, 2. Red avatar deposits "NeedsID" pheromone, 3. ASVid is more attracted to NeedsID pheromone than lawn pheromone and climbs gradient to ID target.

4.1 Surveillance and Patrol Algorithm

The surveillance and patrol algorithm must be able to control Autonomous Surveillance Vehicles (ASV's) surveying one or multiple Areas of Interest (AOI) with a certain revisit frequency. The algorithm has been nicknamed the "lawn cutting" algorithm due to certain features of the algorithm that resemble the act of cutting a lawn.

For this algorithm, the place agents within an AOI continually emit an attractive pheromone (the "lawn") that propagates through the other place agents forming a pheromone gradient leading to the AOIs (Figure 1). The Blue walkers controlling the ASV's (referred to henceforth simply as the "ASV") are attracted to these pheromones. Their interpreting equation will cause them to probabilistically climb the gradient towards the AOI containing the highest concentration of attractive pheromone (longest grass). Once an ASV has visited a place agent within the AOI, all the attractive pheromone is removed (cutting the grass) and the place agent stops emitting any more attractive pheromone for a time inversely proportional to the desired surveillance frequency for that AOI. After that time, the pheromone pump is turned back on and attractive pheromone accumulates in the place agent again (grass re-grows).

The ASV deposits repulsive pheromone in the next place agent it plans to move to. This keeps other ASV's away from that same place agent to avoid duplication of effort.

Table 1 gives the pheromone settings for surveillance and patrol. The interpreting equation is

$$V(p) = s(\Phi_l, p) - s(\Phi_v, p)$$

4.2 Target Acquisition Algorithm

The swarming algorithms for positioning sensors for data acquisition were not included in this study. These algorithms have been studied elsewhere [26]. For this study as long as the swarming entity is within sensor range of the target, then the target is detected with a probability defined by an experimental parameter *Pd*. A dwell time experimental parameter was also used that defined how long the sensor had to remain with the target before it could confirm a target's identification.

Table 1: Settings for Surveillance and Patrol

Pheromone	Update	Prop	Evap	Threshold	Deposit
Visited (Φ_v)	12	0	0.3	1.00E-35	2
Lawn (Φ_l)	1	0.75	0.03	1.00E-300	20

The only target acquisition swarming function modeled in this study was the requirement to cue additional sensors to confirm or identify a target. The algorithm is described in Figure 2. There are two kinds of ASVs in this scenario. *ASVdet* has a sensor that can detect, but not identify targets. *ASVid* has a sensor that can both detect and identify targets. If ASVdet detects a target then it creates a Red avatar to represent the target. The Red avatar begins depositing *NeedsID* pheromone.

ASVid is more attracted to NeedsID pheromone than to Lawn pheromone, so in the presence of NeedsID it will climb the NeedsID gradient and identify the target. Once identified, the Red avatar will stop pumping NeedsID pheromone. ASVid deposits a large amount of Visited pheromone to keep other units from being attracted to the remaining (decaying) NeedsID pheromone in the area. After identification the target is set up for tracking or ignored if it is found to be harmless.

Table 2 lists the pheromone settings for target cueing.

ASVid uses a different interpreting equation than ASVdet:

ASVid:

$$V(p) = s(\Phi_l, p) - s(\Phi_v, p) + 10s(\Phi_t, p) + 2x10^{11} s(\Phi_n, p)$$

ASVdet:

$$V(p) = s(\Phi_l, p) - s(\Phi_v, p) + 10s(\Phi_t, p)$$

4.3 Target Tracking Algorithm

The target tracking algorithm is designed to allow targets to be tracked continuously or intermittently by the ASVs. A Red avatar estimates the behavior of the Red target in between sensor contacts. The Red avatar continually deposits Tracking pheromone. However the ASV deposits a large amount of Visited pheromone when the Red target is detected (see Figure 3).

Table 2: Pheromone Settings for Sensor Cueing in Target Acquisition

Pheromone	Update	Prop	Evap	Threshold	Deposit
Tracking (Φ_t)	6	0.75	0	1.00E-300	100
Visited (Φ_v)	12	0.1	0.3	1.00E-35	2
Lawn (Φ_l)	1	0.75	0.03	1.00E-300	20
NeedsID (Φ_n)	1	0.8	0.25	1.00E-300	200

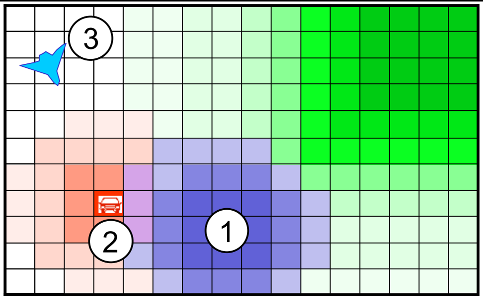

Figure 3: Pheromone Tracking Algorithm – 1. ASV acquires target and large Visited pheromone deposit repelling other ASVs, 2. Red avatar estimates movement, deposits "Tracking" pheromone that eventually overcomes strength of Visited pheromone, 3. Nearby ASV is more attracted to Tracking pheromone than repelled by Visited pheromone and climbs gradient to reacquire the target.

This "kicker" deposit is designed to cause ASVs to stay away from the Red target after it has been acquired. Once the kicker evaporates, or the Red avatar moves out from under its protective cloud, its Tracking pheromone will again attract ASVs to come and establish another contact to update its track. By varying the amount of the Visited kicker deposit one can vary the revisit frequency for the tracking.

Table 3 gives the pheromone settings. These settings result in a revisit period of 200 seconds for a Red target moving at 3 kph and 20 Blue units moving at 90 kph on a 200 x 200 grid.

The interpreting equation is:

$$V(p) = s(\Phi_l, p) - s(\Phi_v, p) + 10s(\Phi_t, p)$$

5. Experiments and Results

This section describes a series of experiments that were conducted to verify that the swarming algorithm could meet the functional requirements of the military scenarios.

5.1 Surveillance Coverage Perrformance

Purpose: Determine the performance of Blue ASVs in covering all the places in a surveillance area.

Area: 20 km x 20 km, gridded with 0.1 x 0.1 km squares

Blue ASV units: 30 @ 90 kph, corner start position

Blue behavior: surveillance algorithm.

Total Runs: 100 random seeds

The coverage results are plotted in Figure 4. The plot of the fixed pattern search shows how a pre-programmed coverage pattern (where each unit is assigned an area to sweep) would perform under the same circumstances. Initially it performs worse than the pheromones as the units move into their starting positions, but eventually it performs better since there is no overlap in their paths. The average time to reach 95% coverage by the swarming algorithm is about 50% longer than a programmed path. A stochastic swarming algorithm will never perform as well as a programmed pattern that can be designed for minimal overlap. But the performance of the swarming algorithm shows some good characteristics:

- It has a fairly tight bound on performance (note the narrow bands until coverage reaches 80%)
- The performance is linear in most of the region until coverage reaches about 80%.

Table 3: Pheromone Settings for Target Tracking

Pheromone	Update	Prop	Evap	Threshold	Deposit
Tracking (Φ_t)	6	0.75	0	1.00E-300	100
Visited (Φ_v)	12	0.1	0.3	1.00E-35	2
Lawn (Φ_l)	1	0.75	0.03	1.00E-300	20

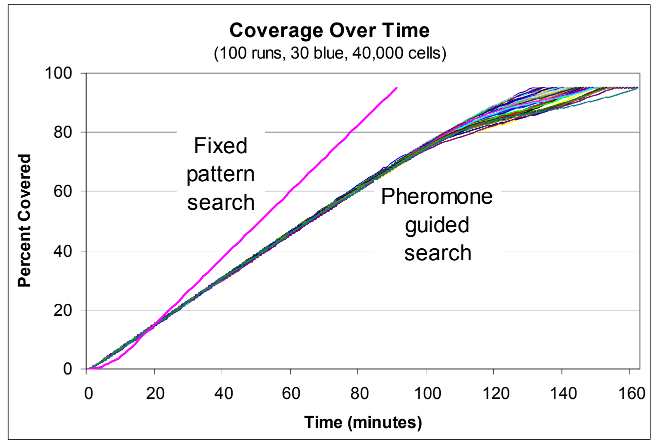

Figure 4: Surveillance Area Covered Over Time - the chart shows each of the 100 runs individually. The fixed pattern search partitions the area equally the ASVs that then perform a sweep pattern search.

It should be noted that when surveying for mobile targets it is not important to visit every point in the surveillance space. In fact a major advantage of a stochastic algorithm over a programmed pattern is that it is unpredictable and an adversary is less likely to be able to determine when to remain hidden.

Figure 5 shows a series of snapshots of the simulation as it progresses. All units begin in the bottom right corner of the grid. They expand out diagonally across the space sweeping

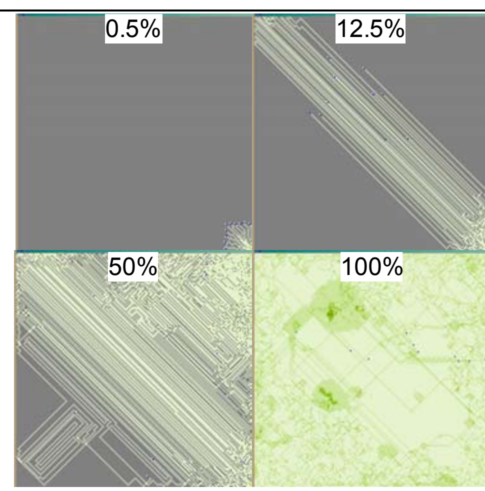

Figure 5: Swarm Surveillance Coverage - 10 ASVs survey a 200 x 200 grid. The swarm begins sweeping diagonally, eventually doing loops and other maneuvers as it seeks out remaining unsurveyed areas. The darkness of the color indicates the number of times a place has been visited. The number on each plot indicates the level of coverage achieved.

back and forth, emergently executing loops and other maneuvers as more of the area is covered.

5.2 Target Acquisition

Purpose: Determine ability of Blue ASVs to find mobile Red units.

Area: 20 km x 20 km, gridded with 0.1 x 0.1 km squares

Blue units: 10, 20, 30 @ 90 kph, random start positions

Red units: 10 at 3kph, random start positions

Blue behavior: Use surveillance algorithm. When Blue lands in a sector containing a moving Red, the Red is considered detected with probability Pd.

Red behavior: Red picks a random point in the area to move to. Red moves at 3 kph until it reaches that point and then rests for one hour. At-rest detection probability: 0. In-move detection probabilities: 0.5, 0.75, and 1.0

Total Runs: 81 = 9 random seeds x 3 Blue swarm sizes x 3 Pd's.

Figure 6 shows all the results. Increasing the number of Blue ASVs can have a dramatic improvement in performance. The time to find 10 Red units in a 40,000 cell grid was decreased from 20 hours to 5.5 hours by increasing the number of units from 10 to 30. There is also a fall off in performance that would be expected. Eventually adding more ASVs will not significantly reduce the time to detect all the Red units. The effect of varying the probability of detection is roughly linear as would be expected. It should be noted that since Red is hiding on average 33% of the time, the effective Pd's for this experiment are 33%, 50%, and 66%.

5.3 Discontiguous Area Surveillance

Purpose: Determine how effective Blue is at distributing assets and covering discontiguous areas of interest (AOIs).

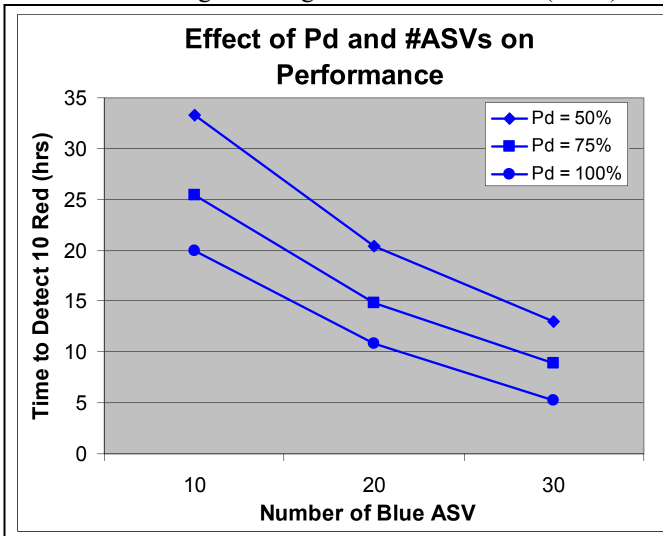

Figure 6: Effect of number of Blue and probability of detection on detecting all 10 mobile Red units – The chart shows the average time to detect all 10 Red mobile units in a 40,000 grid area using different numbers of Blue ASVs. Each plot represents a different probability of detecting Red when Red is not hidden and a Blue ASV occupies the same cell as Red.

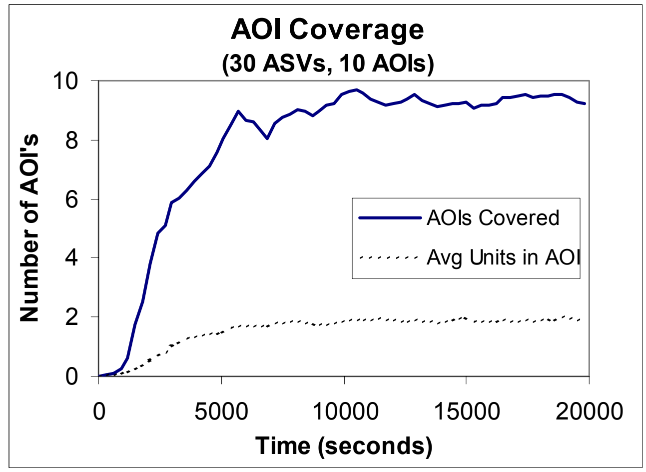

Figure 7: Coverage of the 10 AOIs Over Time (20 point moving average).

Area: 20 km x 20 km, gridded with 0.1 x 0.1 km squares

AOIs: 10, each 1 km x 1 km in size, random positions

Blue units: 30 @ 90 kph, corner start position

Blue behavior: Use surveillance algorithm.

Total Runs: 13 random seeds

Figure 7 shows the average coverage of the ten AOI's over time. The system converges exponentially to the asymptotic state. The asymptote itself is high, maintaining coverage of nine or ten of the AOI's most of the time. On average 2 ± 0.27 ASV's are assigned to each AOI. The low standard deviation shows the effective allocation of ASV's over the AOI's. On average 20 ASVs are within the AOIs and 10 ASVs are surveying either immediately outside or between the AOIs. This provides a population that is able to keep the AOIs covered in the case of failure.

Figure 8 depicts the simulation as it progresses. One can see the units finding and then surveying the different AOIs.

Some of the challenges in this problem include:

- Need to keep too many units from remaining in AOI in

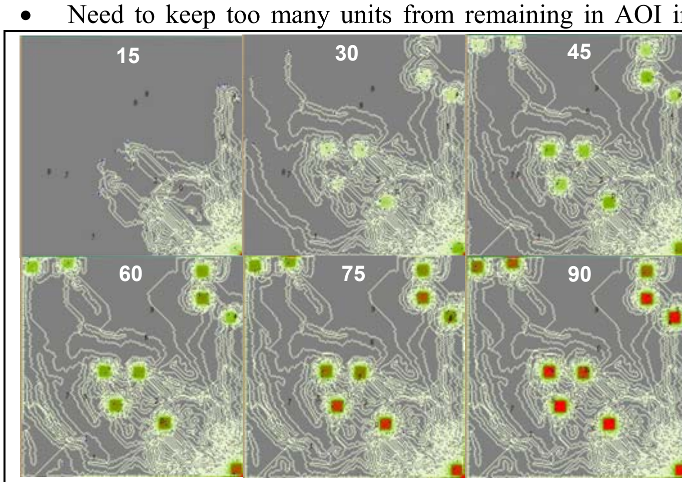

Figure 8: Surveillance of 10 AOIs - The six displays show the history of ASV coverage. The number elapsed minutes is shown. The colors indicate the number of ASV's that have visited that location. By 45 minutes all AOIs are under surveillance

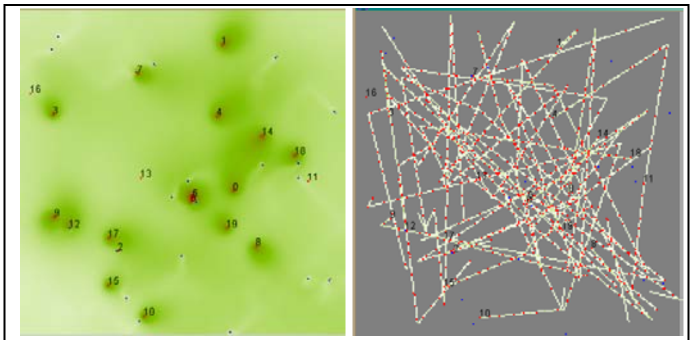

Figure 9: Blue Tracking Result - Left pane shows the tracking pheromone field being emitted by 17 Red units (numbered) being tracked. The right pane shows the Red tracks. Dark sections on the track indicate where the unit was acquired by Blue and white is the interpolated movement in between acquisitions.

the lower right where the all the units start

- Make sure the two AOIs in upper left are found and attract enough units for surveillance.

The algorithm performed well on a difficult problem: trying to find all the AOIs and then balance the number of ASVs that were allocated to surveying each. None of the swarming units knew the number of AOIs or where they were located. Still they were able to self organize, find the AOIs and distribute their numbers across them to maintain regular surveillance.

5.4 Intermittent Tracking

Purpose: Determine effect of number of Blue on efficiency of tracking Red.

Area: 20 km x 20 km, gridded with 0.1 x 0.1 km squares

Blue units: 10, 20, 30 @ 90 kph, corner start position

Red units: 20 @ 3 kph, random start positions

Blue behavior: Use tracking algorithm. When Blue occupies same cell as moving Red, Red is detected with 100% probability. If the Red unit does not have an avatar, then Blue requires an additional 600 seconds of identification time and a Red avatar is created to track the Red unit.

Red behavior: Red picks a random point in the area to move to. Red moves at 3 kph until it reaches that point and then rests for one hour. At-rest detection probability: 0. In-move detection probability: 1.0.

Red Avatar: Use tracking algorithm. If Blue has not revisited the Red unit within 900 seconds the Red avatar is removed and the Red unit is no longer in the tracked state.

Total Runs: 27 := 9 random seeds x 3 Blue unit sizes

Figure 9 shows images from the simulation. The left pane shows the tracking pheromone being emitted by the 17 Red units currently being tracked. Units 11, 13, and 16 have not yet been detected by Blue. The other dots are the Blue units. The right pane shows the Red tracks in two colors: dark sections indicate where Blue acquired the target and white indicates periods in between acquisitions. The fairly even spacing of the acquisition points along the track indicates that Blue is maintaining a good track on the vehicles.

Figure 10 shows the percentage of the 20 Red units tracked over time. 30 Blue units were able to find and track 90% of the Red units with only a 0.5% track loss (percentage of times that a reacquisition failed to occur and the track was lost). 20 Blue units trying to find and track 20 Red units is a fairly difficult problem (since the Blue units need to continue surveying when they are not trying to find Red again for another track). But the

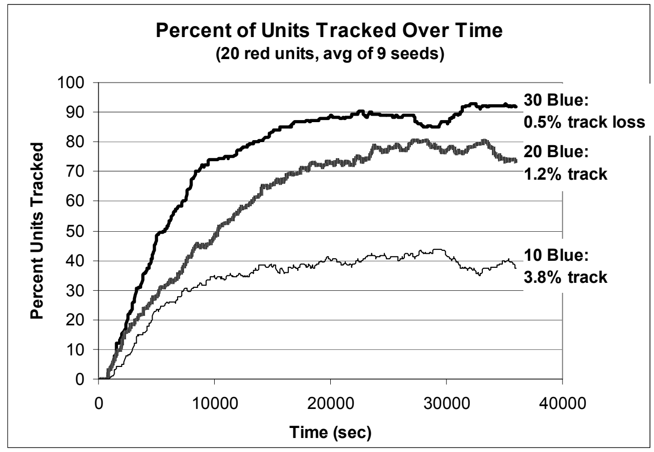

Figure 10: Blue Efficiency in Tracking Red Over Time

algorithm performed well, tracking 80% of the Red units with less than 1.2% track loss. When the number of Blue units is reduced to 10, only 40% of Red is tracked with a 3.8% track loss. There appear to be two factors contributing to the 40% figure: Blue is distracted with the tracking task and hence unable to continue surveying to find the other Red units and once they are found, they are more likely to lose the track.

Other experiments demonstrated that by varying the amount of the "kicker" deposit made by Blue, one can fairly accurately control the time between re-acquisitions. This allows one to vary the frequency of how often a target is revisited based on how important that target is.

5.5 Target Cueing

Purpose: Evaluate the performance of the cueing algorithms. Some Blue entities need to 'cue' other Blue entities to perform a task. Blue swarm entities are divided into two types. One type has a simpler sensor package that can only detect Red units, the other type can both detect and identify Red units.

Area: 20 km x 20 km, gridded with 0.1 km x 0.1 km squares (i.e., 200 x 200 grid)

Blue units: 10, 20, 30 - with 50% being 'ASVdet' and 50% being 'ASVid', 90 kph, corner start position

Red units: 20 @ 3 kph, random start positions

Blue behavior: Use tracking algorithm. When Blue occupies same cell as moving Red, Red is detected with 100% probability. If the Red unit does not have an avatar a Red avatar is created to track the Red unit and Blue deposits 'NeedsID' pheromone if the Blue unit is 'ASVdet' otherwise Blue waits 600 seconds to complete identification.

Red behavior: Red picks a random point in the area to move to. Red moves at 3 kph until it reaches that point and then rests for one hour. At-rest detection probability: 0. In-move detection probability: 1.0.

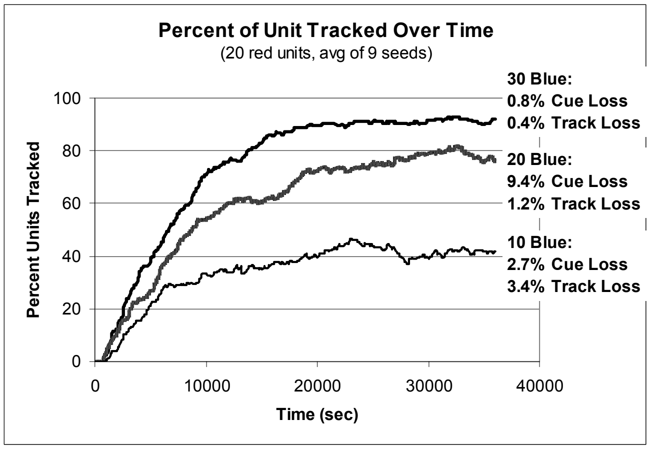

Figure 11: Sensor Cueing - chart shows percentage of Red units tracked over time when initial identification requires a special sensor confirmation. Only half of Blue units have the sensor that can identify Red.

Red Avatar: Use tracking algorithm. If Blue has not revisited the Red unit within 900 seconds the Red avatar is removed and the Red unit is no longer in the tracked state.

Total Runs: 27 := 9 random seeds x 3 Blue swarm sizes

In this experiment only half of the Blue units have the required sensor to identify Red. The other Blue units can detect, but not identify Red. When ASVdet finds a Red unit, an ASVid must come over and positively identify the Red target before it can be placed on the tracking list. Once it has been identified, and as long as the track is not lost, any Blue unit can reacquire the target to update the track (i.e. identification is only required for new targets found or rediscovered old targets).

Figure 11 shows the same information as Figure 10: tracking performance over time. The additional requirement of the second sensor for identification does not appear to affect either the tracking percentage (90%, 80%, and 40% in both this experiment and in the tracking experiment) or the track loss percentage. The number of Blue does have a noticeable impact on the number of cueing requests (requests for identifying sensor) that were not fulfilled. With 30 units only 0.8% of the cue requests went unsatisfied, while with 10 Blue units 12.7% of the cue requests went unsatisfied. An unsatisfied cue request does not count as a lost track since the track was never established. But an unsatisfied cue request does mean that a unit previously detected, will now be lost again and have to be reacquired through surveillance before it can be tracked. So a failure in cueing should result in a slightly lower percentage of total Red units being tracked. This effect must be small, since it does not appear in the data as shown.

6. Demonstration

In October 2004 Altarum, Johns Hopkins University APL, and the Army Research Laboratory demonstrated the use of these swarming algorithms to control a heterogeneous population of air and ground unmanned vehicles in an urban combat scenario at Aberdeen Proving Grounds.

The demonstration used four robots controlled by APL's algorithms, a mock urban area, and two Unmanned Air Vehicles (UAVs) controlled by Altarum's digital pheromone technology. The demonstration focused on the swarming algorithms that control and coordinate the behaviors of the heterogeneous mix of vehicles.

The unmanned ground vehicles were research quality robots made by iRobot, Inc. All four robots used short range fixed acoustic sensors, laser range finders for obstacle detection and avoidance, and commercial GPS receivers for localization.

The air vehicles were modified Mig 117 Bravo target drones with a 6 ft wingspan. The basic airframe was fitted with a modern engine, an autopilot by Micro-Pilot, and low light or infrared camera. The autopilot was taught to take-off, hand launch, fly, and land completely autonomously.

Altarum's pheromone algorithms controlled and coordinated the flight of the two UAVs as they performed continuous surveillance over an urban area looking for potential adversaries. The two air units worked together to ensure even, thorough, and continuous coverage of all areas in the surveillance region while avoiding any collisions. They also provided patrol coverage of a mock convoy as it moved through the area.

While the UAVs surveyed a broad area over the airfield, the ground robots surveyed and patrolled around some mock buildings set up for the demo. During the demonstration, one of the ground robots failed. The other ground robots were able to dynamically readjust their patrol patterns to accommodate the missing unit without any intervention by the operator. This unplanned event helped to demonstrate the robustness of these algorithms to unexpected events.

The demonstration showed cooperative behavior between the air and ground units when the identity of a potential adversary detected by one of the UAV's was automatically confirmed by one of the ground robots with a special sensor capable of target identification.

The actions of the vehicles were not scripted as evidenced by their adapting to the unplanned failure of one of the ground robots. Rather than specify each vehicle's task, the operator simply gives a high level command to the whole swarm, such as "survey this area and track any identified targets" or "patrol around this convoy". The robots autonomously configured themselves to determine which robot would perform what task in order to accomplish the overall objective. The operator was free to monitor their behavior, receive their reports, and provide additional guidance as needed when priorities or mission objectives changed. The swarm did not need any special configuration to meet a wide variety of mission requirements, respective of the operating environment or the number and type of vehicles involved.

7. Conclusions

At the start of this study there was concern about the whether the wide range of scenarios and the requirements they placed on the swarm would result in a large variety of algorithms being required. This study was able to demonstrate that a single pheromone mechanism can be used to perform all the functions required by these scenarios. The surprising versatility arising from such simple mechanisms is one of the more promising aspects of this new class of algorithm.

The mechanism proved to be surprisingly robust to large variations in the parameter settings. Certain parameters (such as the Lawn evaporation and propagation factors), had a greater influence than others, but the mechanism performed well even when those were varied by a factor of 10 or 100.

Adding a new function typically involved at most

- Adding a new pheromone
- Adding a new term to the interpreting equation
- Conducting some experiments to get the right settings

The stigmergic swarming algorithms appear quite promising as a means to control a wide variety of important behaviors for a swarm. They are robust against a wide variety of scenarios, do not require extensive tuning, and are effective in controlling both large and small swarms distributed over large areas. This study has laid the groundwork for future studies and implementation tests.

Acknowledgment

This study was supported by the Decision Support Center of ASD NII. The views and conclusions in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Department of Defense, or the US Government. This study was supported and monitored by the Air Force Research Laboratory (AFRL) under contract number F30602-02-C-0196.

References

[1] J. Arquilla and D. Ronfeldt. Swarming and the Future of Conflict. DB-311, RAND, Santa Monica, CA, 2000. Available at http://www.rand.org/publications/DB/DB311.

[2] E. Bonabeau. Swarm Intelligence. In *Proceedings of Swarming: Network Enabled C4ISR*, ASD C3I, 2003.

[3] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. New York, Oxford University Press, 1999.

[4] S. Brueckner. *Return from the Ant: Synthetic Ecosystems for Manufacturing Control*. Thesis at Humboldt University Berlin, Department of Computer Science, 2000.

[5] S. Camazine, J.-L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau. *Self-Organization in Biological Systems*. Princeton, NJ, Princeton University Press, 2001.

[6] J. Dubik, R. Richards, and G. Trinkle. Joint Concept Development and Experimentation. In *Proceedings of Swarming: Network Enabled C4ISR*, ASD C3I, 2003.

[7] S. J. A. Edwards. Swarming on the Battlefield: past, Present, and Future. MR-1100-OSD, RAND, Santa Monica, CA, 2000.

[8] S. J. A. Edwards. Military History of Swarming. In *Proceedings of Swarming: Network Enabled C4ISR*, ASD C3I, 2003.

[9] D. Inbody. Swarming: Historical Observations and Conclusions. In *Proceedings of Swarming: Network Enabled C4ISR*, ASD C3I, 2003.

[10] H. V. Parunak. Making Swarming Happen. In *Proceedings of Swarming: Network Enabled C4ISR*, ASD C3I, 2003.

[11] H. V. D. Parunak. 'Go to the Ant': Engineering Principles from Natural Agent Systems. *Annals of Operations Research*, 75:69-101, 1997.

[12] H. V. D. Parunak and S. Brueckner. Swarming Coordination of Multiple UAV's for Collaborative Sensing. In *Proceedings of Second AIAA "Unmanned Unlimited" Systems, Technologies, and Operations Conference*, AIAA, 2003.

[13] H. V. D. Parunak and S. A. Brueckner. Stigmergic Learning for Self-Organizing Mobile Ad-Hoc Networks (MANET's). In *Proceedings of Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'04)*, pages 1324-1325, 2004.

[14] H. V. D. Parunak, M. Purcell, and R. O'Connell. Digital Pheromones for Autonomous Coordination of Swarming UAV's. In *Proceedings of First AIAA Unmanned Aerospace Vehicles, Systems,Technologies, and Operations Conference*, AIAA, 2002.

[15] SMDC-BL-AS. Swarming Unmanned Aerial Vehicle (UAV) Limited Objective Experiment (LOE). U.S. Army Space and Missile Defense Battlelab, Studies and Analysis Division, Huntsville, AL, 2001. Available at https://home.je.jfcom.mil/QuickPlace/experimentation/PageLibrary85256AB1003BBEA7.nsf/h_0036FB98FFD2ACCA85256AB2004161B0/D7680995272C266B85256B20004E1BF0/?OpenDocument.