2009

Genetic stigmergy: Framework and applications

Joshua A. Brandoff

GENETIC STIGMERGY: FRAMEWORK AND APPLICATIONS

BY

JOSHUA ADAM BRANDOFF

BS, Binghamton University, 2008

THESIS

Submitted in partial fulfillment of the requirements for
the degree of Master of Science in Systems Science
in the Graduate School of
Binghamton University
State University of New York
2009

Accepted in partial fulfillment of the requirements for
the degree of Master of Science in Systems Science
in the Graduate School of
Binghamton University
State University of New York
2009

December 1, 2009

Hiroki Sayama, Department of Bioengineering, Binghamton University

Craig Laramee, Department of Bioengineering, Binghamton University

Harold Lewis, Department of Systems Science, Binghamton University

Abstract

Stigmergy has long been studied and recognized as an effective system for self-organization among social insects. Through the use of chemical agents known as pheromones, insect colonies are capable of complex collective behavior often beyond the scope of an individual agent. In an effort to develop human-made systems with the same robustness, scientists have created artificial analogues of pheromone-based stigmergy, but these systems often suffer from scalability and complexity issues due to the problems associated with mimicking the physics of pheromone diffusion. In this thesis, an alternative stigmergic framework called 'Genetic Stigmergy' is introduced. Using this framework, agents can indirectly share entire behavioral algorithms instead of pheromone traces that are limited in information content. The genetic constructs used in this framework allow for new avenues of research, including real-time evolution and adaptation of agents to complex environments. As a nascent test of its potential, experiments are performed using genetic stigmergy as an indirect communication framework for a simulated swarm of robots tasked with mapping an unknown environment. The robots are able to share their behavioral genes through environmentally distributed Radio-Frequency Identification cards. It was found that robots using a schema encouraging them to adopt lesser used behavioral genes (corresponding with novelty in exploration strategies) can generally cover more of an environment than agents who randomly switch their genes, but only if the environmental complexity is not too high. While the performance improvement is not statistically significant enough to clearly establish genetic stigmergy as a superior alternative to pheromonal-based artificial stigmergy, it is enough to warrant further research to develop its potential.

Dedication

To my parents, Barbara and Craig Brandoff, my sister Amanda and my lovely Polly. I would not be where I am today without your love, support and understanding.

Acknowledgements

Firstly, I wish to thank Professor Hiroki Sayama for his ability to mix brilliant scholarship with patient mentorship during the development of this thesis. Without his assistance, this work would not exist. I also wish to thank Craig Laramee and Harold Lewis for agreeing to sit on my committee and taking the time to review my work.

I was very thankful to receive the guidance, support and friendship of many members of the Binghamton University community. Ken McLeod was a persistent force of optimism and helped spark within me the fires of innovation. Without Ellen Madison, I would not have made it past my first semester. Thank you Ellen for helping me with all those essential tasks and for your warmth and geniality, which made the department feel like a second home. On his way to changing the world, Guruprasad Madhavan always spared some time to share his vision and inspiration and helped me find my own. David Sloan Wilson helped me understand the true power of evolution as a framework for understanding and bettering the world. His classes were among my most educational, enlightening and enjoyable experiences in graduate school. Justin Garcia, Tom Raway, Kathryn Fletcher, Alda Mizaku and Ashley Raba all deserve my gratitude for their camaraderie and for helping me keep my humorous side alive and well.

I was also fortunate to receive guidance from members outside the University community. Thanks to Chrystopher Nehaniv, Daniel Polani, the program committee and attendees of the 2009 IEEE Symposium on Artificial Life, and the Icosystem Corporation for their thoughtful comments on my past and present work.

Table of Contents

List of Figures

List of Tables

Chapter 1: Background

1.1 Introduction

Originally described by Pierre Huber in 1810 (Holldobler and Wilson, 2009) and named by Pierre-Paul Grassé in 1959 (White, 2005), stigmergy is a system of coordination whereby collective action is achieved through indirect interactions between agents via modifications to their local environment. Unlike many presently engineered human-made systems, stigmergic systems are able to self-organize through simple local interactions and without the guidance of a central coordinator. Stigmergy relies upon a number of interacting feedback loops that define how agents and their environment change as a result of interaction (Figure 1).

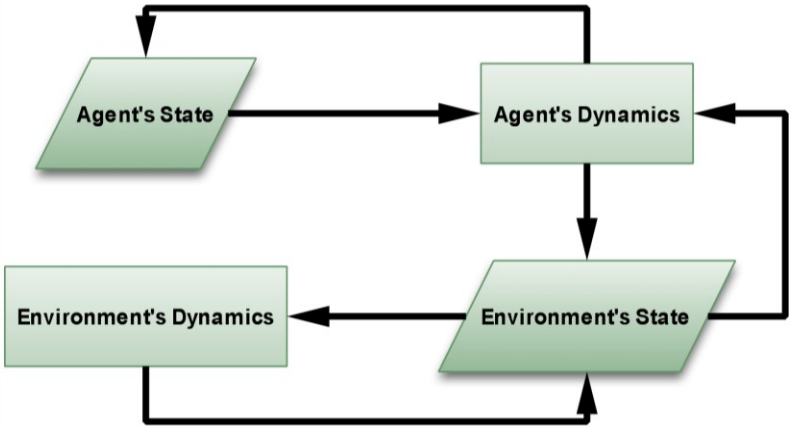


Figure 1: Flow-chart describing the feedback loops in a stigmergic process (based on figure in Van Dyke Parunak (2006), p. 164).

Each agent possesses internal and external states, with the former invisible to the perception of other agents. Agents are able to perceive and modify their environment through a (usually) small number of sensors and actuators. Guiding the agent is a controller program that modifies an agent’s actions depending on the sensed local environment and the agent's internal state. In addition, the controller program itself may be guided by a separate program that changes the agent's interaction dynamics themselves as a function of time or other internal information.

In nature, stigmergy is most visible in social insects, which have graced the Earth for at least 50 million years. More than 90% of the signals used in communication by these insects are through chemicals called pheromones, which can trigger various behaviors in other insects of the same species depending on their type and intensity (Holldobler and Wilson, 2009). Through the process of natural selection, these insects gained the ability to create simple "algorithms" that can use pheromone traces to collectively achieve beneficial actions. These collective actions are “satisficial” rather than optimal in nature. The term “satisficing” is defined as the achievement of an adequate or satisfactory outcome rather than the best possible outcome (Simon, 1956). In nature, it is usually impossible to acquire the amount of information necessary to achieve a globally optimal solution (i.e. finding the best food source in the *entire* forest as opposed to one that is “good enough”). Social insects have evolved stigmergic communication to find the most efficient way to complete a task rather than the best way.

The stunning amount of organization possible through pheromonal stigmergy means colonies of social insects can act as a type of *superorganism*, a term often used by evolutionary biologist E.O. Wilson (Holldobler and Wilson, 2009) to describe the emergence of complex collective behavior at a higher scale than that of an individual organism. Up until Grassé's formal introduction of stigmergy, this kind of collective behavior was thought impossible without the guidance of a central controller. It has since attracted the interest of scientists who wish to learn how local stimuli like pheromones are "organized in space and time to ensure the emergence of a coherent adaptive structure and to explain how [social insects] could act independently yet respond to stimuli provided through the common medium of the environment of the colony" (White, 2005).

While stigmergy is apparent in organisms as diverse as bacteria, slime-molds, and fish (White, 2005), ants have emerged as a primary species of study. Ants have a set of internal "algorithms" that allow them to modify their local environments based on what is immediately apparent (mostly through the use of pheromones) (Holldobler and Wilson, 2009), such as the dead bodies of their kin. Ant species *Lasius niger* and *Pheidole pallidula* are known for building cemeteries through the use of pheromonal stigmergy. If dead ants are initially scattered randomly throughout an environment, their living relatives will "smell" them and start clustering them together (Dorigo, Bonabeau & Theraulaz, 2000). These clusters emerge because of positive and negative feedback loops, which are intrinsic parts of any stigmergic process (Holland & Melhuish, 1999; White, 2005). An ant will tend to put more bodies where bodies already exist because the collective "smell" of larger clusters attracts the living ants. Such clustering indicates that small differences in the initial concentration of pheromones can be amplified over time. Another example of this is seen with ant foraging (Holldobler and Wilson, 2009). While initially the search for food sources is somewhat random, pheromone "trails" left by ants returning from good sources will be reinforced as other ants join to take their share. The stronger the pheromone trail gets, the more ants follow it until the pathways to less plentiful resources diminish and disappear.

Like ants, termites can use similar "winner-takes-all" stigmergic processes to build nests. Termites may initially deposit pheromone-impregnated soil pellets randomly, but the probability of depositing another mud ball in a given location increases with the sensed presence of other mud balls and associated pheromones (Backers, Holland & Deneubourg, 1994; Dorigo, Bonabeau & Theraulaz, 2000). Eventually, mud columns emerge that are further altered through stigmergic processes resulting from the interaction of various concentrations of chemical pheromones, water vapor and carbon dioxide.

Wasps and bees use a combination of pheromones and vision to build complex nests out of hexagonal modules (Figure 2). They can recognize elements of nest construction in process and then, using a small number of internal rules, augment the existing "construction site" in a given way. For instance, Theraulaz et. al. found that with nest building in bees, the probability of adding a cell to a three-wall site is about ten times higher than the case of a two-wall site (Theraulaz & Bonabeau, 1999). After one insect leaves, another can come take its place and make another adjustment using the same internal algorithm with a slightly different local environment. Through the collective interaction of hundreds or thousands of wasps or bees, a full nest structure can emerge.

Stigmergic processes are by no means limited to social insects; examples of stigmergy exist in the human world as well. Holland and Melhuish describe a simple example where several drivers are attempting to negotiate a muddy track. If one car finds the mud in an area on the track too deep, his deep trail marks will act as a sign that alerts other conscientious drivers to avoid that area (Holland & Melhuish, 1999). More refined examples include social networking services like Wikipedia and YouTube, where consumer posts change the "environment of interest" around a piece of media which may, in turn, attract the attention of other users (Parunak, 2006). Presently, scientists and engineers are continuing to design and develop engineered systems that mimic the dynamics of successful stigmergic systems seen in nature. By focusing on the collective actions of relatively identical classes of agents, they hope to remove the need for processes to be performed by highly specialized (and costly) agents and increase robustness to system failure in more mission-critical applications.

1.2 Survey of Work

In recent years, stigmergic frameworks have been applied to everything from the routing of data in mobile telecom (Roth & Wicker, 2003) and Peer-to-Peer networks (Mamei and Zambonelli, 2005), to data mining (Ramos & Abraham, 2004) and even the development of military swarm robots (White, 2005). The dynamics of these systems are often closely modeled after the physics of pheromone dispersal seen in termites and ants. Agents in these systems can deposit different types of "virtual pheromones" in their local environments which can be physical, simulated, or even network constructs (White & Salehi-Abari, 2008) where pheromone concentrations can be assigned to nodes or edges. Just like real pheromones, these virtual analogues can be programmed to decay over time. Agents themselves can be programmed to deposit these pheromones at varying rates and increase or decrease their sensitivities depending on the nature of the application or

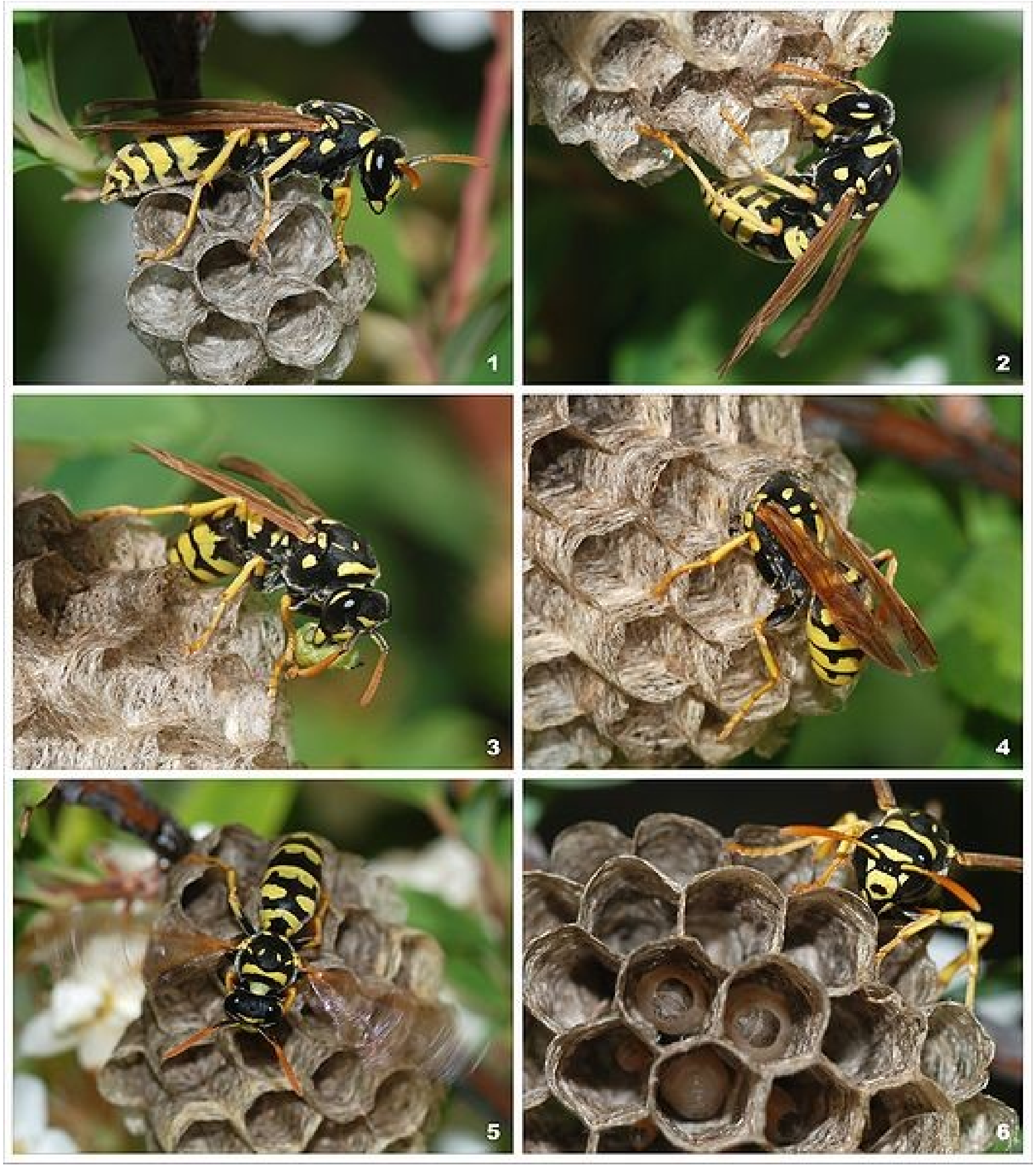


Figure 2: An illustrative example of wasp nest building via stigmergic interactions. Individual wasps analyze the existing structure of the honeycomb and add one hexagonal structure, altering the local environment. Other wasps can come by and further alter the environment based on the recent changes (by: Joaquim Gaspar through Wikipedia under the Creative Commons Attribution-Share Alike 3.0 Unported license).

desired interaction between agents (Parunak, 2006).

Entire classes of *ant algorithms* have been created to use simulated pheromones and ant-like agents to solve distributed optimization and control problems such as vehicle routing, network routing and graph coloring. Ant algorithms are especially adept at addressing Traveling Salesman problems (TSP) where an agent has to find a closed tour of minimal length while hitting every city or node in a network. Ant System (AS), Ant Colony Optimization (ACO) (Dorigo, Bonabeau & Theraulaz, 2000) and Ant-Based Control (ABC) (White, 2005) are just a few of the many types of ant algorithms created to address TSP where virtual ants leave an artificial pheromone trail on the edges that they have crossed once they finish a tour. These pheromones increase the likelihood that other ants will follow the trail and find a destination. Pheromone evaporation is employed to lessen the influence of initial trails (when there is no existing pheromone to influence decision-making) and to allow the system to forget trails that prove ineffective. In one application where an ABC scheme is used for routing calls in a telephone network, "older" virtual agents are even programmed to leave less pheromone over time if it takes them longer to get to their destination (White, 2005). The group size and pheromone dispersion must also be programmed carefully to prevent an overwhelming amount of pheromone to be deposited along paths. These algorithms often produce more optimal paths in TSP-systems than those found using general-purpose algorithms like evolutionary computation or simulated annealing.

Other stigmergic frameworks, such as Ulieru et. al's functional Stigmergic Medical Diagnostic System (SMDS) (Ulieru & Unland, 2006) can be applied to problems that don't fall in the same class as TSP. SMDS is designed to get more accurate medical diagnoses through collective intelligence, rather than relying on the limited or biased perception of one agent (or doctor). First, a request for diagnosis is placed on a virtual blackboard environment where different virtual "diagnosing agents" (specialized for certain classes of ailments) can decide to make an attempt at classifying the problem if it is in their sphere of expertise. If one agent positively comes to a conclusion, its decision is registered in a tree-like format on the blackboard. Other agents, with more specific expertise in that diagnosis class, can then come, examine the existing tree and see if they should tag onto the diagnosis hierarchy (if their own pheromone type is similar enough to the one on a given branch of the tree). The finished diagnosis tree can then be used for more correct medical care.

In the physical realm, swarm robotics has been a prominent test bed for stigmergic frameworks since indirect communication can help ameliorate problems with interference. With swarm robotics, many small agents with limited processing capabilities can interact to achieve beneficial collective behavior. While attempts have been made to use real pheromones in these systems (Wagner, Lindenbaum & Bruckstein, 1999), much research involves the use of virtual pheromones distributed in a physical medium, such as Radio-Frequency Identification (RFID) cards or tags. Robots can read and write information to these objects and the information they create can be read or changed later by other robots or humans (Mamei, Quaglieri & Zambonelli, 2006; Mamei & Zambonelli, 2007). Most RFID-robotic research focuses on using the cards as a means of localizing objects in the environment (Kim & Chong, 2007, Mamei, Quaglieri & Zambonelli, 2006; Mamei & Zambonelli, 2005; Mamei & Zambonelli, 2007; Milella, Cicirelli & Distante, 2008; Patil et. al., 2008) or tracking the location or pose of the robot itself (Bekkali, Sanson & Matsumoto, 2007; Chen et. al., 2007; Howard, Parker & Sukhatme, 2006; Lee & Lee, 2006; Roussos et. al., 2007) or some combination thereof using SLAM (Simultaneous Localization and Mapping) techniques (Kleiner & Dornhege, 2007; Kleiner, Prediger & Nebel, 2006), where passive RFID tags are used by robots to build a "map" of a volatile environment and use it to orient themselves or find human victims (Carbone, Finzi & Orlandini, 2008). In other mapping applications, virtual pheromones are used to prevent trajectory overlap by individual robots (Mamei & Zambonelli, 2007) in an attempt to increase performance. Much of this work focuses on decreasing localization error through statistical techniques like Kalman filtering (Bekkali, Sanson & Matsumoto, 2007), fuzzy inference techniques (Milela, Cicirelli & Distante, 2008) or even through the use of multiple directional RFID antennas (Kim & Chong, 2007).

Other collective robotics applications, such as construction, have a decreased emphasis on pheromone manipulation but still make use of the spatial sorting and clustering seen in the building of termite nests and bee hives (Holland & Melhuish, 1999). A physical nest building implementation was designed where robots were programmed to grip thin circular "pucks" and drop them into clusters (Backers, Holland & Deneubourg, 1994). In this instance, robots essentially ignore each other and only focus on manipulation of local pucks. Interactions in such construction environments can be made more complex by giving the building materials themselves the ability to "talk back" to the robots that are handling them (Werfel & Nagpal, 2006). This is potentially useful in situations where the system must be guided towards a specific structural layout.

For other situations where certain *classes* of structures are more desirably than others, researchers such as Bonabeau et. al. attempt to use genetic algorithms to understand which agent instructions produce "better" structures (based on a pre-defined fitness) and what those instructions have in common (Bonabeau et. al., 2000).

1.3 Observations

As has already been established, the primary benefit of a pheromone-based stigmergic framework is robustness. If individual agents fail, their "traces" or local information will often still be left behind in the environment and not immediately lost (White, 2005), giving the system time to adapt. In addition, no matter how large or dynamic an environment gets, because agents only interact locally they are not overwhelmed. (Parunak, 2006) No agent necessarily needs a global picture because they can work very effectively in parallel to produce a collective behavior (Ramos & Merelo, 2004).

The robustness of pheromone-based stigmergy in nature encourages many researchers to design analogous frameworks in man-made systems. Unfortunately, many of these researchers fall victim to the biomimicry version of “not being able to see the forest for the trees”. Efforts to artificially mimic the physics of pheromone diffusion has led to new classes of problems needing to be solved, such as error minimization (Herianto, Sakakibara, & Kurabayashi, 2007; Parunak, 2006) and the management of "autocatalytic snowball effects" (Dorigo, Bonabeau & Theraulaz, 2000), where, due to runaway feedback processes, virtual pheromones concentrate or diffuse too quickly for proper behaviors or structures to emerge. The cost of true-to-nature artificial analogues of pheromonal stigmergy may be the very robustness they were designed to sustain. If the scientific community instead takes a step back and uses nature as a guide instead of a blueprint, it can open the door for more creative stigmergic frameworks. Thus, researchers may be better served by focusing less on stigmergy as it exists in nature and more on stigmergy "as it could be."

Chapter 2: Presentation of Framework

Stigmergy “as it could be” means developing stigmergic frameworks that are inspired by, but do not currently exist in nature. It is an attempt to reap the benefits associated with natural stigmergy without replicating its constraints. Natural stigmergy evolved in the context of the natural world, not in the world of artificially created systems. Thus, it is reasonable to conclude that the most robust or effective forms of stigmergy for a given human-made system may only conceptually resemble their natural cousins.

This work describes one possible man-made alternative that combines elements of natural stigmergy with the information constructs used in genetic evolution. The goal of this hybridized framework—called “*genetic stigmergy*”—is to encapsulate behaviors in a fully portable, gene-based fashion that frees them from the identity of an individual agent. Such a framework allows for a degree of collective adaptability impossible in natural stigmergy, artificial or otherwise, and thus its potential deserves to be explored.

2.1 Overview

Genetic stigmergy is an indirect communication framework where agent behavioral algorithms—represented as collections of virtual "genes"—can be shared, in part or whole, via an external medium. The discretization of algorithms into spatially distributable genes provides a uniform "currency" that agents can use to quickly swap in behaviors that are found to be locally adaptive by other agents. This “hot swapping” allows for real-time optimization of collective behavior without prior knowledge of an environment. Genetic stigmergy differs from pheromonal stigmergy in that the information exchange is not limited by the paradigm of chemical physics and the problems associated with its mimicry. In addition, the genetic information exchanged is more complex than a simple trace or marker and lends itself very easily to evolutionary manipulation.

An agent may write its own genes to the external medium (heredity), potentially with some minor changes added to them at small probability (allowing for variation). When another agent accesses genes from the medium, it may adopt the genes as its own code (allowing for selection) at another probability that may depend on the "openness" of the previous genes as well as the quality of the new genes written in the medium. Through artificial analogs of heredity, variation and selection, it is possible to include evolutionary processes in a genetic stigmergy framework through such techniques as genetic algorithms or evolutionary programming.

The framework may also be implemented in such a fashion that no restrictions will be imposed on evolvable agent behavior. In such an open-ended system, it is expected that "selfish" individual behaviors that are good at spreading within a population but inconsistent with collective interest may emerge and thereby reduce the collective performance of the population. The removal of non-cooperatives phenotypes may be achieved by specifically programming the protocols to remove them from the swarm, or by implicitly suppressing the spread of non-cooperative phenotypes through evolutionary means. External thresholds may also be applied that limit when an agent has access to locally-stored information, or when an agent can use accessed information to modify its own algorithms.

2.2 Important Terms and Phrases

The following terms are used frequently throughout this thesis and are defined explicitly here to limit confusion:

Gene: A single piece of data or element of a behavioral algorithm.

Chromosome: A collection of multiple genes carried by an agent that collectively influences its phenotype in a direct or indirect fashion.

Phenotype: The external behavior of an agent arising from instructions represented by its behavioral gene or chromosome.

Locus: A discrete location on an agent's chromosome that contains genes which correspond to an agent's behavior in different types of situations.

Allele: A variant of genes on a specific locus that corresponds with alternative reactions to a *specific* type of situation or external stimuli.

2.3 Comparisons with Non-Genetic Stigmergy

Genetic stigmergy stands apart from other forms of artificial stigmergy by moving away from "stigmergy as it is" in nature to "stigmergy as it could be." Much of the present research in artificial stigmergy focuses on mimicking the mechanics of pheromonal communication (Dorigo, Bonabeau & Theraulaz, 2000; Herianto, Sakakibara, & Kurabayashi, 2007; Parunak, 2006; Wagner, Lindenbaum & Bruckstein, 1999). While artificial, pheromonal-based stigmergy has the benefit of being modeled after a natural process with millions of years of evolution behind it, researchers often get bogged down in attempts to mimic the physics of pheromone deposition and diffusion, sometimes adding unnecessary complexity to the system. In addition, genetic stigmergy potentially allows for greater persistence of agent states and thus greater robustness. If an individual agent learns a unique way of solving a problem, it can deposit its entire behavioral algorithm (or a representation of it) for other agents to use if it is lost or destroyed.

Chapter 3: Objective of Thesis

3.1 Hypotheses and Context

The objective of this thesis is to explore the efficacy of genetic stigmergy as an indirect communication framework in the context of a swarm robotics application by experimentally testing the following hypotheses:

A. The introduction of genetic stigmergy among a swarm of robotic agents will make their collective intelligence more robust and efficient than through a purely probabilistic framework.

B. The performance improvement of robotic agents by genetic stigmergy requires appropriate mechanisms hard-wired into each agent for promoting and maintaining beneficial, task-oriented behavior within the swarm.

Swarm robotics is a relatively new field of research with a focus on mutually interacting, self-organizing robots that can collectively achieve tasks through the use of decentralized local mechanisms. Stigmergy in swarm robotics has so far been limited to exchanging minimal information among agents, such as concentration of virtual pheromones. The recent development of economical, high-capacity Radio-Frequency Identification (RFID) cards has opened up a new opportunity for stigmergy. Through these cards, robotic agents can dynamically exchange more complex, logical information, such as a genetic code that controls their behavioral rules. Dynamic, real-time modification of agents' behavioral "genes" may increase the adaptability of a swarm to a complex system, which is useful for tasks such as collective exploration of an unknown environment. Certain behaviors may be more adaptive in various areas of an environment (i.e. better at navigating the area more quickly). Using genetic stigmergy, robotic agents do not have to communicate with each other directly and would not need complicated algorithms to manage the physics of pheromone diffusion. RFID cards can be distributed throughout an environment for robots to record their genetic codes. Through modification of the probability of interaction with the RFID cards or by directly limiting interaction to a certain "window", the collective behavior of the swarm can be guided in a beneficial direction.

3.2 Physical Basis

In this thesis' experiments, the use of a physical robotic swarm made of 8 OPEN-ROBOTs (Figure 3) designed by Abraham Howell of Abe Howell's Robotics (Binghamton, NY) is assumed. Unlike other commercially available robots such as the e-Puck (Herianto, Sakakibara, & Kurabayashi, 2007) or LegoBots (Mamei & Zambonelli, 2005), the OPEN-ROBOTS are highly functional yet still extremely cost-efficient. The flexibility of these robots has already been demonstrated in several previous experiments, including a simulation of complex foraging behavior (Howell et. al., 2006). The robots cost approximately $450 each and can be constructed in less than five hours with off-the-shelf materials.

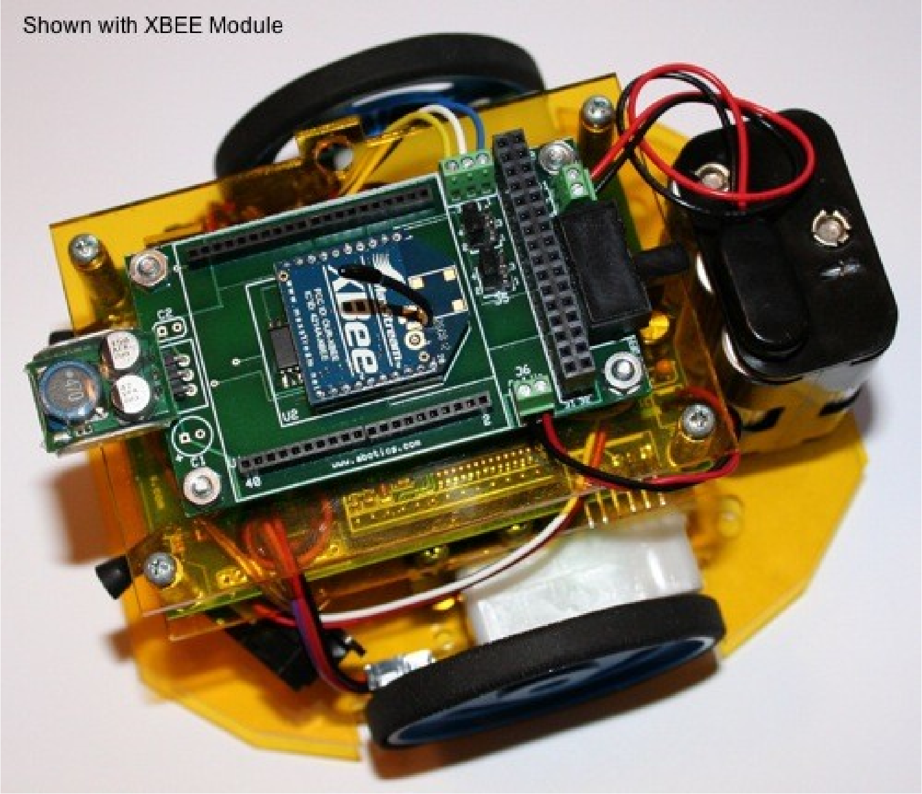


Figure 3: OPEN-ROBOT with wireless XBee R ZigBeeTM communication module and RFID tag reader (not visible).

The robot, powered by 6 rechargeable AA batteries, measures 11.43 centimeters in width, 14.6 centimeters in length, and 8.89 centimeters in height. It can move at approximately 9.2 cm/s and rotate at 92 degrees/second. Each of its two wheels are controlled by separate GM8 motors which are controlled by a PIC18F4520-based controller board with an integrated H-Bridge chip (allowing the motors to move forwards or reverse). The robot has five Sharp GP2D120 infrared sensors--three located on the front left, center and right of the robot, with the other two arranged in the rear. These infrared sensors are the primary tool with which the robots can detect and avoid obstacles. Two cadmium sulfide light sensors sitting above the front infrared sensors are used to detect ambient light conditions. To communicate with other agents or a central computer (if desired), the robots are equipped with XBee® ZigBee^{TM} module. This module consumes less power and is significantly cheaper than a Wi-Fi module. On the undercarriage of the robot is a 5.1 cm x 7.6 cm rectangular antenna used to read and write data to the 3.8 cm diameter antennas in the RFID cards. The expected read/write cycle is 0.17 seconds.

The robot is controlled by a modifiable, boot-loadable firmware that exists in the PIC18F4520-based controller board mentioned above. This firmware allows the OPEN-ROBOT to be controlled through a serial-based command set. Serial commands can be sent wirelessly from a central computer to one or more robots. The robot can also react autonomously, relying solely on the behavioral rules loaded in its firmware.

3.3 Virtualization

For the design of different implementations of genetic stigmergy, computer simulation of swarms is almost a necessity because it drastically reduces the time and cost for experimental testing of the implementations under consideration. Using physical OPEN-ROBOTs as a model, two sets of experiments are performed simulating virtual OPEN-ROBOTs in realistic environments of varying complexity. The experiments test how different variations of simple, non-evolutionary genetic stigmergy affect the overall swarm's capacity to fully explore open space in an unknown environment. One class of experiments examines the efficacy of a single-gene framework (robots can only exchange one behavior gene indirectly), while the other examines the efficacy of a multi-gene framework (robots can exchange multiple genes indirectly).

Chapter 4: Experiment I: Single-Gene Genetic Stigmergy

The goal of the single-gene experiment is to determine, on average, what proportion of the open space in a given environment a swarm of robots could cover in a given time period. Different implementations of single-gene genetic stigmergy are tested by using a global control parameter (stored locally in the robot) that controls how robots make use of behavioral genes deposited in the environment by other robots. Through manipulation of this parameter, robots can be encouraged to adopt or avoid the genes they find. The usefulness of various parameter settings is then compared against random switching of genes, as a control. Because genetic stigmergy involves various implementations of ordered information exchange, randomization is used as a control to determine if a given "ordered" system of communication is statistically better than a purely unordered system. This experiment was originally presented at the 2009 IEEE Symposium on Artificial Life in Nashville, TN (Brandoff & Sayama, 2009), but has since been expanded for the purpose of this thesis.

4.1 Simulation Platform

The simulator supporting this virtual world uses 3DRad (3DRad, 2008), a free video game development environment with a strong 3-D physics simulation engine. 3DRad has advanced rendering capabilities and can import 3D object data in various formats. The original 3D CAD data of the OPEN-ROBOTs, RFID cards and environment are imported to create a true-to-life virtual world (Figure 4). The behavioral rules of the robotic agents are written in AngelScript (similar in function and syntax to C) and integrated into the physics simulation. 3DRad simulates its virtual physics in discrete time steps at the rate of 75 frames-per-second. In view of this refresh rate, the OPEN-ROBOT's linear and angular velocities in the simulated world can be rescaled using a simulation speed factor so that the simulated motion is the same as or faster than that of the actual robot observed in the real world. Collisions between robots and other objects are detected by invisible "rays" that protrude approximately five centimeters from the locations of virtual infrared sensors on the front of the robot. When the surface of another object crosses this ray, an imminent collision is detected and reacted to according to the behavioral rules given to each agent. Details of the collision detection process are calibrated to match the actual behavior of the physical infrared sensors.

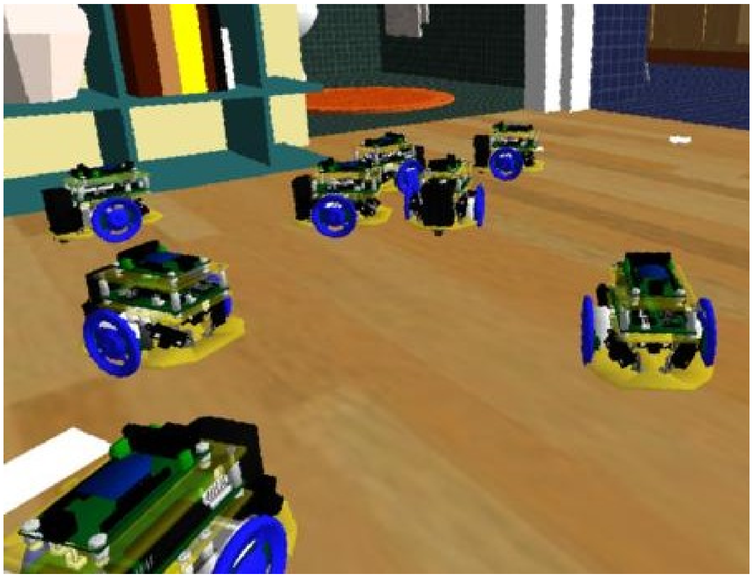


Figure 4: Eight robots exploring a hallway with two nearby RFID tags.

A simulation is initialized by first randomizing the positions and orientations of the robots by the “entrance” to an environment, followed by all RFID cards throughout the entire environment. All objects are placed in open areas inside the environment and do not overlap with other objects or the barriers. RFID cards may, however, be placed under the robots' initial positions. Robots must be spaced at least 8 centimeters from one another while cards must be spaced at least, approximately 10 centimeters from other cards. These random initializations are used to prevent any behavior anomalies that come from the starting positions of the objects, rather than the emergent behavior from their interaction. After environmental initialization, the simulation sequentially checks for collisions between the robots, environment and RFID cards. Each robot is also initialized with a single integer value corresponding to a given behavioral gene (that can later be deposited to an RFID card). This genetic information, along with the current position of every robot, is written to an external data file at each time step of the simulation.

4.2 Experimental Setup

For all experiments, a swarm of 8 robots and 80 RFID cards are used. Their initial positions and orientations are determined in a random fashion. The simulated environment is a 54 square-meter, single-floor, five-room house (see Figure 5) acquired from Google 3D Warehouse (Google 3D Warehouse, 2008), an online repository where users can upload 3D models using Google's free SketchUp modeling software. The assumption of eight agents in a 54 square-meter environment is fairly conservative compared to other experiments where anywhere from ten (Mamei & Zambonelli, 2005) to eighty (Howard, Parker & Sukhatme, 2006) agents are used in environments as large as 600 square-meters.

Each of the eight robots, situated in random positions near the front door of the home, are randomly initialized with single genes corresponding to one of four possible exploration behaviors: *Wander*, *SpiralOut*, *WanderSpiral* and *RandomReaction*. All behaviors contain basic obstacle avoidance. If a robot detects an obstacle to its left or right, it will rotate away from that obstacle before continuing with its specific behavior.

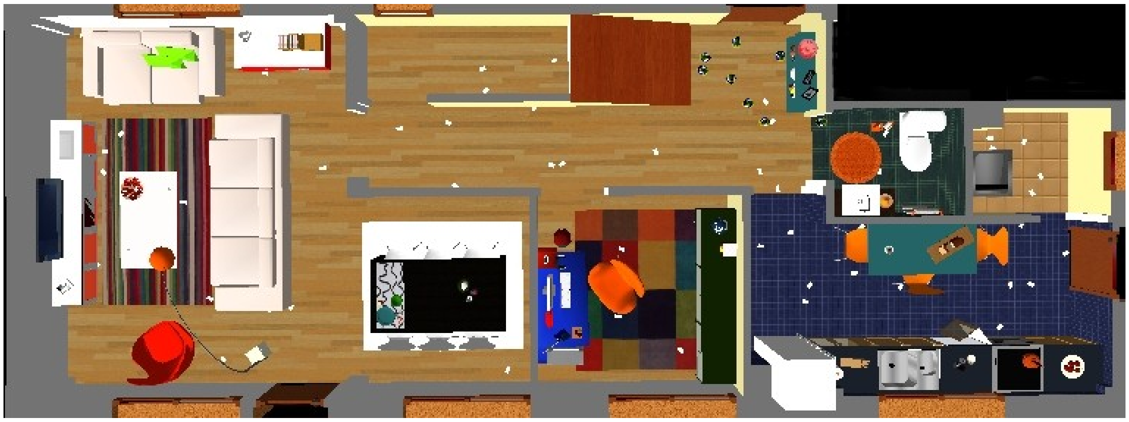


Figure 5: Eight robots exploring the ground floor of a virtual home filled with 80 randomly distributed RFID cards.

Each behavior determines what the robot will do when it is not avoiding obstacles. With the *Wander* gene, a robot simply moves forward in a straight trajectory. With *SpiralOut*, the robot makes a discrete rotation right, followed by a discrete motion forward. The size of these discrete steps (occurring during each time-step) depends on the simulation speed factor mentioned earlier. With *WanderSpiral*, the robot randomly chooses to either move forward or rotate right during each time step. Finally, with *RandomReaction*, the robot randomly chooses to either move forward, rotate right, or rotate left during each time step. Here, the resulting behavior is very similar to a random walk.

The primary experimental parameter, *K*, determines how all robots will interact with the RFID cards (see Figure 6 for an overview of robot-RFID card interaction). If the RFID card has not been written to yet during a given time-step, the robot will write its gene to the card. If a card has already been written to and *K* is positive, the robot will (with *K* probability) adopt the behavioral gene referred to on the card. If *K* is negative, the robot will adopt (with $|K|$ probability) a random gene that is not present on the card.

A series of five Monte Carlo simulations are performed for each *K*. Coverage trajectories for each trial, composed of hundreds of thousands of individual trajectory points, are examined through Monte Carlo integration using 100,000 sample points (with radii corresponding to that of the robots) extracted from a rectangular space encompassing the entire home. The area of this space is larger than the actual area of the house (53.9 meters squared). Because of this, the results are normalized by finding the proportional difference between the true area and the rectangular space used for Monte Carlo integration and divided the coverage values by this number. Simulations are performed at 10x real-time, so a simulated hour-long trial only takes six real minutes.

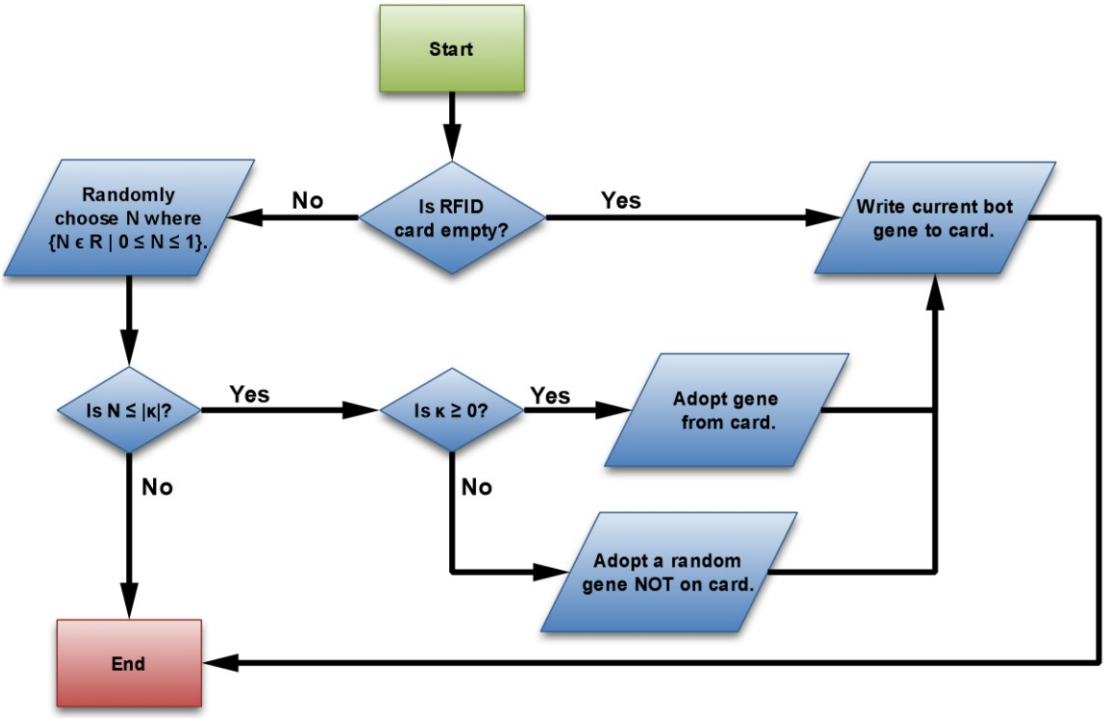

Start
Is RFID card empty?
No
Yes
Randomly choose N where {N ϵ R | 0 ≤ N ≤ 1}.
Write current bot gene to card.
Is N ≤ |κ|?
Yes
Is κ ≥ 0?
Yes
Adopt gene from card.
No
No
Adopt a random gene NOT on card.
End

Figure 6: Flowchart describing interaction between a robot and an RFID card in a single encounter event. Here, N is a random number between 0 and 1 belonging to the set of all Real numbers (indicated by R) and Κ is the probability of a robot modifying its gene based on interaction with an RFID card.

4.3 Experimental Results

Figure 7 illustrates an example of the characteristic trajectories of robots when *K* is set to a positive value. Unlike other work where maps are created by identifying boundaries between open spaces (Kleiner & Dornhege, 2007) and techniques are developed to prevent overlapping trajectories (Howard, Parker & Sukhatme, 2006), the maps created by the swarms in these experiments *depend* on overlapping trajectories to fill in empty space. Areas without trajectories emerge as obstacles and boundaries in these maps.

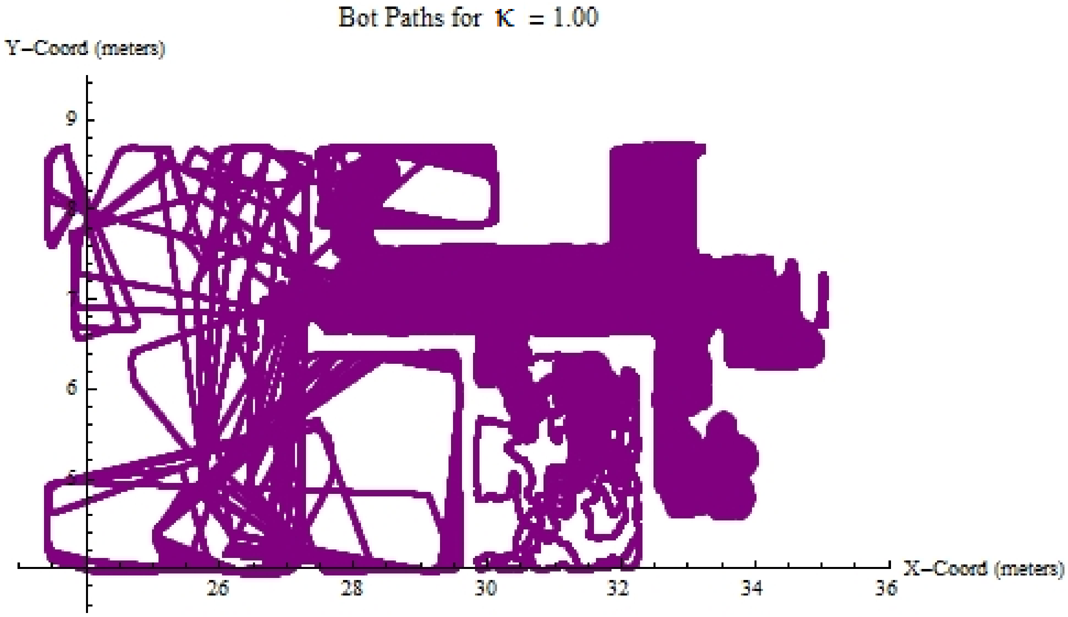


Figure 7: An overlay of the trajectories of a swarm of 20 robots, each initialized with a random exploration strategy, over an hour in the virtual home covered in 60 RFID cards with Κ = 1. The image, resulting from an earlier test of the simulator, demonstrates how robots with certain genes (identified by the shapes of their characteristic trajectories) tend to self-organize into different areas of the house. While a 20-to-60 robot-to-card ratio is used in this example, the behavior is also characteristic of simulations with an 8-to-80 robot-to-card ratio.

Even though the robots are randomly initialized with a certain gene, robots with a given behavior tend to self-organize into the same rooms during each simulation. Robots with the *Wander* gene usually end up in large rooms (the living room/TV room) with open spaces and obstacles with open space between them. Robots with the *RandomReaction* gene usually end up in smaller rooms (the bedroom and kitchen) with more closely spaced obstacles. Robots with the *Spiral* and *SpiralOut* genes tend to dominate the area near the front door of the house and the long adjacent hallway with no obstacles in it.

Upon the completion of the experiments, the data is organized into a series of scatter plots showing the individual and average coverage trajectories for each trial and their corresponding coverage values for each *K* value. Figure 8 indicates a negative correlation between average proportion of coverage and *K*. Figure 9 shows that, with both positive and negative *K* values, high coverage rates are possible. However, with more negative *K* values, the lower bound of possible coverage values is restricted such that the range of coverage values for the simulations with lower negatives are concentrated around the higher coverage values.

Figure 10 illustrates the robot trajectories from each of the five trials at three representative *K* values. At *K* = 1, room coverage ranges from 26.4% to 70.0% while at *K* = -1, coverage ranges from 54.4% to 67.3%. While the highest coverage value, 70.0%, emerges when *K* = 1, the most consistent performance occurs when *K* = -1. = -1, 0 and 1 and demonstrates an increase in coverage consistency as *K* was adjusted negatively.

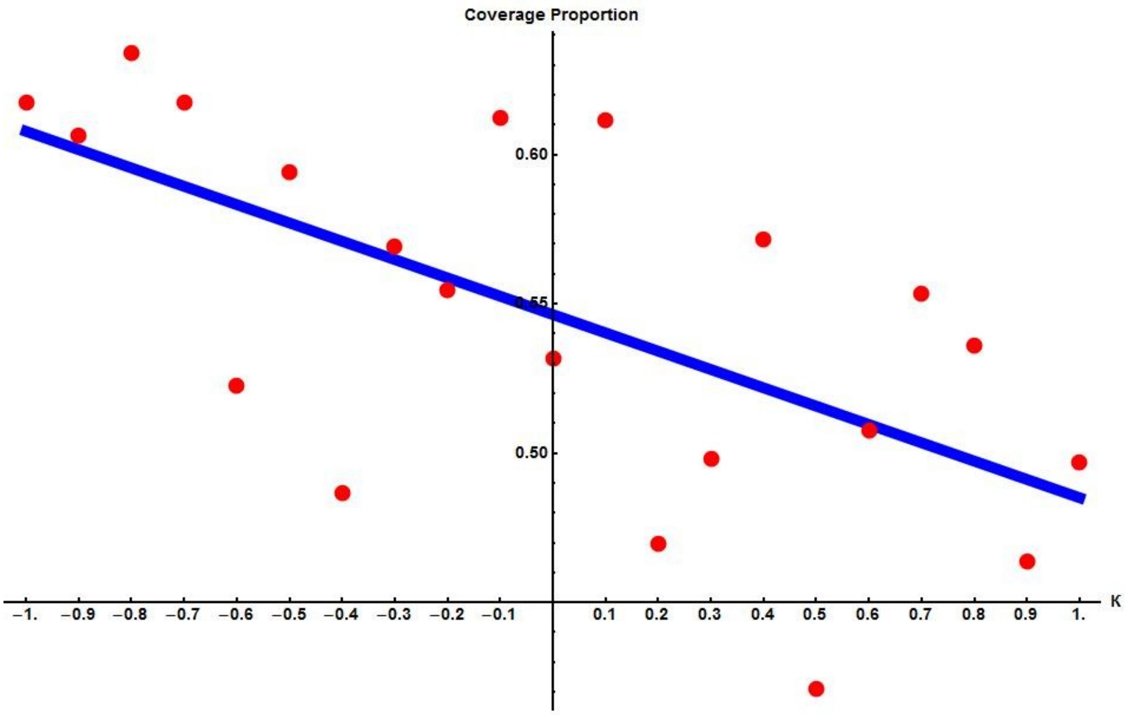


Figure 8: The average proportion of open space covered for the five trials of each K with a best fit-line described by y = 0.546415 - 0.0613712x, with all parameter p-values (given y=b-ax) <.001 and R^2 = 0.40.

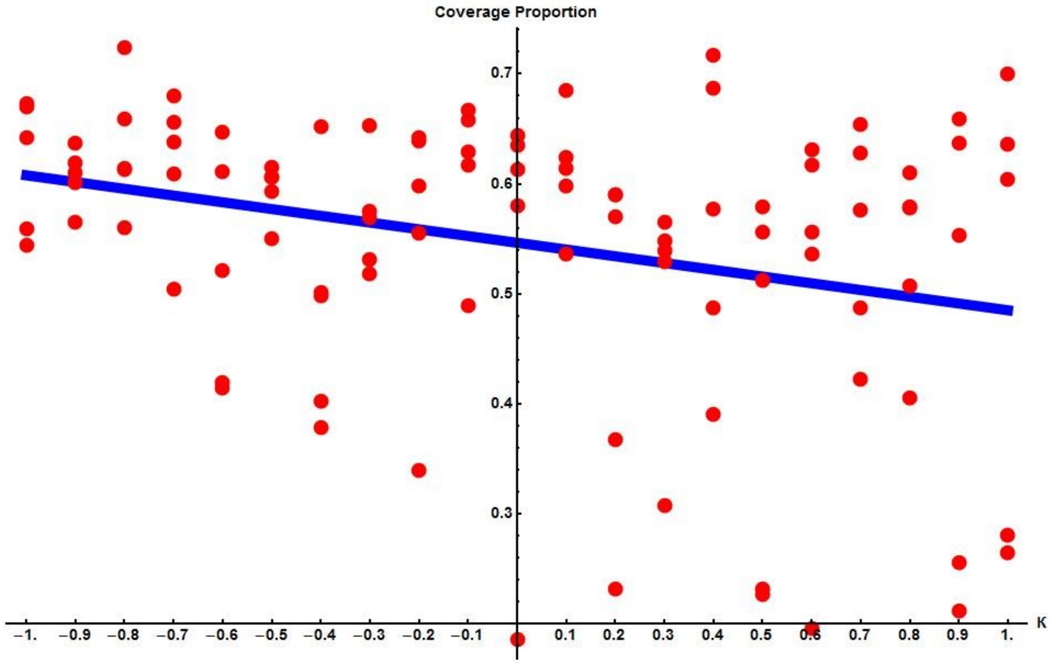


Figure 9: A scatter plot of 105 coverage values, resulting from 5 trials with each of the K values with a best fit line described by y = 0.546415 - 0.061371x, showing a statistically significant correlation (p < 0.005). The correlation coefficient of K and the coverage proportion is -0.294594.

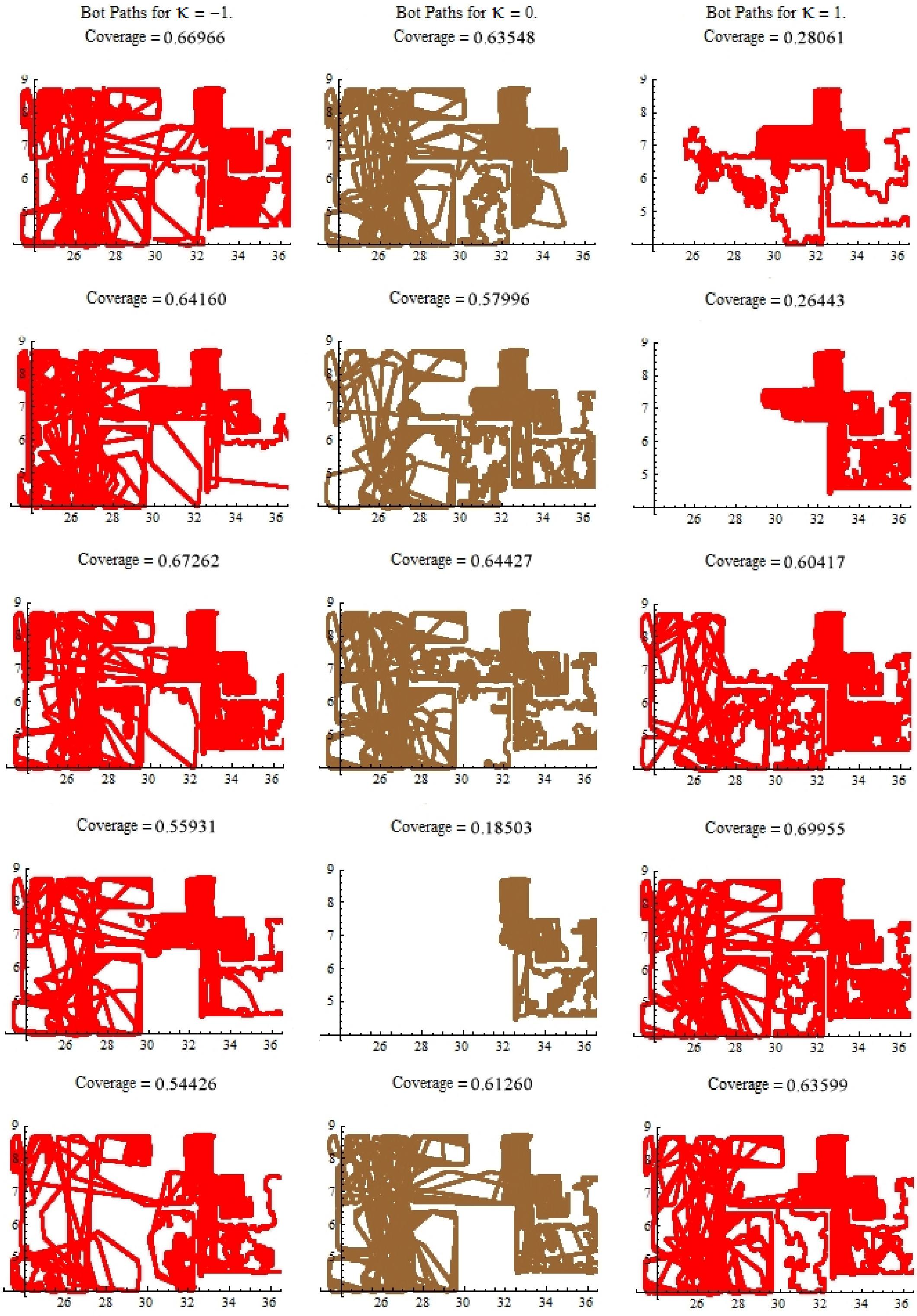


Figure 10: Aggregated trajectories of robots in five trials with Κ = {-1,0,1}. Axes correspond to X- and Y-coordinates of trajectories in meters. Coverage values are the proportion of space filled in environment (where 1.0 = 100% coverage).

Figure 11 indicates a positive correlation between the *K* values and the variance of the five trials. The lowest *K* value, -1, has a variance close to zero, while higher *K*. As opposed to indicating dominant strategies within the swarm, this figure demonstrates that the low performance variance associated with negative *K* values corresponds with rapid shifting of genes by the swarm, whereas the high performance variance associated with positive *K* values corresponds with significantly less frequent shifting of genes. Figure 12 highlights the dynamics of strategy adoption across various values of negative and positive *K*.

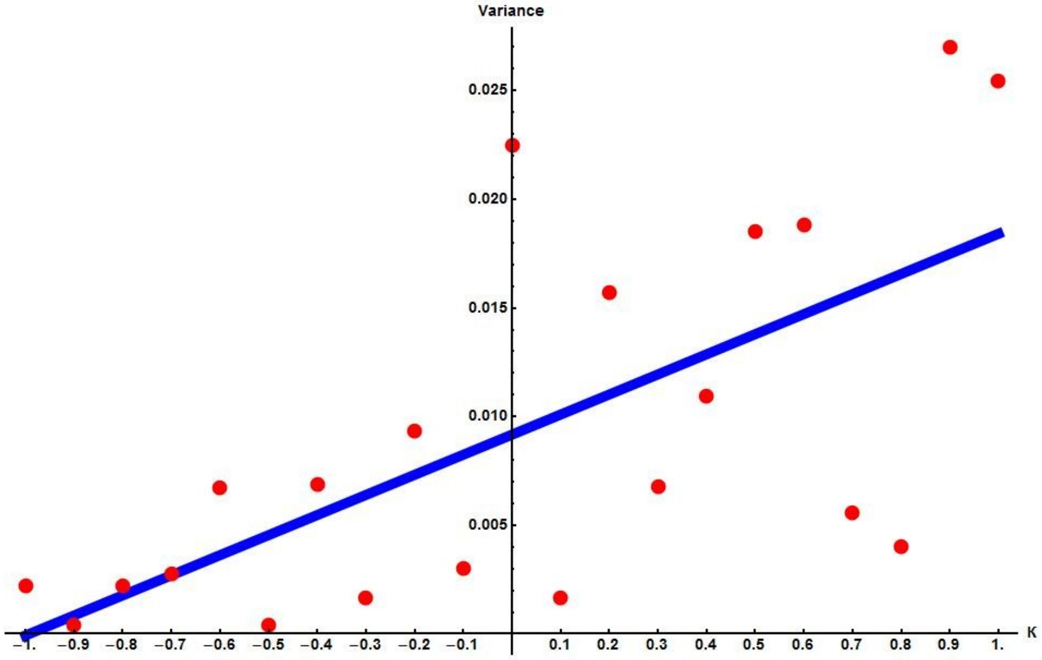


Figure 11: Variance of the five trials for each K as a function of K with a best-fit line described by y = 0.009168 + 0.009245x, showing a statistically significant correlation (p < 0.001). The coefficient of correlation between K and the variance is 0.665875.

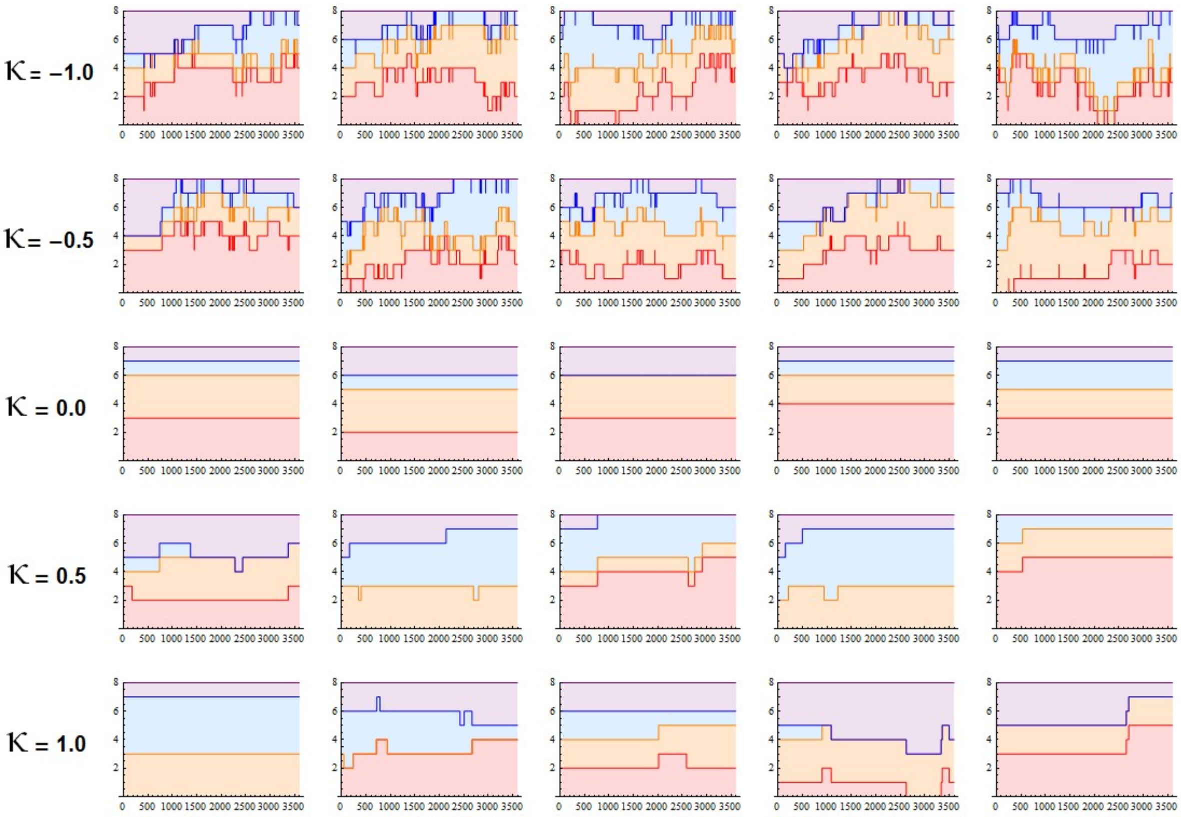


Figure 12: Stacked line plots of the number of robots using each possible gene in the five trials for each given K value. Simulated time is represented on the X-axis in seconds and the number of robots using a given gene is indicated by the Y-axis. The four shades in each graph (from bottom to top) correspond with the genes Wander, SpiralOut, WanderSpiral and RandomReaction respectively. Since it is possible for a swarm to be randomly initialized with less than four genes, not all shades may be present in all graphs.

The exploration performance of the *K*-based algorithm is then compared with random switching, where a robot randomly adopts a new gene (and thus a new behavior) every time it encounters an RFID card. Figure 13 shows the coverage maps associated with this new set of experiments. Shockingly, these results proved to be better than those arising from -*K* values by having a higher range of coverage values and higher average overall coverage while maintaining relatively low performance variance. In the random trials, coverage ranges from 61.2% to 75.6% with an average coverage of 68.1% and performance variance of 0.003. At *K* = -0.9, coverage ranges from 56.5% to 63.7% with

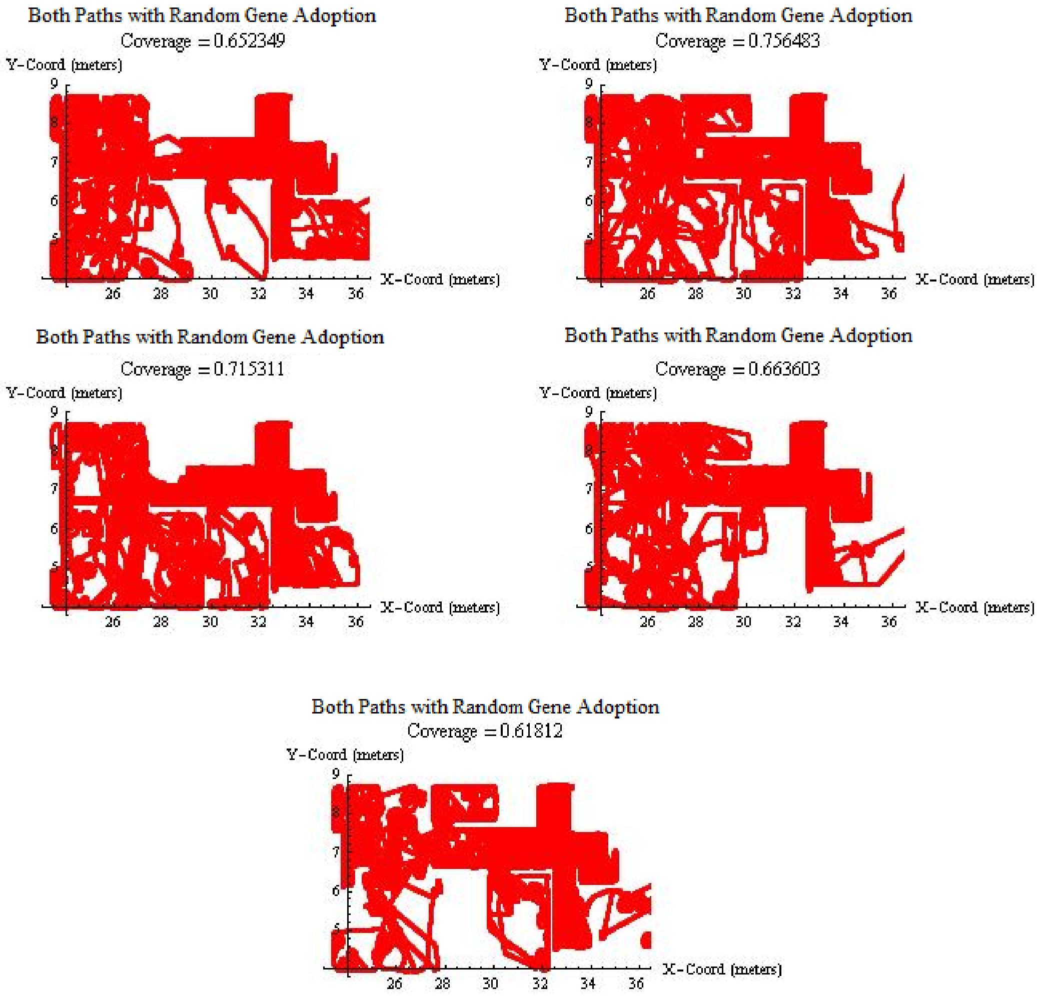


Figure 13: Aggregated trajectories of robots in five trials where robots randomly adopt a new gene every time they encounter an RFID card. Axes correspond to X- and Y-coordinates of trajectories in meters. The variance in coverage is approximately 0.003.

an average coverage of 60.1% (the greatest average coverage of all *K* values) and performance variance of 0.0004 (the second lowest variance of all *K* values).

The superior performance of the random switching trials indicates that single-gene stigmergy, as described in this thesis, is too simplistic and uses RFID cards too naively to engender non-trivial results. Low RFID card density and the severely constricted number of genes likely weaken the effect of genetic stigmergy. In addition, because only one environment is used, there is no way to determine the sensitivity of genetic stigmergy to environmental complexity. To more effectively explore the potential genetic stigmergy, a more complex framework is needed that uses more realistic genetic structures comprising of greater than one gene.

Chapter 5: Experiment II: Multi-Gene Genetic Stigmergy

The goal of the multi-gene experiment is to demonstrate that a more thoughtfully constructed genetic stigmergy framework will provide improvements in a swarm's mapping performance beyond those arising from the random switching of genes. In this experiment, robots carry a more complex, multi-locus chromosome where genes at each locus control a robot's reaction to different types of external stimuli. Multiple alleles of the genes at each locus allow for much greater diversity in the robot's response to a specific type of stimuli than was possible in the single-gene experiments. The density of RFID cards is increased and the cards are now able to record the frequency of genes deposited to them. Different implementations of multi-gene genetic stigmergy are tested where robots are encouraged to adopt ("Majority Seeking") or avoid ("Minority Seeking") frequently used genes or randomize their chromosome. In addition, the robustness of multi-gene genetic stigmergy is demonstrated by exploring the different implementations in environments of varying complexity and through the usage of "accessibility windows" that control when a robot can access the genetic information on a card.

It is important to note that while the conclusions of the single-gene experiment are incorporated into the multi-gene experiment, the multi-gene experiment uses a different experimental framework. The number of parameters added or changed in the multi-gene experiment limits continuity from the previous experiment.

5.1 Simulation Platform

For these experiments, a more flexible simulation platform is designed using the Python programming language to quickly and effectively examine the swarm's exploration behavior in environments of varying complexity. Like the first class of experiments, the generated virtual environment is composed of OPEN-ROBOTS, RFID cards and a to-scale environment within which the robots interact (Figure 14). By using the Python-based platform on a multiple-CPU system, four simulations can be run at up to 3600x real-time with different parameter settings.

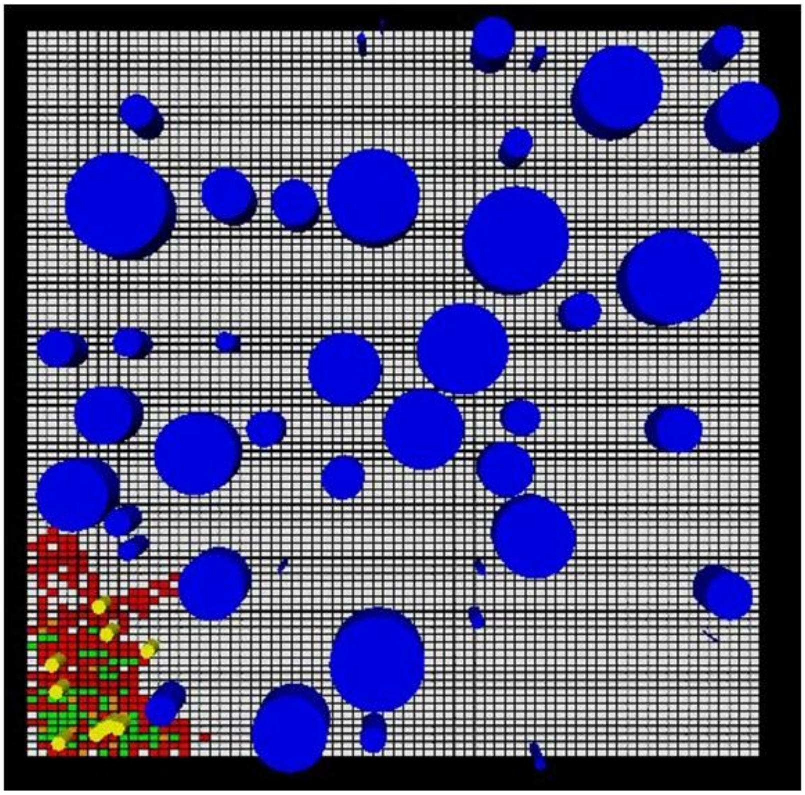


Figure 14: An "aerial view" of 10 swarm robots (in yellow) exploring a randomly generated virtual environment with a dense grid of RFID cards embedded in the floor. The blue cylinders are obstacles and RFID cards change color depending on their current state.

Unlike the single-gene simulations, experiments performed using the Python-based simulator use only two-dimensional mathematical representations of robots, obstacles and RFID cards. Robots and obstacles are represented as circular constructs with appropriate radii (7.25 cm for robots, approximately corresponding with their physical counterparts). When visual confirmation is needed to confirm that an algorithm is working properly, the freely available VPython package is incorporated into the simulator. For aesthetic purposes, obstacles and robots are given arbitrary height values (Figure 14).

The "reduced" nature of the Python platform necessitates a modification of how robots detect collisions. A robot will register the existence of an obstacle if its "sensor circumference" (a radius of 18.5 cm beyond its virtual embodiment) overlaps with the circumference of a given obstacle, robot or passes beyond the boundary of the environment. Collisions with RFID cards are calculated by testing to see if the robot's center is within the perimeter of a given card.

5.2 Experimental Setup

Because the environments of the new simulations are meant to mimic homes or factories with dense, non-overlapping grids of RFID cards in the floor, approximate coverage is determined by counting the total number of RFID cards the swarm interacts with. For all experiments, a swarm of 10 robots are initialized with randomized initial positions in the lower left corner of a square (also see Figure 14), 54 square-meter environment filled with a grid of 6,840 RFID cards (approximately 130 cards per square-meter or 12 cards per square-foot).

Every environment also contains a total of 10 square-meters of obstacles. How this allotment is divided is controlled by a control parameter that sets the maximum possible radius for a given obstacle. The radius of a newly generated obstacle is determined by choosing a number from a uniform distribution between zero and the "max radius" parameter: “R”. If R is low, many more obstacles are needed to reach the allotted overall obstacle area. If R is high, fewer obstacles are needed to reach this allotment. Experiments are performed using R values ranging from 0.2 meters to 2.0 meters in increments of 0.2 meters. Examples of relatively "simple" and "complex environments” are highlighted in Figure15.

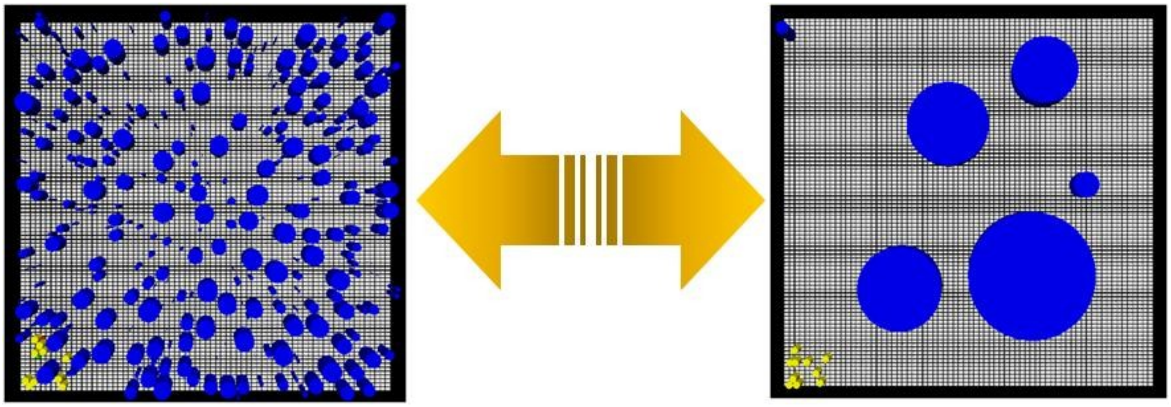


Figure 15: Example manifestations of environments with R values between 0.2 meters (left) and 2 meters (right). Environments with smaller R values are considered complex to navigate, while environments with larger R values are considered simple.

Instead of carrying one of four possible genes, robots now carry a chromosome with three loci, each containing a gene that controls the robot's reaction to environmental stimuli (see Table 1). The first locus contains genes that control how a robot reacts when it senses a potential collision with another robot. The second locus contains genes that control how a robot reacts to potential collisions with a static obstacle or the environmental boundaries. The third locus controls how a robot behaves when it does not sense any obstacles or other robots in its vicinity. Alleles can be combined in 128 different ways, giving the robot much more algorithmic flexibility than in the single-gene experiments. At the beginning of each simulation, all ten of the robots are initiated with chromosome (1,1,0). The resulting phenotype causes the robots to rotate away from other robots and obstacles, but to otherwise move forward. This chromosome encourages the robots to spread out across the environment instead of staying clumped up in the corner.

	Locus 1 **(Reactions to robots)**	**Locus 2** **(Reactions to Obstacles)**	**Locus 3** **(Reactions to Open Space)**
0	Rotate towards robot	Rotate towards obstacle	Move forwards once
1	Rotate away from robot	Rotate away from obstacle	Move backwards once
2	Back away from robot	Back away from obstacle	Move forwards twice and rotate left
3	Randomly choose between rotating towards, away or backing away from robot	Randomly choose between rotating towards, away or backing away from obstacle	Move forwards twice and rotate right
4			Move backwards twice and rotate left
5			Move backwards twice and rotate right
6			Randomly choose between rotating left, right or moving forwards once
7			Randomly choose between rotating left, right or moving backwards once

Table 1: Breakdown of the chromosomal structure controlling each robot's behavior and the potential genes that control behaviors in specific situations. Note that robots move and rotate in discrete increments, 9.2 cm and 92.1 degrees respectively (the maximum distance a physical OPEN-ROBOT could move or rotate in a single second).

The RFID grid acts as a distributed counter system that records the number of times cards have been hit by robots and what genes each robot was carrying. Every time a robot is within range of an RFID card, the RFID card will increment its 'hit' counter and counters corresponding to the individual alleles the robot is carrying. Over time, the RFID card will generate a tally of locally used allele frequencies. Whether a robot can access the information on a card depends upon accessibility windows, another control parameter. These windows are implemented to prevent a robot from adopting new genes too rapidly and essentially “jittering” in place by instantaneously switching behaviors. As indicated by Figure 16, a robot can only modify its chromosome based on the RFID card's information when the card's hit counter is within a "hit window" defined by the experimenter. Depending on the experiment, a window can open after between 1 and 9 hits and can close after between 2 and 10 hits. Different permutations of these windows allow for 45 different accessibility windows. Figure 17 describes the interaction algorithm between robots and RFID cards. While robots can only access information within a given window, they can "reset" the cards global hit counter to zero after the accessibility window has closed.

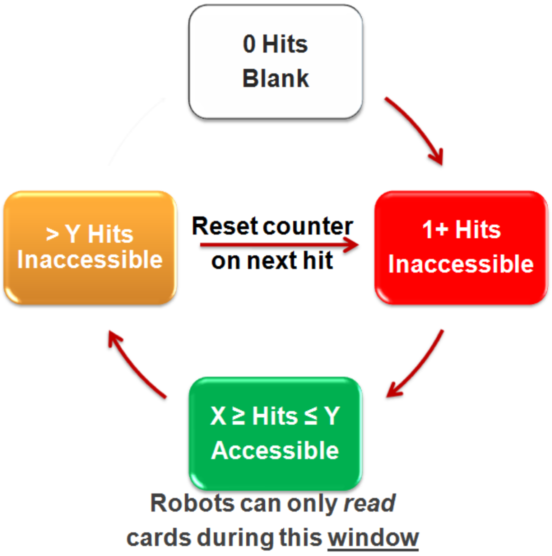


Figure 16: In the multi-gene framework, access to RFID cards is controlled by virtual "hit counters" stored on the card. Depending on the number of hits, an RFID card can cycle between one of several different accessible or inaccessible states (and colors).

Before a simulation begins, the positions and orientations of ten robots are randomly initialized by the lower left corner of the environment. Afterwards, obstacles of random radius are generated such that their total area is equal to 10 square meters (RFID cards are created in a massive grid that lies directly underneath all robots and obstacles).

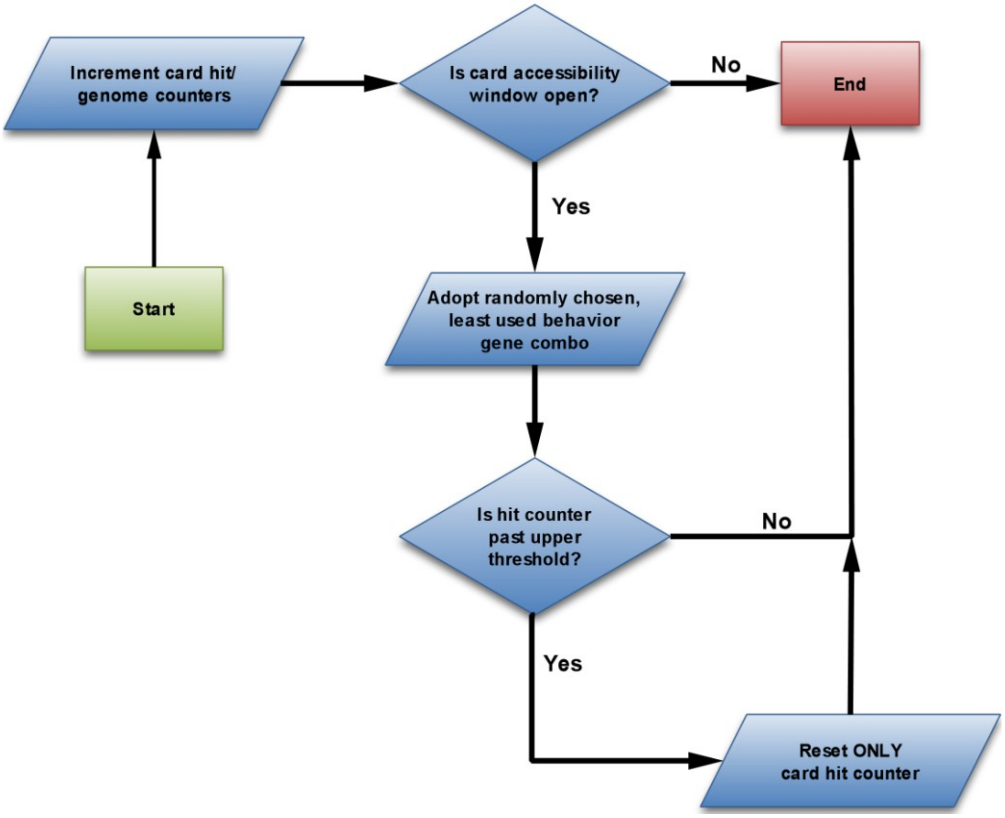

Increment card hit/ genome counters
Is card accessibility window open?
No
End
Yes
Start
Adopt randomly chosen, least used behavior gene combo
Is hit counter past upper threshold?
No
Yes
Reset ONLY card hit counter

Figure 17: Flowchart describing interaction between a robot and an RFID card in a single encounter event under the "Minority Seeking" interaction paradigm, where a robot adopts the least used combination of behavior genes based on the information available on the card. Here, the "accessibility window" refers to the card hit thresholds within which a robot can access the genetic information stored on the RFID card.

All objects are placed in open areas inside the environment and do not overlap with other objects are barriers. Robots must be spaced such that their "sensor circumference" does not overlap with those of other robots. These random initializations are expanded upon from the single-gene experiments to prevent any anomalies that would arise from any given static environment. After environmental initialization, the simulation continuously (during each time-step) checks for collisions between the robots, environment and RFID cards. At the end of each simulation, information is written to an external text file that shows the state of the system at each second.

To determine the most efficacious implementation of the multi-gene framework, four robot-RFID card interaction paradigms are tested: “Minority Seeking”, “Majority Seeking”, “Randomization” and “No Threshold”. The “Minority” and “No Threshold” paradigms both encourage robots to adopt the genes least frequently recorded to a card, but the “Minority” paradigm restricts card access to a given accessibility window. “Majority Seeking” and “Randomization” also use accessibility windows but “Majority Seeking” encourages robots to adopt the genes *most* frequently recorded to a card, while “Randomization” forces the robots to randomize their genes (and ignore the information on a card). The effectiveness of each paradigm is tested by performing a series of 30 Monte Carlo simulations for each combination of possible RFID accessibility window and R value. Simulations are performed at 3600x real-time, so a simulated hour-long trial takes less than one real second.

5.3 Experimental Results

The performance of the “Minority Seeking”, “Majority Seeking”, “Randomization” and “No Threshold” paradigms are compared using a fixed accessibility window of 5 to 10 card hits for values of R from 0.2 meters to 2.0 meters. Figure 18(a) demonstrates that the “Majority Seeking” and “No Threshold” paradigms produce dismal performance, while the “Minority Seeking” and “Randomization” paradigms achieve the highest average performance. Figures 18(b) and 18(c) indicate that for both paradigms, coverage increases arc tangentially as a function of increasing R. Table 2 shows that, with the exception of $R = 1.6$ meters, the “Minority” paradigm is the best performing implementation for values of $R \geq 1.0$ meters. For $R < 1.0$ meters, the “Randomization” paradigm is the highest performing implementation. The one-sided p-values resulting from a mean difference test comparing the “Minority” and “Randomization” paradigms indicates that while there is an absolute performance difference, this difference cannot be demonstrated statistically with enough confidence as only three p-values are less than 0.05.

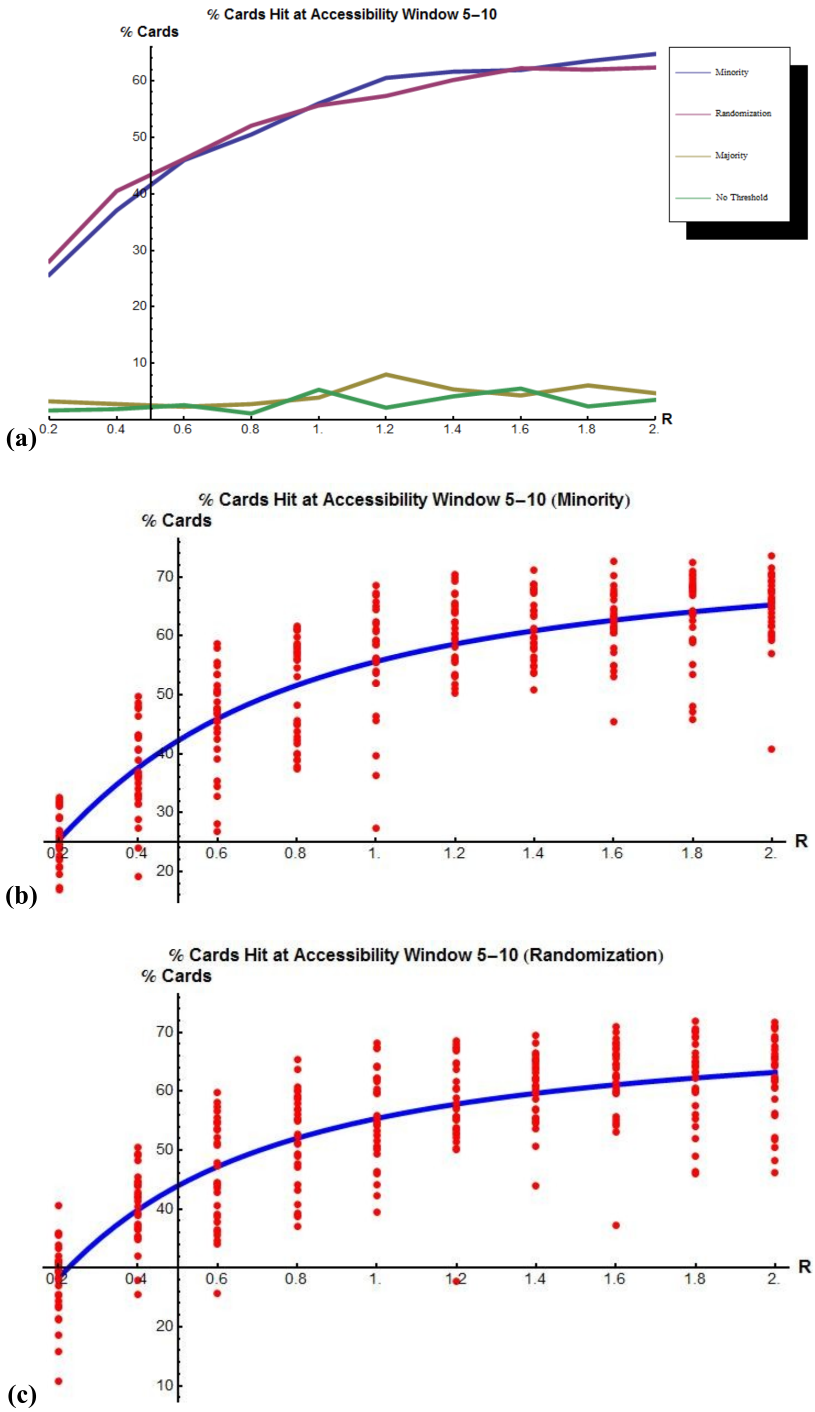


Figure 18(a)-(c): (a) Plot showing the relative average performance of different implementations of multi-gene genetic stigmergy. All implementations use accessibility windows of 5-10 hits, except for the "No Threshold" which allows robots to immediately use genetic information on the RFID cards. (b) Scatter plot of coverage percentages from all 30 "Minority Seeking" trials at each R value with a best fit-line described by y = 48.8787*ArcTan(0.172003+1.99357*x), with all parameter p-values <.005 and R^2 = 0.984. (c) Scatter plot of coverage percentages from all 30 "Randomization" trials at each R value with a best fit line described by y = 46.1076*ArcTan(0.231348+2.34793*x), with all parameter p-values <.005 and R^2 = 0.982.

R	**0.2**	**0.4**	**0.6**	**0.8**	**1.0**	**1.2**	**1.4**	**1.6**	**1.8**	**2.0**
“Minority”	25.7	37.1	46.0	50.5	**56.0**	**60.5**	**61.6**	61.9	**63.5**	**64.8**
“Randomization”	**28.1**	**40.5**	**46.2**	**52.1**	55.7	57.3	60.2	**62.2**	62.0	62.4
One-Sided P-Value	.048	.029	.463	.235	.435	.042	.160	.420	.213	.084
“Majority”	3.3	2.8	2.3	2.8	3.9	8.0	5.4	4.3	6.1	4.7
“No Threshold”	1.7	1.9	2.6	1.1	5.3	2.1	4.1	5.5	2.4	3.5

Table 2: Average percentage of cards covered for all R using accessibility window 5-10. The highest coverage values for each R are in bold red lettering. The one-sided p-values result from a mean difference test between coverage results of the “Minority” and “Randomization” trials at each R using a null hypothesis: μ_1-μ_2=0. Only three values (highlighted in yellow) were significant ($p<.05$).

Closer examination of the “Minority Seeking” paradigm at R ≥ 1.0 meters allows for a greater understanding of how it influences robot behaviors over the course of a simulation and compares to the “Randomization” paradigm. Figures 19(a)-(c) provide tentative evidence that the swarm is collectively reacting to plateaus in the acquisition of “newly found cards” by increasing the rate at which they diversify their behavior and then decreasing it when the plateaus are overcome. In plot (a), for example, card discovery plateaus around t = 40 s until t=50 when the rate of behavioral change increases. Eventually, the rate of new card discovery increases as well, out of a plateau.

Figures 20, 21 and 22 all show the average gene usage for all robots at each locus averaged at each second of the overall simulation. The values R = 1.2 and 2.0 meters are used because at these values, the “Minority Seeking” paradigm outperforms the “Randomization” paradigm by the largest margins (3.2% and 2.4% respectively). The value R = 1.4 meters is used as an intermediary value for continuity. The initialization of all robots with the [1,1,0] chromosome (causing them to rotate away from other robots and obstacles and move outwards), appears to skew the initial allele distribution on all loci. The patterns of allele distribution at each locus act as signatures that help differentiate the “Minority Seeking” paradigm from the “Randomization” paradigm, where the alleles are *always* uniformly distributed regardless of initial conditions.

After exploring the “Minority Seeking” paradigm at a fixed accessibility window, further experiments are performed to examine the effect modifying the window has on coverage. Figure 23 displays the accessibility windows that produce the optimal performance at a given R value. None of the windows begin below 4 hits and, with the exception of R = 1.4 meters, all windows begin at 5 hits or higher. Most windows have a width of 3 hits. As Table 3 indicates, if the highest performing (or optimal) accessibility windows for “Minority Seeking” are used, the “Minority Seeking” paradigm always outperforms the “Randomization” paradigm, on average, by approximately 1-4%. However, as with the fixed windows, a statistical difference between the two paradigms cannot be demonstrated with enough confidence as only three p-values are less than 0.05.

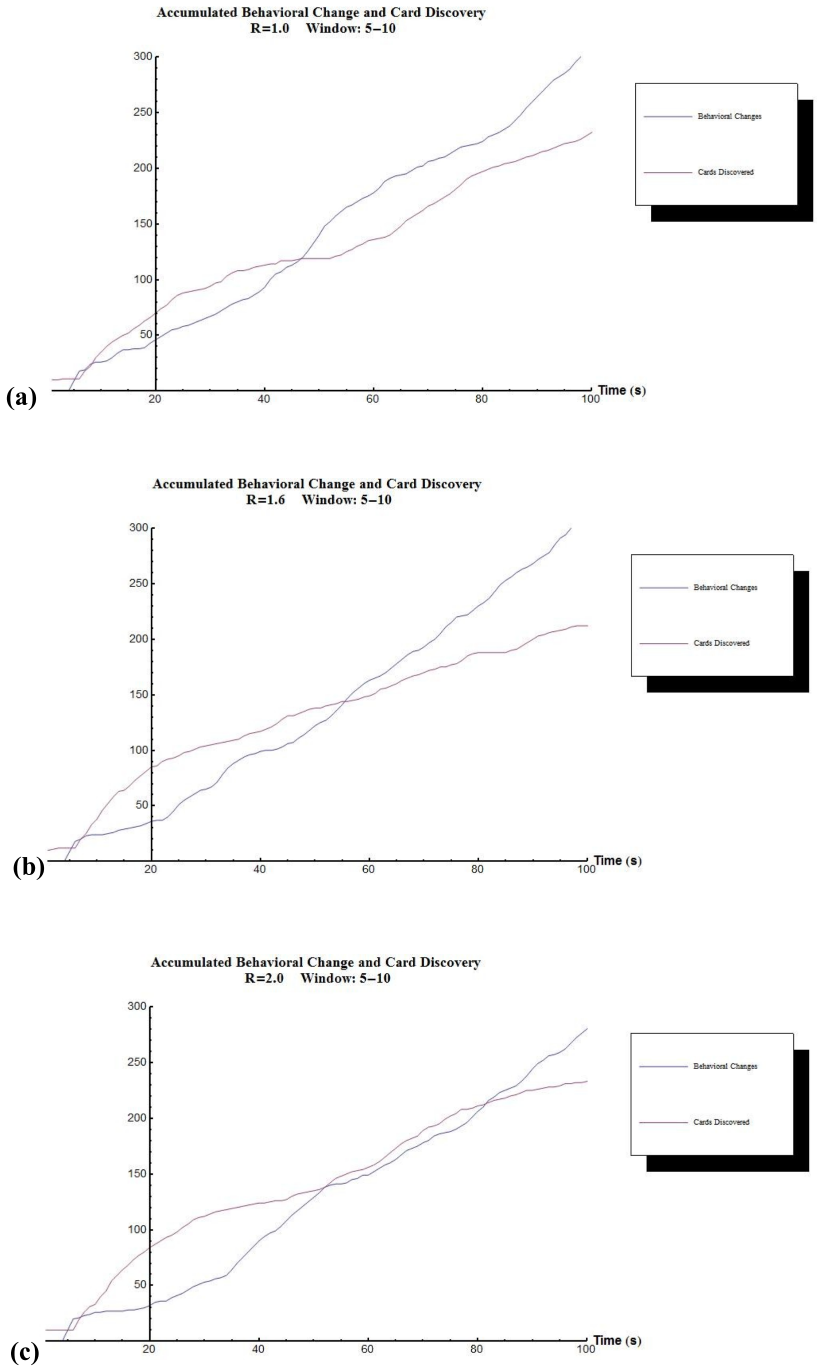
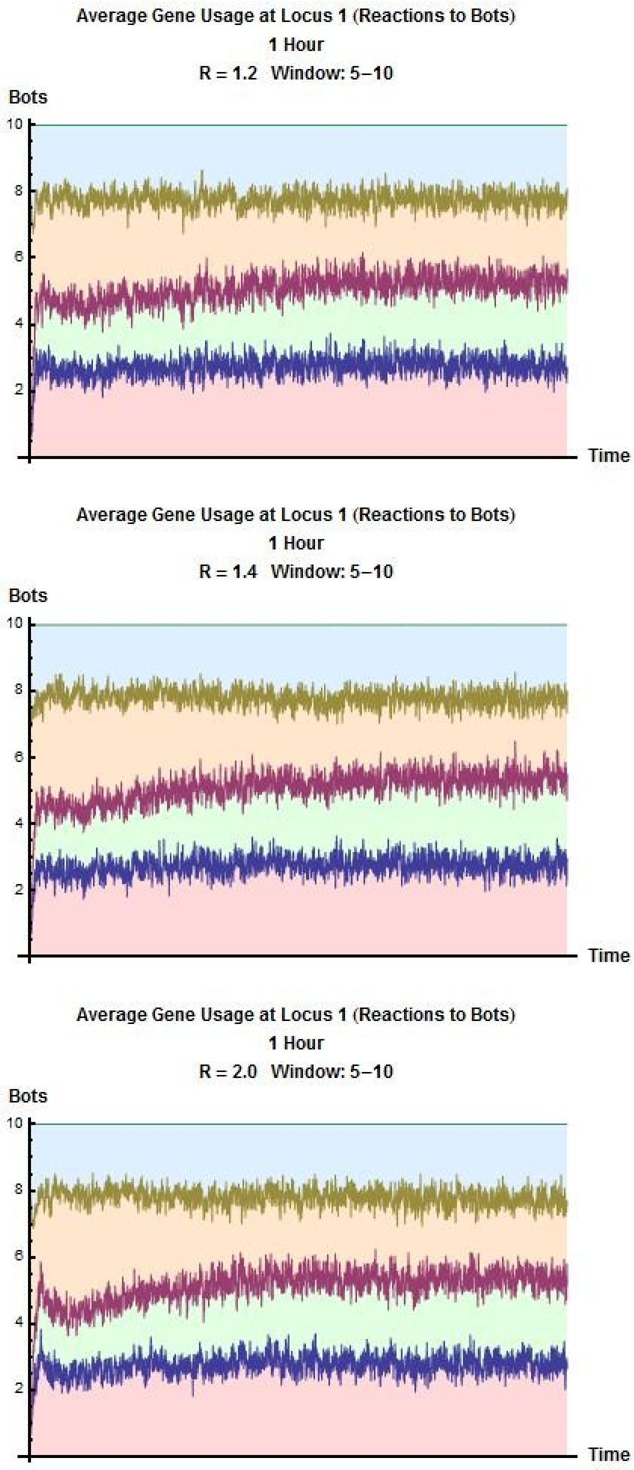


Figure 19(a)-(c): Time-series plots of the cumulative number of changes in robot behaviors and the cumulative number of cards discovered by robots over 100 seconds at accessibility window 5-10 and R = 1.0, 1.6 and 2.0 meters. Plateaus in accumulation are visible at various time scales where the rate of card discovery or behavioral change slows down. The "Behavioral Changes" curve is blue and the "Cards Discovered" curve is purple.

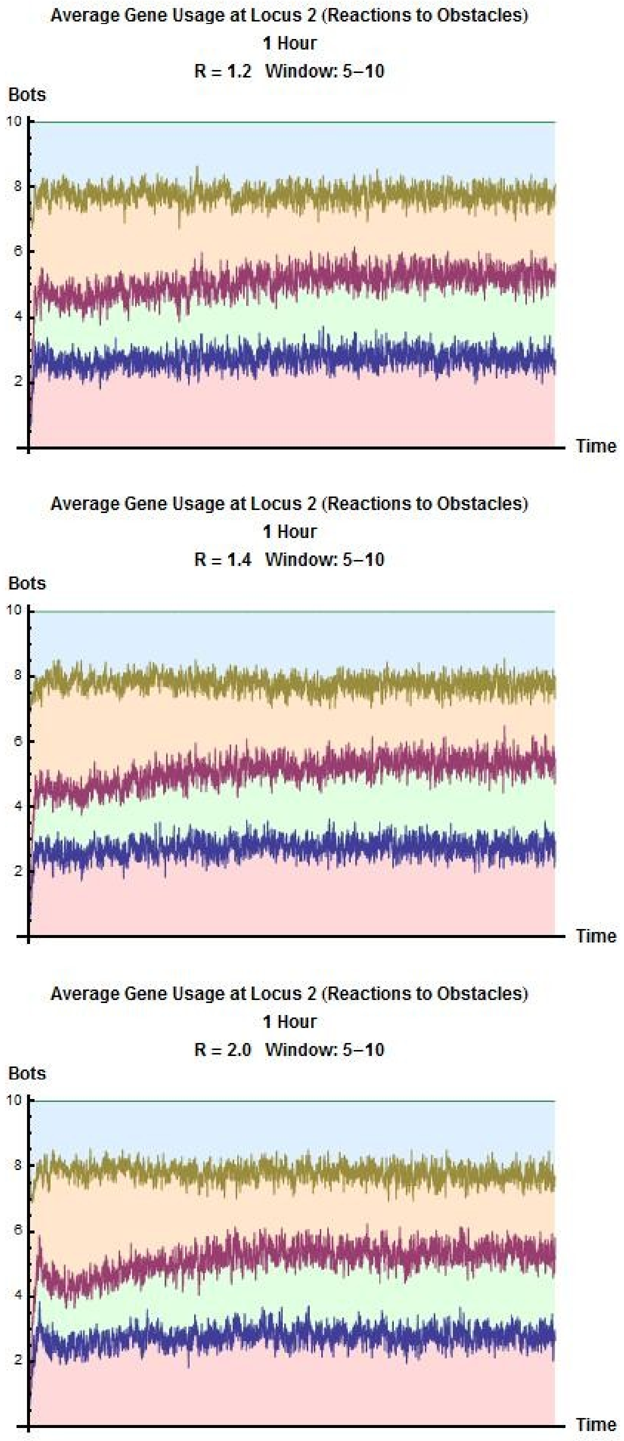


Average Gene Usage at Locus 1 (Reactions to Bots)
1 Hour
R = 1.2 Window: 5–10

Average Gene Usage at Locus 1 (Reactions to Bots)
1 Hour
R = 1.4 Window: 5–10

Average Gene Usage at Locus 1 (Reactions to Bots)
1 Hour
R = 2.0 Window: 5–10

Figure 20: Average gene usage for all robots at locus 1 (controlling reactions to other robots) for entire simulation (1 hour). Selected R values 1.2, 1.4 and 2.0 are used with accessibility windows of 5-10. Colors correspond to the four different possible alleles such that red indicates "rotate towards", green indicates "rotate away", orange indicates "back away" and blue indicates "random reaction".



Average Gene Usage at Locus 2 (Reactions to Obstacles)
1 Hour
R = 1.2 Window: 5–10

Average Gene Usage at Locus 2 (Reactions to Obstacles)
1 Hour
R = 1.4 Window: 5–10

Average Gene Usage at Locus 2 (Reactions to Obstacles)
1 Hour
R = 2.0 Window: 5–10

Figure 21: Average gene usage for all robots at locus 2 (controlling reactions to static obstacles) for entire simulation (1 hour). Selected R values 1.2, 1.4 and 2.0 are used with accessibility windows of 5-10. Colors correspond to the four different possible alleles such that red indicates "rotate towards", green indicates "rotate away", orange indicates "back away" and blue indicates "random reaction".

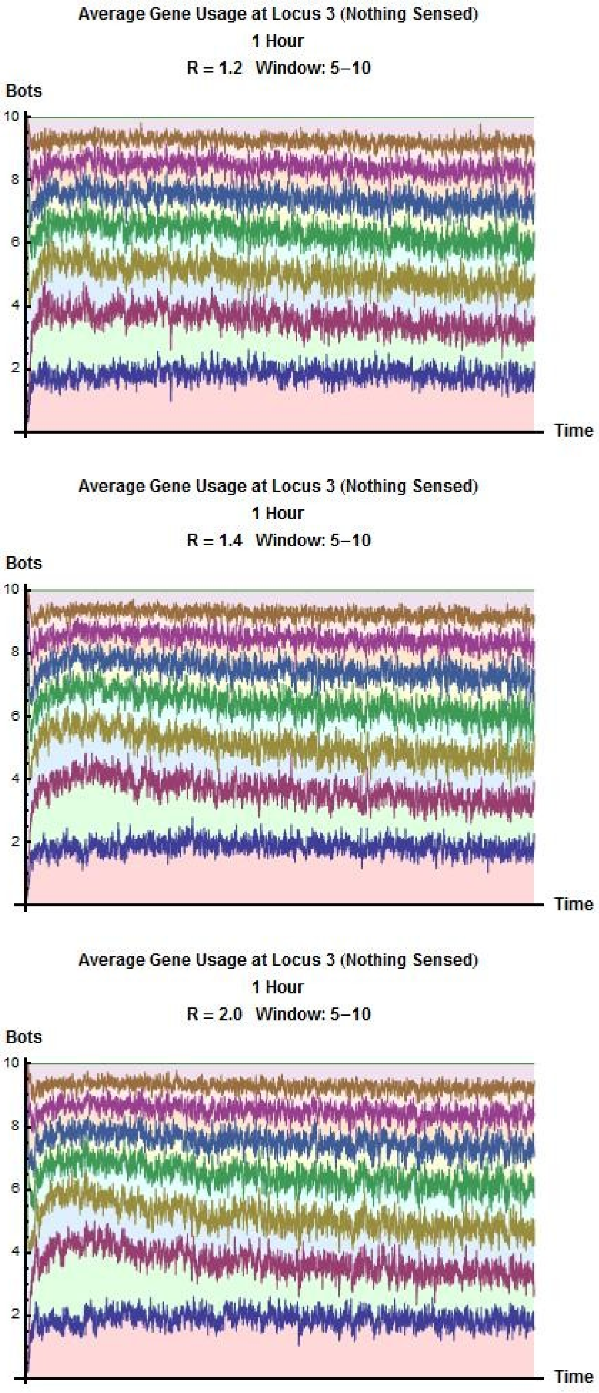


Figure 22: Average gene usage for all robots at locus 3 (controlling motions when no obstacles/robots) for the entire simulation (1 hour). Selected R values 1.2, 1.4 and 2.0 are used with accessibility windows of 5-10. Colors correspond to the eight different possible alleles such that red, green, blue, cyan, yellow, orange, pink, and purple correspond to "move forwards", "move backwards", "move forwards twice and rotate left", "move forwards twice and rotate right", "move backwards twice and rotate left", "move backwards twice and rotate right", "random rotate or forwards motion", and "random rotate or backwards motion" respectively (see from bottom up).

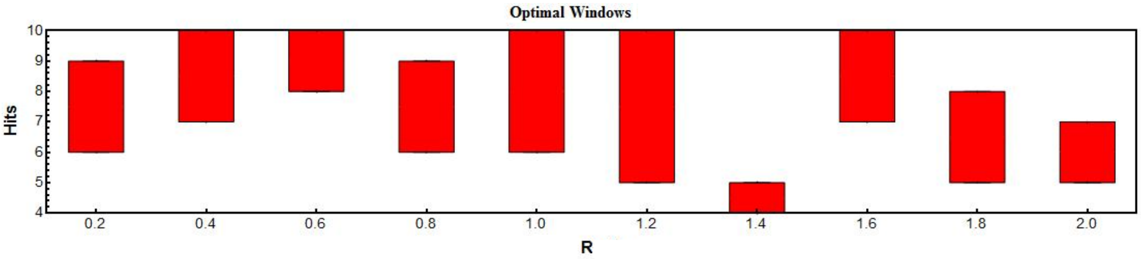


Figure 23: Optimal accessibility windows for "Minority Seeking" paradigm at all R values.

R	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2.0
"Minority"	29.5	39.5	50.0	56.2	59.4	61.0	61.8	64.4	65.0	65.3
"Randomization"	28.3	36.4	48.2	52.5	57.3	58.5	60.0	61.6	63.1	62.0
Absolute Difference	**+1.2**	**+3.1**	**+1.8**	**+3.7**	**+2.1**	**+2.5**	**+1.8**	**+2.8**	**+1.9**	**+3.3**
One-Sided P-Value	.222	.060	.111	.040	.181	.042	.072	.176	.100	.019

Table 3: Table of average coverage percentages for R values between 0.2 meters and 2.0 meters at their respective optimal accessibility windows. At every R, the "Minority Seeking" paradigm outperformed the "Randomization" paradigm. The net performance differences between the two averages are given in the third row by the green bolded values. The one-sided p-values result from a mean difference test between coverage results of the "Minority" and "Randomization" trials at each R using a null hypothesis: μ_1-μ_2=0. Only three values (highlighted in yellow) were significant (p<.05).

Chapter 6: Discussion

6.1 Analysis of Experimental Results

The experimental results acquired from testing the “single-gene” and “multi-gene” frameworks refute the first hypothesis and support the second hypothesis described in Chapter 3. In the “single-gene” experiments, randomization outperforms the *K*-based algorithm. In the “multi-gene” experiments, though there are absolute performance differences between the “Minority-Seeking” and “Randomization” paradigms, mean differences tests (see Tables 2 and 3) indicate there is not enough statistical evidence to indicate a difference for all R values. Thus, the first hypothesis that genetic stigmergy (as implemented in this thesis) is an improvement over random switching is not satisfied. However, the failure of the “No Threshold” paradigm to produce coverage results anywhere near those of the “Minority Seeking” paradigm indicate that hard-wired restrictions on when agents can access information are needed to ensure collective success. Thus, the second hypothesis that indicates hard-wired “meta-rules” are necessary for effective task-oriented behavior is satisfied.

6.1.1 “Single-Gene” Framework

Through manipulation of *K* values in the single-gene genetic stigmergy experiments, a significant correlation between probability of accepting diverse genes and the degree of variance in coverage values is demonstrated. When *K* is negative, the robots intentionally avoid redundancy by avoiding the adoption of a gene already written to an RFID card. The more negative K is the more likely robots are to avoid redundant genes.

With $K \geq 0$, greater K values correspond with higher coverage variance. When the simulations are started, there usually are one or more RFID cards beneath the swarm. Behaviors corresponding with the genes *SpiralOut* and *SpiralWander* do not cover nearly as large a space as other genes, but they are very efficient at almost completely covering the local space around the robots. When all the robots are close together in the beginning, this means that robots with "spiral" genes tend to hit RFID cards under the swarm before other robots. With high positive K values, a majority of the swarm is more likely to adopt these globally inefficient genes and constrict them to the space near their starting positions. This "winner-takes-all" approach allows certain genes to become locally adaptive, but it also makes the swarm very sensitive to initial conditions and less robust at mapping an environment.

With $K < 0$, increasingly negative K values increase the probability of a robot adopting a gene different from that on a card. By preventing the "winner-takes-all" scenario, the overall variance of the coverage decreases significantly without detrimental effects on performance. The "diversity" of the swarm is effectively increased while avoiding the pitfalls seen in other experiments where specialized task completion algorithms can cause swarm behavior to become stuck in local performance minima (Nouyan, Campo & Dorigo, 2007). In these instances, the "diversity" is manually added by a human operator. This diversity is illustrated by Figure 12, which indicates that negative K values prevent any one gene from dominating the swarm. Positive K values prevent such dynamic shifting in gene usage, so performance can vary greatly depending on which gene is dominant within the swarm.

6.1.2 "Multi-Gene" Framework

The multi-gene framework is built on the limited success of the single-gene framework in that its primary incarnation, the "Minority-Seeking paradigm", encourages local diversity of genetic material while the "Majority-Seeking" paradigm encourages a "winner-takes-all" course of events. As in the single-gene experiments, the "Majority-Seeking" paradigm produces poor coverage results on average, while the "Minority-Seeking" paradigm and "Randomization paradigm" produce similarly high coverage rates.

The split performance dominance between the "Minority" and "Randomization" paradigms makes more sense when considered in the context of how humans organize their domestic and working spaces. Some research (Crabtree & Rodden, 2004) examining how people move about their homes suggests that, in general, "obstacles" or fixtures in a space tend to be aligned more along the edges of an environment than in the middle. This more "open" environmental layout generally corresponds with R > 1.0, where there are larger obstacles more sparsely distributed. With R < 1.0, there are many tiny obstacles densely distributed in the environment. With such environmental complexity, there is likely no better way to explore than stochastic penetration via simple random motions.

Figures 19(a)-(c) demonstrate that the "Minority Seeking" paradigm (unlike the "Randomization" paradigm) allows robots to collectively react to plateaus in new card discovery by more rapidly modifying their genes. However, while the plateaus described in the results are promising, further experimentation is necessary to determine the existence and strength of any casual relationship present. Also, due to the variance in robot positions, environmental complexity and environmental layout, the scale of plateaus may vary widely over time and space. This may make causal confirmation more difficult.

Figures 20, 21 and 22 help explain why the “Minority Seeking” paradigm outperforms the “Randomization” paradigm when $R \geq 1.0$. At loci 1 and 2 (Figures 20 and 21), which control how robots react to other robots and obstacles, there appears to be an initial bulge of gene usage associated with the “back away” behavior. This may indicate that the “back away” behavior is useful in finding new RFID cards in the beginning of a simulation (preventing robots from avoiding the gene). After some time, the distribution settles so that all genes are more-or-less equally represented among the swarm at any given time. At locus 3 (Figure 22), the “move forwards” and “move backwards” genes are initially represented somewhat more than other alleles. Later, the allele distribution also settles, but the “move forwards” gene maintains a relatively larger presence in the swarm.

In the “Randomization” paradigm, the haphazard shuffling of genes would, on average, produce an allele distribution such that all genes were represented equally. In the “Minority Seeking” paradigm, this even distribution only appears that the end of the simulations. Even then, the distribution is often slightly skewed in favor of effective genes (such as “move forwards”). The fact that the “Minority Seeking” paradigm allows for adaptive genes to be “over-represented” in the allele distribution is likely the reason why it generally outperforms the “Randomization” paradigm at R > 1.0.

The “Minority Seeking” paradigm outperforms the “Randomization” paradigm for all R values when optimal accessibility windows are used, but this is not very useful from a practical standpoint. One may not know which accessibility values to use before deploying robots. The environmental complexity must be specifically quantified, but this is not easy to accomplish “on the spot” in the real world.

While the genetic stigmergy framework appears promising, several potential flaws or confounding errors in the experimental design must be considered. In the multi-gene genetic stigmergy experiments, the order in which robots acknowledge sensor readings indicating obstacles may affect their overall motion. The construction of the simulated environments themselves may pose an issue. The fact that many RFID cards are hidden under obstacles or locked away in permanently inaccessible regions (in the highly complex environments) may skew coverage results. Also, the uniformity of the obstacles (all where circular) and consistently square environment shape may not be fully representative of environments in the real world.

In an effort to more convincingly demonstrate the potential of genetic stigmergy, the framework must be thoroughly examined in the context of other simpler and more complex techniques to determine its true usefulness in practical applications. More advanced techniques may affect better performance at the cost of increased time and effort spent developing individual learning processes. However, this cost-benefit compromise may change depending on the environment and local constraints placed on the system. In addition, a more robust testing of genetic stigmergy under a unified experimental framework is necessary. The “multi-gene” experiments added many new parameters that did not exist in the “single-gene” experiments and, most importantly, jumped from using one gene per agent to three. Future experiments should run multiple Monte Carlo simulations for all algorithm paradigms using one to three genes, an increasing density of RFID cards (from very sparse to highly dense) and varying environmental sizes with different arrangements and shapes of obstacles. Only by thoroughly testing each parameter in a consistent fashion can it be determined which parameter(s) is/are most important in the successful implementation of genetic stigmergy. Also, aside from space filling, it is likely there are other engineering applications where genetic stigmergy can more clearly differentiate itself from other techniques. Other application possibilities should be thoroughly explored to determine what is genetic stigmergy’s “killer application”.

Potential improvements to the system include time-stamping the genetic information deposited to RFID cards or "smarter" RFID cards that can exchange information locally with other RFID cards to help coordinate robots (similar to Werfel & Nagpal, 2006, where blocks and robots can communicate). In addition, allowing for multiple simultaneous accessibility windows in different areas of the map depending on local need (as in Dorigo, Bonabeau & Theraulaz, 2000) may help if, and only if, the swarm can autonomously determine the proper thresholds on the fly. Allowing for genetic evolution within the robotic swarm may help in this regard. Through the introduction of such operators as mutation, novelty can be introduced to the system that helps robots discover more adaptive behavior or ways of interacting with the environment.

Individualized manipulation of accessibility windows through the use of evolutionary operators may be the most important factor in significantly improving the performance of genetic stigmergy. The importance of thresholds has already been thoroughly examined by Dorigo and Bonabeau (Dorigo, Bonabeau & Theraulaz, 2000), who note that simple threshold models have limitations due to their fixed nature and are only valid over short-time scales. In the longer term (perhaps as a function of time), accessibility windows should change and differentiate to allow for agent specialization. Future work will determine if such mutability of thresholds is feasible or realistic.

6.2 Other Applications

If genetic stigmergy can be made more robust, it has the potential to positively influence many fields beyond swarm robotics. By abstracting the framework, it can be applied to everything from “web crawlers” prowling the Internet, to human beings perusing in retail environments. Two particularly exciting applications are decentralized advertising and green energy management, where effective genetic stigmergy would sidestep the need for central management without limiting the value or efficacy of the application.

6.2.1 Decentralized Advertising

Advertisers are constantly looking for ways to make their ads more relevant to consumers through ever-more personalized recommendations. They achieve success through the monitoring of a customer's web and social networking habits, but often at the cost of that customer’s privacy. With ever-more intrusive technologies (such as eye and face-scanning technologies) in consideration, many people are worried that their information will be stolen by identity thieves or worse. However, the idea that further customization must always come at the cost of privacy is not necessarily true.

With genetic stigmergy, users can carry simple devices with information that indicates their consumer preferences and privacy settings. When the customers pass stores or ad kiosks, their buying patterns and preferences are anonymously recorded and used to pull up proper advertisements or discounts. A kiosk can even modify its own behavioral genes to produce "advertising phenotypes" that better match the tastes or moral preferences of customers in the area. The benefit of this system is that it is totally decentralized (given that the kiosk has an adequate reservoir of base ad material for adaptation) and the only humans with direct access to consumer preferences are the consumers themselves. They could even "turn off" their devices and be totally invisible to customization in a retail setting. This framework can also be hybridized with pheromonal stigmergy such that a consumer’s preference can be set to evaporate after a certain period.

6.2.2 Green Energy Management

Increasing world demand for energy is spurring the research and development of smarter grid management technologies. Normal grids are very inefficient and centralized, making it hard for them to react to subtle changes in demand in local areas. Li, Poulton & James, 2007) review current smart approaches, such as centralized planning algorithms or agent-based market-oriented algorithms (where agents carry out negotiations with resources to fix prices) but find that many run into problems of scalability.

Stigmergic systems, properly applied, allow for greater scalability. ANT algorithms (mentioned in the Background) can help support efficient power routing by building up virtual pheromone traces at nodes (homes or utilities) along the most efficient path. Genetic stigmergy could complement these ant algorithms or virtual agents by augmenting their capacity for behavioral development. Each home or utility would send out a number of virtual "ants" that negotiate (according to their genes) with other ants to determine the best way to route power, or even to set prices. If a certain number of ants (from different 'owners') find a particularly beneficial way of negotiating with another ant from a utility, it can leave a portion of its genetic code in some local environment to be read later by other ants looking for better negotiating tactics. Ant behaviors could adapt to the local structure of a local portion of a grid and the desire of its residents and might even co-evolve or form a larger collective to represent their interests, potentially as a virtual species.

Chapter 7: Conclusion

Genetic stigmergy holds great promise as an alternative to pheromonal-based artificial stigmergy for the achievement of collective action through self-organization. Unlike pheromonal-based stigmergy, genetic stigmergy is not mired in unnecessary complications due to attempts to mimic chemical diffusion. Limited experimentation indicates that multi-gene genetic stigmergy may be an effective tool in such fields as swarm robotics, but much theoretical work remains to be done to demonstrate this framework’s robustness in robotics and elsewhere. In addition, the implementation of behavior meta-rules to control agent access to local information appears necessary to direct a swarm's emergent behavior to useful ends. These interaction restrictions are even more important in a system where agents can evolve their behaviors in real-time. Future work and further experimentation will address these issues and help develop genetic stigmergy into a viable platform for decentralized communication.

Notes

To acquire the underlying code for the simulations in this thesis, please contact Joshua Brandoff at josh.brandoff@gmail.com

Bibliography

3DRad. (2008). http://www.3DRad.com.

Howell, A. (2008). Abe Howell's Robotics. http://www.abotics.com.

Beckers, R., Holland, O.E., & Deneubourg, J.L. (1994). From local actions to global tasks: stigmergy and collective robotics, *Artificial Life IV*, pages 181–189. MIT Press.

Bekkali, A., Sanson, H. & Matsumoto, M. (2007). RFID indoor positioning based on probabilistic RFID map and Kalman filtering, *WIMOB '07: Proceedings of the Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, pages 21, Washington, DC, USA. IEEE.

Bonabeau, E., Silvain, G., Snyers, D., Kuntz, P. & Theraulaz, G. (2000). Three-dimensional architectures grown by simple "stigmergic" agents, *BioSystems*, 56:13–32.

Brandoff, J., & Sayama, H. (2009). Cultural transmission in robotic swarms through RFID cards, *Proceedings of the Second IEEE Symposium on Artificial Life (IEEE-CI-ALIFE '09*), pages 171–178, Nashville, TN, USA. IEEE.

Carbone, A., Finzi, A. & Orlandini, A. (2008). Model-based control architecture for attentive robots in rescue scenarios, *Autonomous Robots*, 24(1):87–120.

Chen, P.Y., Chen, W.T., Wu, C.H., Tseng, Y.C. & Huang, C.F. (2007). A group tour guide system with RFIDs and wireless sensor networks, *IPSN '07: Proceedings of the 6th International Conference on Information Processing in Sensor Networks*, pages 561–562, New York, NY, USA. ACM.

Crabtree, A. & Rodden, T. (2004). Domestic routines and design for the home, *Computer Supported Cooperative Work (CSCW)*, 13(2):191-220.

Dorigo, M., Bonabeau, E. & Theraulaz, G. (2000). Ant algorithms and stigmergy, *Future Generation Computer Systems*, 16(9):851–871.

Google 3D Warehouse. (2008). http://sketchup.google.co/3dwarehouse.

Herianto, Sakakibara, T., & Kurabayashi, D. (2007). Artificial pheromone system using RFID for navigation of autonomous robots, *Journal of Bionic Engineering*, 4(4): 245–253.

Holland, O. & Melhuish, C. (1999). Stigmergy, self-organisation, and sorting in collective robotics, *Artificial Life*, 5(2):173–202.

Holldobler, B. & Wilson, E.O. (2009). *Superorganism: The Beauty, Elegance and Strangeness of Insect Societies,* W. W. Norton & Company, Inc., New York, NY, USA.

Howard, A., Parker, L.E. & Sukhatme, G. S. (2006). Experiments with a large heterogeneous mobile robot team: exploration, mapping, deployment and detection, *International Journal of Robotics Research*, 25(5-6): 431–447.

Howell, A., McGrann, R., Eckert, R., Sayama, H. & Way, E. (2006). Using RFID and a low cost robot to evolve foraging behavior, *Late Breaking Paper of the 2006 Genetic and Evolutionary Computation Conference*.

Kim, M. & Chong, N.Y. (2007). RFID-based mobile robot guidance to a stationary target. *Mechatronics*, 17(4–5):217–229.

Kleiner, A. & Dornhege, C. (2007). Real-time localization and elevation mapping within urban search and rescue scenarios: field reports, *Journal of Field Robotics*, 24(8-9): 723–745.

Kleiner, A., Prediger, J. & Nebel, B. (2006). RFID technology-based exploration and SLAM for search and rescue, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4054–4059, Beijing, China.

Lee, H.J. & Lee, M.C. (2006). Localization of mobile robot based on radio frequency identification devices, *SICE-ICASE International Joint Conference*, pages 5934-5939.

Li, J., Poulton, G. & James, G. (2007). Agent-based distributed energy management, *AI 2007: Advances in Artificial Intelligence*, pages 569–578.

Mamei, M., Quaglieri, R., & Zambonelli, F. (2006). Making tuple spaces physical with RFID tags, *SAC '06: Proceedings of the 2006 ACM Symposium on Applied Computing*, pages 434–439, New York, NY, USA. ACM.

Mamei, M. & Zambonelli, F. (2005). Physical deployment of digital pheromones through RFID technology, *AAMAS '05: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*. New York, NY, USA: ACM, pp. 1353–1354.

Mamei, M. & Zambonelli, F. (2005). Programming stigmergic coordination with the TOTA middleware, *AAMAS '05: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 415–422, New York, NY, USA. ACM Press.

Mamei, M. & Zambonelli, F. (2007). Pervasive pheromone-based interaction with RFID tags. *ACM Transactions on Autonomous and Adaptive Systems*, 2(2):4.

Milella, A., Cicirelli, G. & Distante, A. (2008). RFID-assisted mobile robot system for mapping and surveillance of indoor environments. *Industrial Robot*, 35(2): 143–152.

Nouyan, S., Campo, A., & Dorigo, M. (2007). Path formation in a robot swarm, *Swarm Intelligence*, 2(1): 1–23.

Parunak, H.V.D. (2006). A survey of environments and mechanisms for human-human stigmergy, *Environments for Multi-Agent Systems II*, 3830: 163–186.

Parunak, H.V.D. (2006). Evolving swarm agents in real time. *Genetic Programming Theory and Practice III*. Chapter 2, pages 15-32.

Patil, A., Munson, J., Wood, D. & Cole, A. (2008). Bluebot: Asset tracking via robotic location crawling. *Computer Communications*, 31(6):1067–1077.

Ramos, V. & Abraham, A. (2004). Evolving a stigmergic self-organized data-mining. http://arxiv.org/abs/cs.AI/0403001.

Ramos, V. & Merelo, J.J. (2004). Self-organized stigmergic document maps: environment as a mechanism for context learning. http://arxiv.org/abs/cs.AI/0412075.

Roth, M. & Wicker, S. (2003). Termite: ad-hoc networking with stigmergy, *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, (5)2937–2941.

Roussos, G., Papadogkonas, D., Taylor, J., Airantzis, D., Levene, M. & Zoumboulakis, M. (2007). Shared memories: a trail-based coordination server for robot teams, *RoboComm '07: Proceedings of the 1st International Conference on Robot Communication and Coordination*, pages 1–4, Piscataway, NJ, USA. IEEE Press.

Simon, H.A. (1956). Rational choice and the structure of the environment, *Psychological Review*, 63(2):129-138

Theraulaz, G. & Bonabeau, E. (1999). A brief history of stigmergy. *Artificial Life*, 5(2):97–116.

Ulieru, M. & Unland, R. (2006). A Stigmergic approach to medical diagnostics, *MAS*BIOMED'06: Second International Workshop on Multi-Agent Systems for Medicine, Computational Biology and Bioinformatics*, pages 87-103, Hakodate, Japan. AAMAS.

Wagner, I.A., Lindenbaum, M. & Bruckstein, A.M. (1999). Distributed covering by ant-robots using evaporating traces. *IEEE Transactions on Robotics and Automation*, 15:918–933.

Werfel, J. & Nagpal, R. (2006). Extended stigmergy in collective construction. *IEEE Intelligent Systems*, 21(2):20–28.

White, T. (2005). Expert assessment of stigmergy: a report for the Department of National Defence. Technical report A382144. Department of Computer Science, Carleton University, Ottawa, ON, Canada.

White, T. & Salehi-Abari, A. (2008). A swarm-based crossover operator for genetic programming. *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, New York, NY, USA. ACM.