

Evolving Adaptive Pheromone Path Planning Mechanisms

John A. Sauter
ERIM

PO Box 134001
Ann Arbor, MI 48113-4001
1-734-623-2513

jsauter@erim.org

Robert Matthews
ERIM

PO Box 134001
Ann Arbor, MI 48113-4001
1-734-623-2510

rmatthews@erim.org

H. Van Dyke Parunak
ERIM

PO Box 134001
Ann Arbor, MI 48113-4001
1-734-623-2509

vparunak@erim.org

Sven Brueckner
ERIM

PO Box 134001
Ann Arbor, MI 48113-4001
1-734-623-2529

sbrueckner@erim.org

ABSTRACT

Agents guided by synthetic pheromones can imitate the behavior of insects in tasks such as path planning. These systems are well suited to problems such as path planning for unmanned robotic vehicles. We have developed a model for controlling robotic vehicles in combat missions using synthetic pheromones. In the course of our experimentation, we have identified the need for proper tuning of the algorithms to get the desired behavior. We briefly describe the synthetic pheromone mechanisms for dynamically finding targets and planning safe paths. Genetic algorithms for automatically tuning the behavior of the pheromone equations are described.

Keywords

Ant algorithms, pheromones, path planning, unmanned vehicle control, mobile robots, genetic algorithms, fine-grained agents.

1. INTRODUCTION

Synthetic pheromones have been applied to a number of domains. In DARPA's JFACC program [12], we use pheromones to control air combat missions, with special emphasis on unmanned air vehicles. In the course of our research, we identified several difficulties with generalizing this technique so that human commanders could easily apply them under a number of battlefield scenarios. We report on recent research that has identified promising techniques to address these challenges. This report describes pheromone-based movement control (Section 2), reviews the two evolutionary mechanisms we have developed for tuning these systems (Section 3), and discusses the results (Section 4).

2. SYNTHETIC PHEROMONE MECHANISMS

Agility in military operations requires units to learn of changes in opportunities and threats quickly. Missions must be re-planned dynamically to take advantage of new information. Funneling intelligence into a centralized location that computes a new

solution is too slow and subject to failure. Ant colonies are able to quickly adapt to changing food sources without such centralization. They construct networks of paths with pheromones (evaporative scent markers) that connect their nests with food sources. Mathematically, these networks form minimum spanning trees [9], minimizing the energy ants expend in bringing food into the nest. This optimal structure emerges from the actions of individual ants acting on their environment.

The potential of insect models for multi-agent coordination and control is receiving increasing attention. [13] and [2] outline several mechanisms, including pheromones, that lend themselves to practical application. [5] and [7] offer theoretical discussions with simple applications, and [6] shows how these techniques can play a credible game of chess.

The most mature practical use of pheromone techniques is in routing telecommunications packets (e.g. [1] and [10]). Application of these techniques to moving physical entities can be traced to the Cascade system [11], a self-routing modular material handling system. The application of these techniques to routing and load balancing is being evaluated in the ESPRIT MASCADA project [14]. Steels proposed similar mechanisms for coordinating small robots used in exploring remote planets [17]. Dorigo and colleagues [4]; [8] have applied these mechanisms to a range of optimization problems including the traveling salesperson problem and the quadratic assignment problem.

Pheromone fields are a type of potential gradient. The notion of movement guided by a potential gradient has been applied to robot navigation [15], which automatically maps from a given distribution of targets and obstacles to a movement plan. Many different fields are used to represent different classes of targets and obstacles.

We are interested in using a pheromone field to guide unmanned robotic vehicles (URV's) through the battlespace that includes targets and threats. In these scenarios, robotic vehicles seek to find safe ingress and egress paths to targets that are defended by various threats. To be useful in warfighting, a fielded system requires four characteristics:

Diverse.—It must fuse information of various types and from various sources, including targets to be approached, threats to be avoided, and the presence of other URV's with whom coordination is required.

Distributed.—Centralized processing of a potential field imposes bottlenecks in communications and processing, and presents single points of failure. Ideally, the potential field should be distributed and stored close to the point it is generated and used.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Conference '00, Month 1-2, 2000, City, State.
Copyright 2000 ACM 1-58113-000-0/00/0000...\$5.00.

Dynamic.—The battlespace is an uncertain and rapidly changing environment, and the methods and architecture used to construct and maintain the field must be able to incorporate and react to such changes rapidly.

Easy to Use.—The commander thinks in terms of mission and task objectives. He seeks to maximize the objectives obtained and minimize the losses while acting within the rules of engagement.

The architecture described in [12] inspired by insect pheromones satisfies the first three requirements, while the research we discuss here addresses the fourth. These approaches will eventually allow the commander to specify their objectives to the system and the system will evolve a solution that dynamically meets those objectives.

2.1 Basic Mechanisms

Our implementation of synthetic pheromones has four components: a distributed network of *place* agents that maintain the pheromone field and perform aggregation, evaporation, and diffusion, the Red *surrogate* agents representing the enemy targets and threats, the *walkers* representing the Blue or friendly URVs, and *ghosts* which wander over the place agents building pheromone paths. In JFACC, we tile the physical space with hexagons, each representing a place agent with six neighbors but in principal both regular and irregular tiling schemes can be employed. The underlying mathematics of the pheromone field, including critical stability theorems, is described in [3].

Battlefield intelligence from sensors and reconnaissance activities causes the instantiation of Red surrogate agents 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 walker agent is associated with one place agent at any given time. It can read the current strength of pheromones at that place and at each of its neighbors, and can deposit its own pheromones into the place. A walker moves from one place to another by spinning a roulette wheel whose segments are weighted according to this set of strengths.

The walkers continually send out ghost agents to travel over the distributed network of place agents and find the safest, shortest path to a target. While walkers move from place to place at the speed of the URV they represent, ghost agents move at the speed of the network. Ghost agents can also sense and deposit pheromones. They look for targets and then build and reinforce a pheromone path from the walker to the target. The walker then follows the path laid down by the ghost agents to the target. The mechanism employed by the ghosts to build paths to targets under different scenarios is the subject of this research.

2.2 Ghost Mechanism

In our applications, the ghost's choice function weights the various input pheromones to create a single "net pheromone" or attractive force for each neighbor that is used in weighting the roulette wheel for determining the ghost's next move. The basic pheromone flavors are:

- *RTarget*: deposited by the Red surrogate for a target (e.g. Red headquarters).
- *RThreat*: deposited by the Red surrogate for a threat (e.g., Red air defense)

- *GTarget*: deposited by a ghost that has encountered a target and is returning to the URV.
- *GNest*: deposited by a ghost that has left the URV and is seeking a target.

In addition, the ghost may know *Dist*, an estimate of the distance to the target when the target location is known.

We experimented with several different forms of the equation. Manual manipulation of the equation yielded the current form:

$$\frac{\theta \cdot RTarget + \gamma \cdot GTarget + \beta}{(\rho \cdot GNest + \beta)(Dist + \phi)^{(\delta + \alpha(RThreat+1))} + \beta}$$

Table 1 lists the tunable parameters in the equation and the effect that increasing the parameter has on the ghost's behavior. 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. Even if a skilled practitioner were able to tune the equation by hand, the system would still be impractical for end users who don't think of their problem in terms of α , β , and γ . It was this that led us to investigate the possibility of using evolutionary methods to tune the parameters of the equation. The next section describes the results of several evolutionary approaches to tuning the equation.

Table 1: Tunable Parameters and their Effect on Ghosts

Parm	Effect of Increasing
α	Increases ghost repulsion to threat at longer distances to target
δ	Narrows the beam of ghosts moving to a specific target
ϕ	Increases threat avoidance near target
ρ	Increase ghost exploration (by avoiding <i>GhostNest</i> pheromone)
θ	Increases attraction to <i>RTarget</i> pheromone
β, γ	Avoid discontinuities, offsets null pheromone levels

3. Evolving the Parameters

[16] describes a number of approaches that could be taken to evolving parameter assignments and presents some preliminary results. We consider two approaches here: some simple evolutionary strategies and a genetic algorithm.

In the original JFACC program over a dozen different test scenarios were used. Each scenario consisted of a particular configuration of targets and threats representing various degrees of difficulty for the ghosts. Tests were performed with static environments and dynamic environments (Red and Blue units mobile) and with complete and incomplete knowledge (knowledge of Red forces incomplete and uncertain). No single set of parameter values could solve all the test cases. Manual tuning was required to find the best set of parameters for each scenario.

For the purposes of the evolutionary experiments, we chose six representative static tests with complete knowledge. Figure 1 shows four of the cases we report here. The shortest safest path is

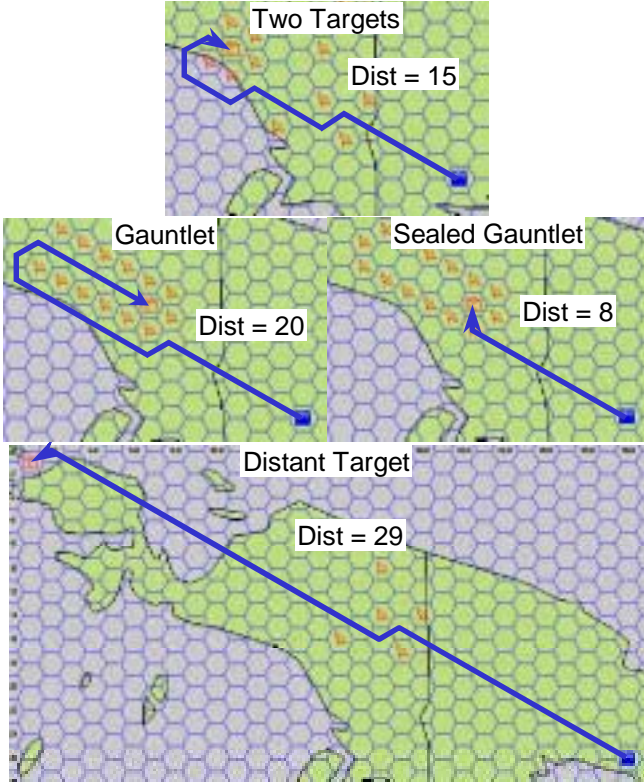


Figure 1: Four of the test scenarios used

marked along with the distance of that path. Two Targets presents two equidistant targets with different strengths (the second target is not shown). Gauntlet is a challenging problem since ghosts must find the entrance some distance from the target and then similarly find their way back out. This problem proved to be intractable with conventional potential field methods. The sealed gauntlet forces the agents to eventually decide to pierce the threat barrier to reach the target. Finally the Distant Target scenario tests the ability of the ghosts to find distant targets and then build and maintain a stable path over that distance. The other two cases not shown were sealed versions of the Two Targets and Distant Target scenarios.

3.1 Evolutionary Strategies

The first algorithm tested was a simple evolutionary strategy. The walker sends out three new ghosts every ten time steps. Ghosts are initially created with random values (uniform distribution about a mean) for each parameter. Ghosts have a life of 1000 time steps so the total population of ghosts reaches a maximum of 300. The ghosts that return to the nest after finding a target are placed in a queue for the remainder of their life. While there are two or more ghosts in the queue, new ghosts are created by randomly selecting parameters from two parent ghosts in the queue. Every 100 choices, a random value is chosen. Parents were either chosen from the head of the queue only (labeled “Queue ES”) or from a round robin of all the ghosts in the queue (“Round Robin ES”). Ghosts that found the shortest paths would remain in the queue longer and therefore pass on more of their parameters to offspring. The algorithm also included a means to explore values outside the initial range of the parameters in case that range was too narrow.

These experiments were compared to the results achieved when all the ghosts used the best set of hand-tuned parameters and when all the ghosts were given random parameter settings.

Despite the simplicity of the algorithm and the lack of accounting for the effect of threats, these ghosts performed surprisingly well, solving the scenarios correctly in almost every trial. Figure 2 plots the level of G_{Target} pheromone next to the walker for the Two Targets test case. This is a measure of the strength of the path and the guidance provided by the ghosts to the walker. The

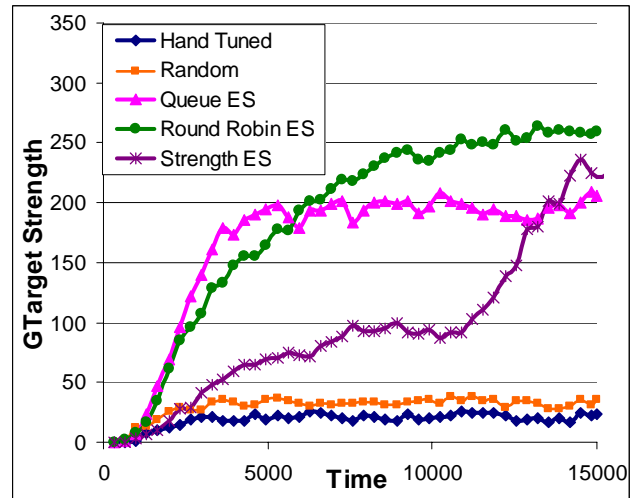


Figure 2: Simple evolutionary strategies outperform the best hand tuned results on Two Targets

evolutionary strategies far outperformed the hand-tuned results building stronger paths in less time. The round robin was slightly better than the queue strategy. In fact the random settings slightly outperformed the best hand-tuned result.

A third evolutionary strategy was developed because the initial strategies did not always solve all six configurations nor did they take into account threats. In this strategy ghosts are given a strength value that is decremented by encounters with threats according to the following equation:

$$s_i = \frac{1}{\sum d_j}$$

where d_j is the strength of a threat (defending unit) encountered by the ghost. The ghosts that return are again placed in a queue for the remainder of their life. But the parents are selected from the two ghosts in the queue with the highest remaining strength (labeled “Strength ES”). With this strategy the ghosts were able to correctly and consistently solve all six test cases, but as shown in Figure 2, they were slower in developing a strong path.

Figure 4 plots the G_{Target} strength of the Strength ES strategy for the four cases depicted in Figure 1. The result of one of the genetic algorithms described below is also shown for reference. Figure 5 shows the strength of the pheromone field at the end of the experiment. One can clearly see that a stronger path to the target has been formed by the Strength ES strategy than by the hand tuned parameters.

All runs were stopped after 40,000 time steps. At the end of the run, the Strength ES strategy had not always settled on a single set of parameters. There were often several “species” of individuals that seemed to evolve out of the population. Figure 3 plots the evolution of the α parameter over the course of the experiment for each of the four test cases. For Two Targets, only two values remained. For the others there were several parameter values at the end of the run. The Distant Target shows an initial period of random parameter selection followed by a gradual narrowing of the number of values of α in the population. The expansion of the initial range can also clearly be seen in this plot.

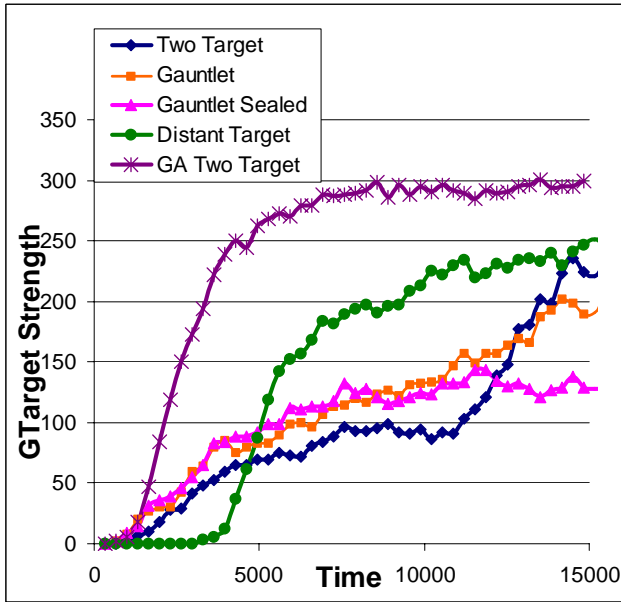


Figure 4: Final ES performance on four of the test cases

3.2 Genetic Algorithm

A genetic algorithm was developed for the ghosts. The first 300 ghosts are created with random values uniformly distributed over

the range 0-50. Ghosts are given a fitness value based on their ability to find targets and avoid threats. The fitness of a ghost is defined by the following equation:

$$f_i = \frac{\sum t_i + N}{\sum d_j}$$

where t_i is the strength of a target found by the ghost and d_j is the strength of a threat (a defending unit) it encounters. N is 1 until the ghost returns back to the walker (the “nest”) when it is set to 10. For our experiments the ghosts sought only one target. The target strength was 10. All threats had strength of 5. When the ghost returns to the walker it no longer moves, but remains in the breeding population for the remainder of its life.

After the first 300 ghosts have been created, the rest of the ghosts are created from a genetic operation. The values for the parameters are encoded as a bit string with 16 bits per parameter for a chromosome of 128 bits (the upper and lower β were encoded separately). Two parents are selected based on a roulette wheel selection from all ghosts in the population weighted by their fitness. The probability that the i^{th} ghost is selected for breeding is $f_i / \sum f$. A single point crossover operation is used to generate the offspring. Single bit random mutations occur with a probability of 1%.

The genetic algorithm was able to solve all six of the test cases, though there was one change required on the penalty function for encountering threats in order to consistently solve the gauntlet problem. The original threat strength of 2 was insufficient to keep the ghosts from occasionally building paths that pierced the threat barrier. Increasing it to 5 solved the problem.

Figure 6 shows the $GTarget$ strength next to the walker. The genetic algorithm was better at building a stronger path faster than the evolutionary strategies in all test cases.

The genetic algorithm also resulted in different species. These species were not always as well defined as the evolutionary strategies. The crossover operation and the bit mutation both

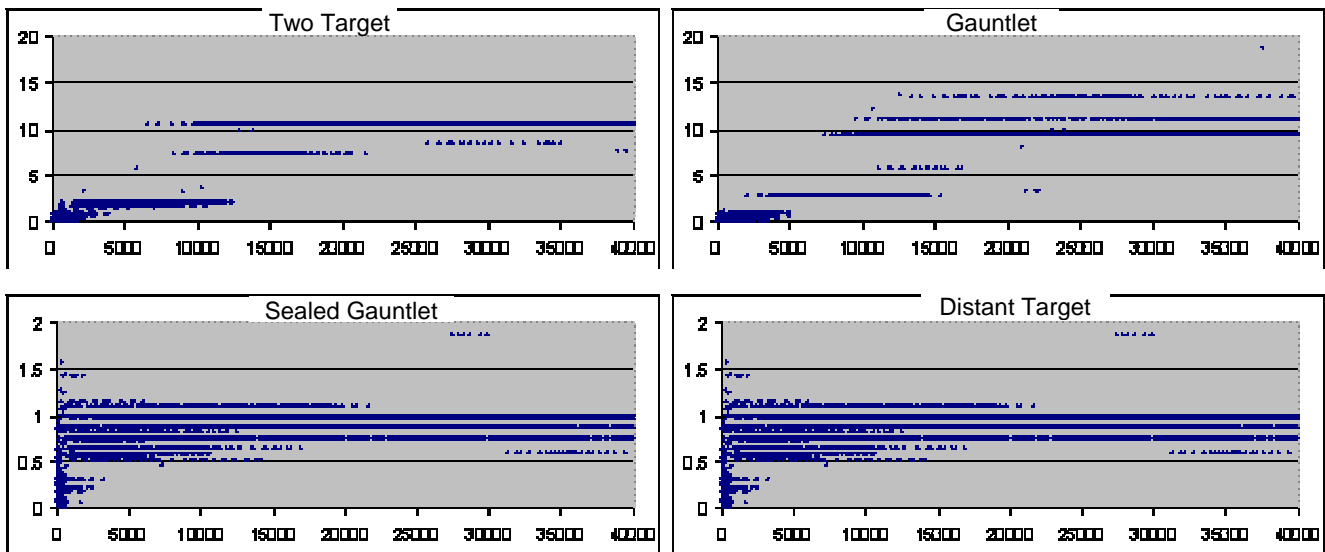


Figure 3: Evolution of the alpha parameter resulted in several dominant individuals

generate new parameters values for the GA population. The ES must wait for the rare bit mutation before it sees a new parameter value appear in the population. Thus the GA explores more values and does not settle as quickly onto a few individuals as the ES does.

Figure 5 shows the final *GTarget* pheromone concentration for the Two Targets scenario using three methods: hand tuned, Strength ES and GA. The Strength ES and GA paths are clearly defined at the end of the experiment (and in fact have been well established long before that time).

Table 2 compares the values for each parameter for the three primary strategies studied: hand tuned (HT), Strength ES (ES), and the genetic algorithm (GA). For the ES and GA columns, the values represent those of the dominant species in the final population. For the distant target scenario, there was no single dominant individual in the final GA population. Surprisingly, though both ES and GA found good solutions, there is little resemblance in the final set of parameters they have chosen.

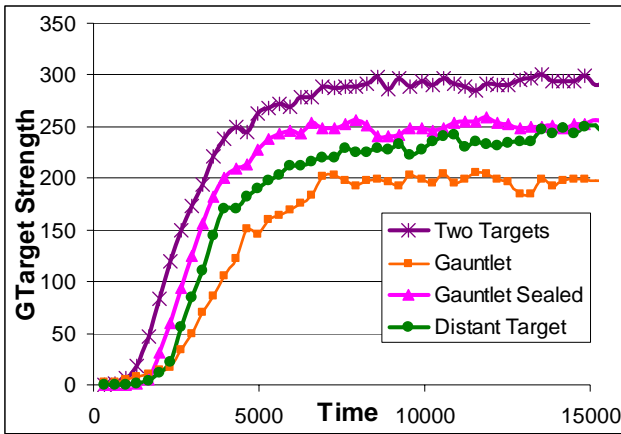


Figure 6: GTarget strength using the genetic algorithm

4. DISCUSSION

The experiments have demonstrated that both the Strength ES and the GA strategies were very effective in solving all six of the test cases. The GA strategy was slightly more successful at building a stronger path more quickly than the Strength ES strategy. Both of them were vastly superior to the hand tuned parameter settings. The results of the experiments were obtained in a matter of

Table 2: Comparison of parameter values for the dominant species

Parm	Two Targets			Gauntlet			Sealed Gauntlet			Distant Target	
	HT	ES	GA	HT	ES	GA	HT	ES	GA	HT	ES
α	10.0	11.1	5.1	10.0	9.5	11.0	3.0	1.0	1.2	10.0	8.0
δ	0.0	1.8	42.1	0.0	1.0	7.9	4.0	6.0	15.3	9.0	25.1
φ	1.5	0.8	3.4	1.5	0.2	0.1	1.5	0.1	0.7	4.0	0.4
ρ	1.0	0.1	7.6	1.0	0.2	47.7	3.0	0.3	35.8	1.0	6.9
θ	1.0	0.5	4.7	0.0	0.8	46.6	1.0	0.3	15.3	0.0	15.2
β	0.5	0.2	24.2	0.5	0.1	36.2	0.5	0.7	26.8	0.5	12.8
γ	0.5	0.7	7.1	0.5	0.8	25.3	0.5	0.6	40.0	0.5	18.2

minutes, while the hand-tuned parameters took over a month to develop.

It is interesting that the GA population tends to reduce its exploratory behavior over time while the ES population still shows some wandering (evidenced by the broader deposit of pheromone in Figure 5). This is good if the intent is to focus on a single target, but it could be a problem if you want to keep exploring and looking for other alternatives.

For example, in the Two Targets dynamic experiment, where the walker follows the path laid down by the ghosts, the ghosts maintained paths to both targets throughout the run, with the stronger of the two paths leading to the stronger of the two targets. The second path acted as a “cache” of useful information. If something happened with the first target (say the appearance of a previously undetected threat in the path, or the destruction of the stronger target by some other force) then the ghosts would be able to quickly reinforce the path to the secondary target since they wouldn’t need to start from scratch. This exploratory behavior is an important component of the adaptability of the population and the GA strategy would need to be investigated further to determine the extent to which it narrows down on a single solution excluding other possibilities.

The GA approach described here is a centralized algorithm. A natural extension is to restrict the breeding population to just those ghosts that make it back to the walker as in the evolutionary strategies. This would keep the computation completely decentralized.

There are some interesting differences between this GA and

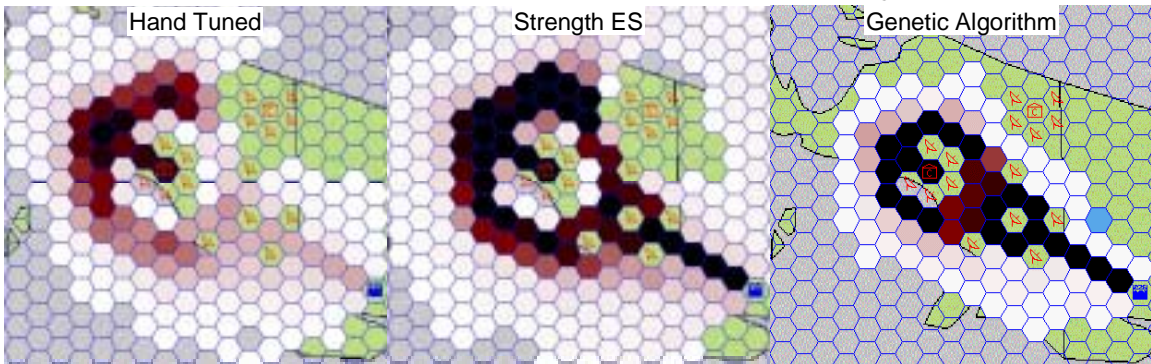


Figure 5: Final pheromone strength for the TwoTargets test case for the three main strategies

traditional GA. Normally in GA the entire population is synchronized. All breeding occurs at the same time and the entire population is replaced in one generation. In our approach, breeding occurs in parallel with the evaluation of the ghosts and only 1% of the population is replaced in each generation. Since the evaluation of an individual can take 100 – 300 time steps (the round trip distance with room for wandering) forcing a complete evaluation cycle before breeding would probably have slowed down the algorithm considerably.

Evaluation of the fitness of an individual is normally performed separately for each individual in the population, or in a tournament where individuals from separate populations compete with each other. Our GA does neither. The ghosts are part of a mixed population. Each of them is depositing pheromones and reacting to pheromones in a common environment. Thus, unfit individuals are depositing pheromones in the same environment being sensed by fit individuals potentially causing the fit individuals to score lower than they would otherwise. This fact initially concerned us. We weren't sure whether ES or GA would even work under those circumstances. However, this particular problem appears to have a number of reasonable solutions (as evidenced by Table 2) so the effect of having a mixed population did not prevent the algorithms from identifying and rewarding the better individuals.

These results lead to an intriguing idea. Can we use evolutionary algorithms to go directly from a statement of a requirement to a solution? In our path planning system ideally the commander would specify commands such as, "Find the strongest target in Sector H and create the shortest path where the probability of kill is less than 0.5%". This information could be fed directly into the objective functions that weight the ghosts for selection as parents. As ghosts encounter threats they would determine their probability of being killed through simple kill probability tables that are readily available. If their probability of kill rises above the threshold specified, they would die and not participate in breeding. Ghosts would be awarded points based on the strength of the target they found in the specified sector. Breeding would be weighted towards the ghosts that returned with the most points, and the least probability of being killed. Their tenure in the breeding pool would be determined by their life remaining so that ghosts that found shorter paths would breed longer and produce more offspring. Now rather than having to figure out how to hand tune the parameters to meet the specified constraints and objectives, the system evolves the parameters that meet those objectives.

5. CONCLUSION

The evolutionary algorithms explored in these experiments have demonstrated the ability to automatically tune the parameters of a pheromone-based path planning system so it can successfully function in a number of test scenarios. These solutions consistently outperformed the best hand tuned parameters that took skilled programmers over a month to develop.

Fine-grained agent systems exhibit complex behaviors. Though they have demonstrated their ability to solve complex problems such as path planning, programming the individual behaviors of the agents to achieve an overall system behavior has proven to be challenging. For this particular system, evolutionary methods seem capable of moving us closer to the goal of specifying the

system behavior and letting the agents evolve their individual behavior to reach that goal.

6. ACKNOWLEDGMENTS

This work was supported in part by the DARPA JFACC program. Our research team included O. Gilmore, M. Nandula, J. Posdamer, E. Feibush, and E. Greene.

7. REFERENCES

- [1] E. Bonabeau, F. Henaux, S. Guérin, D. Snyers, P. Kuntz, and G. Theraulaz. Routing in Telecommunications Networks with "Smart" Ant-Like Agents. Santa Fe Institute, Santa Fe, NM, 1998.
- [2] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. New York, Oxford University Press, 1999.
- [3] S. Brueckner. Return from the Ant: Synthetic Ecosystems for Manufacturing Control. Thesis at Humboldt University Berlin, Department of Computer Science, 2000.
- [4] M. Dorigo, V. Maniezzo, and A. Coloni. The Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics*, Part B, 26(1):1-13, 1996.
- [5] A. Drogoul. De la Simulation Multi-Agents à la Résolution Collective de Problèmes. Ph.D. Thesis at University of Paris IV, 1993.
- [6] A. Drogoul. When Ants Play Chess (Or Can Strategies Emerge from Tactical Behaviors? In *Proceedings of Fifth European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW '93)*, pages 13-27, Springer, 1995.
- [7] J. Ferber. *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Harlow, UK, Addison Wesley Longman, 1999.
- [8] L. M. Gambardella and M. Dorigo. HAS-SOP: Hybrid Ant System for the Sequential Ordering Problem. Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Lugano, Switzerland, 1997. Available at <ftp://ftp.idsia.ch/pub/luca/papers/SOP-tr-idsia-11-97.ps.gz>.
- [9] S. Goss, S. Aron, J. L. Deneubourg, and J. M. Pasteels. Self-organized Shortcuts in the Argentine Ant. *Naturwissenschaften*, 76:579-581, 1989.
- [10] M. Heusse, D. Snyers, S. Guérin, and P. Kuntz. Adaptive Agent-Driven Routing and Load Balancing in Communication Networks. ENST de Bretagne, Brest, FR, 1998.
- [11] H. V. D. Parunak, J. Kindrick, and B. Irish. Material Handling: A Conservative Domain for Neural Connectivity and Propagation. In *Proceedings of Sixth National Conference on Artificial Intelligence*, pages 307-311, American Association for Artificial Intelligence, 1987.
- [12] H.V.D. Parunak, S. Brueckner, J. A. Sauter, and J. Posdamer. Mechanisms and Military Applications for Synthetic Pheromones. *Workshop on Autonomy Oriented Computation*, Montreal, Canada 2001.

- [13] H. V. D. Parunak. 'Go to the Ant': Engineering Principles from Natural Agent Systems. *Annals of Operations Research*, 75:69-101, 1997. Available at <http://www.erim.org/~van/gotoant.pdf>.
- [14] P. Peeters, P. Valckenaers, J. Syns, and S. Brueckner. Manufacturing Control Algorithm and Architecture. In *Proceedings of Second International Workshop on Intelligent Manufacturing Systems*, pages 877-888, K.U. Leuven, 1999.
- [15] E. Rimon and D. E. Kodischek. Exact Robot Navigation Using Artificial Potential Functions. *IEEE Transactions on Robotics and Automation*, 8(5 (October)):501-518, 1992.
- [16] J. A. Sauter, H.V.D. Parunak, S. Brueckner, and R. Matthews. Tuning Synthetic Pheromones With Evolutionary Computing. In *Genetic and Evolutionary Computation Conference Workshop Program (GECCO 2001)*, San Francisco, CA, 2001.
- [17] L. Steels. Cooperation between Distributed Agents through Self-Organization. Vrije Universiteit Brussel AI Laboratory, 1989.