OPTIMIZATION OF SWARM ROBOTIC CONSTELLATION COMMUNICATION FOR OBJECT DETECTION AND EVENT RECOGNITION

By

Matthew R. Proffitt

A Thesis
Submitted to the
Faculty of the Graduate School
of
Western Carolina University
in Partial Fulfillment of
the Requirements for the Degree
of
Master of Science in Technology

Committee:

______________________________ Director

____

____

______________________________ Dean of the Graduate School

Date: __________________________

Spring 2011
Western Carolina University
Cullowhee, North Carolina

OPTIMIZATION OF SWARM ROBOTIC CONSTELLATION COMMUNICATION FOR OBJECT DETECTION AND EVENT RECOGNITION

A thesis presented to the faculty of the Graduate School of
Western Carolina University in partial fulfillment of the
requirements for the degree of Master of Science in Technology.

By

Matthew R. Proffitt

Director: Dr. Brian Howell
Assistant Professor
Department of Engineering Technology

Committee Members:
Dr. Peter Tay, Department of Engineering Technology
Mr. Paul Yanik, Department of Engineering Technology

April 2011

ACKNOWLEDGEMENT

I would like to thank my director Dr. Brian Howell for his assistance, encouragement, and enthusiasm. No matter where my life's path takes me, I will never forget the lessons learned through his guidance and honest friendship. I would also like to thank my committee members Dr. Peter Tay and Mr. Paul Yanik for their tireless work and help over my academic career. Their insight and backing have helped me stand against daunting odds.

I would also like to thank Dr. Erin McNelis for selflessly giving her time and effort to my research. I also extend my thanks to Dr. James Zhang for his teachings and for convincing me to attend graduate school. You have both been a credit to the spirit of higher learning.

I wish to thank my family: my parents Michael and Valencia for their caring and endless support, my grandparents who have always been my refuge, and my brother Jason who has shared every step of my life with me, from tying shoes to MST degrees. You have all influenced the person I am today and I hope to live up to the example you have set.

Last but not least, I would like to thank my fellow graduate student colleagues. In particular A.J. Punch, who's help and friendship I will be eternally grateful for, Christopher Rhoades, Kirke Shouse, Joshua Deaver, LakshmiNarasimhan SrinivasaRaghavan and William Gall.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

OPTIMIZATION OF SWARM ROBOTIC CONSTELLATION COMMUNICATION FOR OBJECT DETECTION AND EVENT RECOGNITION

Matthew R. Proffitt, M.S.T.

Western Carolina University (April 2011)

Director: Dr. Brian Howell

Swarm robotics research describes the study of how a group of relatively simple physically embodied agents can, through their interaction collectively accomplish tasks which are far beyond the capabilities of a single agent. This self organizing but decentralized form of intelligence requires that all members are autonomous and act upon their available information. From this information they are able to decide their behavior and take the appropriate action. A global behavior can then be witnessed that is derived from the local behaviors of each agent. The presented research introduces the novel method for optimizing the communication and the processing of communicated data for the purpose of detecting large scale *meta* object or event, denoted as *meta* event, which are unquantifiable through a single robotic agent.

The ability of a swarm of robotic agents to cover a relatively large physical environment and their ability to detect changes or anomalies within the environment is especially advantageous for the detection of objects and the recognition of events such as oil spills, hurricanes, and large scale security monitoring. In contrast a single robot, even with much greater capabilities, could not explore or cover multiple areas of the same environment

simultaneously. Many previous swarm behaviors have been developed focusing on the rules governing the local agent to agent behaviors of separation, alignment, and cohesion. By effectively optimizing these simple behaviors in coordination, through cooperative and competitive actions based on a chosen local behavior, it is possible to achieve an optimized global emergent behavior of locating a *meta* object or event. From the local to global relationship an optimized control algorithm was developed following the basic rules of swarm behavior for the purpose of *meta* event detection and recognition. Results of this optimized control algorithm are presented and compared with other work in the field of swarm robotics.

CHAPTER 1: INTRODUCTION

1.1 Hypotheses

Being that the interaction and feedback to and from members of a swarm is the basic mechanism of swarm functionality, the optimization of these agent actions and decisions can effectively optimize a behavior that is emulated by the swarm as a whole. To this end it can be hypothesized that if the local interaction and decision making of agents in a swarm can be optimized to emulate small scale behaviors that can supplement an overall task in the case of the research to detect a *meta* event.

1.2 Background

Swarm (Multi-Robot) robotics is a growing field of robotics in which a group of three or more simple robotic members are used in place of a single complex robot to perform actions, a technique which is adapted from natural swarm insects [2]. Each member of the swarm is an autonomous agent which is capable of deciding its own actions based on its perception of an environment through various sensors as described by Russell and Norvig [3]. Where Russell and Norvig classify agents into classes based on intelligence and capability, the presented research deals only with the class of model-based agents where current state information is stored within the agent. The storing of information for a set time allows the agent a structure for perceiving its reality over time. Information of each agent's environment is not limited to the single agent's perceptions, but can also be interpolated with information attained from agent to agent interactions. These intra-swarm interactions are key due to the effects on the behavior of the swarm agents, allowing them the ability to use propagated influences from agent to agent.

By using simple agents together to perform complex tasks, degrees of flexibility, robustness and speed are significantly increased when compared to the capabilities of a single robotic entity [4]. Through the models inspired by natural swarms, the simple behaviors of separation (dispersion), Alignment (flocking), and Cohesion (clustering) can be derived to be implemented to an artificial swarm [5]. The separation behavior, commonly witnessed in nature in insect foraging, allows an artificial swarm the capability to disperse from other agents. This allows the swarm to spread over a quantitatively large area. The alignment behavior, commonly witnessed in nature in schools of fish or flocks of birds, allows an artificial swarm a way for each agent to maintain an optimal distance from other agents. This allows the swarm as a whole to move along a single vector. The cohesion behavior, commonly witnessed in nature with the breading of young and the clustering of dead in insect swarms (most notably ants), allows an artificial swarm to move toward the barycenter of neighboring agents [4]. This allows the sorting and clustering of objects.

Self-organization is referred to as the broad range of pattern formation processes that occur in both physical and biological systems. A narrower constraint of self-organization can be applied to swarm robotics where a self-organized swarm is defined as a system in which an internal organization is present without any guiding influence from outside. Camazine et al. [6] further break down the concepts of self-organization into four principles: positive feedback, negative feedback, amplification of fluctuations, and multiple interactions. Positive being where developed solutions to objectives are utilized or desired behavior is encouraged. Negative controlling the swarm behavior to avoid inefficient solutions to objectives or deadlocks within agent-agent interaction. Amplification of the sensory and interaction fluctuations within a swarm are advantageous in attracting or repelling other agents to or from a point by which new solutions can be exploited. Multiple interactions within the swarm are the driving factor that can generate a high degree of complex outcomes. These in turn can lead to the emergence of a swarm behavior on the global level.

Each agent of the swarm performs tasks that can collectively be construed as a global action, from mapping an unknown space where a single agent maps only a small portion of the entire map [7] to the group transportation of objects where a single agent cannot achieve movement alone [8]. This global *Emergent* behavior can be an ill defined expression. In the concept of the research presented it will be limited to Hamann's definition that "a system has emergent properties (and is then called emergent), if a behavior on the macro-level is not explicitly programmed on the micro-level" [9]. Emergent behavior is key to the effectiveness of a swarm. As emergence derives from the low-level tasks solving low level problems faced by the agent, the emergence that arises can solve more complex problems faced by the swarm. The solving of complex problems through emergence is an important element of swarm intelligence: "Solutions to problems faced by a colony are emergent rather than predefined" [10]. Given the definition that a *meta* event is an event unquantifiable by a single agent within the swarm, a *meta* event can be considered a complex problem that requires emergent behavior to solve.

1.3 Objectives

The objective of the current research is to use known information about robotic swarms to construct an algorithm and framework for the detection of *meta* events. Most research in swarm robotics is centered on dissecting, rationalizing, and optimizing a single objective or aspect of a behavior, as in aggression [11], foraging [12], and object transportation [8].

The objective here is not to optimize a single aspect of a behavior, but to optimize the interactions of the various behaviors that make up the swarm for the purpose of detecting a *meta* event. Where a single behavior system could plausibly achieve a much better result when accomplishing a single objective by the swarm, the detection and recognition of a *meta* event requires simultaneous tasks. These swarm tasks are:

- The quantification of the *meta* event through defining a clamped bounding.
- The optimal coverage of the given field area, avoid clumping and redundancy.
- The efficient use of agent energy, where each agent efficiently achieves goals.

These intricacies of swarm robotics rely on the ability of communication with other agents in the swarm, which will be expanded upon on further in chapter two.

Communication between agents is essential for sharing information gathered by a single agent, thereby allowing each agent further knowledge of the environment or the ability to identify an event using multiple perspectives, as Chazelle [13] shows in triangulation. This communication scheme is the first area to which an optimization of the system can be applied. According to Farinelli et al. [14] it is possible to break down the the level of cooperation, knowledge, coordination, and organization within a swarm. Following the global swarm goals presented above, an optimal communication topology based on the taxonomy with a focus on distributed and strongly coordinated organizations will be chosen to achieve greater optimization for the swarm system.

The challenge in designing and implementing a swarm system as mentioned above is in achieving the emergence of a global swarm behavior. The challenge of “global-to-local programming” as it is defined in Yamins [15] and Yamins and Nagpal [16] is the fact that programming only occurs on the local agent level when the outcome is witnessed on the global swarm level. In fact, as described by Hamman [9], characterizing a global behavior before witnessing the emergent swarm behavior is difficult, if not impossible. This challenge is not specific to traditional systems engineering but encompasses an interdisciplinary investigation into the phenomenon of emergence. Haken notes that, "... despite a lot of knowledge about complex systems the application of this knowledge to the engineering domain remains difficult. Efforts are scattered over many scientific and engineering disciplines" [17]. The main point of conflict when designing a system to produce emergence occurs within the definition of self-organization: outside influence is prohibited. Given that the task of *meta* event detection and recognition is within the scope of this research, the problem of how a swarm system with the principles of self-organization accomplishes this task when outside interaction is not possible. Since traditional system engineering is not applicable for the task of swarm emergence a method of programming *by hand*, as in Hamman [9], must be employed to test abstraction levels of the swarm behaviors in a refinement method until an acceptable algorithm design is produced as explained further in chapter 3.

To assess the concept and capabilities of a swarm system optimized for the detection and recognition of *meta* events a simulation using Matlab will be constructed. Through this simulation the refinement of the swarm control algorithms may processed with the *by hand* technique.

1.4 Significance of Study

The communication of a *meta* event or object is especially helpful in the recognition of large events where single vehicle systems currently in use would be ineffective. Information on disasters (oil spills, radiation fallout, and forest fires), biological events(migrations,plankton blooms, and weather patterns), and large scale missions (security detection, location monitoring, and area mapping) could become quantifiable. This quantifiable information would lead the swarm to take action and/or inform a human population. Where a single vehicle system's *meta* event recognition would be limited to its on-board sensor capabilities and allotted loiter time around the event, a swarm system would be able to determine through inter-swarm communication the presence of a *meta* event in real time.

From the current research, and verified through simulations in chapters 4 and 5, progress may be made into the field of swarm intelligence. Through the control algorithms described the local behaviors of a swarm agent can be optimized to the mission of detecting *meta* events. Further work on these algorithms can refine and fully implement this work into a cognitive field-ready system capable of completing missions dependent upon the recognition of a *meta* event.

1.5 Delimitations of Study

The operation of swarm robotics can be divided into three main sub-problems:

1. The sensing of the member's local space.
2. The interaction and communication of the members within the swarm.
3. The resulting swarm behavior derived from each member.

Given that the swarm must be able to detect its surroundings to detect events within an environment, various sensors must be included in the robot's make-up. With these sensors, such as ultrasonic, acoustic, magnetic, and Global Positioning Systems (GPS), the member is able to characterize its surrounding. As objects move and events occur within the environment each member is able to detect abnormalities to be processed. In the presented research it is understood that each member has a visual sensory input from its environment in terms of distance to environmental boundaries and event presence. The relationship aspect of the swarm is defined as how the robotic swarm agents interact with every other agent as defined by Farinelli et al. [14]. The dynamics of the relationship between each agent being that they may or may not actively communicate and may or may not work cooperatively. Where cooperation between agents is considered the effect of influencing others to a shared goal.

A new dynamic of recent interest in the field of swarm research is the addition of the heterogeneity (dissimilarity) to the physical make-up of swarm members. The addition of heterogeneity allows a further degree of flexibility and robustness to the entire swarm as shown in Bonabeau et al. [4]. Heterogeneous teams have already proved successful in many applications. Advantages over homogeneous (similar members) system can already be witnessed in computing [18] and in robotic mapping and exploration [19]. Due to the advantages and overall optimization qualities presented by a heterogeneous aspect, it was determined that a completely optimized swarm should employ heterogeneity. The design and optimized behavior of a homogeneous swarm will be the focus of the presented research.

Further research into swarm technology would logically venture into the realm of multiple dimensions, where an agent would be capable of three dimensional movement. Real world problems would then become applicable to a swarm of these agents in an aquatic, aerial, or weightless environment. Given that the multi-dimensional implementation of a robotic swarm system in a dynamic environment has yet to be employed as of the current research it can be gathered that this would be an avenue for future work, but is not within the scope of the presented research.

Though the eventual goal of the proposed research is to create a swarm of deployable robotic agents to perform *meta* event detection, the current research is focused on the development of the agent control system. The challenges that arise from both agent-agent interaction and behaviors will be addressed and extended through the Matlab simulations. Discussion on the future implementations of a heterogeneous swarm make-up, aspects of a three dimensional spatial environment, and physical implementation will be discussed in chapter 5.

CHAPTER 2: LITERATURE SURVEY

This literature review covers the basic concepts of swarm robotics related to this research. Addressed by this review will be the overall aspects of designing multi-agent systems: the macroscopic-microscopic interaction, self-organization, and emergent behavior. Further expansion of these basic concepts introduces key specific aspects of *meta* detection: the local agent perspective, swarm coordination, and local decision making. Discussions will also include a introduction into avenues of future work for this research in tracking *meat* events and using heterogeneous members. The current research is supported by the previous research presented below and is framed within the scope presented in chapter one for detecting and recognizing *meta* events.

2.1 The Micro-Macro Effect

The microscopic to macroscopic problem was originally defined not in the field of engineering, but in the discipline of sociology. Defined in Alexander et al. [20] and Schillo et al. [21], the micro-macro link or micro-macro problem consists of the interactions between individuals in a society and the macro-structures they create. This micro or macro phenomena is studied from a sociologist's point of view, but many of the observations may be transposed into the field of swarm robotics as both deal with population dynamics. Observations include the reciprocal influence that the micro-interactions can generate on the macro-scale as well as the backpropagation influence the macro-scale has on the micro-scale. This closed loop system of interaction raises the question proposed by Alexander et al. as to how a determination is made to which level the influence originated from. Schillo et al. also attempts to bridge the gap between the disciplines of the sciences and sociology through the relation of distributed artificial intelligence (DAI) to other forms of artificial intelligence. Using the studies on micro and macro agent theories, a substantial grasp on the concepts that can contribute to understanding abnormalities in the creation and adjustment of local behaviors can be achieved.

2.2 Local Behavior

Interest into the micro, or local, behaviors of swarm agents was initially begun by Reynolds [5]. In his modeling simple swarm behaviors Reynolds defines three rules that must govern the behavior of swarms, flocks, herds, and schools in simulations:

1. Separation
2. Alignment
3. Cohesion

Reynolds' classification of local behavior achieved a simplistic and easily implemented model of these behaviors which has been the standard for further behavioral swarm designs. These simple behavioral rules are defined on the microscopic level. It is possible by adhering to these behaviors for a swarm to generate their own self-organizing behavior.

The expanding field of swarm intelligence in artificial systems, its inspiration from the natural world, and its implementations within technology is classified by Bonabeau et al. [4] with respect to insect behavior and how to apply models in the design of complex systems. This work focuses on research on swarm intelligence in artificial systems and has spawned an enormous amount of research in multiple disciplines. By drawing connections from the natural world the authors are able to explain the benefits of implementing swarm designs into non-biological systems to increase optimization and control. By identifying characteristics that have been implemented in artificial swarms, this work has influenced subsequent research in communication networks, division of labor, data analysis, and basic swarm attributes.

2.3 Self Organization

Self-Organizing systems are described as systems in which members organize themselves by an internal design or organization without any stimulus from a outside factor. Originally discussed through a multidisciplinary effort by Nicolis and Prigogine [22] in thermodynamics, Ashby [23] and von Foerster [24] in cybernetics, the proposed systems are able to maintain constant or decrease their own entropy, dissipating excess energy to the surroundings. Ashby further notes that a self-organizing system evolves to a state of equilibrium regardless of the starting conditions of the system. Camazine et al. [6] attempt to explain behavior in natural systems by identifying the four principles that govern self-organization. These principles, through the biological structures that develop through interactions among its organisms, are:

1. Positive Feedback
2. Negative Feedback
3. Amplification of Fluctuations
4. Multiple Interactions

This allows a system to become driven by its own components though agent-agent interactions and explains how a system of order is established based on local rules and interactions. To take advantage of features of swarm robotics, self-organization must occur. Properties typically ascribed to self-organization are: increase in order, autonomy, adaptability, robustness, and dynamics (Wolf and Holvoet, [25]). Self-organization as described above differs from the concept of emergence by the fact that a system may be self-organizing but could or could not display emergence behaviors.

2.4 Emergent Behavior

Emergent Behavior has proven to be difficult to define, both in engineering as well as other disciplines due to its vague fundamental philosophical concepts and the extensive possibilities which exist. The difficulty in creating a full and concise definition of emergence is a continuing para dime. Despite its importance and seemingly omnipresent creation the concept of emergence as Holland [26] describes it is a topic that more wondered at than examined. The definition of emergent behavior as pertaining to the presented research will adhere to a simplified version as presented by Hamann [9]. Hamann describes emergence with relation to the macro-micro problem in which a system has emergent properties if macro behaviors arise without being explicitly programmed on the micro-level. According to Dorigo et al. [27] and Dorigo and Şahin [28] this emergence is key to a swarm's effectiveness as the problems faced by a swarm are complex and not predefined. This is of importance because swarms are not an optimal solution to a simple problem, but rather to a complex and sometimes ill-defined problem. The concept of emergence, according to Beni [29], Şahin [30], and Bjerknes et al. [31], can also serve as an indicator that an algorithm is consistent to the principles of swarm robotics. As illustrated in the work of Bjerknes et al. where a swarm of relatively simple agents capable of only local sensing and communication are implemented. From their results the authors conclude that the resulting behavior is truly emergent from their algorithm.

2.5 Swarm Intelligence

Swarm intelligence is introduced and defined by Beni and Wang [32] as a section of artificial intelligence that focuses on the decentralized behavior of a swarm system that shows the properties of self-organization. Expanding on Beni and Wang's definition, Millonas [33] defines five basic properties of swarm intelligence systems:

- Proximity Principle: The agents execute simple computations concerning space and time.
- Quality Principle: The agents respond to quality factors, such as determining the safety of a location.
- Principle of Diverse Response: The agents distribute themselves and their resources in a variety of ways instead of concentrating on a narrow focus of behavior.
- Principle of Stability: The swarm is stable against repetitive fluctuations in the environment and does not oscillate.
- Principle of Adaptability: The swarm is sensitive to changes in the environment that require a change in the swarm behavior.

Millonas' work, corroborated by the works of Kennedy and Eberhart [34] and Bonabeau et al. [4] on defining these principles of swarm intelligence is also furthered by the commonly assumed statement that a decentralized collective behavior is based on the local information (Hamann [9]). Where this local information is limited to an agent's local neighborhood and global communication is typically disallowed. Based on the limitation in knowledge Hamann defines a sixth principle of swarm intelligence, the principle of local information. This principle relies on the theory of particle swarm optimization [34] [35], where the concept of a local best agent position based upon its position in its local neighborhood (LBEST) and a global best agent position based upon its position in the global swarm (GBEST) are introduced. These concepts are especially helpful for movement in the agent to achieve an LBEST position which in turn propagates an effect on the swarm as a whole where the agents' movements to an LBEST constitute a swarm movement to the GBEST swarm positions. In their optimization of a nonlinear function using particle swarm methodology where a swarm can be considered a nonlinear implementation, Kennedy and Eberhart are able to not only detail an optimized use of local neighbors, but give a very simple and inexpensive design for achieving a GBEST position, in terms of both memory requirements and speed. The use of the local neighborhood allows the use of a limited number of other agents within the swarm. This in turn leads to lower communication cost as well as lower calculation cost. The authors also introduce a nearest neighbor velocity matching scheme where velocities become dynamically adjusted according to their difference from their LBEST position. Through this velocity matching the swarm is able to achieve further optimization toward an optimal GBEST.

2.6 Taxonomy

Farinelli et al. [14] propose a classification of the previous research of the coordination in the field of swarm robotics. The authors propose the main aspect of the creation of a swarm robotic system is the coordination relationship between each robotic agent in the network. The classification of the various applications and behaviors to control coordination is imperative. By creating a taxonomy for swarm behavior the relationship of a system can classified according to its communication characteristics. This taxonomy has varying levels based on the agents relationships, as shown in Figure 2.1 where the focus on this research's swarm relationship has been highlighted.

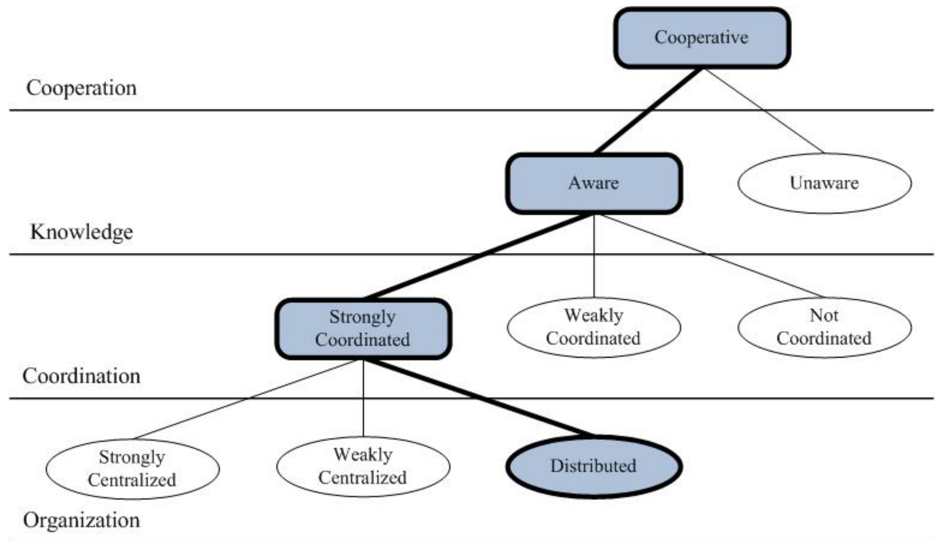

Figure 2.1: Multiple Robot Taxonomy

This classification focuses first on the level of cooperation between agents. A cooperative system is where agents share information to assist in the completion of a goal. Competitive systems focus on the acquisition of a goal by the single agent where agents compete against each other to accomplish the goal. A cooperative cooperation system can then be sorted into aware or unaware systems. Aware system agents have knowledge of other agents that constitute the swarm. Unaware system agents are unaware of being in a swarm and thus do not communicate. Functionality in an unaware system is accomplished through associating other agents as environmental abnormalities and performing actions with regard to them. Aware knowledge can then be broken into 3 degrees of coordination where each degree corresponds to how information is communicated. Strongly coordinated systems have predefined communication rules which are are followed by each agent. Weakly coordinated systems do not rely on these communication rules, instead are capable of of all combinations of communication rules. Not coordinated systems are aware of other agents but are not allowed to take into account their actions toward the goal. A strongly coordinated system can then be separated into the system's organization for how decisions are reached. A strongly centralized system contains a statically defined *leader* agent that decides all actions taken by the swarm. A weakly centralized system dynamically designates a *leader* based on each agent's relative swarm status where each agent has the the possibility to become the *leader*. In place of a central *leader* agent a distributed system allows each agent to decide its own actions.

The presented research centers about the choice of a distributed active communication topology in favor of a strongly coordinated active communication topology. It is hypothesized that an optimal distributed swarm communication scheme will be preferable for the purpose of detecting the *meta* event within the environment to that of a optimized strongly coordinated swarm communication scheme, concerning the metrics of speed, accuracy and swarm members. The decentralization of a swarm, and its comparison to a strongly centralized organization is also addressed in Trianni [36]. Trianni notes that a distributed system controller is a normal practice due to the infeasibility of a large scale centralized approach. Given the need to rely on communication to and from the *leader* the lack in flexibility and robustness in a centralized swarm is apparent. The feasibility of of a centrally coordinated system can only be seen in cases where the number of agents within the swarm is very small as in Matarić et al. [37]. Failure of the central controller in a centrally coordinated system would lead to a cascade failure of the entire swarm architecture in which the swarm topology evaporates [36]. In contrast to centrally organized systems Trianni praises the distributed organization with its impressive reduction in complexity of control systems, robustness to agent and information loss, and allowance of emergence behavior. Trianni does highlight the main problems of stagnation and deadlocks encountered through a distributed system which will be discussed on in chapter 3. Stagnation and deadlock situation arises from the fact that efforts of many individuals may reciprocally cancel each other. Avoidance of deadlock equilibrium is of paramount importance in the design of a swarm system.

2.7 Localization

A major issue with the member communication is the canonical problem of localization and tracking as addressed in Zhao et al. [38]. Given that localizing and tracking events and objects within the known environment is the focus of the swarm network, the critical problem defined by Zhao et al. is to dynamically define and form sensor groups based on task requirements and resource availability. The problem of tracking an object is also presented as the authors focus on discovering the object entering their sensory field, query processing (in which the processor turns member queries and data modification commands into a query plan - a sequence of operations to perform) the information towards regions of interest, collaboratively processing the target's location though the use of other neighboring members' information, and communicating the target object's movements to needed members.

Localization is also addressed by Grabowski et al. [19], where a swarm of centimeter-scale robots were able to collaborate for the purpose of mapping and exploring an unknown environment. The authors present a well founded solution for defining and overcoming the localization problem without the need to rely on GPS. Given the current limitations of GPS accuracy, a group of robots within close proximity would require an alternative means of localizing their position. Assuming a known position and orientation of all robots at time t_0 the question of localization would be to determine the position and orientation of all robots at time t_1. To accomplish localization the authors implement a maximum likelihood estimator based on previous positions and orientations through the use of dead reckoning and distance measurements. Grabowski et al. [19] state that dead reckoning can be expressed as the estimation of location by integrating encoder signals and using knowledge of "vector commands" (rotation in place over an angle, followed by a forward straight-line motion over a distance) with the likelihood that movement occurred over an angle and distance. Distance measurements, being an error correction tool for the drift typically seen in localization algorithms based solely on dead reckoning, require all robots to halt movement and "ping" so relative location can be calculated. By incorporating these localization techniques a conclusion is drawn that three robots were sufficient to achieve and maintain localization. With the addition of a fourth robot to the swarm a moving member could unambiguously localize itself with respect to each other member. The addition of a fifth member provided the robustness of a swarm, where the loss of one agent did not effect the ability of others to determine their own position.

2.8 Information Delegation

Issues in cooperative multi-robot target observation are discussed and addressed in the critical survey by Cao et al. [39]. Issues in group architecture, resource conflict, origin of cooperation, and the investigation of cooperative robotics used for observation are examined. Other issues that allow a robotic team to maximize object observation time by one or more agents are presented in Parker [40]. Attributes of target observation evaluated by these works contribute to the overall degree of information obtained by each agent. Action recognition of objects within an environment is also addressed in Parker [41]. This work includes an investigation of the extent to which agents of the swarm communicate their actions to other agents and the effect of such communications on cooperative team performance. Parker's investigation of the effects of team size and level of awareness to other agents shows the impact of action recognition and awareness on cooperative swarm designs.

Durring communication an agent must apply a certain amount of knowledge In the communication of the swarm system one agent must, as described by Fukuda et al. [42], apply a certain amount of knowledge for reasoning and then decide upon a reaction to the knowledge. The knowledge of task to be completed, the agent's state, and the environmental state when compared to other member's knowledge through information sharing allows a agent to predict the actions of other members. Knowing the approximate reaction of other agents allows an further degree of cooperation to occur. However errors do arise from uncertain and inconsistent knowledge. One example cites a cellular robotic system (CEBOT) which uses evidential information to describe the robot's environmental states in uncertain terms. Thereby allowing each member to create a map of their known environment through a measure called "evidence mass" where relative position can be calculated and expressed as interval probabilities. As each members evidence mass is combined with that of other agents a coordinated map of the environment is obtained.

2.9 Neural Decision Making

A focus of the current research has been in the area of neural network based control as referenced in Ham and Kostanic [43], Agah et al. [44], and Bishop [45]. The focus of this research is on multilayer supervised learning, backpropagation learning neural networks. Where each layer is comprised of multiple perceptions. A perceptron contains multiple weighted inputs which are summed together and used as an input vector to an activation function $f(x)$. This activation function normalizes the output which is then used as an input into another perceptron layer. Supervised Learning in a neural network with a backpropagation learning rule is carried out by presenting key patterns or exemplars to the network and, based on difference in output, the synaptic weights are changed. Each perceptron in the network performs this weight update during the training phase. Shown

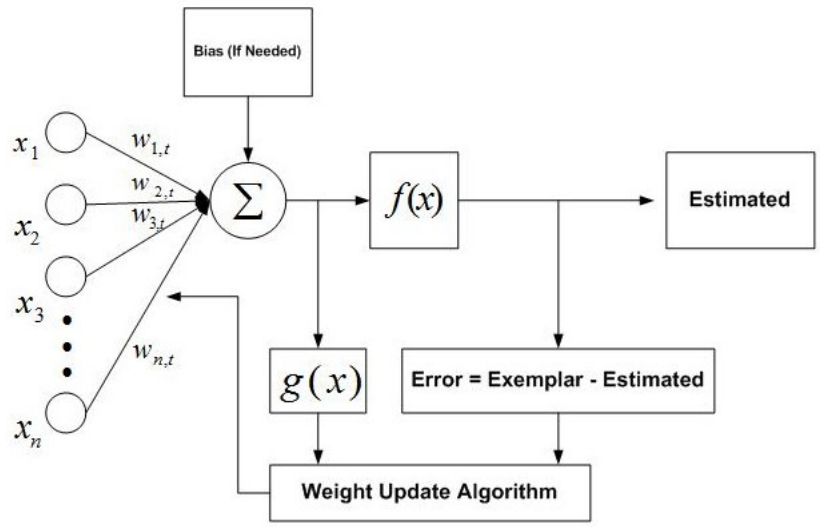

Figure 2.2: Perception with Backpropagation

in Figure 2.2 is the wight update process of a single perceptron.

Within this research a transfer function is limited to the sigmoid function, equation (2.1). During the training of the network the error δ in output can be derived through the use of the derivative of the transfer function , equation (2.2) in conjunction with the output difference to the exemplar, as in equation (2.3)

$$f(x) = \frac{1}{1-e^{-x}} \tag{2.1}$$

$$g(x) = \frac{e^{x}}{(1-e^{-x})^2} \tag{2.2}$$

$$\delta_j^{(s)} = \begin{cases} \left(d_{q,j} - x_{out,j}^{(s)}\right) g\left(v_j^{(s)}\right) & \text{if } s \text{ is the output layer} \\ \left(\sum_{h=1}^{n_{s+1}} \delta_j^{(s+1)} w_{h,j}^{(s+1)}\right) g(v_j^{(s)}) & \text{otherwise} \end{cases} \tag{2.3}$$

where $d_{q,j}$ is the desired output, $x_{out,j}^{(s)}$, $g(\cdot)$, and $w_{h,j}^{(s+1)}$ are defined in Figure 2.2.

Backpropagation is achieved through the update of layer weights through equation (2.4). The weights are updated beginning with the last layer of the network and incremented to the first layer. By incrementally updating the layer weights based on these corrections, a neural network changes its layer weights to identify patterns used in training.

$$w_{j,i}^{(s)}(k+1) = w_{j,i}^{(s)}(k) + \mu^{(s)}\delta_j^{(s)}x_{out,i}^{(s-1)} \tag{2.4}$$

Using this weight update it is possible to train a neural network to associate partial or full information and to determine an acceptable output pattern(Ham and Kostanic [43]). When used in a feed-forward mode, the trained neural network design results in an effective decision making tool. A full diagram of neural network feed-forward mode can be referenced in Appendix A. Applying a neural network to swarm robotics is not a new concept, as seen in Muñoz et al. [46], where a neural network is applied to Khepera Robots to capably explore multiple environments.

2.10 Tracking

Once an event has been detected the problem of tracking that event through the environment then becomes an issue. Zhao et al. [38] approach the problem through the use of a Bayesian filter which computes new beliefs based on previous information. Since the likelihood of an event at a position and the previous positions are non-Gaussian a grid-based non-parametric representation for probability distribution may be used. Through successive "hand-offs" from member to member and manageable communication levels a moving-target scenario *tracking* can be achieved. A challenge in tracking is to track multiple objects. This increases the difficulty exponentially. Challenges also arise from the addition of dimensionality. Here the presence and interaction of multiple objectives which cause the dimension of the under lying state spaces to increase. Requiring the mapping of distributed sensors and the state-space model for estimation within the algorithm.

2.11 Heterogeneity

Heterogeneity can dramatically increase the optimization of a swarm system. The process to segregate other agents in a heterogeneous system must be available so that members within the swarm can correctly identify other agents' capabilities. One approach to segregation as described by Kumar et al. [47] is dependent upon the magnitudes of potential during the interrogation process between agents where agents suited for certain tasks would have more potential than others. RoboCup has been a vivid development bed for swarm research, as various teams compete to develop soccer playing robot teams which compete against each other Farinelli et al. [14]. In this case a team of physically different agents used dynamically assigned roles according to the physical capabilities of each agent to achieve significant performance improvements over homogeneous teams.

CHAPTER 3: DESIGN METHODOLOGY

Since the main objective of the research presented is the quantification of a *meta* event while optimizing the swarm's field coverage and energy consumption, the challenges for developing such an algorithm must be clear. The question becomes not just how to detect a *meta* event in an environment, but how to best allocate the swarm's resources to this event without unnecessary cost to the swarm in both energy used and environment covered. From the definition of swarm robotics we are allowed the use of a number of spacial agents which can be deployed within this environment. Since we are able to implement multiple points of view, the quantification of a *meta* event becomes possible. Where a single point of view system would be unable to quantify the large *meta* event due to its limited field of view, this distributed field of view allows more, and ideally all, of the *meta* event to be recognized at one time. The *body* of the swarm system also becomes a great advantage in the initial detection and coverage of this large environmental area. To optimize the coverage of the environment an equal distance from other nearby agents must be achieved, thus we also identify an evenly covered area as a point of merit. From these abilities and objectives the desired result from a swarm implementation to this problem would be:

- Detecting the *meta* event quickly
- Quantifying or *bounding* the *meta* event
- Covering the environment efficiently and accurately

3.1 Abstract Algorithm Design

In a classical systems engineering approach a formal method of stepwise program refinement is applied to transform an abstract specification into an executable program (Hamann [9]). This approach becomes impracticable when applied to the realm of swarm robotics. There is no globally defined algorithm solving the task step-by-step. Since a swarm is a non-linear approach to a problem in which programming is restricted to only the single agent to allow for the emergence of behaviors, the stepwise refinement of the algorithm can not be completed as the emergence can not be formally derived due to a lack of a constructive step. As the swarm arises from local agent to agent interactions, predicting the resulting emergence from these interactions is a difficult, if not an impossible, step to take before the swarm is deployed. To circumvent this development challenge the programming "by hand" method was applied, a direct approach which does not rely on prediction tools. Programming by hand is considered an iterative approach to the problem, where an initial simple design concept is programmed on the agents and the resultant emergence is witnessed. From this outcome emergence, desired or not, the algorithm can be adjusted and further refined. Large abstract concepts are first applied until an unrefined desired resultant is achieved. From this unrefined algorithm lower abstraction concepts can be further refined until a final algorithm with the desired emergence has been developed.

3.2 Simulation Design

As a development environment was needed for the refinement of the control algorithm, a simulation was designed to not only test possible behavioral characteristics, but their emergence from interactions. The main factors attributing to the *meta* event detection in an applicable situation are narrowed to the following components:

- Field with environmental boundaries
- Agents with embodied characteristics
- *Meta* event with definable boundaries within the environmental limits

To simplify the numerical algorithm to be developed, the movement allowed on the field by the agents was discretized, which is a practice defined by Schner [48] and implemented in other research (Hamann [9]) (Jian et al. [1]). In this discretization time is handled in steps and the field space is quantized into a Cartesian grid. This restricts the movements of agents to a uniform distance of one step in either the X or Y direction. This also allows the acceleration term seen in physical movements to be disregarded for the purpose of the simulation. This discretion of movement also eases the transition through the iterative approach to witness emergence. For convenience of design, environmental boundaries are considered to be the boundaries of the varied rectangular environment. These boundaries within the simulation allowed for a clamped environment in which the emergence of behaviors could be witnessed. Bounding the agents in an asymmetrical field reduced the chance that the witnessed emergence was due to the shape of the environment. In each simulation a number of agents (N-AGENTS) were placed on this bounded field. Since the objective of a system employing this *meta* event detection approach would likely begin in a cohesion or "clumped" position, as would be the case in deployment from a central container or home base, the agents begin the simulations as tightly clumped as possible, as seen in Figure 3.1.

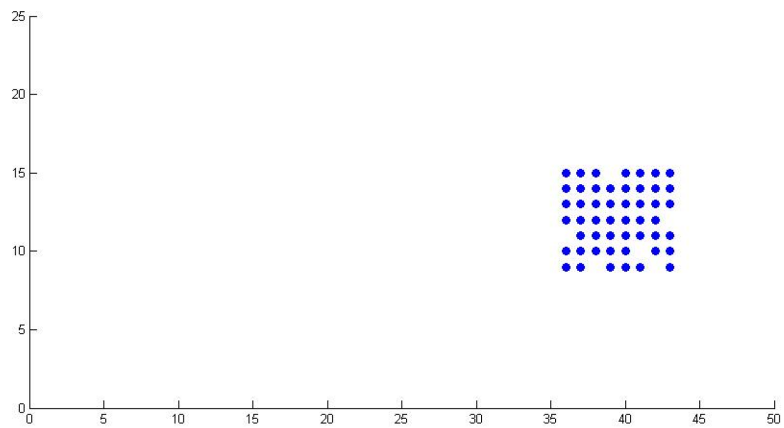

Figure 3.1: Swarm Starting Position on Field

From this clumped starting position the effects of the agent behavior's to spread, cover, and find the *meta* event can be witnessed. Being that the field and agents are now defined in the simulation, a *meta* event is also added, as seen in Figure 3.2.

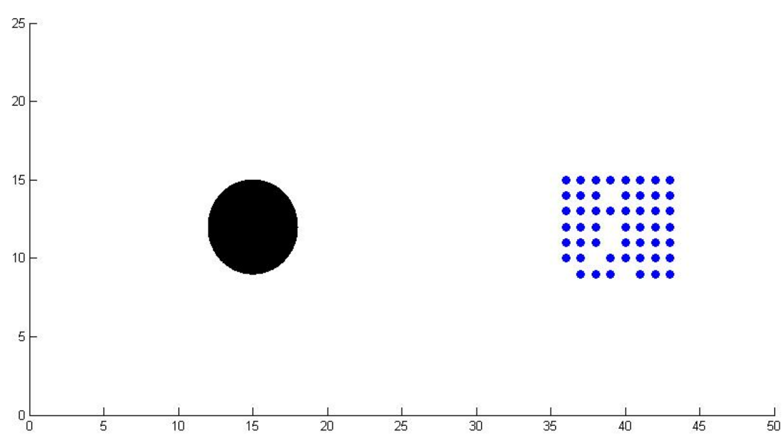

Figure 3.2: Swarm with *Meta* Event on Field

This simulation framework gives the capability of programming behaviors on the swarm agents while allowing the observation of the agent interactions. The design of the simulation was meant to encompass future variances and adaptation which will be discussed in chapter 5.

3.3 Agent Perspective

From the outset the purpose of this research was the optimization of the swarm topology. To this end the assertion of implementing a distributed topology was made due to the reduction in communication cost (Farinelli el al. [14]) and robustness when compared to other methods (Trianni [36]). A distributed organization allows the presence of self-organization as defined by Camazine et al. [6] in the swarm system, particularly the parallel search of possibilities to optimize the encirclement of the *meta* event. Results from both distributed and strongly centralized organization through simulations are presented in chapter 4. To enable emergence from this distributed organization, the agent to agent interactions must be investigated. Therefore, a break is made where the focus turns from the global swarm perspective to the local agent level perspective according to the *local to global* problem (Yamins [15]). The local agent has its own frame of reference to other agents and environment, its local neighborhood. Any communication done by the agent is limited to this local neighborhood, where ability to communicate is defined as being within the agent's communication range and where interaction with the environment is limited to a specific sensor payload. In this simulation each agent was limited to four nearest neighbor agents. This limitation in agent neighbors allowed the necessary attributes of swarm behavior of triangulation, collision avoidance, and accurate information sharing, while keeping network traffic and computational cost to a minimum. By utilizing their distinct local neighborhood a local best position can be achieved for each agent, where the movement toward a local best results in the swarm movement toward a global best (Kennedy and Eberhart [34]).

3.4 Behavior Through Neural Decision

Each agent in the swarm can be thought of as an individual within a population. The emergence that is required from the swarm to detect the *meta* event comes from these interactions between these individuals. The difficulty arises from having the agent express the optimal behavior to influence these interactions when the local neighborhood has so much variation. Because artificial neural networks are effective when inferring and generalizing based on the slight changes in stimulation, it was decided that each agent would incorporate a neural network for decision making. The neural network approach also achieves a scalability factor for the use of multiple stimuli, both local and communicated, which allows for a multitude of sensor payloads. An array of sensor inputs comprised of the agent (A) and its local neighborhood (S) are applied to the neural network. Where the neural network input array is

$$\Big\{ \{A_i(t), A_i(t-1), \cdots, A_i(t-c)\},$$

$$\{S_i^{(0)}(t), S_i^{(0)}(t-1), \cdots, S_i^{(0)}(t-c), \cdots, S_i^{(k)}(t), S_i^{(k)}(t-1), \cdots, S_i^{(k)}(t-c)\} \Big\}$$

where i is the current agent, k is the target neighbor, and c is the size of data cache.

Based on a training set of local scenarios and corresponding desired behaviors $(b_0, b_1, \cdots, b_n)$ a three layer neural network was trained. The size of the first hidden layer was chosen to be $2n+1$, where n is the size of the input layer. This allows more interconnectivity between neurons, which increases the accuracy of the neural network. Following the processing of data through the layers of the neural network, an output is produced that maps the behaviors the agent can perform. The greatest resultant of this output value is then used by each agent to determine an appropriate behavior. In this case the agent performs the behavior of the highest output from a feed-forward neural network with a multiple perspective input. The same trained network was applied to each agent, giving each of them the ability to decide their own course of action.

3.5 Local Neighborhood Force

For the agent to decide the best direction to take, a process that makes use of the local neighborhood was decided upon. Since the distance and location from each other member in the agent's local neighborhood is known, the local best position can be calculated at the current time. Figure 3.3 shows an example of the position relationship between agent *i* and on of its neighbors, agent *k*.

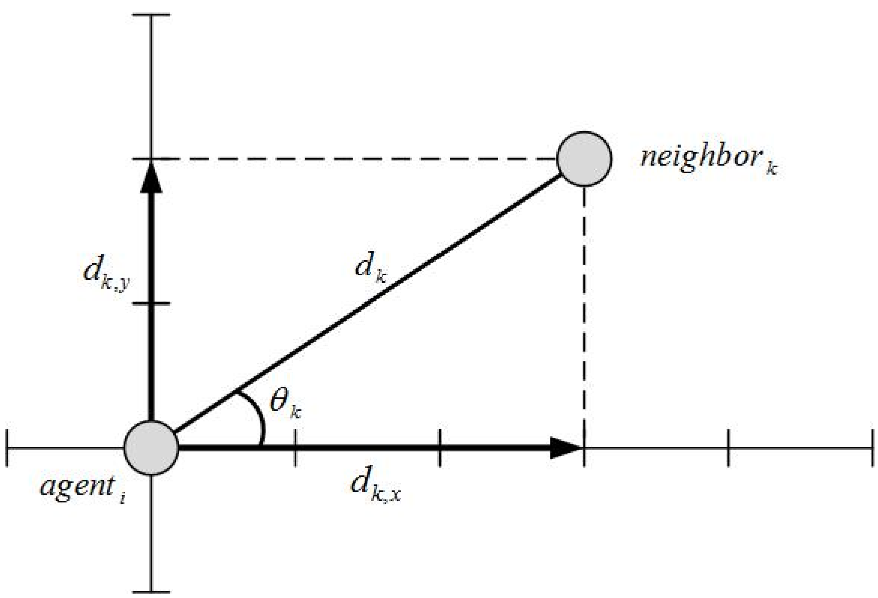

Figure 3.3: Agent and neighbor position relationship

To calculate the local best vector ($\overrightarrow{v}_{(i,desired)}$) for an agent's movement, the forces of each agent are calculated with respect to distance (d), angle (θ), and feedback polarity to each neighbor based upon equation (3.1)

$$\begin{aligned}
\overrightarrow{v}_{(i,desired)} &= -\sum_{k=0}^{N_{1,R}} \frac{1}{\overrightarrow{v}_k} + \sum_{k=0}^{N_{1,A}} \frac{1}{\overrightarrow{v}_k} \\
\overrightarrow{v}_k &= d_k e^{j\theta_k} \\
i &= 1, 2, \cdots, N \\
N_1 &\subset \{1, \cdots, c\}
\end{aligned} \tag{3.1}$$

where	N	- number of agents in swarm,
	N_1	- number of agents in neighborhood of the i^{th} agent,
	$N_{1,R}$	- number of neighbors emitting negative feedback to the i^{th} agent,
	$N_{1,A}$	- number of neighbors emitting positive feedback to the i^{th} agent,
	d_k	- distance to k^{th} neighbor,
	θ_k	- angle to the k^{th} neighbor,
	c	- size of data cache.

Based upon the aforementioned discretization of movements, equation (3.1) is discretized to produce (3.2). The forces acting upon an agent in the X direction is calculated by equation (3.3) and in the Y direction by equation (3.4), respectively.

$$\overrightarrow{v}_{(i,desired)} = \left(F_{(i,x)}, F_{(i,y)}\right) \tag{3.2}$$

$$F_{(i,x)} = -\sum_{k=0}^{N_{1,R}} \frac{cos\theta_k}{d_k} + \sum_{k=0}^{N_{1,A}} \frac{cos\theta_k}{d_k} \tag{3.3}$$

$$F_{(i,y)} = -\sum_{k=0}^{N_{1,R}} \frac{sin\theta_k}{d_k} + \sum_{k=0}^{N_{1,A}} \frac{sin\theta_k}{d_k} \tag{3.4}$$

The calculations of forces acting upon the agent are thus the inverse of the distance vectors to the members of its local neighborhood. By applying simple vector math to these forces a direction is chosen which would lead to a local best position relative to the agent's neighbors. An example of an agent repulsing away from its neighbors to optimize space covered is shown in Figure 3.4.

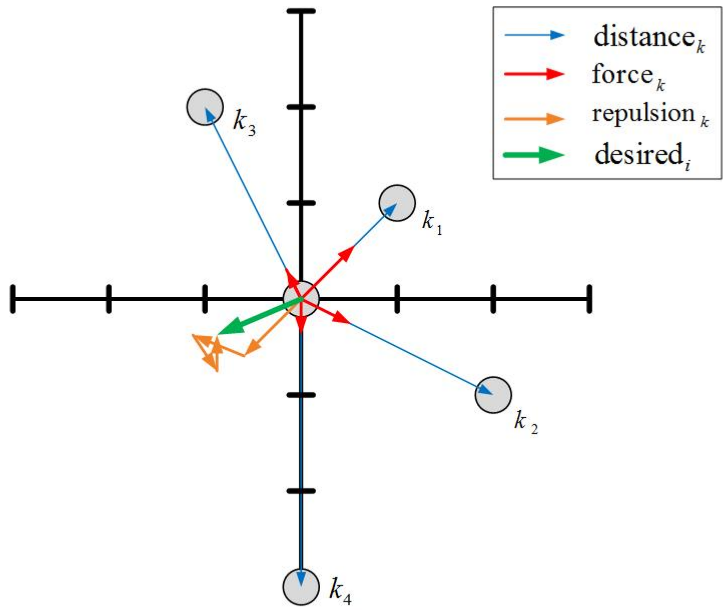

Figure 3.4: Best Heading from Local Forces

3.6 Design of Emergence

Applying the use of force calculations based on local neighborhood feedback leads the swarm to create emergent behaviors. With the objective of the swarm being the optimal detection and recognition of a *meta* event, a number of behaviors for the agents to emulate were developed based upon scenarios seen in Table 3.1.

Initial Behaviors		
Description	Action	Anticipated
Meta event present	Keep standoff from event, send negative feedback	Keep other agents from event for optimal resource allocation
Event possible	Keep standoff from event, send positive feedback	Gather more agents to quantify event
Neighbor event detection	Go to event detection, send positive feedback	Achieve more agents at event
Propagate event information	Repulse from neighbors, send positive feedback	Send more agents to event detection while avoiding overcrowding
Disperse from cluster	Repulse from neighbors, send negative feedback	Spread swarm over environment while avoiding clustering
Environmental Boarder	Keep standoff from border, send negative feedback	Keep agents from environmental obstacles

Table 3.1: Initial Agent Behaviors

Since there was no way to predict the emergence created by the behaviors and agent to agent interactions, these initial agent behaviors became the abstract starting point where refinement would start. After training the behavioral neural network it was witnessed that "*Meta* event present" behavior was an unreachable state. This was due to the limitations of the local neighborhood system and the behavior was thus discarded as a main behavior. With the remaining behaviors intact the simulation was conducted in a limited fashion with no event present in order to witness the emergence and effectiveness of the dispersal behaviors, shown in Figure 3.5.

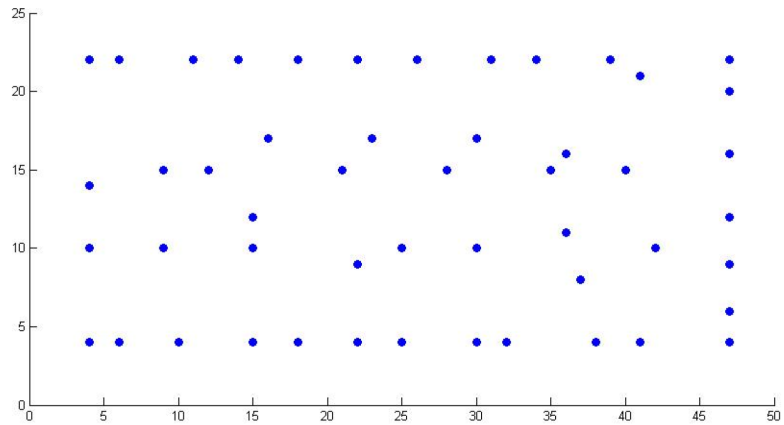

Figure 3.5: Initial Dispersion Behavior Emergence

Visually from Figure 3.5 the inclusion of an environmental behavior is an unnecessary and inefficient use of the swarms capabilities. The agents began to cluster on the environmental boundaries and are unable, based upon the local neighborhood forces, to exit the behavior. This emergence also led to the inclusion of the environment to the forces calculated in the local neighborhood, where equation (3.2) would now become updated to include the environmental force (cf) as in equation (3.5)

$$\overrightarrow{v}_{(i,desired)} = \left(F_{(i,x)} + x_{(i,cf)}, F_{(i,y)} + y_{(i,cf)}\right) \tag{3.5}$$

where cf is environment dispersal force.

Propagating communication of event detection over multiple local neighborhood systems also produced detrimental effects, as seen in Figure 3.6. Given that the distributed system in place does not allow for the communication of initial event detection, agents were drawn to non-event agents.

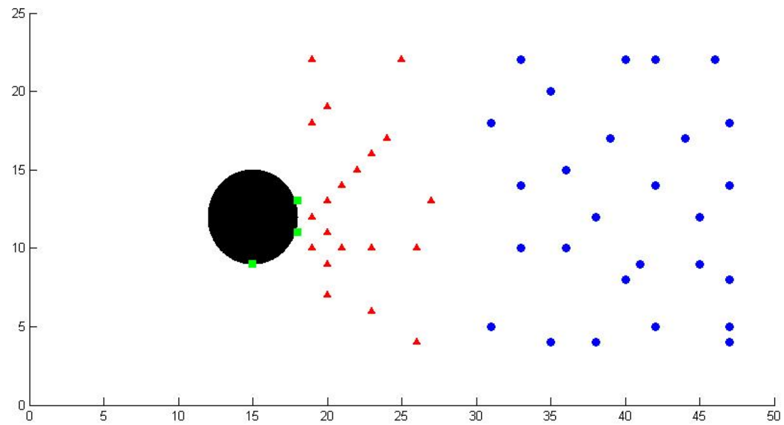

Figure 3.6: Event Barycenter Attraction Behavior Emergence

Unintended clustering based on the propagation of information leads to the assumption that, over a simple coordination scheme like the local neighborhoods, swarm direction to the event can not be influenced in this method.

3.7 Behaviors

Taking the iterative method of rejection and acceptance based on the emergent swarm behavior, the simple agent behaviors were narrowed and adjusted. In keeping with the guiding principles set by Reynolds [5], the evolution of the behaviors were kept as simple as possible to reduce the complexity needed for each agent. From this effort three simple behaviors were achieved as shown in Table 3.2

Resultant Behaviors		
Description	Action	Anticipated
Event Behavior	Keep standoff from event. Send positive feedback. Send negative feedback if *Meta* status achieved.	Draw other agents to event. Once enough information collected, send agents away to cover more effectively
Avoidance Behavior	Send negative feedback. Repulse from negative feedback. Attract to positive feedback.	Gather more agents to quantify event while keeping coverage and resources optimized.
Dispersion Behavior	Send negative feedback. Repulse from all feedback.	Quick and normalized movement from starting cluster.

Table 3.2: Resultant Agent Behaviors

Once the final agent behaviors were decided upon, a three layer perceptron neural network with supervised learning using backpropagation was trained with scenarios and corresponding desired behaviors. This neural network was then applied to all agents. In the feed-forward mode the agents consult the resultant of their neural network and proper behavior is followed based upon the greatest return value. For example, in Table 3.7 the greatest return value is the Dispersion behavior so the agent performs the actions associated with the Dispersion behavior.

Output	**Behavior**
0.0425	Event
0.2660	Avoidance
0.7203	Dispersion

Table 3.3: Choice of Behavior through Neural Network Output Example

For visual reference, agents emulating a specific behavior can be identified and separated through Figure 3.7.

Figure 3.7: Agent Behavior Color and Shape Reference

At the point of swarm initialization ($t = 0$), before any behavioral decisions are calculated, each agent follows the procedure outline in Figure 3.8.

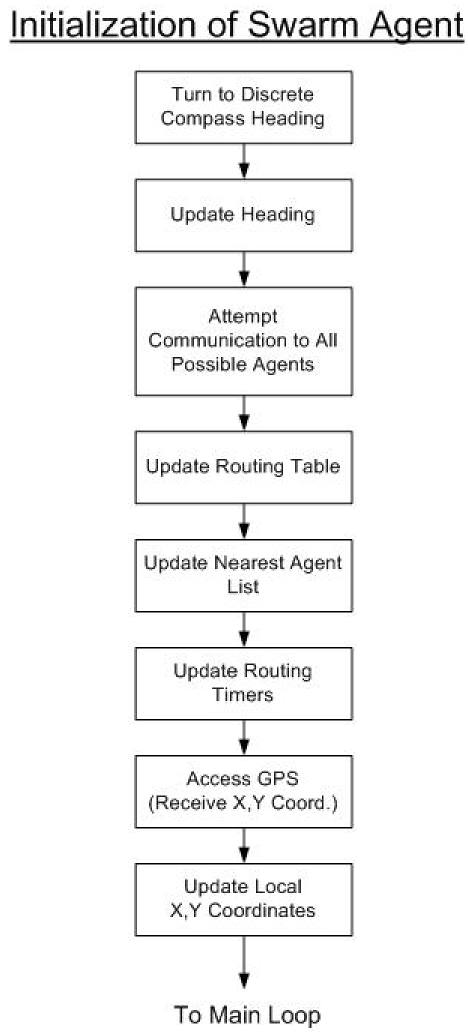

Figure 3.8: Initialization Procedure

Once initialized each agent enters a main loop illustrated in Figure 3.9. This structure allows each agent to decide proper behavior, perform basic maintenance, and dictates communication protocol between the local neighborhood, Figure 3.10.

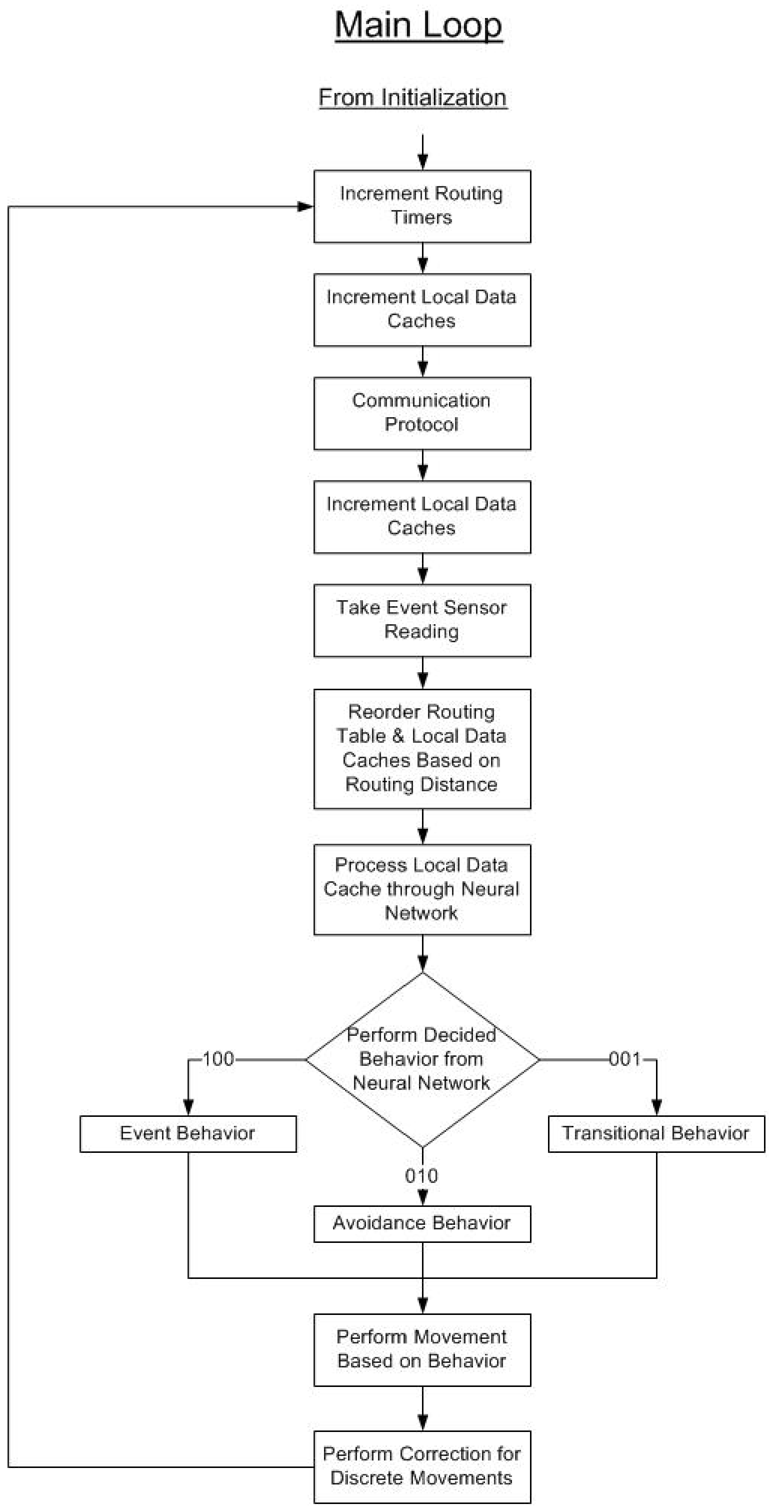

Figure 3.9: Main Procedure

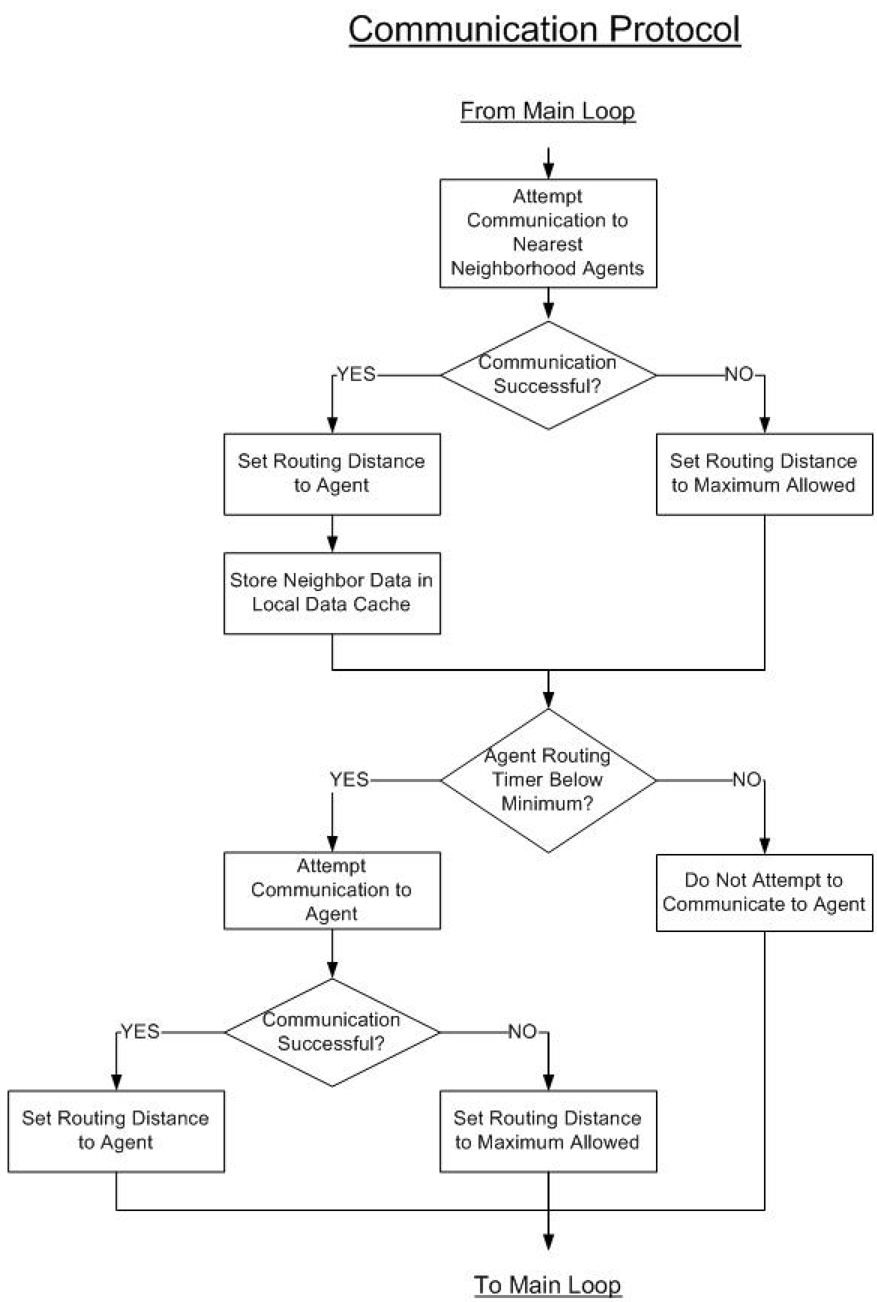

Figure 3.10: Communication Procedure

3.7.1 Dispersion Behavior

Considering the initial goal of finding the *meta* event as fast as possible from a clustered beginning swarm state, a dispersion behavior became mandatory. Based entirely from negative feedback to and from its local neighborhood, an agent emulating the dispersion behavior will achieve a local best position by repelling its neighbors. Illustrated in Figure 3.11 a Dispersion agent will receive negative feedback from any agent within the repulsion range D_r until the target neighbor is not within repulsion range or has been replaced in the local neighborhood.

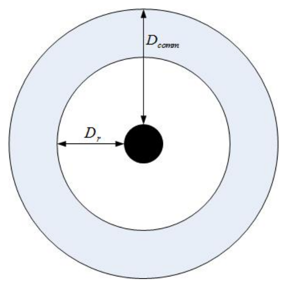

Figure 3.11: Dispersion Distance with Communication

By pushing itself away from others, Dispersion agents are capable of making the swarm cover a distance in a normalized and speedy fashion. Illustrated in Figure 3.12 is a swarm that has achieved full dispersion within a boundless environment where communication and local neighborhoods are kept intact. Dispersal in this fashion also contributes to the field coverage seen in Figure 3.12 where coverage is shown through a region of influence surrounding each agent. This formation of agents has reached a steady-state that requires no movement and extremely sparse network traffic as a whole.

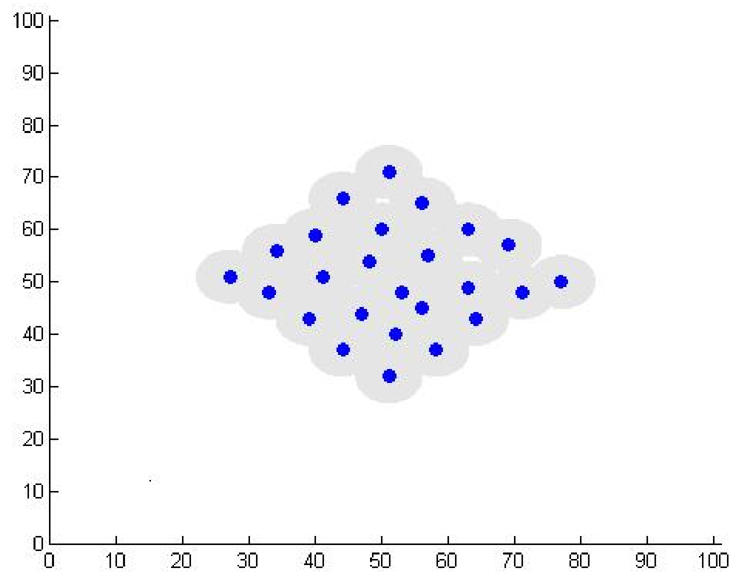

Figure 3.12: Dispersion Behavior Within Unbounded Environment

Agents that decide through their neural network to behave as Dispersion follow the procedures mentioned above, as outlined in Figure 3.13.

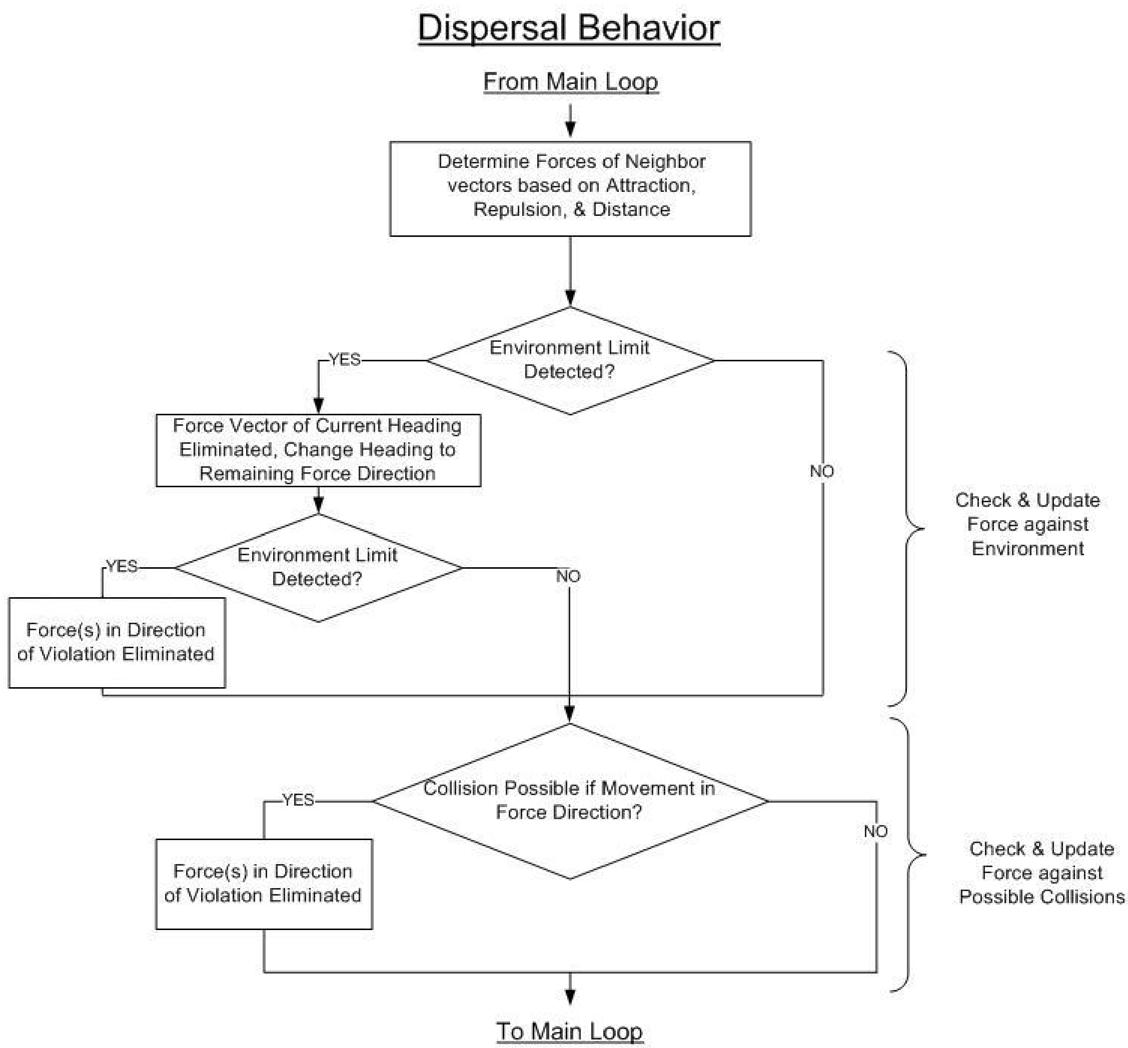

Figure 3.13: Dispersion Behavior Procedure

3.7.2 Event Behavior

As stated before, a single agent system is incapable of quantifying a *meta* event. Hence a behavior to attract other agents to a detected *meta* event is necessary. By broadcasting positive feedback to its local neighborhood, Event agents are able to influence their neighborhood to move toward its detected *meta* event. Instead of relying on its neighborhood's possible negative feedback, an Event agent attempts to achieve a set standoff distance to the detected *meta* event. In attempting to achieve this ideal orbit from the *meta* event an Event agent must adhere to a minimal distance to other agents D_{min}, illustrated in Figure 3.14.

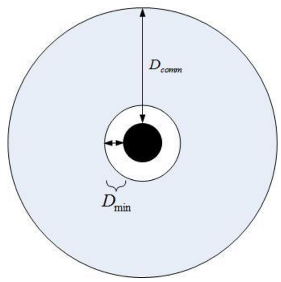

Figure 3.14: Minimum Distance with Communication

By keeping this minimum distance and standoff requirements, Event agents are able to successfully attract other members of the swarm and keep a portion of the *meta* event bounded. Agents that decide through their neural network to behave as Event agents follow the procedures mentioned above, as outlined in Figure 3.15.

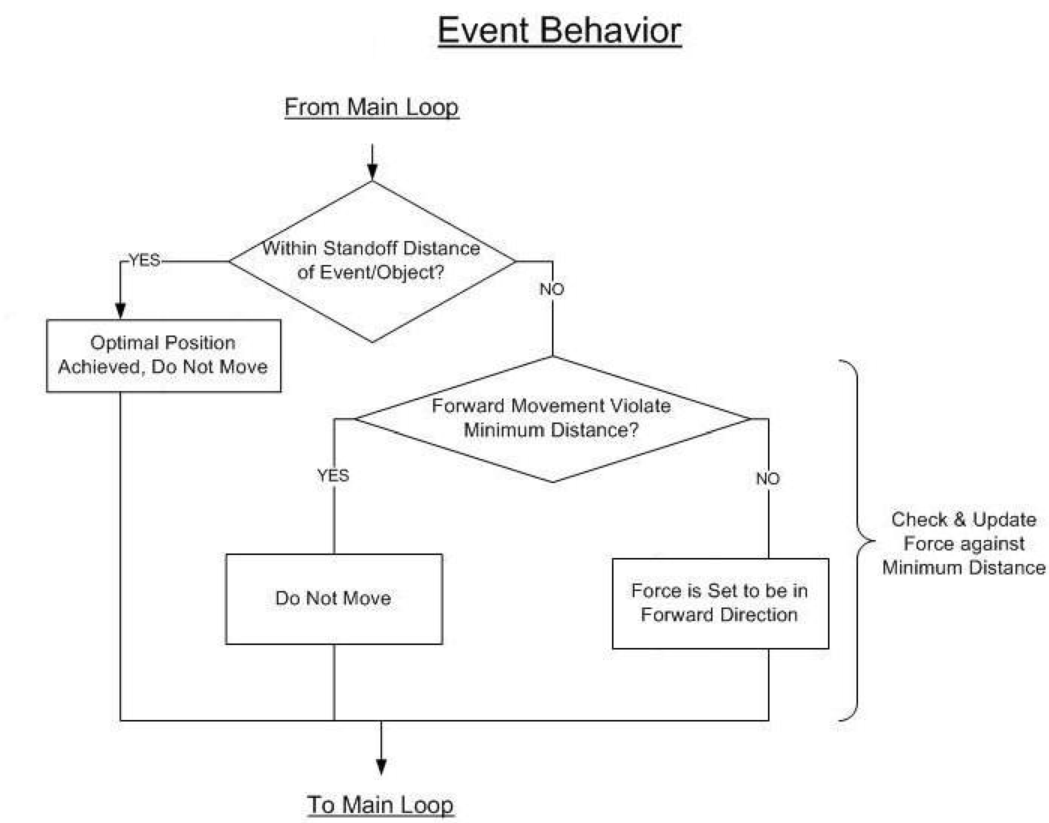

Figure 3.15: Event Behavior Procedure

Beyond the Event behavior is a sub-behavior, a *Meta* behavior of the Event state, which is bound by the rules of D_{min} just as an Event agent would be. Once an Event agent establishes another Event or *Meta* agent within its local neighborhood, an agent identified token is passed. When an Event or *Meta* agent receives a token, it has the choice of either passing it to another acceptable agent or holding it until a suitable destination can be discovered. An acceptable agent to receive a token must be an Event or *Meta* agent that has not possessed this token for a number of hops, to prevent unintended errors in reciprocal token passing over abnormalities in the event boundary. Such errors result in the false transfer of boundary information. If an Event agent receives its identified token, the agent saves the number of passes of the token and stops sending positive feedback to the swarm, instead sending negative feedback. This change in the Event behavior is identified as the transition to *Meta* behavior. Token passing does not cease when *Meta* status is achieved, but is limited. Instead of initializing a token at the first opportunity, a delay is added to the time each agent reinitializes their identified packet into the system. A *Meta* agent remains in the *Meta* behavior until, within the saved increment for passing the token through the network, the token fails to return. This ability to change feedback allows the swarm to more efficiently allocate resources toward and away from the *meta* event.

3.7.3 Avoidance Behavior

A bridge between the Dispersion and Event behaviors was also required; one that could not only advance toward the agents communicating the event detection, but also keep the resources of the swarm optimized between event bounding and area coverage. Classified as Avoidance, this behavior relies both on the positive feedback from Event agents as well as the negative feedback from Dispersion agents while following the rules of D_{min} and D_r. Initially another rudimentary behavior, Avoid agents produced a greater swarm presence at the *meta* event, but did not allow the swarm to encircle it. Being drawn to the Event agents caused a stall of swarm movement around the *meta* event when contact was made, shown in Figure 3.16.

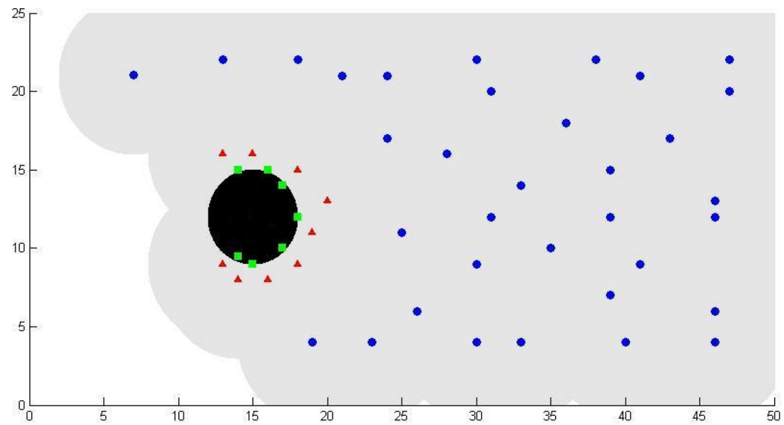

Figure 3.16: Simulation at Steady-State, No Momentum

To circumvent this problem in functionality, a factor of neighborhood velocity (V) originally introduced in Kennedy and Eberhart [35] was adapted for use with the local communication system. Where the local best position calculated in equation (3.5) is implemented within equation (3.6) to incorporate the neighborhood momentum agitation, equation (3.7)

$$\overrightarrow{v}_{(i,desired)_{t+1}} = \overrightarrow{v}_{(i,desired)_t} + V \tag{3.6}$$

$$V = c_2 \sum_{k=1,\neq i}^{n} V_k - c_1 V_i \tag{3.7}$$

where c_1 and c_2 are user defined constants.

The trajectory of an Avoid agent as it clusters to Event agent(s) can then be altered to further encircle the *meta* event. Illustrated in Figure 3.17, the Avoid agent's perception of forces is skewed by the momentum of the Dispersion agents, who are attempting to achieve their local best position through repulsion which in turn draws the Avoid agent to the *meta* event and continues the chain reaction of bounding the *meta* event. An added positive emergent behavior from the addition of momentum is the perturbation of stagnant Avoid agents by the rest of the non steady-state swarm. This allows gaps in bounding to be filled.

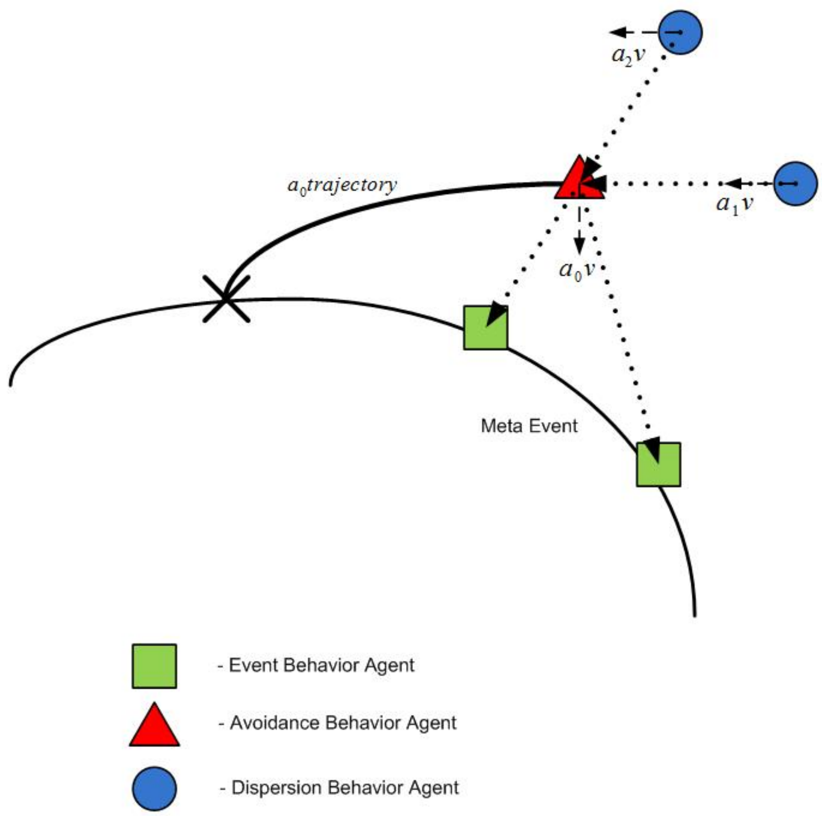

Figure 3.17: Trajectory Altered by Neighborhood Momentum

Agents that decide through their neural network to behave as Dispersion follow the procedures mentioned above, as outlined in Figure 3.18.

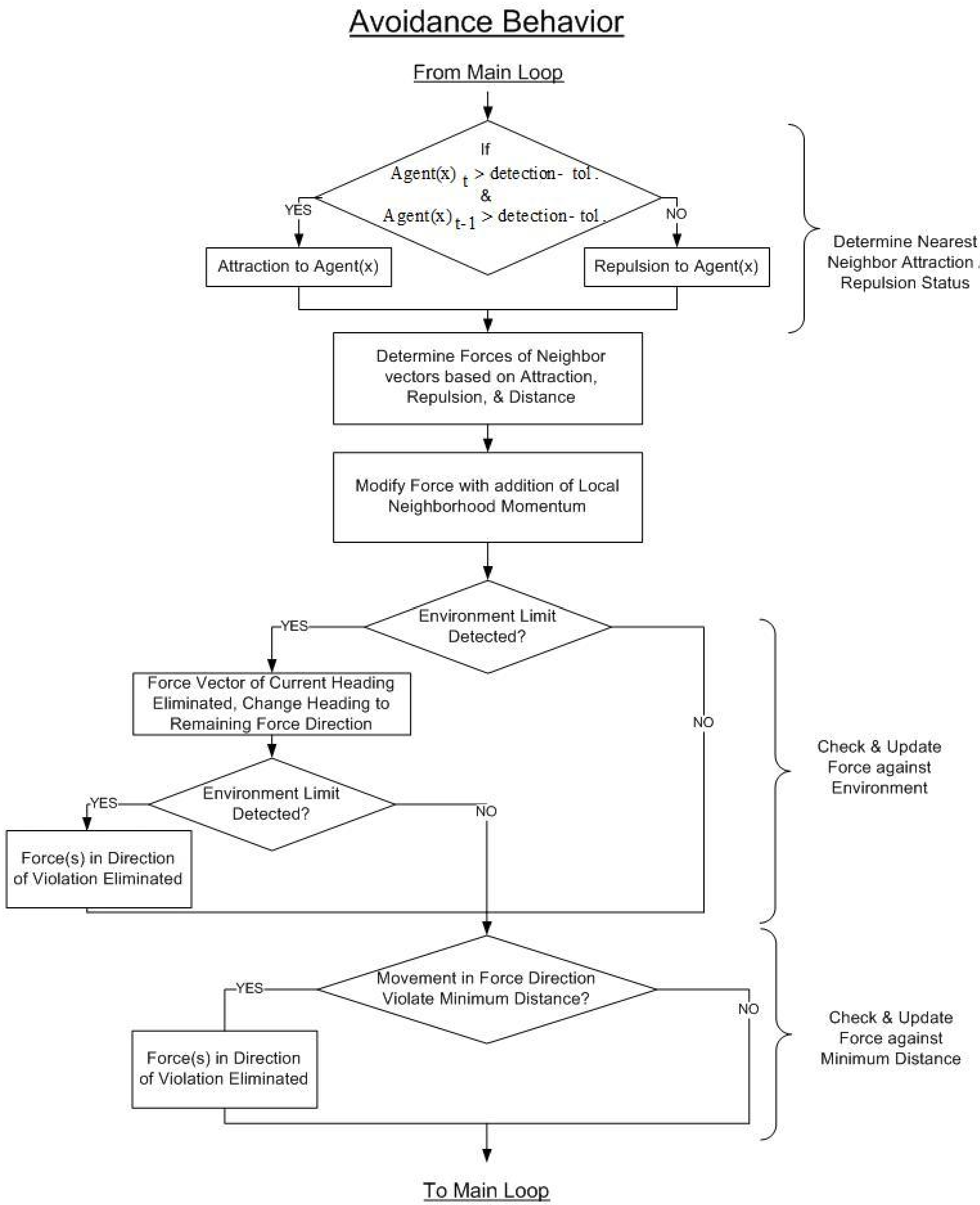

Figure 3.18: Avoidance Behavior Procedure

CHAPTER 4: RESULTS

4.1 Global Simulation

Following the parameters set in Table 4.1, results from a single simulation until full coverage and *meta* event bounding is reached are presented in Figures 4.1 - 4.7. This simulation agents are limited to detecting the *meta* event when their Cartesian coordinates intersect the *meta* event circumference. In the physical domain this simulation would represent a swarm of agents capable of two dimensional movement finding a two dimensional and unmoving *meta* event. The *meta* event does not restrict the agent's movement on or across it and is detected through the use of an inboard sensor that yields a yes or no detection rate.

Simulation Parameters	
Variable	**Value**
Number of Agents	50
Field X Length	50
Field Y Length	25
Meta Event Shape	Sphere
Meta Event Area (πr^2)	$(\pi \cdot 11^2)$
Meta Event Center Coordinates	$(15, 12)$
Swarm Cluster shape	Rectangle
Swarm Cluster Area $(X \cdot Y)$	$(8 \cdot 7)$
Swarm Cluster Center Coordinates	$(40, 12)$
Communication Range (D_{comm})	8
Dispersion Range (D_r)	5
Minimum Range (D_{min})	2
Environment Standoff	2
Meta Event Standoff	0
Sensor Range	1
Data Cache Size (c)	4
Neighborhood Momentum (c_2)	80%
Agent Momentum (c_1)	20%

Table 4.1: Simulation Parameters

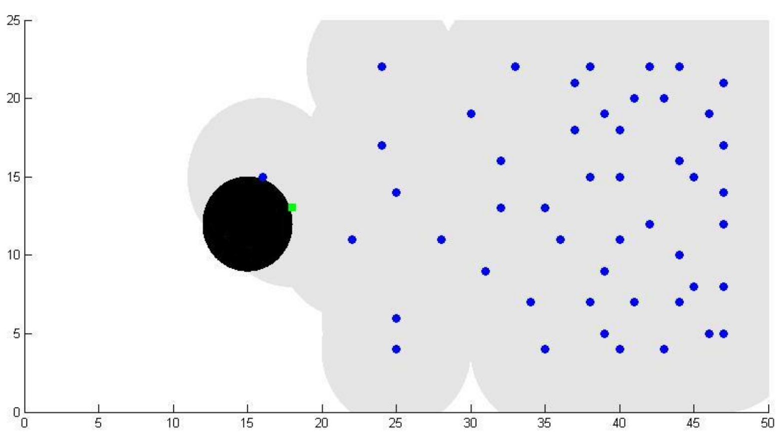

Figure 4.1: Global Simulation $(t = 10)$

At $t = 10$ in Figure 4.1, all swarm members have begun to disperse away from the starting cluster for further area coverage. A single agent has decided to change behavior to that of Event due to contact to the *meta* event though its sensor input.

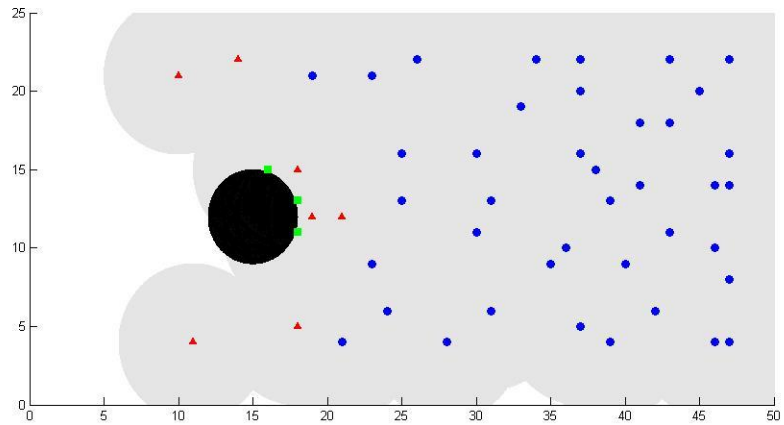

Figure 4.2: Global Simulation ($t = 20$)

At $t = 20$ in Figure 4.2, the swarm is continuing to disperse over the field area. There are now more Event agents covering the *meta* event and in turn affecting their neighbors to become Avoid agents. Momentum from the local neighborhoods of the outlying Avoid agents are keeping the cluster emergence from the first detection of the *meta* event, keeping the swarm from halting after contact is made.

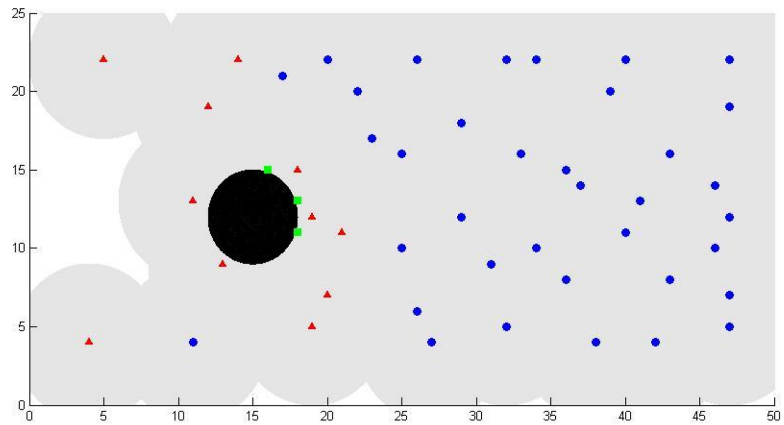

Figure 4.3: Global Simulation ($t = 30$)

At $t = 30$ in Figure 4.3, the swarm agents have begun to encircle the *meta* event due to the positive feedback from the Event agents and the momentum affecting the Avoid agents. Effects of the neural decision process can be seen as the outlying agents that have moved beyond Event agent communication are still trying to achieve a local best that would bring them to the *meta* event, based on past communicated information.

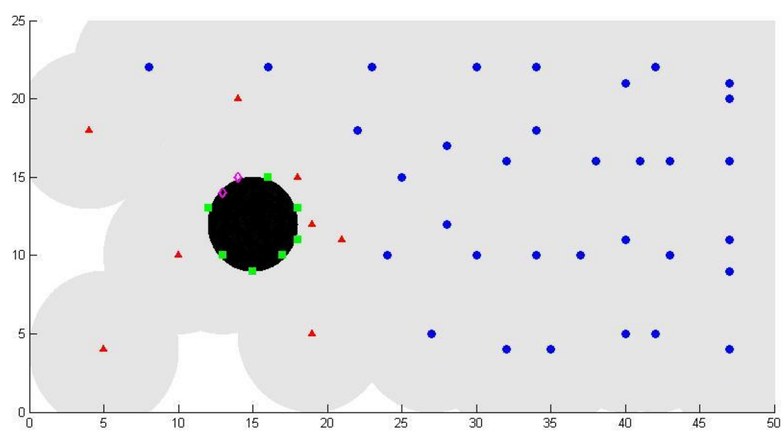

Figure 4.4: Global Simulation ($t = 40$)

At $t = 40$ in Figure 4.4, full encirclement of the event has been achieved by Event agents. Due to the Event agents passing of *meta* tokens two Event agents have decided to enter the sub-behavior of *Meta* behavior, reducing draw of the swarm to the *meta* event.

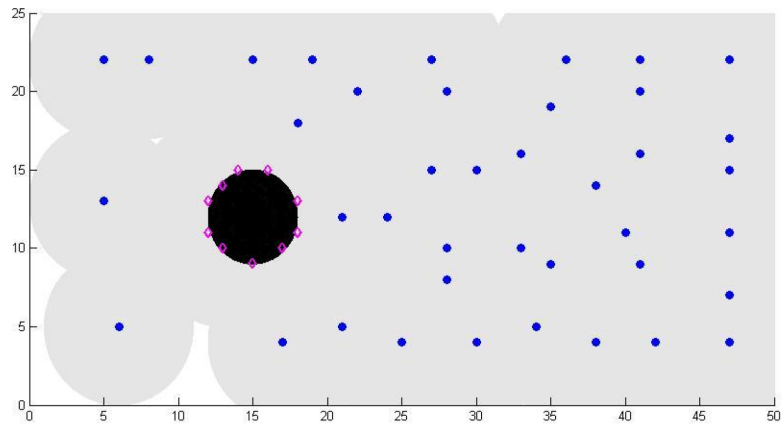

Figure 4.5: Global Simulation ($t = 50$)

At $t = 50$ in Figure 4.5, all Event agents have entered *Meta* behavior, repelling other agents away from the completed task. Being that the Dispersion agents making up the remainder of the swarm have not achieved an optimal distance from their neighbors, thus not achieving an optimized coverage of the field. The swarm remains in an active state of achieving a global best.

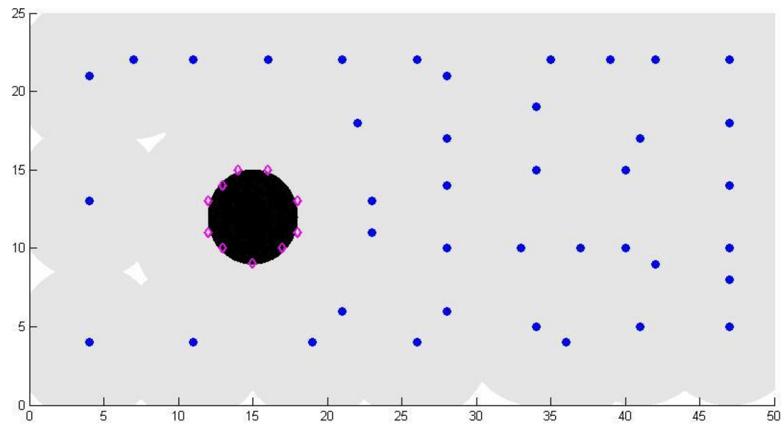

Figure 4.6: Global Simulation ($t = 60$)

At $t = 60$ in Figure 4.6, the agents bounding the *meta* event remain allocated to their task, while the rest of the swarm continues to achieve a better coverage of the field.

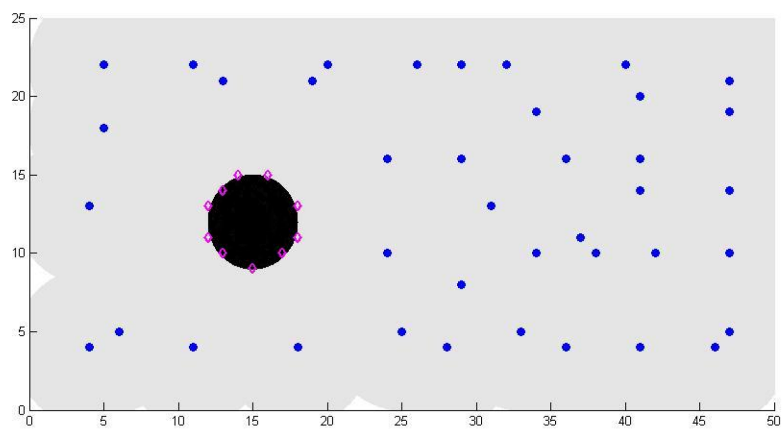

Figure 4.7: Global Simulation ($t = 70$)

At $t = 70$ in Figure 4.7, a global best has approximately been achieved. Oscillation continues to minimize throughout the Dispersion agents as equal spacing between agents, according to the neighborhood force, is accomplished. At the given iteration greater than

98% of the field in question has been covered and the *meta* event has been successfully bounded by the swarm, thereby completing the overall mission.

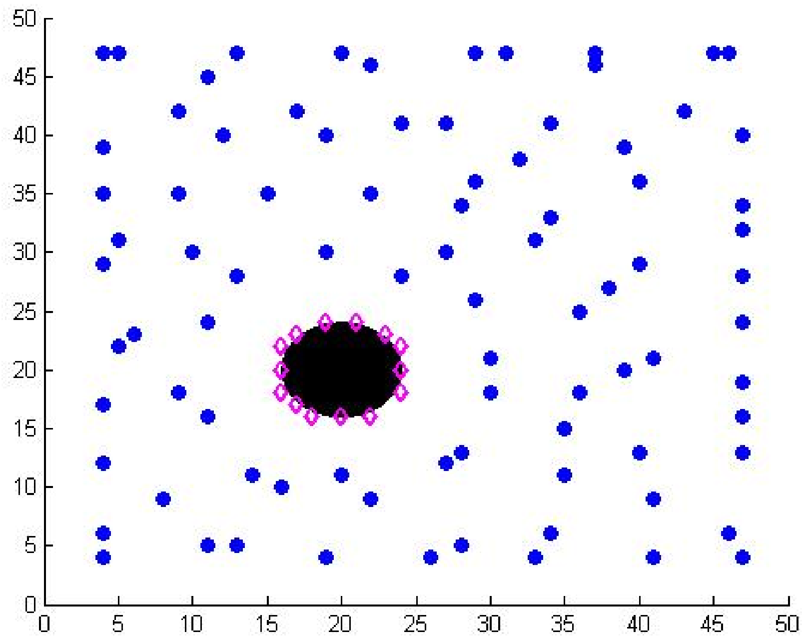

Figure 4.8: Global Simulation Square Field

A resultant of a square field is also presented in Figure 4.8, where the swarm size is increased to 100, the starting coordinates of the swarm are $(40,40)$, and the *meta* event is centered at $(20,20)$.

4.2 Desired Emergence

All subsequent figures within this section are taken from a population of 30 individual simulations that adhere to the parameters specified in Table 4.1. Each result is a component which corresponds to a desired emergent behavior: area coverage, bounding of the *meta* event, traffic over the network, and movements taken.

As a standard for comparison, an optimized random movement algorithm inspired by natural swarms and based on the work of Cohen [49], Hayes et al. [50], and Schmickl et al. [11] was also simulated. This random movement algorithm incorporates the same structure as the proposed algorithm, but does not use the forces generated by the local neighborhoods for dispersion and only has two behaviors: dispersion from other agents and detection of event. In this random algorithm agents will determine a random direction and proceed in that direction until dispersion from its local neighborhood is achieved, an event is detected, or a collision is possible. When a collision is possible, either to the environment or another agent, the agent stops, choses a new direction, and continues if possible. The ideal movement of all agents in straight lines by this random movement spreads out the swarm dispersing from the starting cluster.

Since the decision of a distributed organization for the proposed algorithm was considered the optimal approach, a comparison is also made against a strongly coordinated system where the simulation parameters for the strongly centralized system are also specified in Table 4.1.

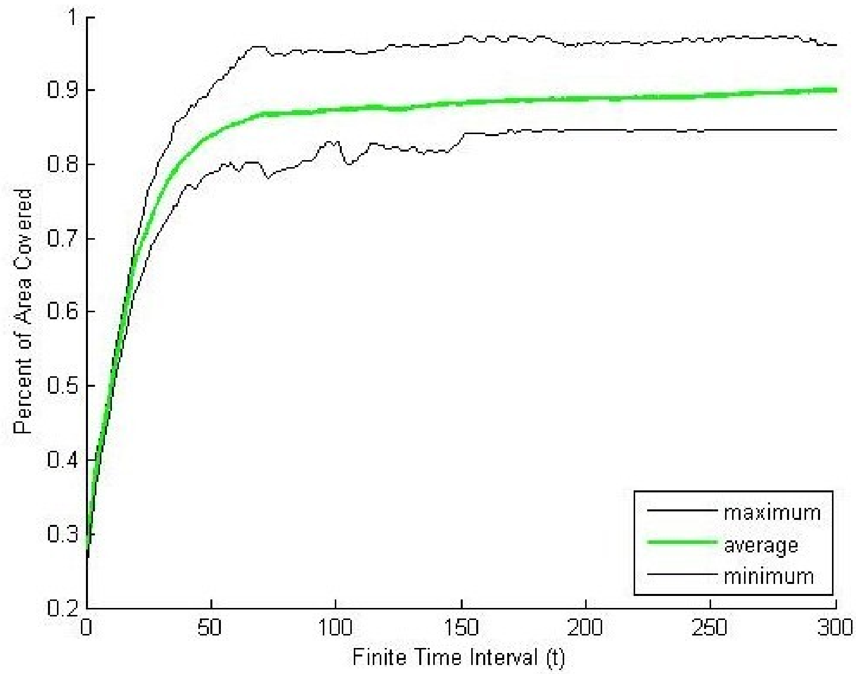

Figure 4.9: Area Covered: Algorithm (30 Replicates)

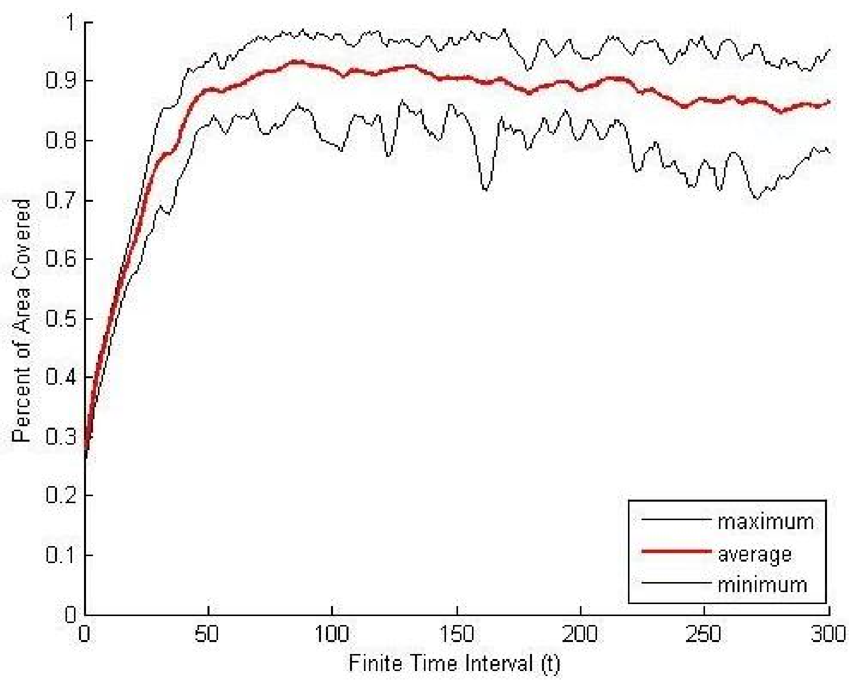

Figure 4.10: Area Covered: Random (30 Replicates)

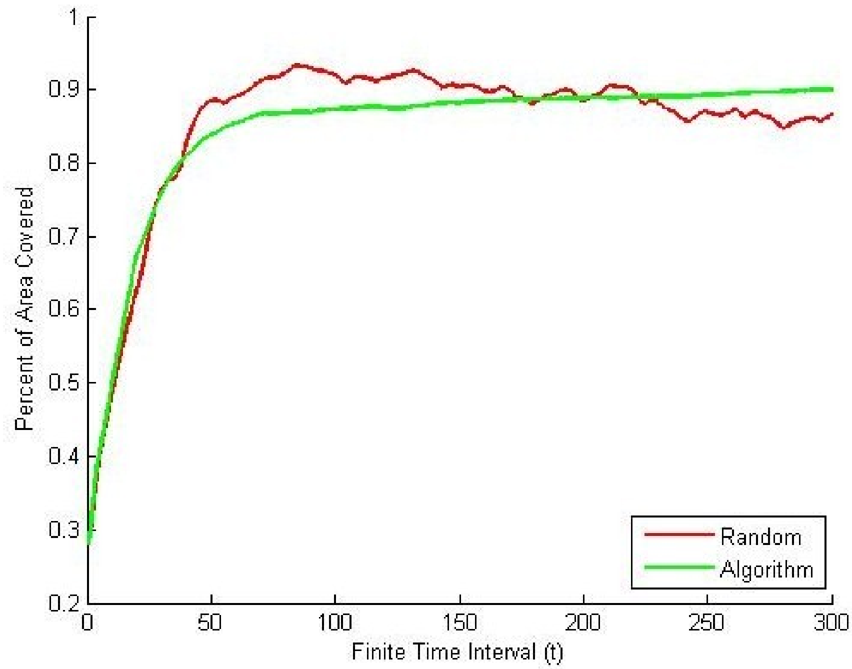

Figure 4.11: Average Area Covered: Algorithm vs. Random

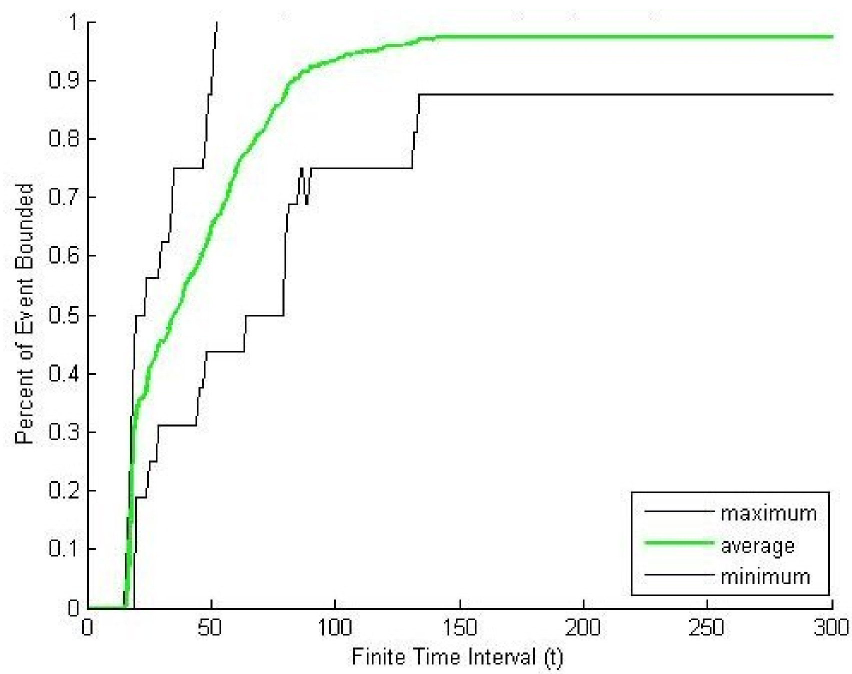

Figure 4.12: Event Bounded: Algorithm (30 Replicates)

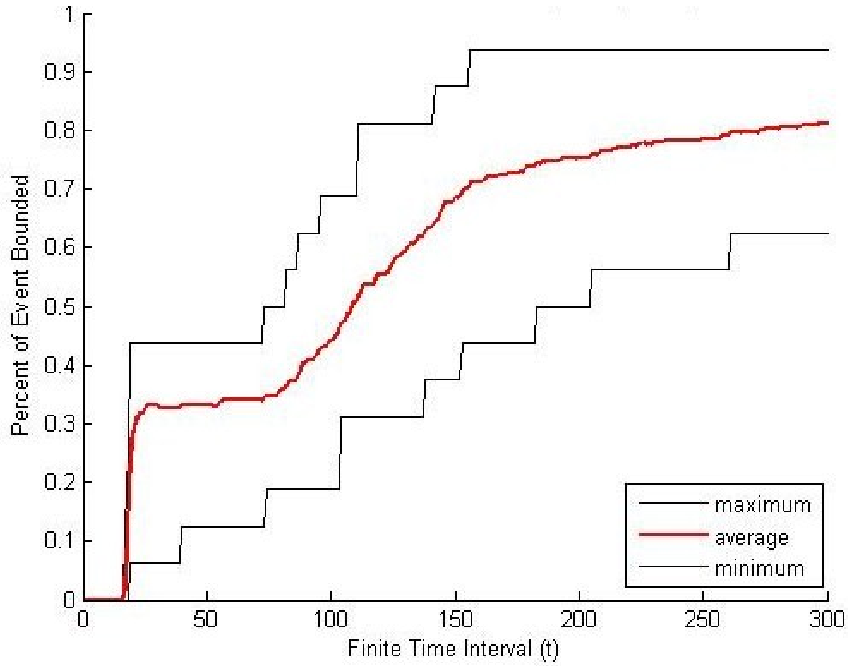

Figure 4.13: Event Bounded: Random (30 Replicates)

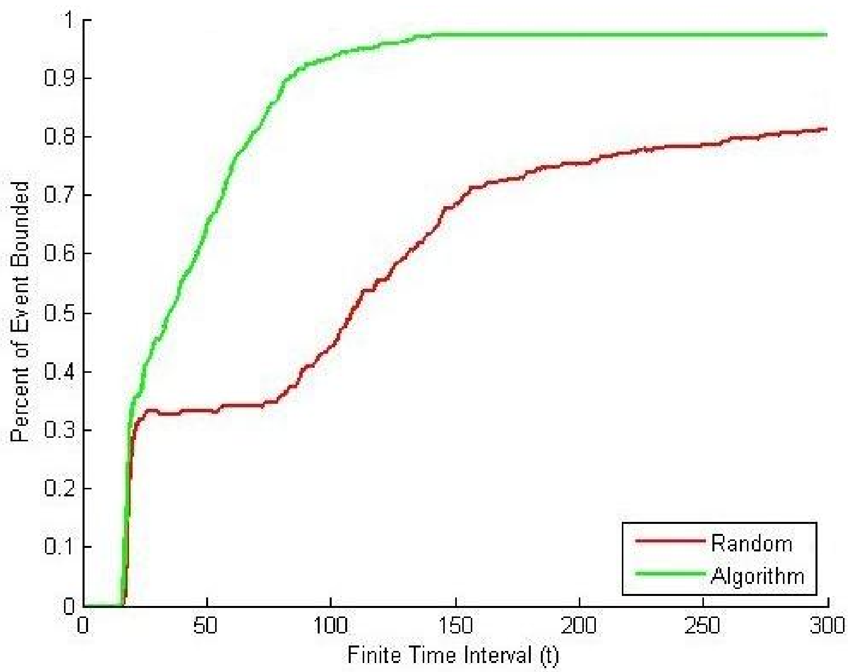

Figure 4.14: Average Event Bounded: Algorithm vs. Random

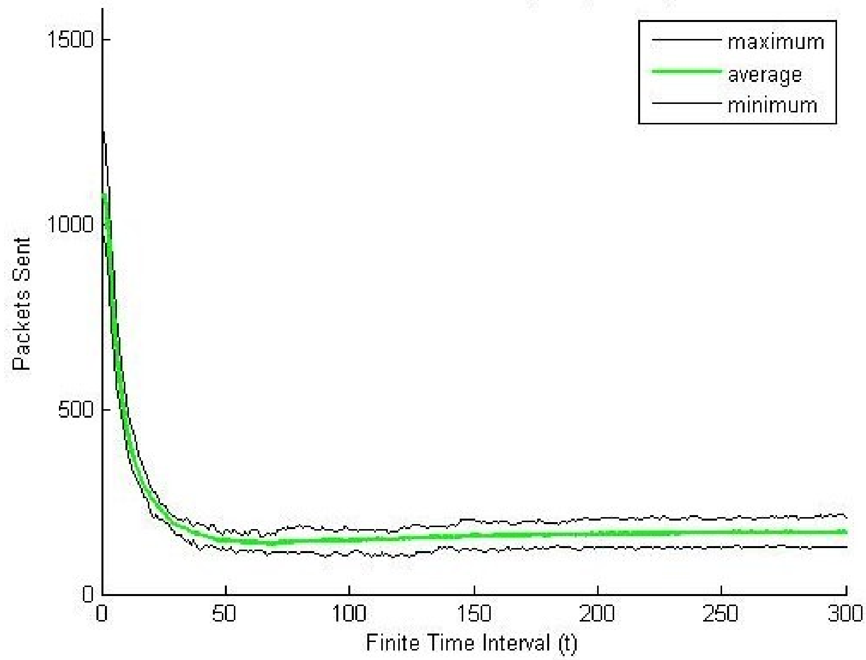

Figure 4.15: Network Traffic: Algorithm (30 Replicates)

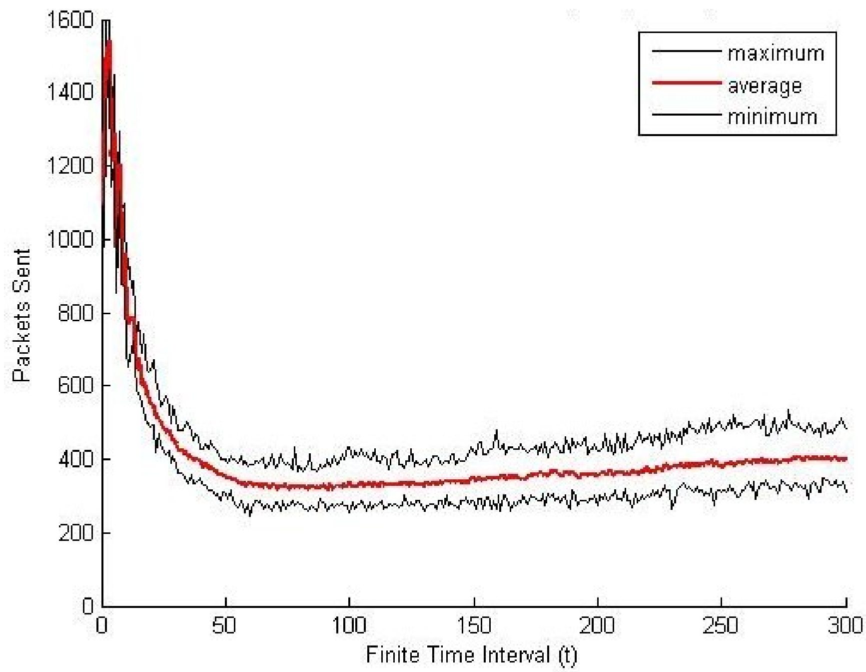

Figure 4.16: Network Traffic: Random (30 Replicates)

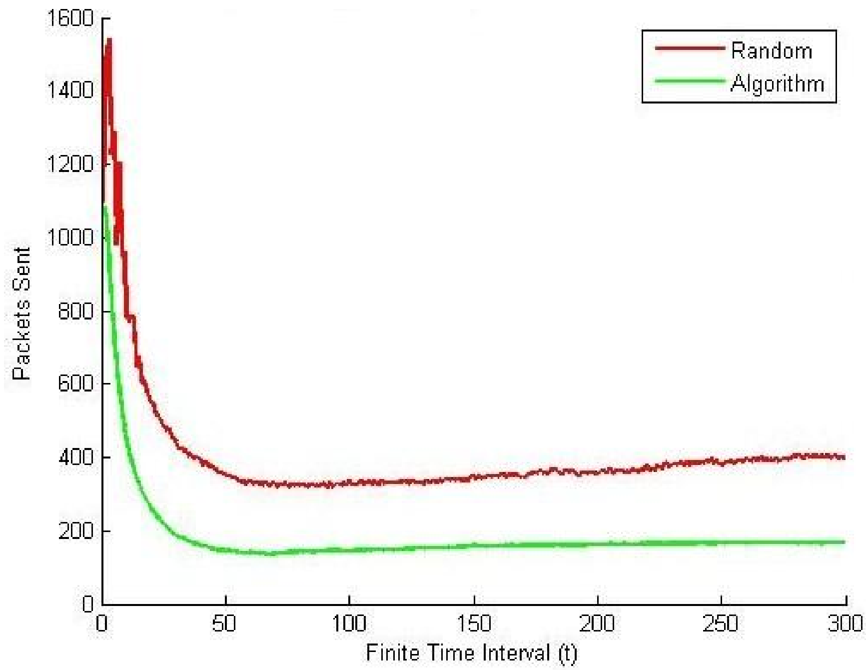

Figure 4.17: Average Network Traffic: Algorithm vs. Random

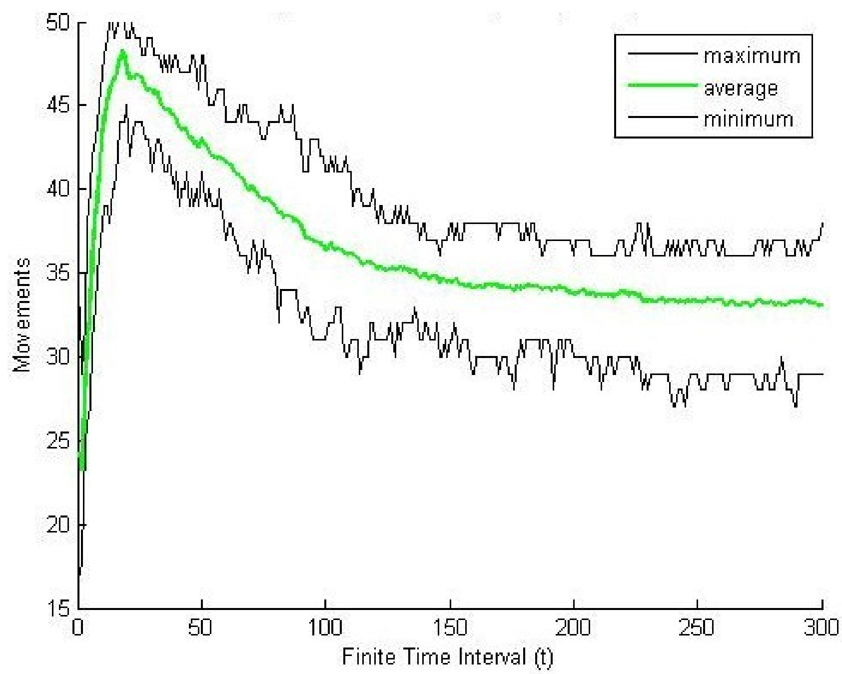

Figure 4.18: Movements Taken: Algorithm (30 Replicates)

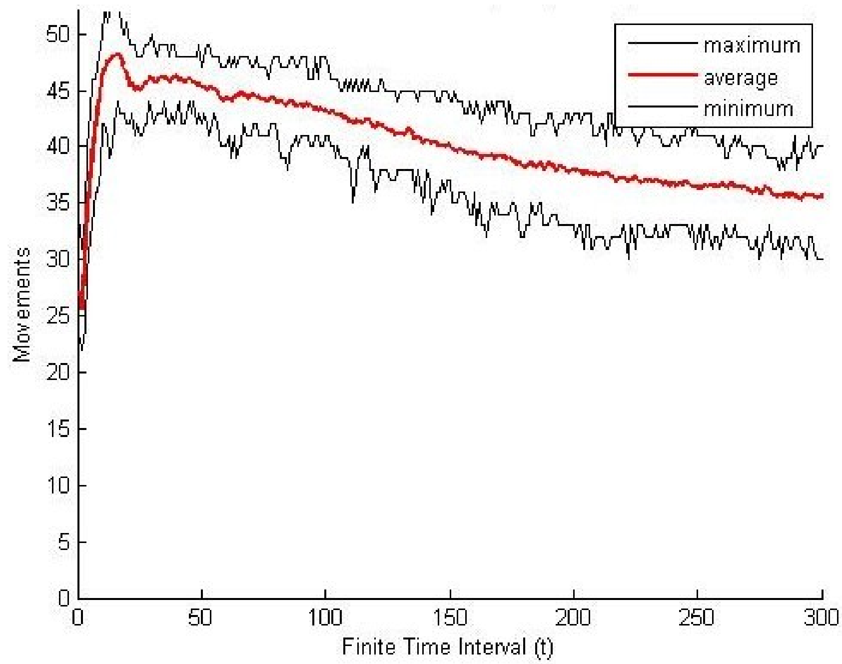

Figure 4.19: Movements Taken: Random (30 Replicates)

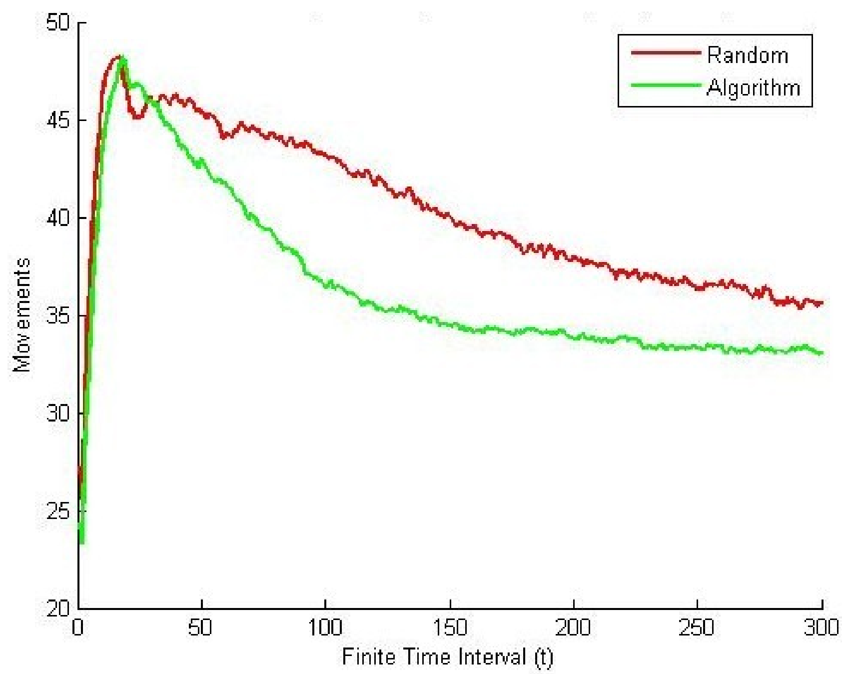

Figure 4.20: Average Movements Taken: Algorithm vs. Random

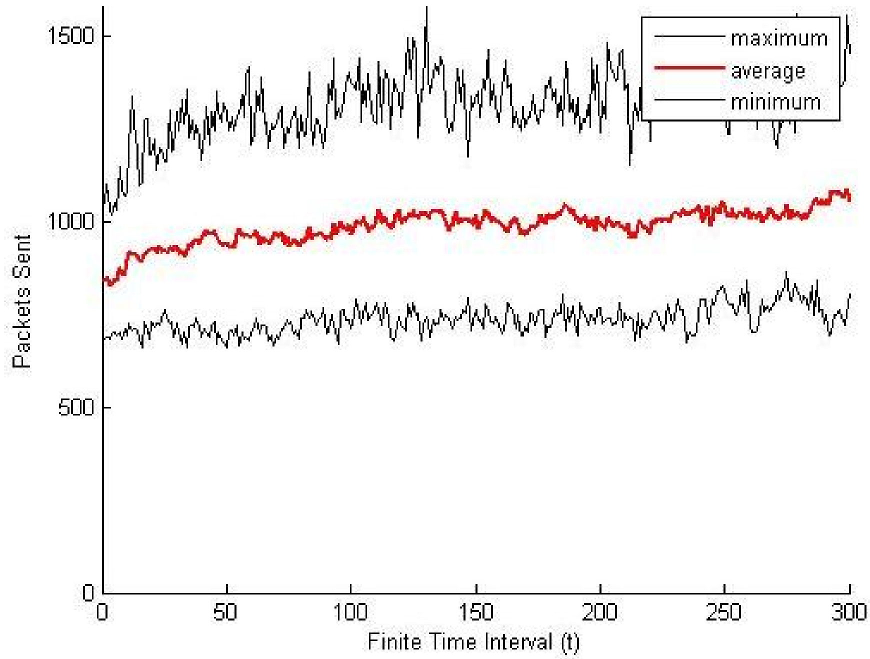

Figure 4.21: Network Traffic: Strongly Centralized (30 Replicates)

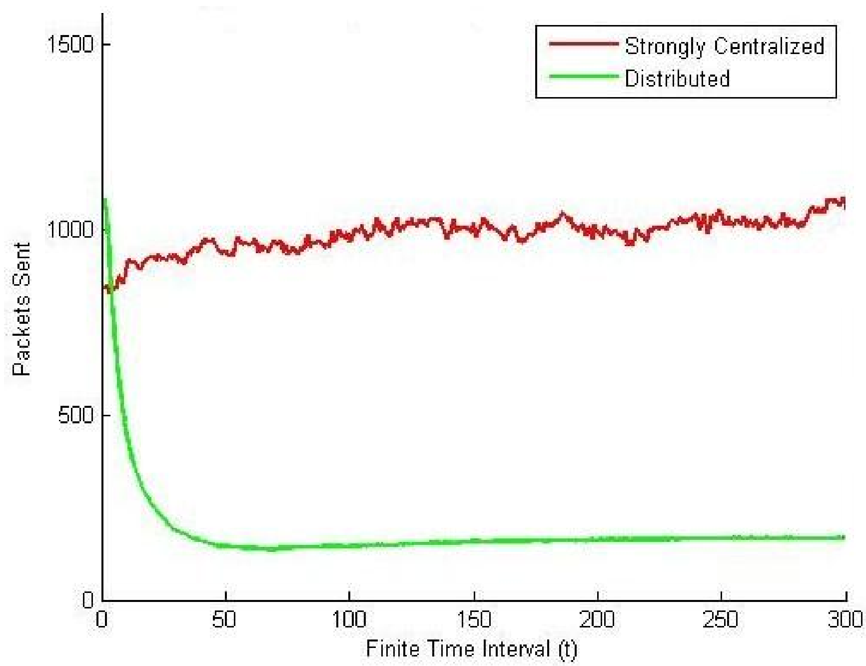

Figure 4.22: Average Network Traffic: Distributed vs. Strongly Centralized

4.3 Dispersal from Initial

Because a normalize and speedy dispersion is a factor in optimally detecting a *meta* event, the proposed algorithm is compared to a similarly structured algorithm by Jian et al. [1]. This similar algorithm also employs the concept of agent reference based on nearest four neighbors. Both algorithms employ the same parameters (4.2). It is important to note that simulations performed with the proposed algorithm reach full dispersion at or before ($t = 1280$) iterations.

Simulation Parameters	
Variable	**Value**
Number of Agents	60
Field X Start	300
Field X End	-300
Field Y Start	300
Field Y End	-300
Swarm Cluster Center Coordinates	$(0,0)$
Communication Range (D_{comm})	100
Dispersion Range (D_r)	30
Environment Standoff	1

Table 4.2: Dispersion Parameters (Jian et al. [1])

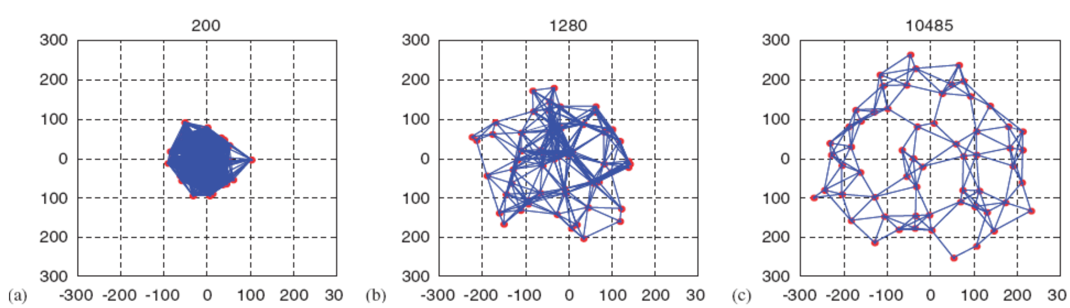

Figure 4.23: Jian et al. [1]: Dispersion of 60 Agents with 6 Desired Neighbors (a) 200 iterations after start (b) 1280 iterations after start (c) 10485 iterations after start

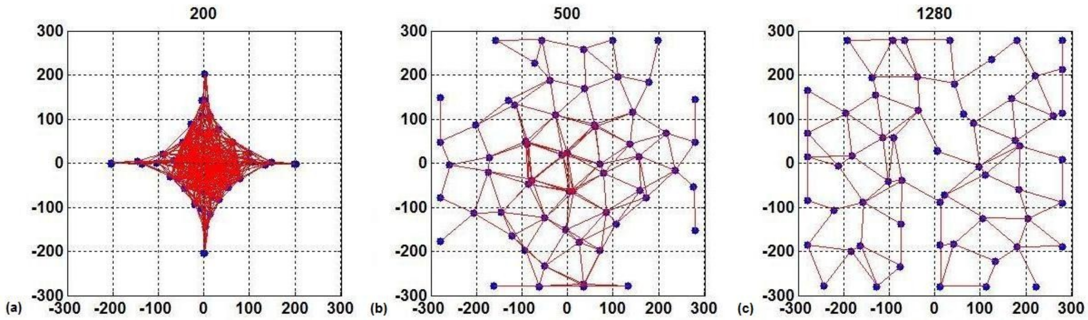

Figure 4.24: Dispersion of 60 Agents with 1 Desired Neighbor (a) 200 iterations after start (b) 500 iterations after start (c) 1280 iterations after start

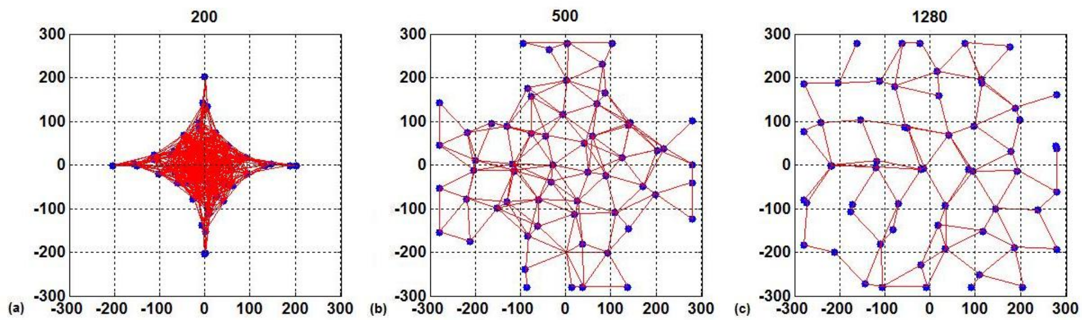

Figure 4.25: Dispersion of 60 Agents with 2 Desired Neighbors (a) 200 iterations after start (b) 500 iterations after start (c) 1280 iterations after start

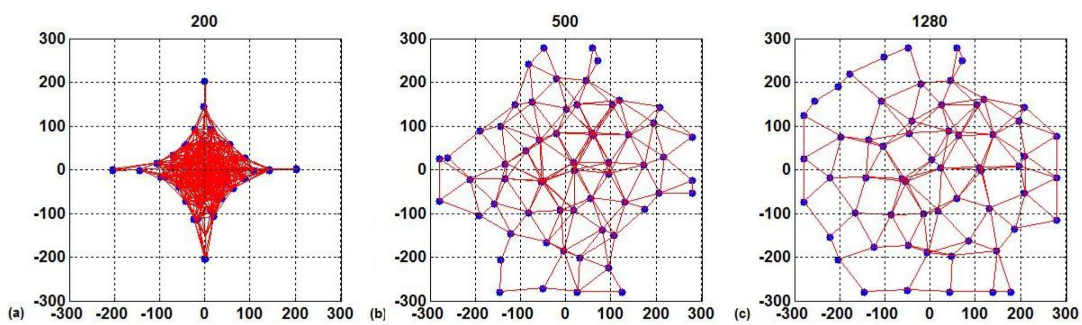

Figure 4.26: Dispersion of 60 Agents with 3 Desired Neighbors (a) 200 iterations after start (b) 500 iterations after start (c) 1280 iterations after start

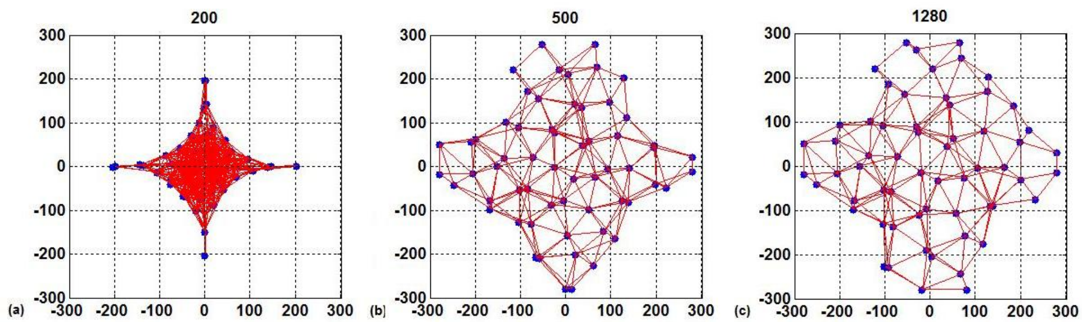

Figure 4.27: Dispersion of 60 Agents with 4 Desired Neighbors (a) 200 iterations after start (b) 500 iterations after start (c) 1280 iterations after start

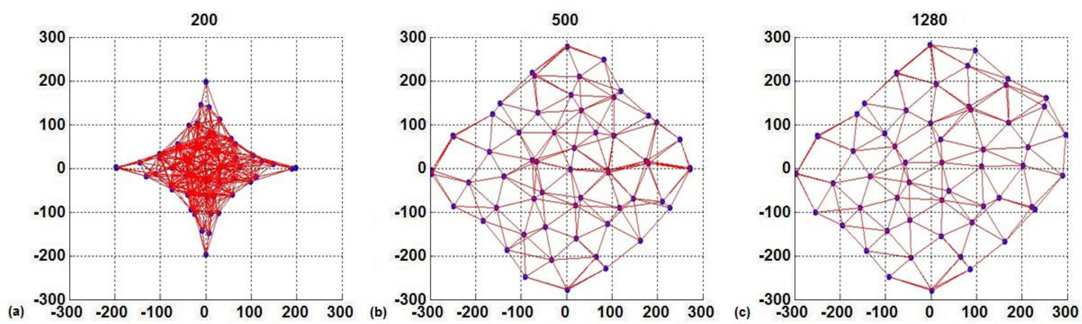

Figure 4.28: Dispersion of 60 Agents with 5 Desired Neighbors (a) 200 iterations after start (b) 500 iterations after start (c) 1280 iterations after start

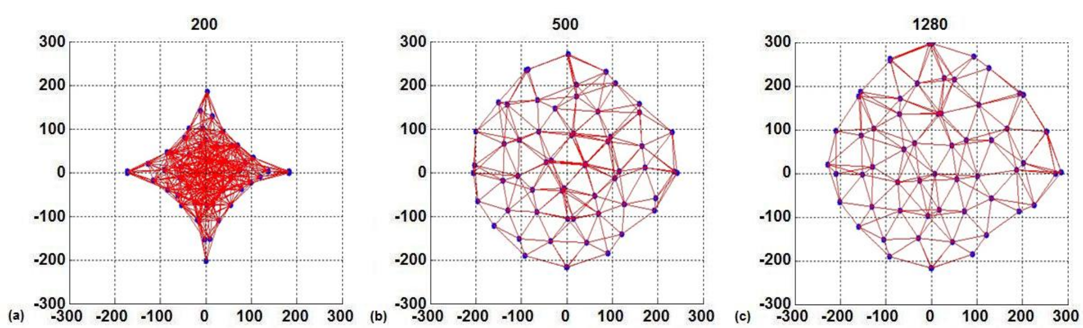

Figure 4.29: Dispersion of 60 Agents with 6 Desired Neighbors (a) 200 iterations after start (b) 500 iterations after start (c) 1280 iterations after start

CHAPTER 5: CONCLUSION AND FUTURE WORK

5.1 Conclusion

The results presented above indicate the proposed algorithm is capable of emulating the desired emergent behavior of detecting and quantifying a *meta* event while distributing agents in an effective manner to cover a large amount of the field. With agents simple enough to require only a minimal amount of intelligence and capabilities, this swarm framework is capable of achieving highly coordinated emergent actions based on the agent to agent interactions.

Employing a local neighborhood interaction allowed the swarm the capabilities of a higher level communication scheme but without the network traffic issues seen with centrally coordinated systems. As shown in Figure 4.22, a distributed communication organization greatly reduces the amount of network traffic. This is due to the necessity of centrally coordinated systems need to contact a leader agent. If an agent is outside of the leader's communication range, multiple hops must be made through other swarm agents. As stated in chapter 2, a centrally coordinated system does show advantages over a distributed system when the system is within the leader's communication range at the initial clustered starting point for the swarm. In contrast as distributed system begins to disperse from the cluster network traffic is dramatically reduced, indicating an optimized organization for the defined mission.

With agents capable of multiple behaviors, chosen by a neural network, the system is able to achieve varying degrees of emergent behaviors that as a whole allow a comparable and more stable form of area coverage. As shown in Figure 4.11, the random movement algorithm is able to achieve a greater coverage of the total field at first, but as more agents become aware of the *meta* event the coverage becomes less than that of the proposed algorithm. In addition the proposed algorithm achieves a relatively stable amount of coverage once dispersion is reached, without the fluctuations witnessed in the random algorithm. The lag in the proposed algorithm's area coverage is due to agents becoming drawn to the *meta* event through positive feedback. The great advantage of implementing the proposed algorithm is in the encompassing of the *meta* event. Shown in Figure 4.2, the algorithm is able to have approximately all of the event bounded at $t = 150$, while the random shows a great drop off as agent begin attempting to encircle the *meta* event and does not achieve a complete quantification of the event within the given time frame. Energy used, through network traffic in Figure 4.17 and movement in Figure 4.20, also shows improvements due to the speed at which dispersion is reached; meaning less is needed for movement as well as information gathering.

Through comparison with a similar communication and movement system, results showed a great increase in speed of dispersion. This is also true when adhering to the 6 desired neighbor links, which assures redundant links to the rest of the swarm for robustness at the cost of speed. Even with this strict interpretation, the proposed algorithm is able to normally and quickly reach a full even dispersion substantially quicker. Through this speedy and stable dispersion a *meta* event is more likely to be detected faster, thus having the swarm capable of reaching a low energy steady-state faster.

5.2 Future Work

Since the final step for any robotic based algorithm is applying it to the physical world, the next logical step would be placing this algorithm on a physical agent. Nuances and error are introduced on this level that could be unaccountable on the simulation level. Differences such as physical limitations, communication interference, as well as agent mortality. Figure 5.1 shows a swarm of proposed physical agents with the capabilities of taking the simulated conciseness of an agent presented in this research directly to the physical realm.

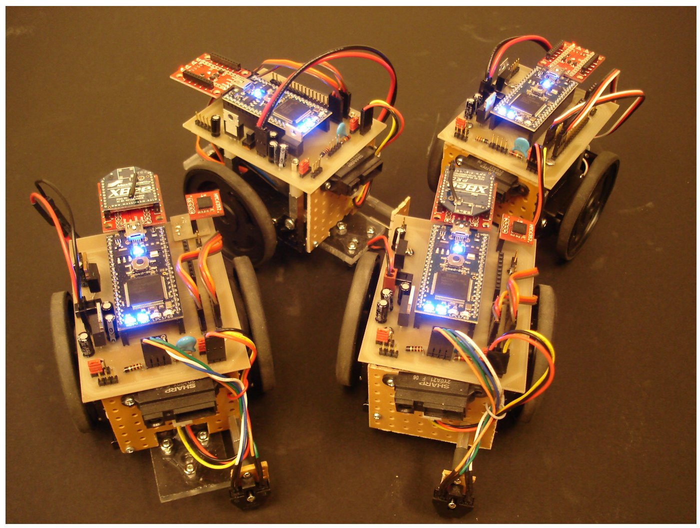

Figure 5.1: Swarm of Proposed Physical Agent Prototype: GERBot

Effectiveness into the algorithms capabilities of quantifying nonuniform *meta* events have been experimented and show promising results, shown in Figure 5.2 and Figure 5.3, but have not been rigorously tested. Additions may have to be made to the local neighborhood structure or the behavioral characteristics to deal with these yet to be discovered adversities.

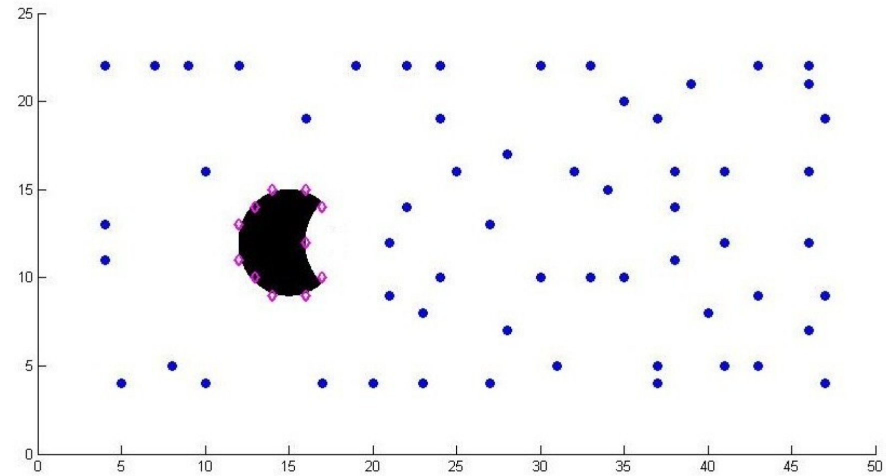

Figure 5.2: Concave *Meta* Event Simulation

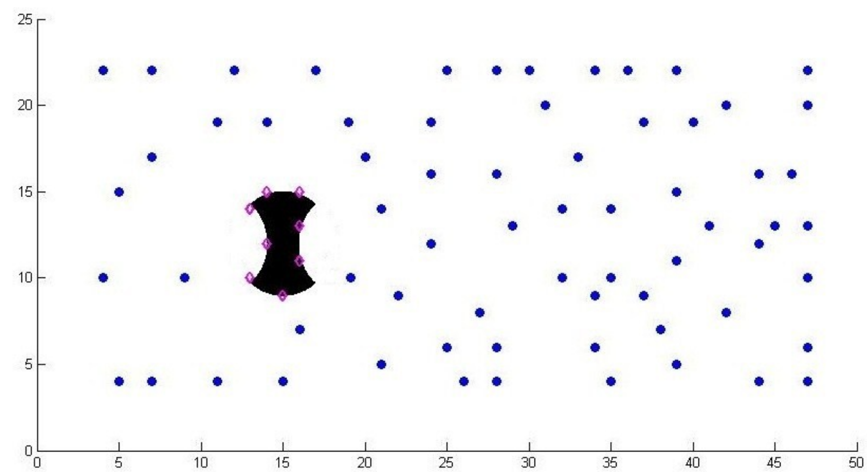

Figure 5.3: Muli Concave *Meta* Event Simulation

Further work in the simulation aspect needs to be completed with the use of dynamic multiple *meta* events, capable of growth and decay, and field size. This research has focused on the ability of a swarm to detect a *meta* event in a 2D Cartesian field. Further extrapolating to a 3D non-Cartesian field would lead to a greater development and characterization of the emergence intelligence witnessed in this research. An advancement to a non-Cartesian movement could also determine if the “star” dispersion emergence is derived from the discretization of the field and not from the internal control algorithm. These dynamics could produce or justify the swarms capability in dealing with a real-world task.

While the current system was limited to a homogeneous configuration, a heterogeneous configuration would allow for another level of optimization. This would allow the use of highly specialized agents. Agents that would function as normal agents in the swarm, but would be capable of completing sub-missions that other agents are incapable of performing. The introduction of this aspect also leads to the ability for agents to self-assemble and create a greater form of agent consciousness within the swarm itself. In essence creating interdependent emergent behaviors on multiple levels of the swarm.

BIBLIOGRAPHY

[1] Y. L. Jian, F. L. Lian, and H. T. Lee, "Deployment of a team of biomimetic searching agents based on limited communication quantity," *Asian Journal of Control*, vol. 10, no. 4, pp. 439–448, 2008.

[2] Y. Mohan and S.G. Ponnambalam, "An extensive review of research in swarm robotics," in *Nature Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, 2009, pp. 140–145.

[3] S. J. Russell and P. Norvig, *Artificial intelligence : a modern approach*, Prentice Hall series in artificial intelligence. Prentice Hall, 1st edition, 1995.

[4] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From natural to artificial systems*, Oxford University Press, Inc., New York, 1999.

[5] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 1987, SIGGRAPH '87, pp. 25–34, ACM.

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

[7] S. Thrun, "Robotic mapping: A survey," in *Exploring Artificial Intelligence in the New Millennium*, G. Lakemeyer and B. Nebel, Eds. Morgan Kaufmann, 2002.

[8] M. Dorigo, “Swarm-bot: an experiment in swarm robotics,” in *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*, 2005, pp. 192–200.

[9] H. Hamann, *Space-Time Continuous Models of Swarm Robotic Systems: Supporting Global-to-Local Programming*, Cognitive Systems Monographs. Springer, 2010.

[10] M. Dorigo, E. Bonabeau, and G. Theraulaz, “Ant algorithms and stigmergy,” *Future Gener. Comput. Syst.*, vol. 16, pp. 851–871, 2000.

[11] T. Schmickl and K. Crailsheim, “Trophallaxis within a robotic swarm: bio-inspired communication among robots in a swarm,” *Auton. Robots*, vol. 25, pp. 171–188, Aug. 2008.

[12] H. Hamann and Heinz W., “Aggregating robots compute: An adaptive heuristic for the euclidean steiner tree problem,” in *From Animals to Animats 10*, M. Asada, J. Hallam, J. A. Meyer, and J. Tani, Eds. 2008, vol. 5040 of *Lecture Notes in Computer Science*, pp. 447–456, Springer Berlin / Heidelberg.

[13] B. Chazelle, “Triangulating a simple polygon in linear time,” in *Foundations of Computer Science, 1990. Proceedings., 31st Annual Symposium on*, 1990, pp. 220–230 vol.1.

[14] A. Farinelli, L. Iocchi, and D. Nardi, “Multirobot systems: a classification focused on coordination,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 34, no. 5, pp. 2015–2028, 2004.

[15] D. Yamins, “Towards a theory of “local to global” in distributed multi-agent systems (i),” in *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, New York, NY, USA, 2005, AAMAS ’05, pp. 183–190, ACM.

[16] D. Yamins and R. Nagpal, “Automated global-to-local programming in 1-d spatial multi-agent systems,” in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems - Volume 2*, Richland, SC, 2008, AAMAS ’08, pp. 615–622, International Foundation for Autonomous Agents and Multiagent Systems.

[17] H. Haken, *Information and self-organization: a macroscopic approach to complex systems*, Springer series in synergetics. Springer, 2006.

[18] E. Gabriel, M. M. Resch, T. Beisel, and R. Keller, “Distributed computing in a heterogeneous computing environment.,” in *PVM/MPI’98*, London, UK, 1998, pp. 180–187, Springer-Verlag.

[19] R. Grabowski, L. E. Navarro-Serment, C. J.J. Paredis, and P. K. Khosla, “Heterogeneous teams of modular robots for mapping and exploration,” *Autonomous Robots*, vol. 8, pp. 293–308, 2000.

[20] J. C. Alexander, B. Giesen, R. Munch, and N. J. Smelser, *The Micro-Macro Link*, University of California Press, California, 1987.

[21] M. Schillo, K. Fischer, and C. Klein, “The micro-macro link in dai and sociology,” in *Multi-Agent-Based Simulation*, Scott Moss and Paul Davidsson, Eds., vol. 1979 of *Lecture Notes in Computer Science*, pp. 303–317. Springer Berlin / Heidelberg, 2001.

[22] G. Nicolis and I. Prigogine, *Self-organization in nonequilibrium systems*, Wiley, New York, 1977.

[23] R. W. Ashby, "Principles of the self-organizing system," in *Principles of Self-Organization: Transactions of the University of Illinois Symposium*, pp. 255–278. 1962.

[24] H.Foerster, "On self-organizing systems and their environments," in *Self-Organizing Systems*, M. C. Yovits and S. Cameron, Eds., pp. 31–50. Pergamon Press, 1960.

[25] T. De Wolf and T. Holvoet, "Emergence versus self-organization: Different concepts but promising when combined," in *Engineering Self-Organizing Systems*, S. A. Brueckner, G. M. Serugendo, A. Karageorgos, and R. Nagpal, Eds., vol. 3464 of *Lecture Notes in Computer Science*, pp. 77–91. Springer Berlin / Heidelberg, 2005.

[26] J. H. Holland, *Emergence: from chaos to order*, Helix Books. Perseus Books, 1999.

[27] M. Dorigo, E. Bonabeau, and G. Theraulaz, "Ant algorithms and stigmergy," *Future Gener. Comput. Syst.*, vol. 16, pp. 851–871, 2000.

[28] M. Dorigo and E. Şahin, "Guest editorial," *Auton. Robots*, vol. 17, pp. 111–113, Sept. 2004.

[29] G. Beni, "From swarm intelligence to swarm robotics," in *Swarm Robotics*, E. Sahin and W. Spears, Eds., vol. 3342 of *Lecture Notes in Computer Science*, pp. 1–9. Springer Berlin / Heidelberg, 2005.

[30] E. Sahin, "Swarm robotics: From sources of inspiration to domains of application," in *Swarm Robotics*, E. Sahin and W. Spears, Eds., vol. 3342 of *Lecture Notes in Computer Science*, pp. 10–20. Springer Berlin / Heidelberg, 2005.

[31] J.D. Bjerknes, A.F.T. Winfield, and C. Melhuish, "An analysis of emergent taxis in a wireless connected swarm of mobile robots," 2007.

[32] G. Beni and J. Wang, "Swarm intelligence in cellular robotic systems," in *Proceedings of the NATO Advanced Workshop on Robots and Biological Systems*, 1989, vol. 102.

[33] M. M. Millonas, "Swarms, phase transitions and collective intelligence," in *Artificial Life III*, C. Langton, Ed. Addison-Wesley, 1994.

[34] J. Kennedy and R. C. Eberhart, *Swarm intelligence*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001.

[35] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, 1995, vol. 4, pp. 1942–1948 vol.4.

[36] V. Trianni, *Evolutionary swarm robotics: evolving self-organising behaviours in groups of autonomous robots*, Studies in computational intelligence. Springer, 2008.

[37] M.J. Matarić, M. Nilsson, and K.T. Simsarin, "Cooperative multi-robot box-pushing," in *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, Aug. 1995, vol. 3, pp. 556–561 vol.3.

[38] F. Zhao and L. Guibas, *Wireless Sensor Networks: An Information Processing Approach*, Morgan Kaufmann Publishers, 2004.

[39] Y.U. Cao, A.S. Fukunaga, A.B. Kahng, and F. Meng, "Cooperative mobile robotics: antecedents and directions," in *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, 1995, vol. 1, pp. 226–234 vol.1.

[40] L.E. Parker, "Cooperative motion control for multi-target observation," in *Intelligent Robots and Systems, 1997. IROS '97., Proceedings of the 1997 IEEE/RSJ International Conference on*, 1997, vol. 3, pp. 1591–1597 vol.3.

[41] L.E. Parker, "The effect of action recognition and robot awareness in cooperative robotic teams," in *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, 1995, vol. 1, pp. 212–219 vol.1.

[42] T. Fukuda and N. Kubota, "Intelligent robotic systems-from a single robot to multiple robotic system," in *Advanced Motion Control, 1998. AMC '98-Coimbra., 1998 5th International Workshop on*, 1998.

[43] F. M. Ham and I. Kostanic, *Principles of Neurocomputing for Science and Engineering*, McGraw-Hill Higher Education, 2000.

[44] A. Agah and G.A. Bekey, "A neural network-based controller for homogeneous and heterogeneous distributed robotic systems," in *Systems, Man, and Cybernetics, 1996., IEEE International Conference on*, 1996, vol. 1, pp. 655–660 vol.1.

[45] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, USA, 1 edition, 1996.

[46] C. Muñoz, N. Navarro, T. Arredondo, and W. Freund, "Cooperative adaptive behavior acquisition in mobile robot swarms using neural networks and genetic algorithms," in *Electronics, Robotics and Automotive Mechanics Conference, 2008. CERMA '08*, 30 2008.

[47] M. Kumar, D.P. Garg, and V. Kumar, "Segregation of heterogeneous units in a swarm of robotic agents," *Automatic Control, IEEE Transactions on*, vol. 55, no. 3, pp. 743–748, 2010.

[48] G. Schner, "A dynamic theory of coordination of discrete movement," *Biological Cybernetics*, vol. 63, pp. 257–270, 1990.

[49] W. W. Cohen, "Adaptive mapping and navigation by teams of simple robots," *Robotics and Autonomous Systems*, vol. 18, no. 4, pp. 411–434, 1996.

[50] A.T. Hayes, A. Martinoli, and R.M. Goodman, "Swarm robotic odor localization," in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, 2001, vol. 2, pp. 1073–1078 vol.2.

[51] R. Grabowski and P. Khosla, "Localization techniques for a team of small robots," in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, 2001, vol. 2, pp. 1067–1072 vol.2.

[52] T. Schmickl, R. Thenius, C. Moeslinger, G. Radspieler, S. Kernbach, M. Szymanski, and K. Crailsheim, "Get in touch: cooperative decision making based on robot-to-robot collisions," *Autonomous Agents and Multi-Agent Systems*, vol. 18, pp. 133–155, 2009.

[53] S. Kornienko, O. Kornienko, A. Nagarathinam, and P. Levi, "From real robot swarm to evolutionary multi-robot organism," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, 2007, pp. 1483–1490.

Appendices

APPENDIX A: NEURAL NETWORK DIAGRAM

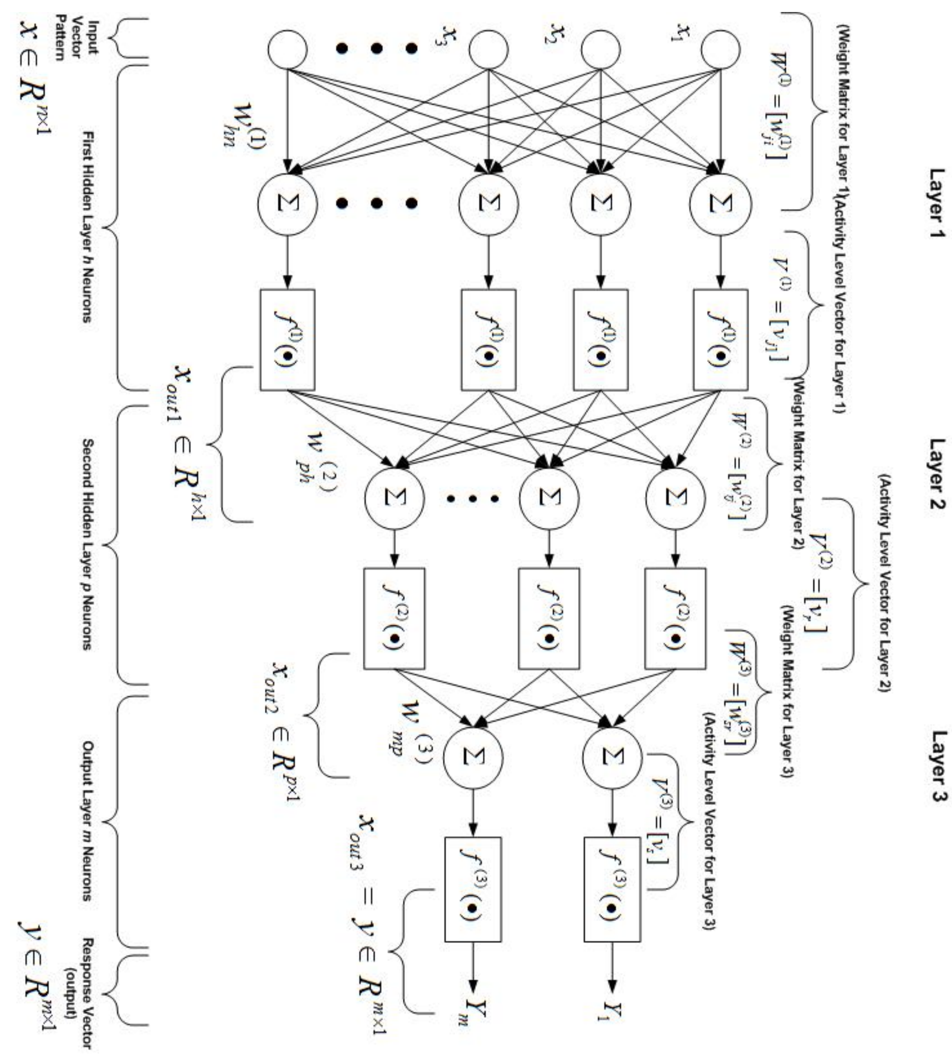

Figure A.1: Three Layer Neural Network: Feed-Forward