2019-04-01

Moderating Influence as a Design Principle for Human-Swarm Interaction

C Chace Ashcraft
Brigham Young University

Moderating Influence as a Design Principle for Human-Swarm Interaction

C. Chace Ashcraft

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Jacob Crandall, Chair
Michael A. Goodrich
Daniel Zappala

Department of Computer Science

Brigham Young University

ABSTRACT

Moderating Influence as a Design Principle for Human-Swarm Interaction

C. Chace Ashcraft
Department of Computer Science, BYU
Master of Science

Robot swarms have recently become of interest in both industry and academia for their potential to perform various difficult or dangerous tasks efficiently. As real robot swarms become more of a possibility, many desire swarms to be controlled or directed by a human, which raises questions regarding how that should be done. Part of the challenge of human-swarm interaction is the difficulty of understanding swarm state and how to drive the swarm to produce emergent behaviors. Human input could inhibit desirable swarm behaviors if their input is poor and has sufficient influence over swarm agents, affecting its overall performance. Thus, with too little influence, human input is useless, but with too much, it can be destructive. We suggest that there is some middle level, or interval, of human influence that allows the swarm to take advantage of useful human input while minimizing the effect of destructive input. Further, we propose that human-swarm interaction schemes can be designed to maintain an appropriate level of human influence over the swarm and maintain or improve swarm performance in the presence of both useful and destructive human input. We test this theory by implementing a piece of software to dynamically moderate influence and then testing it with a simulated honey bee colony performing nest site selection, simulated human input, and actual human input via a user study. The results suggest that moderating influence, as suggested, is important for maintaining high performance in the presence of both useful and destructive human input. However, while our software seems to successfully moderate influence with simulated human input, it fails to do so with actual human input.

Keywords: human swarm interaction, human robot interaction, swarms, robot swarms

ACKNOWLEDGMENTS

This research was sponsored by the US Army Research Lab under grant DOD ARL W911NF-16-2-0226. We also thank the Brigham Young University Department of Statistics for their assistance with this work.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

On September 3, 2014, then United States Secretary of Defense Chuck Hagel delivered a keynote speech at the "Defense Innovation Days" event in which he proposed that robotics and automation would be key in maintaining United States military dominance in the following years [1]. Shortly after, Paul Scharre published a work on how robotics swarms fit perfectly within Chuck Hagel's proposed mission [2]. He claims: "Swarms of robotic systems can bring greater mass, coordination, intelligence and speed to the battlefield, enhancing the ability of warfighters to gain a decisive advantage over their adversaries." They also have the potential to be significantly cost effective, which, while always useful, is particularly important as military budgets fluctuate.

Robotic swarms also appear to have great potential for application to various problems of interest outside of the military due to their robustness properties arising from decentralization. By being decentralized, the swarm is able to incur minimal risk to itself by having a built-in redundancy that helps alleviate failures caused by faults in individual agents or by unexpected environmental events and conditions. This is referred to by Winfield and Nembrini as fault tolerance in robot swarms [3], and is possibly the most significant contributor to robot swarms' robustness.

Until humans can design sufficiently intelligent and capable algorithms to appropriately govern these systems for all applications, human interaction or control of the swarm will likely be required in real-world applications. Kira and Potter state: "Real-time control is a particularly important issue because the locality and unpredictability of emergent behavior makes influencing the entire swarm after deployment both difficult and necessary" [4]. Swarm technology is based on having simple agents perform complex tasks through local sensing and communication between agents. The complex or intelligent behavior is said to *emerge* through the interactions between the swarm agents rather than through a centralized controller. Because the emergent behaviors are often the result of complex interaction rules that are not well understood, it is difficult to design algorithms that ensure that the swarm will exhibit the desired behavior during deployment, especially for large swarms. Part of Winfield and Numbrini's solution to this is extensive testing, validation, and analysis of a swarm system, but a principle problem with their approach is the assumption of a "reasonably well understood operational environment," which we believe will be regularly violated in real-world swarm deployment.

Allowing human interaction will likely provide additional flexibility to a robot swarm in less well-understood environments and in unpredicted scenarios [5]. Also, for military use, the military will likely prefer human interaction because it facilitates maintaining their hierarchical command and control structure and the general public will prefer it due to various ethical questions regarding lethal force with autonomous agents.

Yet, at the same time, utilizing robot swarms presents a new challenge for command and control both in and out of the military. Scharre notes that, as the size of the swarm increases, control must become more focused on the swarm as a whole rather than on the individuals of the swarm. This may seem obvious considering humankind's limited capacity to multitask, [6, 7], but how this is to be done is less so. Winfield and Nembrini state that "A distinguishing characteristic of distributed systems based upon swarm intelligence is that they have no hierarchical command and control structure, and hence no common mode failure point or vulnerability." Ironically, the thing that makes swarms so useful, their decentralization, is what makes human control over them difficult. By adding a human operator to the swarm, one adds an element of centralization in the operator and potentially restricts the autonomy of each agent, which is usually the source of the swarm's robustness and emergent behavior.

Scharre suggests that human control of a swarm could take many forms, hinting at hierarchies of human control and what we call "playbook" style control [8–10], where the human assigns a task to the swarm and completely relies on the swarm's autonomy to perform it. However, Scharre also suggests that mixed types of control mechanisms are likely desirable, which include control paradigms where the human interacts with the swarm at low levels as well as high levels. Yet for the case of low-level interactions, it will likely be challenging for humans to know how to interact with swarms in order to achieve a desired affect. He states: "Human controllers will need to know when to intervene to correct autonomous systems, and when such intervention will introduce suboptimal outcomes" [2].

We take this a step further by asking what form the human-swarm interaction interface should take to ensure correct or beneficial behavior in robot swarms. While training is one way of preventing poor quality or suboptimal outcomes from a swarm [11], another is by controlling how the human will interact with the swarm, perhaps by preventing certain types of interactions, where such interactions would lead to a suboptimal result. In this work, we propose a theory of interaction scheme design to do exactly this by keeping the amount of influence the human has over the swarm at a moderated level, thereby allowing the swarm to take advantage of useful human input and ignore detrimental input. We then create a software implementation of our theory for a specific swarm and interaction method, and show that our implementation can help for at least two types of scenarios using simulated human input. Testing with human input confirms the importance of moderating input based on human skill with the swarm and the information they are provided, but also shows that our implementation to do so fails to perform as desired.

Chapter 2

Related Work

Kolling *et. al.* published a survey in 2016 summarizing work in human-swarm interaction [5], discussing work on commonly studied swarm models and various aspects of human-swarm interaction. Kolling *et. al.* categorize swarm models into four categories: Bioinspired, Control-theoretic, Amorphous Computing, and Physics-inspired. Biological systems are likely some of the earliest inspirations for swarms, and are probably the most commonly thought of by people when they hear the word "swarm." Various algorithms in swarm intelligence, a related but distinct field [12], have already been successfully developed and applied to real world problems [13–16]. Perhaps one of the most popularly implemented and studied model, at least in simulation, is Couzin *et. al.*'s model presented in [17]. Using this model, various researchers have been able to show a variety of interesting spacial behaviors and have also studied their performance with human-swarm interaction [18, 19].

The model we propose to use in this paper is bioinspired, but is different from most models in that the behavior we are interested in is less the spacial positioning of the swarm than their ability to accomplish a certain task; however, there are other bioinspired models with similar interests [15, 20, 21]. Sumpter has published many studies on biological swarms and collective behavior, as well as principles for engineering bioinspired swarm systems [22, 23].

There have been numerous publications in control theory regarding swarms [24–27], many of which are of significant practical value regarding implementation of actual robotics swarms. Benefits of the control-theoretic studies include consideration to physical dynamics often absent in bioinspired work, powerful mathematical tools, and certain theoretic guarantees.

However most of the control-theoretic work in swarms seems only to be concerned with spatial orientation tasks, and the simplifying assumptions needed for the theory are often impractical or restrictive. An explanation of amorphous computing can be found in [28], and while its purpose differs significantly from swarm robotics, the principles governing its implementation are very similar. Applications of amorphous computing to swarm robotics can be found in [29, 30]. Finally, physics-inspired systems attempt to use physics-based laws to produce swarm-like behavior. In this case, each agent can be thought of as an entity that interacts with other entities in the environment based on an established physical law or mathematical equation, like particles in an environment [31–33]. Kolling *et. al.* [5] note that while the results of the physics-inspired swarms are often similar to that of some bioinspired swarms, the perspective is distinct in that interactions in a physics-based swarm are thought of more as passive effects while those in bioinspired swarms are chosen actions.

Two topics that are particularly related to our work are *Neglect Benevolence* and *Levels of Automation*. Neglect Benevolence in swarms is the property that the human-swarm system's performance will improve by having the human delay input [34]. This is opposed to *Neglect Tolerance* [6], which regards how long a system can continue to perform sufficiently well without human input. In their study, Nagavalli *et. al.* ([34]) show that allowing a human operator of a simple swarm to provide input to the swarm too early will result in a sub-optimal outcome. It can further be inferred from that study that input too early or too late can also result in failure of the swarm to achieve a goal.

Level of automation (LOA) scales for human-machine systems were originally suggested by Sheridan and Verplank [35] and then adapted and applied to human-supervisory control systems in [36–38]. Applications to human-swarm interaction include [39–41]. LOA is related to our work in that it is also used to measure the relationship between the human and the machine and to design human-machine interaction, however, there are several minute differences. In particular, there is a strong focus in the LOA on communication between the human and the machine on whether the machine will require human input, which we do not address, and LOA is typically used to design both the machine and the interaction, while we assume the machine is set and focus on the interaction alone.

Shared control is another topic both related and applied to our work. Shared control attempts to improve the performance and capability of a human-machine system by balancing the intent of the operator with the sensors and algorithms that run the machine [42, 43]. For example, Crandall and Goodrich show that sharing control during teleoperation of a single robot increased performance of the human-robot team, reduced the amount of attention the robot needed to function correctly, and was easier to use than manual teleoperation. In 2014, Brown, Jung, and Goodrich studied shared control with a bio-inspired swarm based on Couzin's model [44]. Control is shared with the swarm by allowing the human to control only a small subset of the agents in the swarm. The rest of the swarm consists of agents that are not influenced by the human-controlled agents, and agents that are influenced by both the human controlled agents and the human-immune agents. They then propose graph-theoretic methods for measuring human-influence over the swarm, which they dub *persistence*, *span*, and *connectivity*.

Finally, while much research in swarms is done using techniques such as graph theory and differential calculus as explained above, we have chosen to use an agent based model. Agent based models are often used to examine or study the effects of simple rules or behaviors on a large set of agents interacting in an environment [45], especially when more rigorous methods of analysis become computationally prohibitive. Common areas where this is done include business [46], economics [47], and social science [48].

In this work, we attempt to combine these ideas and build upon them in order to address the problem of humans interacting with large, task-based robot swarms. Much of the current work in swarms, for example, simulate smaller swarms that are not focused on any particular task outside of spatial distribution. We attempt to work with a swarm that is both larger in number of agents and is focused on accomplishing a non-spacial task, to prepare for the eventuality of actual robot swarm implementations. There has also been a large amount of work done to understand how best for a human to interact with a single robot, or a few robots that do not behave as a swarm. We attempt to use the ideas from these works, especially neglect benevolence and shared control, to then enable quality human interaction with large, task-based robot swarms.

Chapter 3

Theory

3.1 Terminology

In order to facilitate discussion of our theory and methods, we define three, not necessarily distinct, parties of interest integral to the development and deployment of robot swarms: the *Problem Holder*, the *Designer*, and the *Operator*. The Problem Holder is the person or group of people who define the purpose of the swarm, fund the Designer, and most likely employ the Operator. The Designer is the person or group responsible for the design and implementation of the robot swarm of interest, and the Operator is the person or group that interacts with the swarm during its deployment. A possible example of the three would be a military entity, a military contractor, and enlisted military personnel.

Another important term for this work is *influence*. Influence is the ability of the Operator to control the behavior of the swarm. Thus, the higher the influence a human operator has over a given swarm system, the more ability said operator has to force the swarm to do his or her bidding. In the context of complex distributed systems, a general definition of influence is difficult to derive. Therefore, in this work, we will only analyze influence comparatively, i.e. we will claim that one interaction scheme provides an operator with more influence than another if it gives the operator greater ability to dictate the actions of the members of the swarm. This imprecise definition is less than ideal, but is sufficient for this work.

3.2 Theory

Due to the complexity of swarm systems, it is typically very difficult for a human to understand the state of the swarm at any particular time, let alone how to interact with it in order to drive it to some desired state. Thus, while it may be desirable in some cases to have a human operator in a swarm system, how best to make use of the human is challenging. For example, allowing an operator to have high influence over the swarm with little training may cause the operator to hinder or block the desirable emergent behaviors that make the swarm useful in the first place. We propose that there are three primary methods to overcome this problem:

1. Human training: Overcome challenges and problems with swarm interaction by making the operator an efficient and knowledgeable user of the swarm. This solution is expensive and time consuming, as each human operator must be trained to expertise, and still may make mistakes.

2. User interface design: Make the user interface sufficiently intuitive and easy to use such that human users can easily know what to do and how to do it. As this type of solution is difficult to implement, it is an active area of research. It also requires extensive testing with potential human operators, and may not generalize to many tasks or environments.

3. User interaction design: Design the interaction method in order to take advantage of useful human input while moderating the effect of bad input. Ideally, this could be done mostly at the Designer level using solid human-swarm interaction principles and the Designer's understanding of the swarm, reducing the need for human training and testing with human operators.

Our research falls into the third category. If able to be done well, this should be a cost-effective and efficient method of creating practical human-swarm systems.

3.2.1 Using Influence to Guide Human-Swarm Interface Design

Some have suggested that the best way to for a human to interact with a swarm is via a "playbook" type interface [8–10], which means that the Operator interacts with the swarm by assigning a task to be performed, and then allowing the swarm to perform that task until termination or until a new task is assigned. We are interested in allowing the Operator to interact with the swarm during the performance of the task in addition to assigning the task. However, with the challenge of understanding the swarm's behavior and what to do to alter it, we believe that a primary consideration when designing a human-swarm interaction scheme should be the influence the Operator has over the swarm.

At a high level, the Problem Holder will have a purpose for the swarm, as well as some measure of performance, but will depend on the Designer and Operator to fulfill that purpose and maximize performance. The Designer can design the swarm to maximize the performance function given by the Problem Holder, but usually requires various assumptions be made, which will most likely be broken at various times during deployment. The Operator can assist the swarm in maximizing its performance by providing information unavailable to the swarm's sensors, providing high-level reasoning the swarm is unable to perform, or overcoming issues caused by the violation of swarm design assumptions. However, the Designer will have implemented the swarm agents' behaviors for a reason, and if the human operator is provided enough influence to sufficiently override those behaviors, then the human can decrease the performance of the swarm with poor input. We propose that finding a balanced level of influence will potentially allow the swarm to take advantage of useful Operator input while being unaffected by detrimental input. A theoretical example of this relationship for some human-swarm system is given in Figure 3.1.

The *sweet spot*, or balanced interval, in Figure 3.1 will vary based on the system, environment, specific operator knowledge and capability, but some general principles for designing balanced interaction schemes would be desirable. We propose *Feedback Based Dynamic Influence* as a general principle, which is the idea that Operator influence should

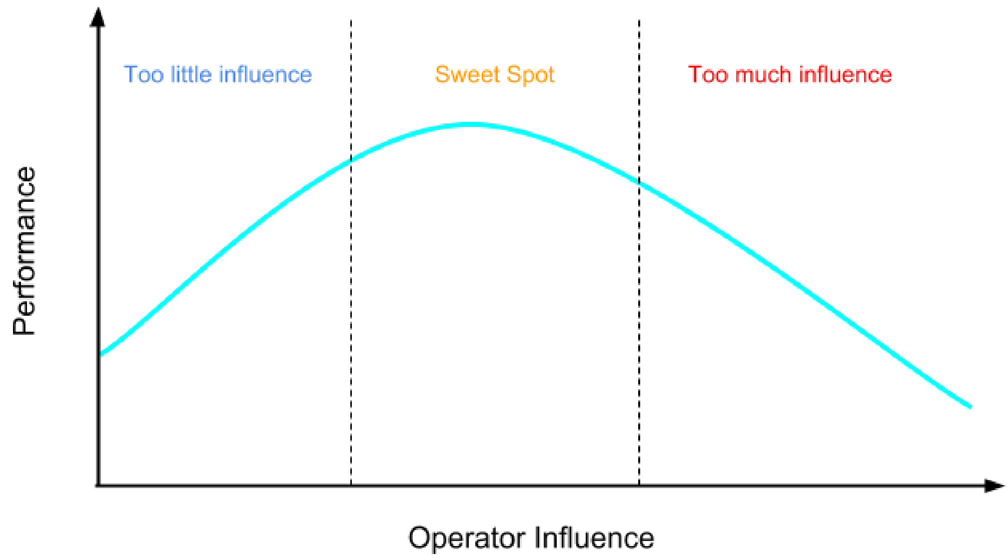


Figure 3.1: Theoretical relationship between Operator influence and swarm task performance for some task and swarm.

change over time based on feedback from the swarm and Operator. The key to this principle is the understanding that Operator behaviors will vary over time and between operators, therefore, no one influence setting will likely suffice to maintain balanced influence in general. By allowing influence to change based on feedback from the swarm and Operator, the Designer can maintain balanced influence for the duration of any deployment for any operator. Exactly how the influence should vary based on the feedback will depend on the swarm, task, and possible feedback, but as the Designer will be designing and implementing all three, these are variables the Designer can control for and use to do so.

In the following sections, we provide results that demonstrate the usefulness of balancing influence in a simulated swarm, we propose a possible implementation of our proposed general principle, and then test our implementation by simulation and a user study.

Chapter 4

Swarm Simulation Tool

In order to study human interaction with a swarm, we implemented a simulator that can both run a simulated swarm and allow human interaction with it. While we hope this research will apply to general swarms, for this study we use a hub-based colony as our model, which is a swarm that revolves around a central location or hub, such as an ant colony, a bee colony, or a termite colony. We are interested in hub-based colonies for various reasons, in particular: 1) Natural hub-based colonies already seem to be task oriented, rather than spatially oriented, 2) Natural hub-based colonies range in size from hundreds to hundreds of millions of individuals [8], and 3) mathematical models for hub-based colonies already exist [5, 22, 49].

The particular hub-based colony we use is a honey bee model inspired by Nevai et. al. [49], and modified for our purposes. In this chapter, we will describe our model and user interface, and then introduce our method of human interaction with this swarm.

4.1 Honey Bee Model

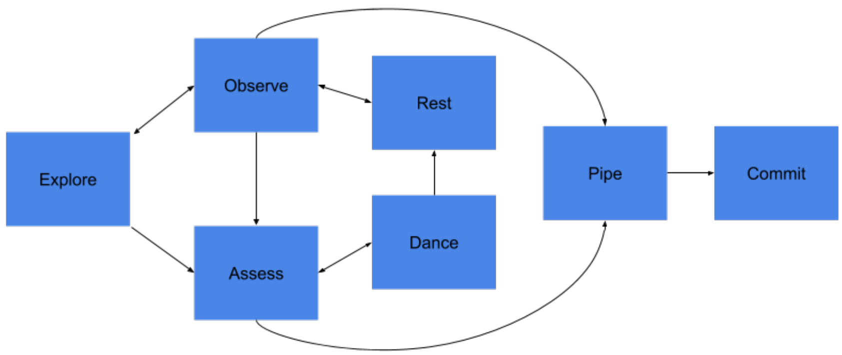
When a honey bee colony hatches a new queen, the colony divides with one part staying in its current location and the other moving to a temporary one, such as a large tree branch, while specific bees, referred to as *scouts*, seek a new nest location [49]. This process is known as *swarming*. While the majority of the new colony stays at the intermediate hub, the scout bees search the area for an acceptable new nest site. When a scout discoverers a site, it assesses its quality, and then returns to the hub to recruit other scouts to assess it further.

The scout recruits other bees via what is known as the waggle dance, as the scout bee does a sort of dance to indicate the relative location and quality of the new site. If an observing scout at the hub (one that is not currently assessing or advertising a site) sees a dancer, it will then attempt to find the advertised site, assess it itself, and return to recruit. Once enough scouts are assessing a single site, the bees 'select' that site, and return to inform the colony of the new nest site location. Additional information about honey bee nest site selection is available in [50–52].

Nevai et al. [49] published a dynamical systems approach to modeling this behavior and showed the existence of several equilibria. However, while we were originally inspired by this paper, we instead use an agent based model in an attempt to more accurately represent swarm behavior with individual robots. The simulated swarm is mostly based around a honey-bee agent that we implemented as a state machine. A state transition diagram as well as a brief description of states and transitions for our agents is given in Figure 4.1.

The task this model attempts to accomplish is high quality site selection in a large environment in limited time. Practical applications of this model are likely few, but could include any task that requires selecting a single location out of many in a large environment, where site quality can be assessed by an agent. Two examples are selecting a location at which to establish a base of operations in a large, unknown environment, and determining the location most likely to contain iron ore on a mountain. Further applications could be plausible by making small modifications to the swarm, such as designing the swarm to work with non-stationary sites, but for simplicity, we work with the model as stated.

Further, given the task this swarm is meant to perform, we assume that the Problem Holder's purpose for this swarm is to select the best site in the environment. Therefore, selection of the best quality site in the environment will be the primary measure we use to gauge our swarm's performance for the entirety of this work. We consider a few others for completeness and because they are interesting, but assume they are all secondary to best-site selection.



Explore	The agent randomly explores the environment for a finite time, seeking potential nest sites.
Observe	The agent returns to the hub (if not there already) and randomly moves about the hub watching for dancers and pipers.
Assess	An agent in this state is attempting to assess the quality of a site. This may come from the agent discovering a site during exploring or by observing a dancer advertise a site.
Dance	After an agent discovers a site and assesses it itself, it returns to the hub to communicate its findings to the rest of the colony through a "dance." Real honey bees perform what is called a *waggle dance*, but we only simulate its effect.
Rest	Biological bees need to rest, and we assume robot bees will need to charge or something similar. Regardless, having agents simply go to the hub and do nothing for a period of time seems important to the total dynamics of nest selection.
Pipe	When the number of agents assessing a site exceeds a threshold, agents begin to pipe. In real bees this is thought to be done by bees vibrating their wings at a certain frequency around the bees at the hub. This "warms them up" and they start doing the same.
Commit	When all agents at the hub are piping, the collective concludes that it has made a choice, and the whole colony moves to the site that was piped for.

Figure 4.1: Honey bee model state transition diagram and table.

4.1.1 Model Implementation

Our implementation of a swarm that follows these dynamics consists of a custom built state machine for the agents and an environment class which contains all of the information about the environment and runs the simulation. The agent implementation closely follows the previously described model, but includes various parameters and low level behaviors that define each agent's movement behavior and how the agent's interact with each other in each state. These parameter settings evolved over the course of our research to create the desired behaviors and performance. For example, the initial settings for convergence often caused the agents to converge too quickly to sites that were near the hub. While this is normal for the honey bee model, because nearer sites are more desirable in terms of the swarm's survivability, we desired site selection based on pure site quality regardless of location. Therefore, we implemented additional convergence criteria to slow the swarm's convergence and increase the likelihood a higher quality site would be chosen regardless of its proximity to the hub.

The environment class contains all the information about the environment the agents are in, such as site information and obstacles, the behaviors governing interactions with the agents and their environments, and the loop running the simulation. It also contains the code for data collection and communication with the user interface.

4.1.2 Model Assumptions

We make various assumptions about our model both in an attempt to make it more realistic, and to make it sufficiently simple to work with. For example, an important assumption we make is that the agents are inexpensive robots. This is reasonable as one of the desirable qualities of a swarm agent is to be disposable. However, the implications we assume result are that the agents are incapable of complex processing, their sensors are noisy, they can only communicate short range, and that they do not carry locating technology like GPS. All of this limits the abilities of the agents, the interactions available to the Operator, and the information available to the Operator. The primary consequences we see in this work are that we assume that the Operator cannot see agent locations on the user interface, swarm agents have noisy site assessments, and that the agents only communicate in the hub or if they are at the same potential nest site. We may also assume that the agent behaviors are limited, as more complex agents could be designed to effectively perform this task without any need for human interaction.

4.2 User Interface

The user interface (UI) is set up as a web server, written in Node.js, that allows a user to initialize simulations in a web browser and then interact with the simulation through a display within the browser. The display is a 2D representation of the simulated environment contained in the previously mentioned environment class. The primary elements displayed include the hub, a radial display providing information about the swarm distribution in space, and potential site locations. The hub is a yellowish circle in the center of the environment, and is where all agents are located at the beginning of the simulation. As the simulation begins, agents leave the hub in different directions and begin to explore the environment for possible nest sites. Again, while the UI is capable of drawing the agents as they move throughout the environment, we assume that the agents will be equipped with GPS, and therefore will not be visible during actual deployments. Therefore, we provide other means of feedback based only on information the agents report when they leave and return to the hub. The magnitude of agents leaving in a given direction and returning in each direction is shown via blue and green dots, respectively, on the radial display (Figure 4.2), which appear as spikes due to lines of the same color drawn to connect the dots. Sites are displayed as colored circles, with the color of the site indicating its quality. Site qualities are real numbers in the left open interval $(0, 1]$, which map to colors between dark red and dark green. The darker the red, the nearer its quality is to 0, and similarly with green and a quality of 1. Qualities near 0.5 appear yellow. An example of the user interface is given in Figure 4.3, which displays the elements just discussed, as well as a possible agent distribution.

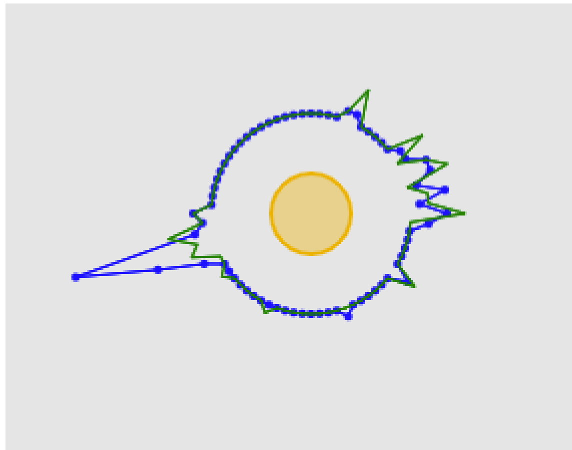


Figure 4.2: Centered on the hub, the radial display indicates the directions agents leave and enter the hub, and in what quantities. Blue dots connected by blue lines create spikes on the display in the direction agents leave the hub. The larger the spike, the more agents are leaving in that direction. The green spikes behave in the same manner, but indicate agents entering the hub rather than leaving. This provides the Operator with an idea of the current swarm state using available information from the agents, as they are not equipped with GPS.

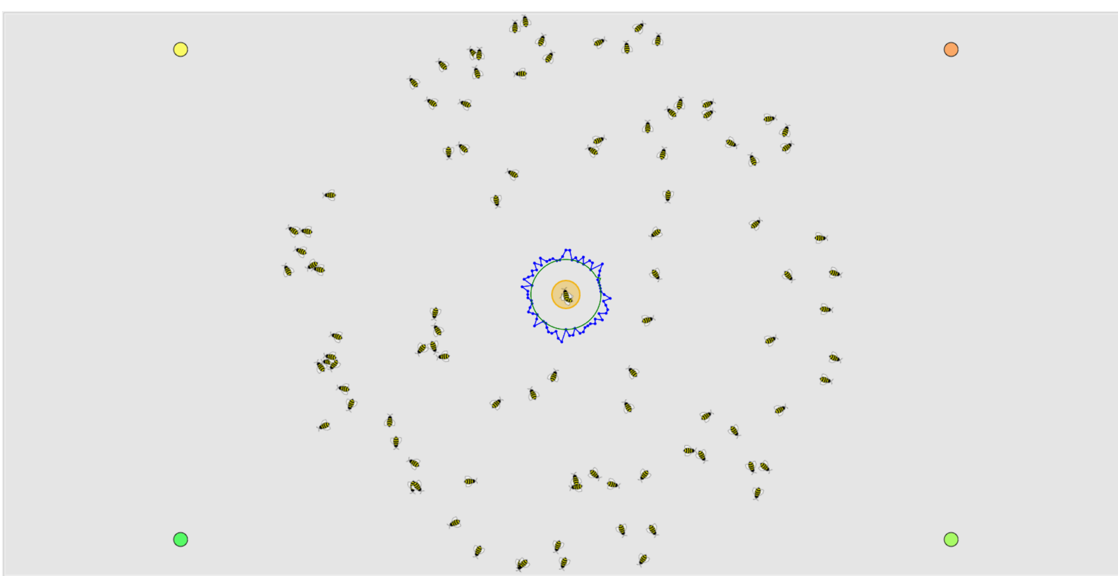


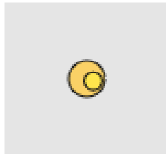
Figure 4.3: Example environment. Agents are shown here as bees (not visible to user), potential sites are shown as colored circles, with more red denoting poorer sites and more green denoting better sites.

4.2.1 Interface Assumptions and Additional Features

A very important assumption we make regarding our interface is that the site information it provides is is imperfect. We believe this is a reasonable assumption, as real environments are subject to change over time and may be inaccurately recorded. Further, if the environment is perfectly known, there is no purpose in deploying the swarm to seek the best potential site. This swarm is meant to explore a large area, to discover potential sites in the environment, and then to choose the best one, The Operator is meant to assist the swarm do so. Therefore, we assume that for each simulation with human interaction, even if the information shown on the UI is accurate, this cannot be assumed by the Operator. Having the Operator make this assumption will be our primary justification for poor Operator input in the following Chapters.

Because we assume agent locations will not be known (and therefore not shown on the display) and the agent feedback so far is rather limited, for Chapter 6, we will provide the Operator with two types of additional agent feedback on the UI that fall within our agent model assumptions. The first we call *agent markers*, which display the locations and qualities of sites being assessed by agents via colored circles (Figure 4.4a) similar to potential sites but much smaller. The location of a marker indicates the location of a site, and the color indicates the site's quality as before. The second is called the *best site indicator*, which indicates the best quality report via a large purple circle with the reported quality printed next to it (Figure 4.4b). This value is "forgotten" every 1,000 time steps to allow newer reports to take precedence and to overcome excessively noisy reports. We felt that these two types of feedback, in additional to the radial display, were sufficient for an operator to understand the current state of the swarm and were reasonable under our model and display assumptions.

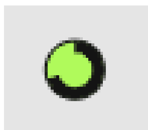
Finally, also for the user study discussed in Chapter 6, to further help human users to understand the operations of the swarm, we add three things to the user interface: 1. a site quality color legend, 2. circular site quality indicators, and 3. a timer. The first two are



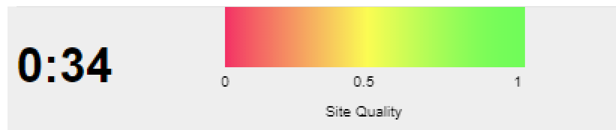
(a) Example of a site with an agent marker. Note that the marker and site are displaying the same quality.

(b) Example of the best site indicator around an agent marker.

Figure 4.4: Examples of UI feedback.



(a) Example of site with with circular quality indicator.



(b) Site quality legend and timer. Located just under the swarm display.

Figure 4.5: UI additions for user study.

to assist the human in comparing site qualities, as it can be difficult to differentiate similar site qualities based on color. The legend is a simple bar showing the color change from site quality 0.0 to site quality 1.0. The circular site quality indicators are thick black arcs around the outside of sites, whose circumference is proportional to the site's quality. Lastly, the timer is provided to show the users how long they have been in the simulation. An example of a site with the circular indicator, as well as an image of the timer and legend, are shown in Figure 4.5.

4.3 User Interaction – Beacons

While the swarm is relatively capable on its own, it does not always find the best quality site in its initial exploration, nor does it always choose the best site if it is found. This is partly by design. Optimizing the swarm and, as mentioned in Section 4.1.2, implementing more complex swarm behaviors practically eliminate the need for human input for this task. By keeping the swarm simple, we adhere to our assumption of inexpensive robots and provide a purpose for a human operator. This allows us to study how altering Operator influence in the interaction scheme affects swarm performance without the difficulties of a more complex swarm system. Thus, for our imperfect swarm, the Operator can assist the swarm through additional exploratory behaviors, additional reasoning about the environment, and by pushing the swarm to converge to higher quality sites.

The method of human interaction that we chose to use was beacon placement. Beacons are placed in the environment by the Operator and interact with the agents that come within its radius of effect. Beacon placement is a relatively simple method for interacting with swarms, and has been used by others studying human-swarm interaction [53, 54]. In our case, the user places a beacon at the desired location in the environment via a mouse click. The beacon is displayed as a green circle without a black lining, as the sites have, and is slightly transparent. The size of the circle shows the beacon's radius of effect. The radii of effect of the beacons will usually be preset for each simulation we discuss, but, in principle, they could be chosen by the Operator at the time of placement.

The primary effect we chose our beacons to have is attraction towards the beacon's center. The idea is to allow the operator to direct the agents towards areas that he or she believes are worth exploring or contain a high quality site. As with the swarm, beacon behaviors evolved over the course of our research. For this work we will only discuss two particular sets of behaviors, referred to as *Attractor 0* and *Attractor 1*. Descriptions of the two sets of behaviors are as follows:

- **Attractor 0**: When an explorer agent enters the radius of effect of Attractor 0, with probability 0.8, the agent will be attracted to the center of the beacon, and all of its movements become biased towards the beacon's center. The resulting effect is that most agents that enter the area of effect wander towards the center with small variations, and once at the center start wandering around the immediate center until the beacon expires. If a site is found within the beacon's radius of effect, the agent becomes an assessor and the beacon no longer has effect on it.

- **Attractor 1**: Attractor 1 has the same base behavior as Attractor 0, but has an additional effect on agents in the hub. For each time step a beacon exists in the environment, there is a 30% chance that the hub will convert an observing agent (also in the hub) to a special type of exploring agent and send it towards the placed beacon. The sent agent's exploring behavior is less random, biased towards moving forward, and the length of time it spends exploring is reduced. The sent agents also will ignore sites not in the radius of effect of a beacon, thus increasing the likelihood that the area the Operator is interested in is explored.

While there are many different methods of human interactions possible, we chose these because of their simplicity, because we believe they provide near-balanced influence to the Operator, and because their similarity simplifies the comparison of influence. We argue that Attractor 1 provides the Operator with more influence than Attractor 0, giving us at least two levels of influence to consider. Seeing why this is the case is relatively straightforward. Attractor 0 only operates on exploring agents that are exploring on their accord, while Attractor 1 creates additional exploring agents and biases their initial movements towards itself. The difference in influence becomes especially obvious when noticing that, later in the simulation, exploring agents are very rare due to the assessing and recruiting processes. Therefore, Attractor 0 tends to only have influence towards the beginning of the simulation, while Attractor 1 maintains some influence until the swarm converges. In the next chapter we provide results demonstrating how Attractor 0, although seemly too low influence to be very useful, allows the swarm to perform better than with Attractor 1.

Chapter 5

Development and Early Testing

In this chapter we provide initial simulation results that lead us to propose a theoretical design principle for appropriately moderating Operator influence. We then propose an implementation of this principle for our swarm model and interaction method, and show that it improves performance over Attractors 0 and 1 alone.

5.1 Model Characterization

To better understand the swarm, we evaluate it's ability to select the best site in the environment under various circumstances. In particular, we exam (a) how the number robots in the swarm and (b) the distribution of potential sites in the environment impacts the swarms ability to find the site with the highest quality.

5.1.1 Quantity Increases Performance

We experimented with various numbers of agents in the swarm in two environments, shown in Figures 5.1 and 5.2, with equidistant sites and various site qualities. Experiments consisted of 100 simulations for each environment/agent-number combination. The results are given in Figures 5.3 and 5.4. These results illustrate that having more agents increases the likelihood the swarm selects the best site. However, more agents also increase the time it takes for the swarm to make a choice. This is because more agents search the environment more thoroughly and with more redundancy, but more agents make it more difficult for the convergence criteria of the swarm to be satisfied.

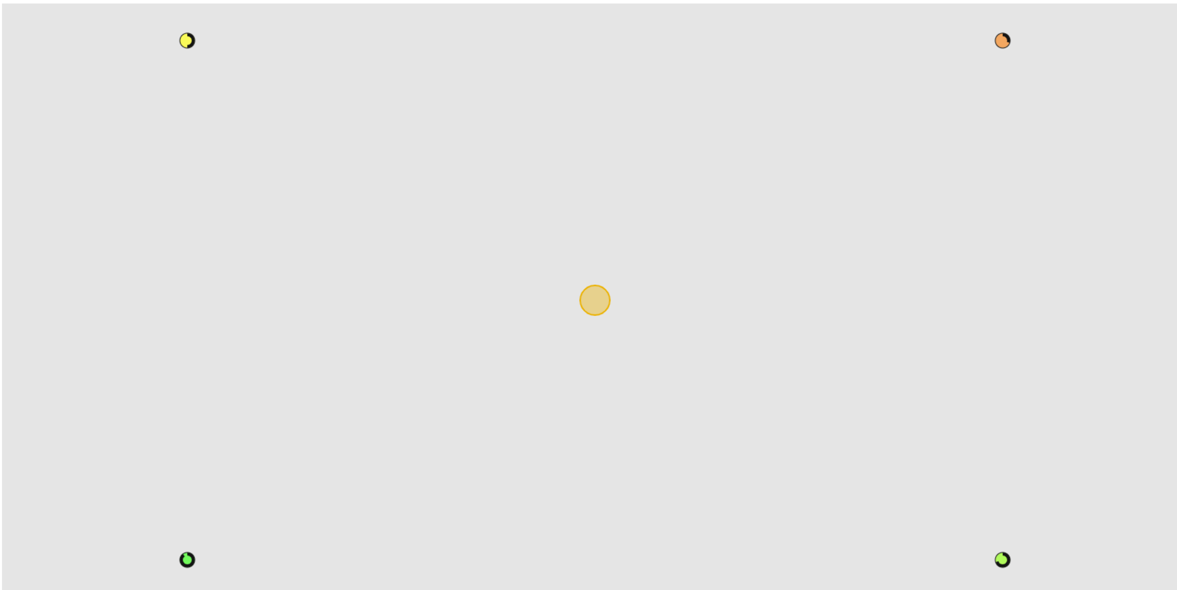


Figure 5.1: Environment 1 in agent quantity experiment.

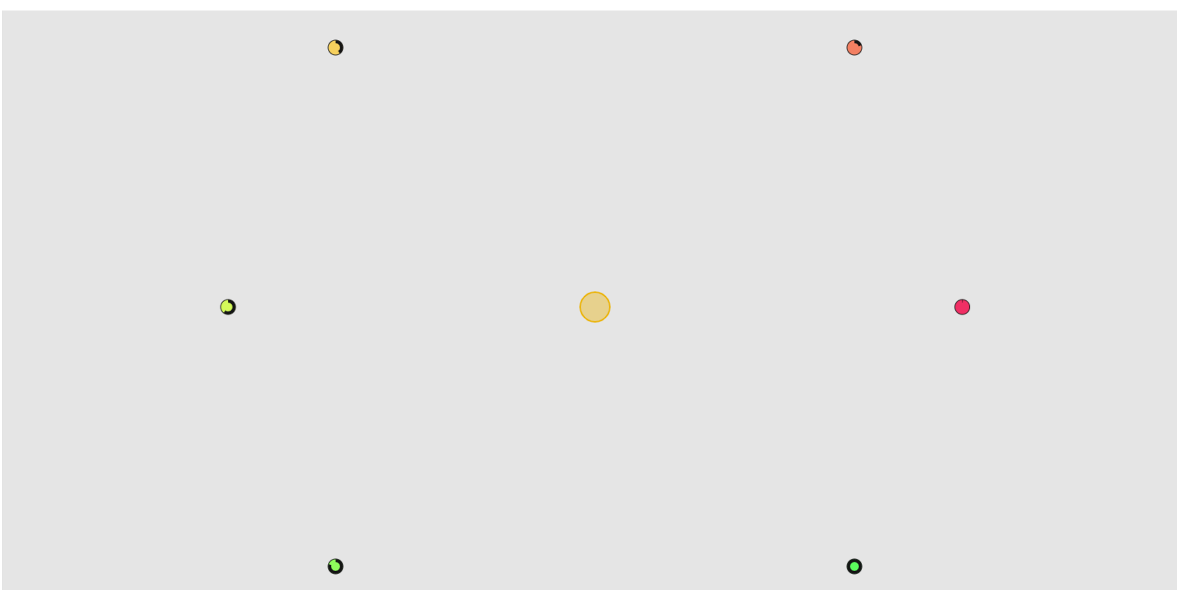


Figure 5.2: Environment 2 in agent quantity experiment.

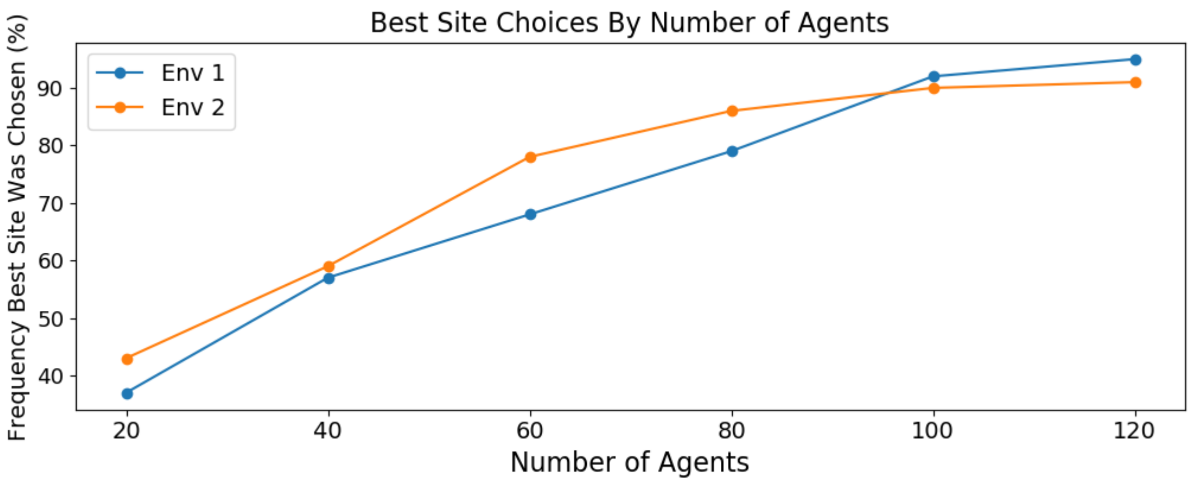


Figure 5.3: Site choices for two different environments, with equidistant sites of various qualities, and six difference numbers of robots.

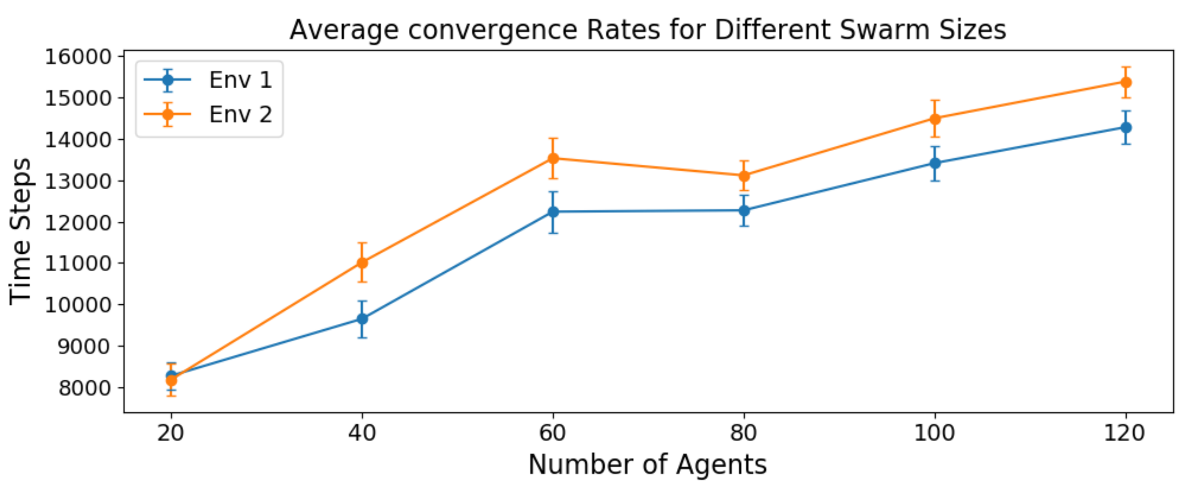


Figure 5.4: Convergence rates for two environments and six different numbers of robots.

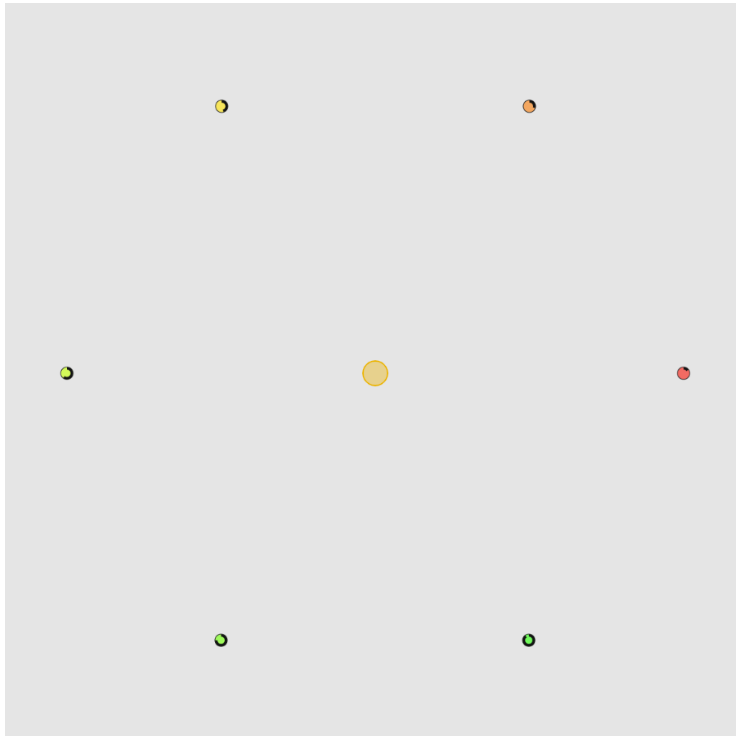


Figure 5.5: Environment with 6 sites with qualities evenly distributed between 0.01 and 0.9.

5.1.2 Effects of Site Distribution

The distribution of sites throughout the environment strongly affects the performance of the swarm. Three particular characteristics of site distribution that we noticed regard the number of sites in the environment, the distance the sites are from the hub, and if sites are *blocked* (by line of site to the hub) by other sites. We tested the swarm in environments with 6, 12, 24, and 48 equidistant sites with qualities evenly distributed between 0.01 and 0.9. Environments with 6 sites and 48 sites are shown in Figures 5.5 and 5.6 respectively. Each experiment was performed with 50 robots, and consisted of 100 simulations. The results, given in Figures 5.7 and 5.8, show a steady increase in convergence time and a steady decrease in best site selection as the number of sites increases.

We also varied the distances of sites from the hub. In one environment (Figure 5.9) we placed 20 sites evenly distributed around the hub and at equidistance from the hub. In a second environment (Figure 5.10), we moved the best site 150% farther from the rest of the

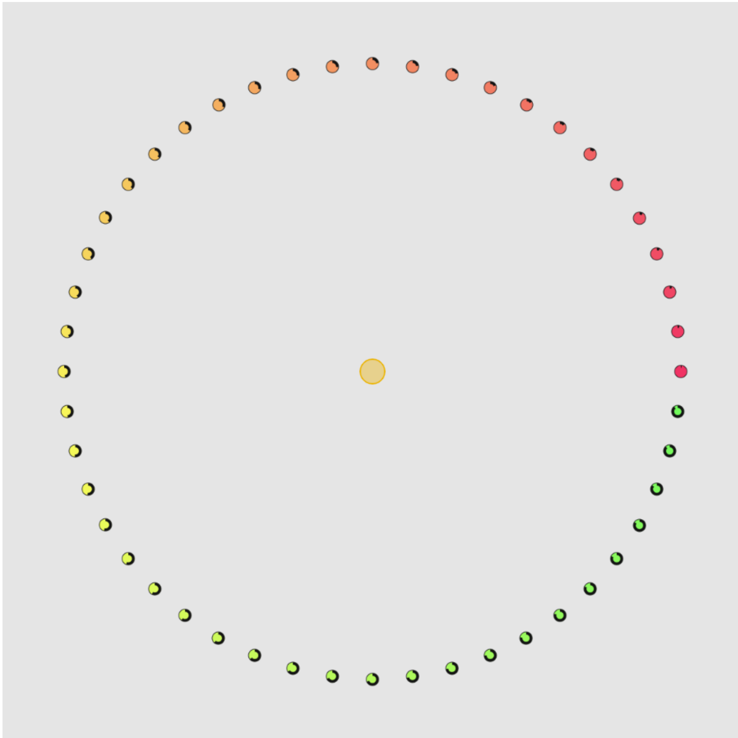


Figure 5.6: Environment with 48 sites with qualities evenly distributed between 0.01 and 0.9.

Figure 5.7: Swarm site selection results for different numbers of sites, including second and third best site selection.

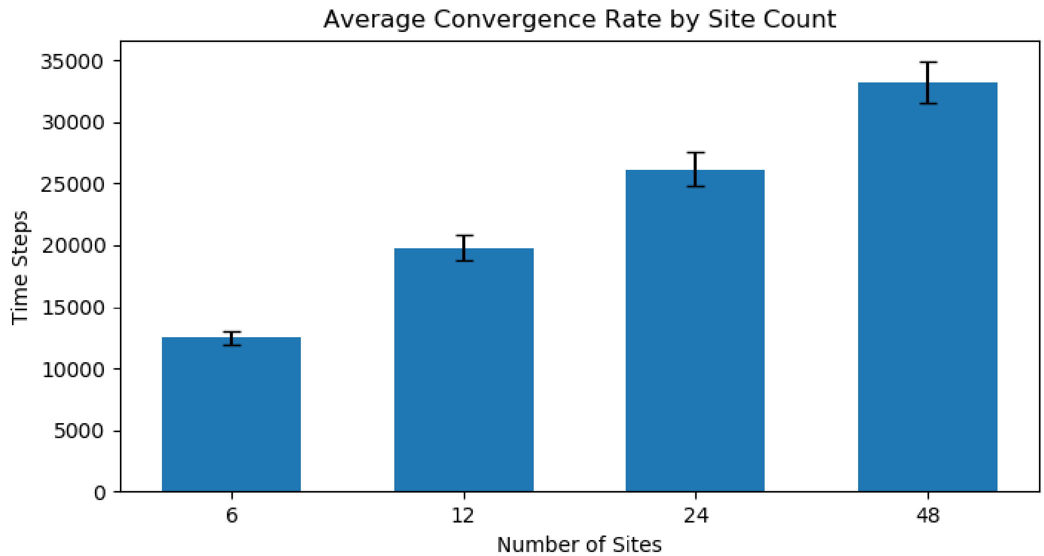


Figure 5.8: Swarm convergence rate results for different numbers of sites, including second and third best site selection.

sites. A swarm with 100 agents exhibited a 52% decrease in best-site selection in the second environment compared to the first, and a 49% increase in average convergence time.

We did a similar test with the environment in Figure 5.11, with and without the (yellow) *blocking* sites, which obscure the higher-quality sites from the hub. This makes the higher-quality sites more difficult for the agents to find since an agent returns to the hub once it encounters a site to report its quality. The results are given in Figures 5.12 and 5.13. In all, the blocking sites reduced the frequency of best site selection from 95% to 55%, and increased the average convergence time by 26%.

We see that when these environment characteristics are combined in more complex environments can drastically effect swarm performance. For example, results for site selection in the environments displayed in Figures 5.14 and 5.15 are given in Figure 5.16, and show that, despite consistent selection of the top three quality sites, best site selection performance for the swarm alone is less than 25%.

These initial results illustrate that this swarm, with a sufficient number of agents, often identifies the best site in the environment under ideal environmental conditions. However,

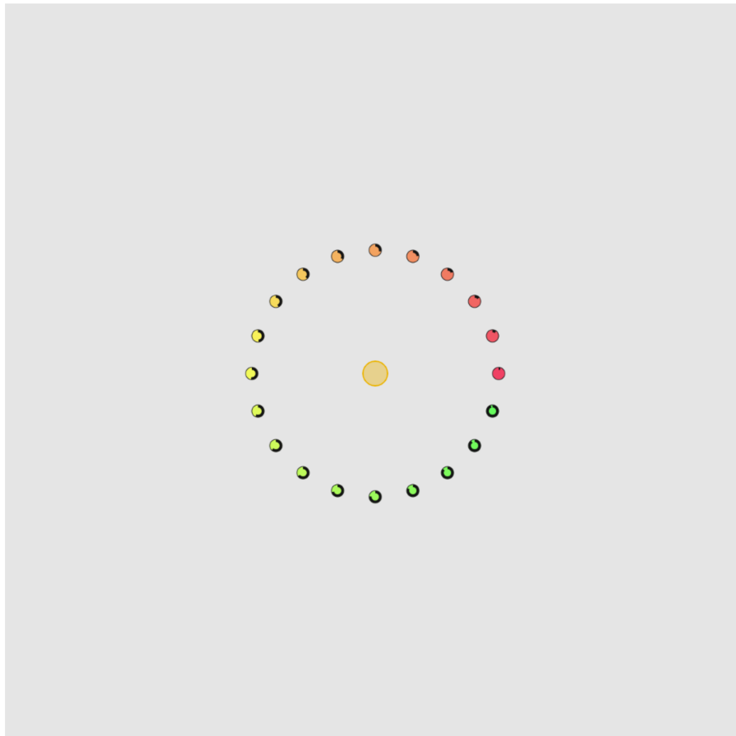


Figure 5.9: 20 equidistant sites with qualities evenly distributed between 0.01 and 0.9.

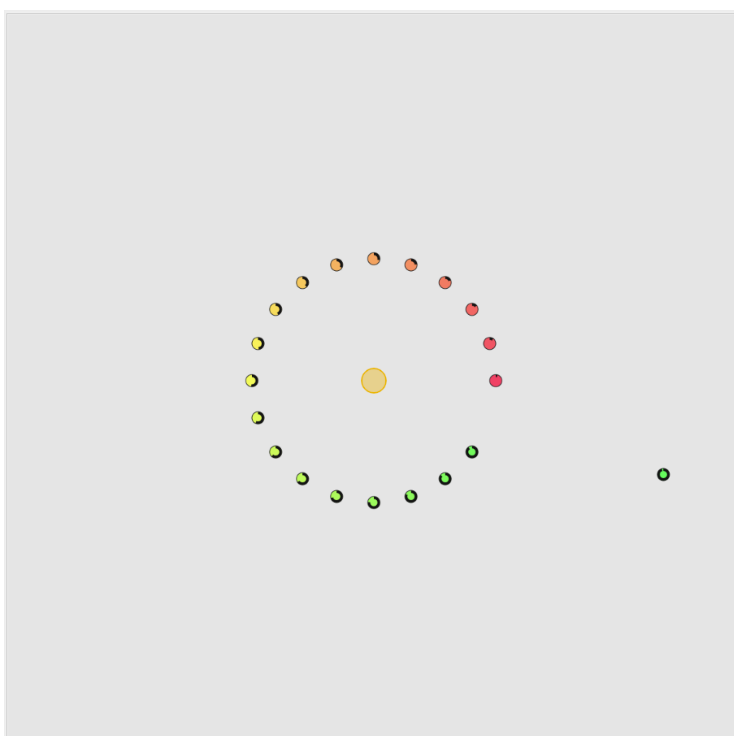


Figure 5.10: Environment in Figure 5.9, with the best site 150% farther from the hub.

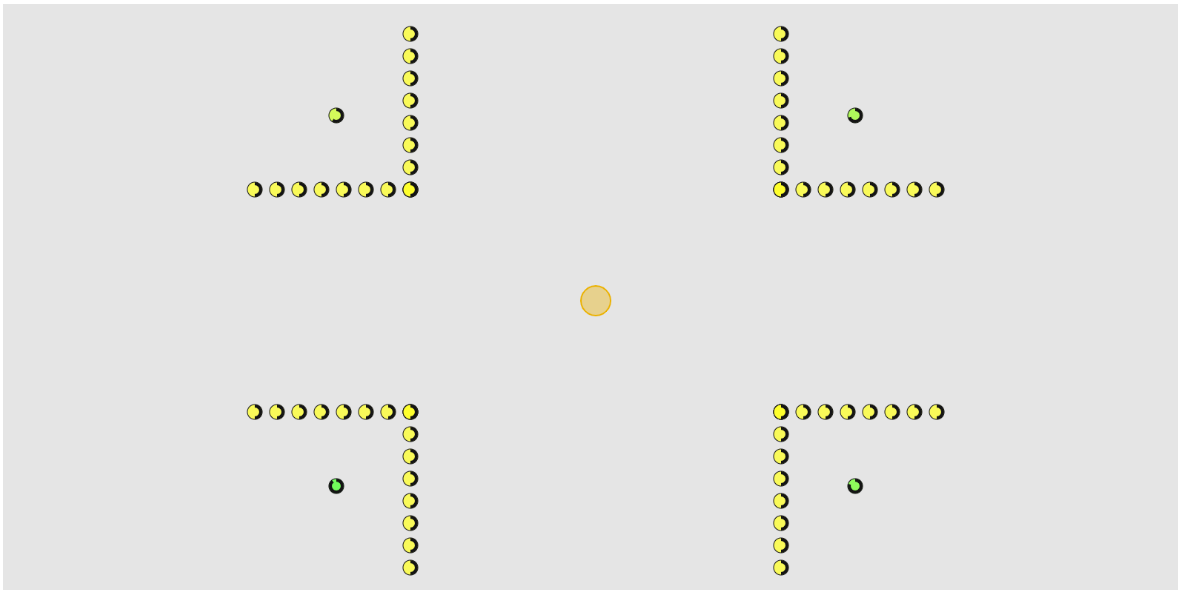


Figure 5.11: Environment with higher-quality (green) sites obscured from the hub by lower-quality (yellow) sites.

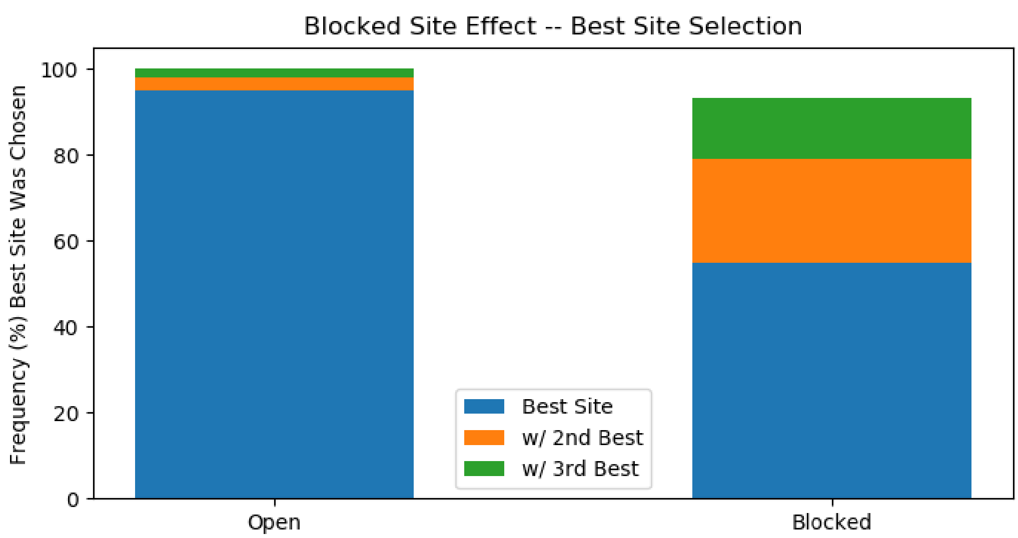


Figure 5.12: Swarm site choice results for the environment in Figure 5.11 with and without the yellow "blocking" sites.

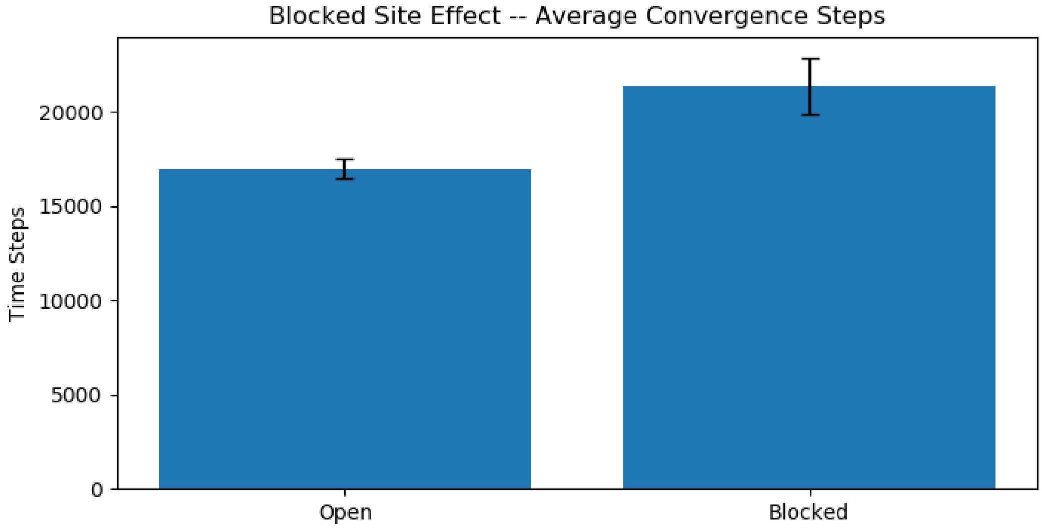


Figure 5.13: Swarm convergence results for the environment in Figure 5.11 with and without the yellow “blocking” sites.

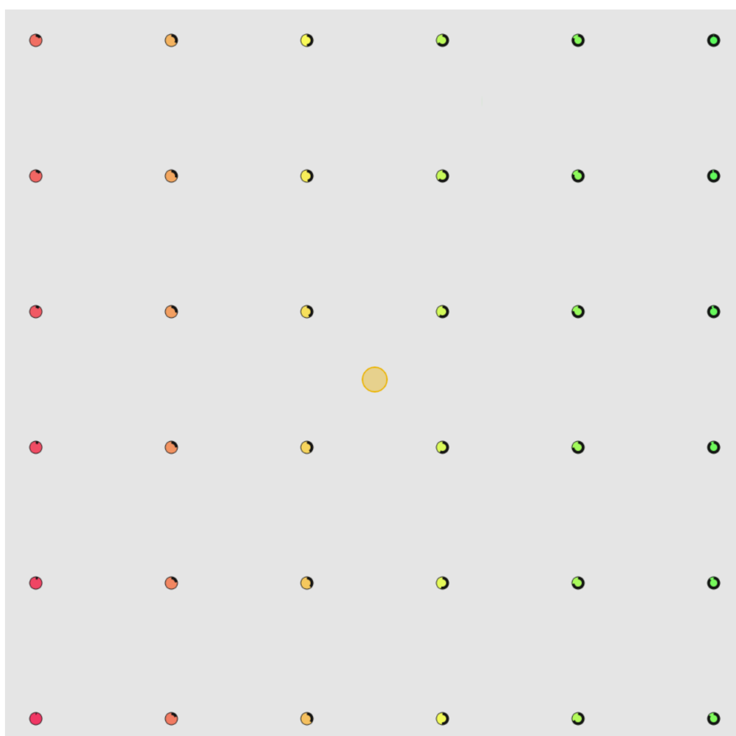


Figure 5.14: Complex environment with sites in a grid pattern. Site quality increases from left to right and from bottom to top.

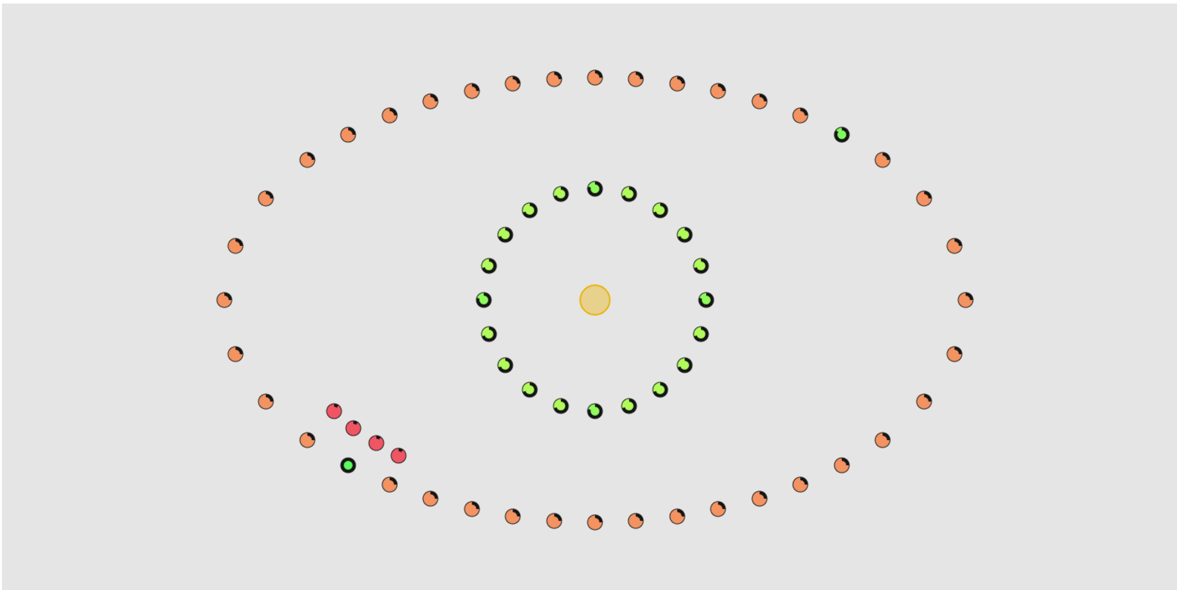


Figure 5.15: Complex environment with the best site blocked by poor quality sites in the bottom left, the second best site in the top right, and 0.7 and 0.8 quality sites closely surrounding the hub.

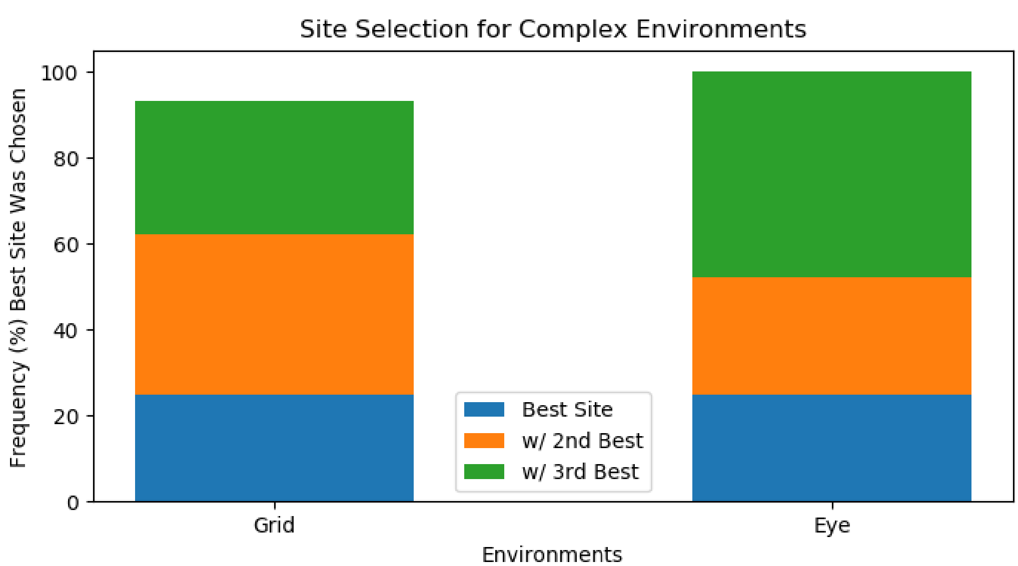


Figure 5.16: Swarm site choice results for the environments in Figure 5.14 and Figure 5.15.

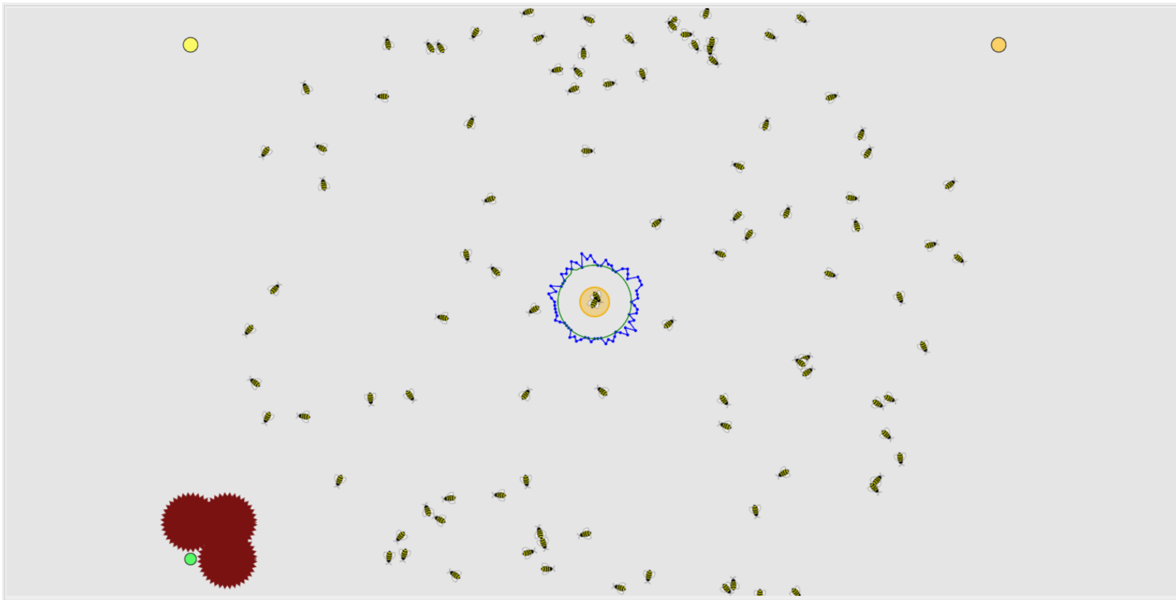


Figure 5.17: Environment with environmental hazards (traps) around the highest quality site.

when the swarm has lower numbers of agents or the environmental conditions are not ideal, the swarm often fails to find the best site.

5.2 Early Results

To justify that moderating human influence could be valuable, we tested a scenario where we use Attractors 0 and 1 in an environment with potential hazards. We implemented environmental hazards we call "traps" in the environment, which kill or destroy agents that enter them. We then simulated the effect of an operator providing poor input to the swarm by attempting to attract agents to traps.

We experimented with this behavior in the environment shown in Figure 5.17 with 100 agents. The best site in the environment is the lower left site, which is essentially surrounded by traps. Because assessing agents travel straight from the site to the hub, it is impossible for agents to go from the site to the hub, or to choose it. However, we assume that the operator, with knowledge of the sites but not the traps, consistently places beacons over the

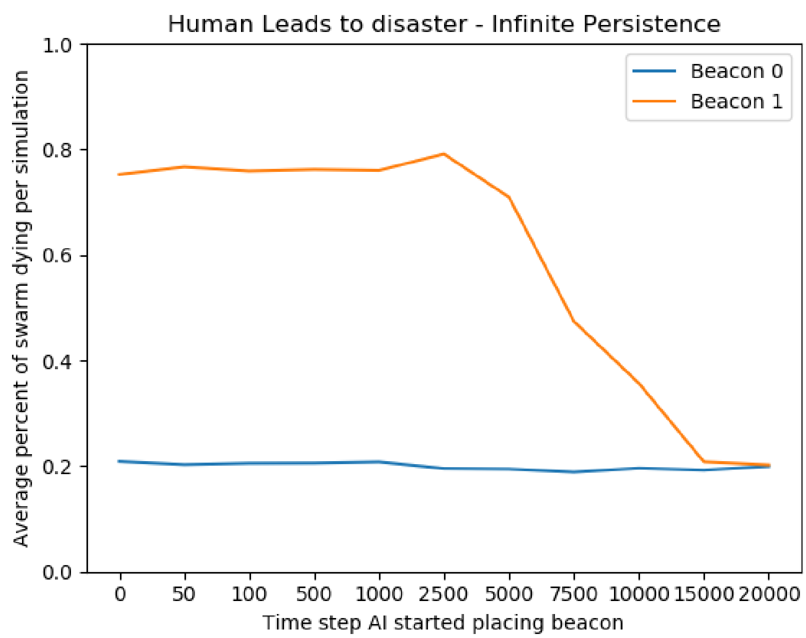


Figure 5.18: Regardless of when the beacon begins being placed with Attractor 0, only about 21% of the swarm agents enter the trap. However, with Attractor 1, if the beacon is placed before the swarm has essentially converged, up to 80% of the agents enter the trap, at which point the simulation terminates.

best site from some point in the simulation until the simulation ends (we called this *Infinite Persistence*). The results are given in Figure 5.18.

In this experiment, higher influence (Attractor 1) leads the simulated operator to usually kill the entire swarm instead of choosing a lower quality, but obtainable, site, while the interaction scheme with lower influence (Attractor 0) only resulted in the loss of a few agents. We further argue that our swarm was not designed to deal with environmental hazards, so this environment violates an assumption of the Designer, and poor Operator input with higher influence made the consequences worse than the same Operator behavior with lower influence. Thus, it seems that the lower-influence setting may be preferred.

However, in other simulations we observed that Attractor 0 provided too little influence to allow the Operator to explore the environment and push the swarm to a better site, even if the Operator knew of one that the swarm did not. This is mostly because the beacons only affect agents in the exploring state, which becomes rare as the simulation progresses and the agents spend most of their time assessing sites, dancing, and observing. On the other hand, Attractor 1 provided enough influence to allow the Operator to force the swarm to choose a sub-optimal site, even when the swarm was assessing the optimal one. One example of this is shown by the results in Figure 5.19, which consists of the simulated Operator dropping beacons on a single site for the duration of the simulation, done 100 times for each site. This Operator behavior may occur if the site quality information on the UI is incorrect, and either there is no feedback from the swarm, or swarm feedback is ignored. Because of the simplicity of the environment (Figure 4.3, all four sites are found 100% of the time. However, with Attractor 1, the swarm always chose the site on which the Operator placed beacons.

5.3 Increasing Performance By Moderating Influence

In context of the previously defined parties of interest, we can interpret the previous results as the Problem Holder desiring a swarm that discovers and selects the best quality site in an environment, the Designer used a modified honey bee model to develop a swarm to do that,

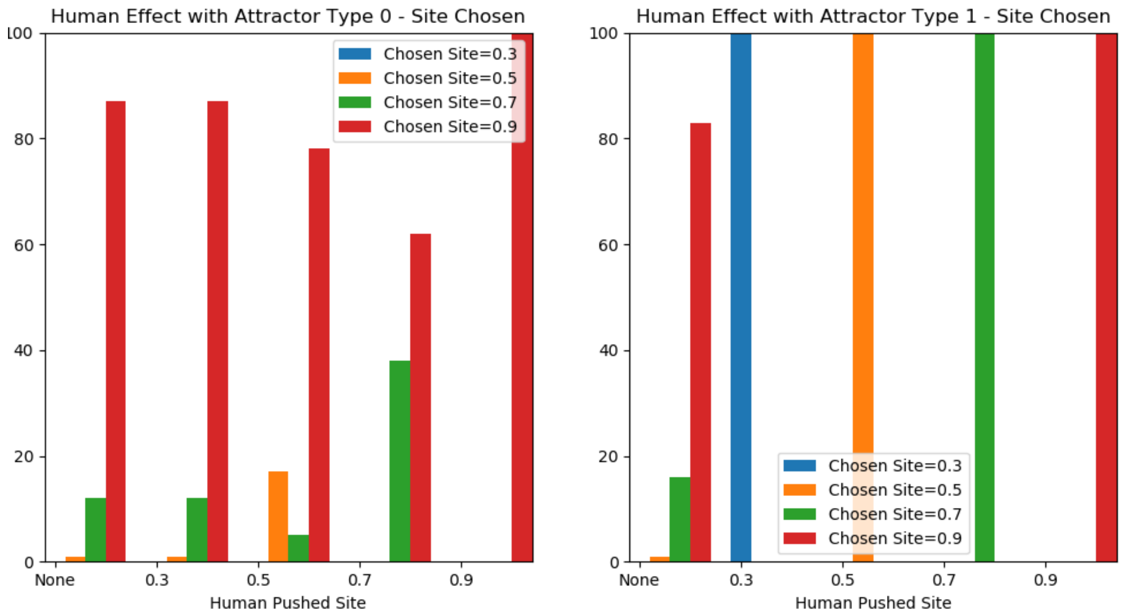


Figure 5.19: Site selection results with the Operator consistently placing beacons on a single site for the whole simulation.

and the Operator uses beacons to assist the swarm and increase its performance. However, an assumption of the Designer was violated by the environment containing hazards, and the Operator, through poor decision making or lack of information, instead of increasing the swarm's performance, decreases it. As previously mentioned, two approaches to overcome this problem would be to further train the Operator to better decision making and plan for environmental hazards, or to augment the UI in some way as to achieve the same effect. Further, one may argue that the Designer could attempt to make changes to the swarm itself in order to overcome these problems. In this case, with such a simple swarm and simple task, it would likely be easy to do any of these, but we would like to attempt to increase performance by instead simply modifying the interaction mechanism between the Operator and the swarm.

There are various ways to modify the Operator's influence over the swarm, including setting beacon delays, beacon durations, and the number of beacons the Operator is allowed to use. We experimented with all of these, but also tried implementing our proposed principle,

Distance	Number of Agents
> 400	5
$(300, 400]$	4
$(200, 300]$	3
$(100, 200]$	2
< 100	1

Table 5.1: Number of agents the IVAM allows to leave the hub in one direction, based on the distance the beacon is placed from the hub. Note that distance is unit-less.

Feedback Based Dynamic Influence, by implementing software that monitors agent information as they enter and exit the hub, and then dynamically adjusts what actions the Operator can take and how the agents respond to them. The result is a constant varying of influence over the swarm depending on the information reported by agents and the actions being taken by the Operator, but intended to keep the influence balanced, or in the sweet spot in Figure 3.1. We call this software the *Influence Verification and Adjustment Module*, or *IVAM*. While we tried various IVAM behaviors, the primary behavior we chose the IVAM to exhibit was to only allow a certain number of agents to leave the hub in a given direction when using Attractor 1, until agents leaving in that direction return to the hub. The number of agents allowed to leave in a direction depends on the distance from the hub the beacon is placed, and is given in Table 5.1. The different number of agents based on distance is to compensate for sites being near to the hub. Beacons placed near the hub are easily found by any agents leaving the hub, resulting in higher influence for close beacons than for farther, more difficult to find beacons. We believe that, by restricting agents in this way, we provide enough influence to the Operator to provide good input at any distance, but restrict the influence enough that the swarm will overcome poor input, even for nearby, low-quality potential sites.

5.3.1 Preliminary IVAM Results

Note that the IVAM immediately fixes the problem with environmental hazards by behaving like Attractor 0 once a few agents die in response to Operator input. So for environments in which the swarm would overcome the traps and choose a site without poor Operator

Figure 5.20: Results for four-sited, simple environment with IVAM algorithm, for various delays between beacon placement.

input, with the IVAM, they will now usually choose a site even with poor Operator input. Additionally, we repeated the simulations for the environment in Figure 4.3, in which we simulate the Operator stubbornly pushing the swarm to a single site, for each site in the environment. The results, shown in Figure 5.20, show that, despite the Operator attempting to force the swarm to choose poorer quality sites, the swarm more often than not chooses the best site. Thus, the IVAM balances influence, supporting the desires of the Problem Holder, without Designer modifications to the swarm, and despite poor Operator behavior.

Chapter 6

User Study

While these results are promising in simple environments with simulated operators, the success of the IVAM depends on its performance with actual human operators, and in more complex environments. Therefore, we perform a user study to confirm our results. We describe the user study in this chapter, and provide the results in Chapter 7.

6.1 User Study Design

In order to further compare types of influence, we define new interaction schemes that arguably have different levels of influence, making *influence* the first independent variable in our study. We also include an additional independent variable that we call *information*. Information refers to the accuracy of the site information shown on user interface. By displaying incorrect site information, we force the human to rely more on agent feedback or potentially provide input that is detrimental to swarm performance. Even with feedback, the incorrect information can confuse human operators, and impact their decision making. Thus, information improves our ability to test human performance under various conditions and with poor human input.

For each independent variable, we designate three possible values. The values for influence are **High**, **Low**, and **IVAM**, while the three possible values for information are **Perfect**, **Missing**, and **Mislabeled**. The values of influence, which will be defined later, vary between subjects, while the values for information, defined in Table 6.1 vary within subjects. The result is a 3x3 user study.

Information Type	Description
Perfect	The information displayed to the user is the same as the true environment.
Missing	Some sites are not displayed to the user, this always includes the best quality site in the environment. Qualities of sites shown are accurate.
Mislabeled	All site locations are shown accurately, but most or all site qualities are incorrectly displayed.

Table 6.1: Description of information settings.

Each user's experience consists of three simulated deployments. Each deployment is in a different environment and is displayed with a different type of information. However, the interaction scheme (influence type) is constant for all three. Environments are designed to be difficult for the swarm to find the best site, but include patterns that the human can potentially recognize and use to provide information outside of the swarm's sensing and reasoning capability (environments are shown in the Appendix). This helps provide the Operator with a purpose in our study. Each simulation is given a time step limit of 65,000 time steps and ran at 70 time steps per second, resulting in a time limit of 15 minutes and 28 seconds per simulation. If the swarm does not converge before the time limit, it is recorded as a failure to converge.

While each user experiences the same environments in the same order, we alter the order of information types for the environments between users. Three types of information allow for six possible orderings of information, and with the three influence types, there are a total of 18 combinations of influence-information orderings for the three environments. We generate the user study by creating all 18 combinations, and then pseudo-randomly assigning one to each of 18 users. Doing this twice results in a total of 36 users required to complete the study, and 108 individual data points.

We have each user use Attractor 1, and isolate the differences between influence variables to the number of beacons allowed to be placed and whether or not the IVAM is

Number of Agents	100
Simulation Speed	70 ts/s
Simulation Duration	65,000 ts
Beacon Type	Attractor 1
Beacon Radius	75
Beacon Duration	500 ts ($\approx 7.1s$)
Probability Agents Ignore a Beacon	0.2

Table 6.2: Constant simulation parameters across all variables. Note that ts stands for "time steps," and s for "seconds."

enabled. Table 6.2 shows the default simulation parameters for all interaction schemes, while the specific influence variables are set as follows:

- **High**: 8 beacons may be placed simultaneously.
- **Low**: 1 beacon may be placed at a time.
- **IVAM**: 1 beacon may be placed at a time, and IVAM is enabled.

We argue that the three influence types are obviously different, and are shown above in descending order of influence, i.e. High provides the Operator with the most influence, and IVAM with the least. However, because of the IVAM's design, we hypothesize that it will provide enough influence to allow the human to help the swarm when it receives useful input, and sufficiently reduce influence so that the swarm can overcome poor human input.

6.2 Assessing Performance

We continue to assume that the performance measure the Problem Holder has designated is whether or not the best site is chosen for a given deployment. Therefore, for the user study, our primary performance measure is the ratio of simulations where the best site was chosen to the total number of simulations ran for each influence-information pair. We also consider the same ratio for the influence variables alone.

In addition to the above performance measure, we record swarm agent locations, beacon placement, best site indicator information, and convergence times. In Chapter 7, we use this information to perform a qualitative analysis of user behavior in cases of non-convergence, and examine performance in terms of near-best-choice selection, beacon usage, and convergence times.

6.3 Expected Results

To support our theory, the IVAM needs to demonstrate that it appropriately balances influence for general environments and Operator behaviors. Our user study incorporates multiple environments with differing information accuracy, and behaviors from 36 different human operators. If the performance measures previously described are higher for the IVAM influence type than for the others, this suggests that the IVAM successfully and appropriately keeps Operator influence balanced for our human-swarm system, and provides evidence for our theory. Otherwise, the IVAM is providing too much or too little influence to the Operator, and our theory and implementation will require further examination.

6.4 Algorithmic Considerations

While the IVAM is meant to implement *Feedback Based Dynamic Influence*, because we attempted to isolate differences in influence, some IVAM functionality was omitted or integrated into the UI for all influence types. For example, originally the Operator could set the size of the placed beacons and the IVAM would restrict the size options based on Operator choices and the state of the swarm. But for the user study, we fixed the size of beacons and eliminated the need for that functionality. The end result is the IVAM behavior restricted to only what was given in Chapter 5.3.

6.5 Simulation Results

Before running the user study with human Operators, we elected to test the user study with simulated operators. To do this, we generated the user study for 90 users, using the same

Environment	Ratio	Probability
1	39/100	0.39
2	38/100	0.38
3	26/100	0.26
Overall	103/300	0.34

Table 6.3: Baseline results for swarm choosing best site, by environment and combined.

methods previously described, and implemented two simulated operator behaviors. We refer to the two operators as *AI 1* and *AI 2*, and assign them the following behaviors:

- AI 1 (poor input): Place beacons on the best site given in the initial information until convergence or the end of the simulation.
- AI 2 (better input): Keep a belief about the best site in the environment, starting with the information provided at the beginning of the simulation, and update that belief based on swarm agent reports. If at any time the best known site indicator reports a site with a higher quality than the currently believed best, update the best known to that marked by best known site indicator. Drop beacons on best known site until convergence, or until the simulation ends.

Both AIs begin beacon placement 150 time steps (about 2.1 seconds) into the simulation. AI 1 then attempts to place a beacon every 150 time steps for the rest of simulation, and AI 2 attempts to place beacons at the same rate until 30,000 time steps have past, at which point it attempts every 225 (about 3.2 seconds). This slightly varies the delay between beacon expiration and beacon placement for the agents.

We also used simulation to establish baseline performance for the swarm without human input. Those results are provided in Table 6.3. Results for the user study with the two simulated operators are given in Table 6.4, and Table 6.5. Again, the performance metric we used was whether or not the best site was chosen in each simulation.

For AI 1, in the case of Perfect information, the persistence of the operator easily drives the swarm to the best site for all influence types, with the exception of one case with

Infl\Info	Perfect	Missing	Mislabeled	Totals
High	30/30 (1.0)	0/30 (0.0)	0/30 (0.0)	30/90 (0.33)
Low	30/30 (1.0)	0/30 (0.0)	1/30 (0.03)	31/90 (0.34)
IVAM	29/30 (0.97)	7/30 (0.23)	11/30 (0.37)	47/90 (0.52)

Table 6.4: Simulated user study results for AI1.

Infl\Info	Perfect	Missing	Mislabeled	Totals
High	20/30 (0.67)	11/30 (0.37)	10/30 (0.33)	41/90 (0.46)
Low	22/30 (0.73)	6/30 (0.2)	10/30 (0.33)	38/90 (0.42)
IVAM	28/30 (0.93)	11/30 (0.37)	18/30 (0.6)	57/90 (0.63)

Table 6.5: Simulated user study results for AI2.

IVAM. However, in the cases of both Missing and Mislabeled information, the IVAM allowed the swarm to overcome the operator's push towards a poor quality site and choose the best one in 18 out of 60 simulations, or 30%, which about as well as the swarm performs without any Operator input according to Table 6.3.

AI 2's input is better, improving performance over AI 1 with both incorrect information types and with both High and Low influences, but still imperfect. AI 2 caused the swarm to choose poorer quality sites in 1/3 of the simulations with Perfect information and High and Low influences. With IVAM influence, AI 2 performed almost just as well as AI 1 with Perfect information, and did just as well or better than High and Low influence and incorrect information. We believe this is this case for two reasons. The first is that the IVAM prevented the swarm from overreacting to the excessive input from AI 2, as excessive input can cause the swarm to fail to converge within the time limit (further discussed in Chapter 7). AI 2 failed to converge in almost 1/3 and 1/4 of the simulations ran with High and Low influences, respectively (see Table 7.6). The second reason is that the IVAM allowed the swarm to maintain focus on the best quality site, even when the operator did not believe it was the best and focused on another.

In both cases the results suggest that the IVAM algorithm helps maintain or increase swarm performance compared to the other two influence types. This is a good indication that the IVAM algorithm is balancing influence as desired. Not only does it solve the problem with traps, but it also allows the swarm to take advantage of useful input, and at least somewhat overcome poor input in the absence of environmental hazards. These results are encouraging, and provided us with sufficient motivation to continue with the user study.

6.6 Format

The user study consisted of a 15 minute slide-show presentation informing the user of the purpose of the study, how the user would contribute, an explanation of the swarm model and the user interface, instruction on how beacons worked and how to place them, and how to understand the various types of agent feedback displayed on the interface. The task they were instructed to perform was to *assist the swarm in finding and choosing the best quality site in the given environment*. This instruction was provided once at the beginning of instruction, and once at the end. Users were informed about the various sources of imperfection in the system, particularly noisy agent reports, that the radial display was restricted to pointing to a finite number of directions, and that the site information displayed may be inaccurate. They were also informed about the time limit, and instructed that the simulation would end in failure somewhere between the 15 and 16 minute mark. At the end of the instruction they were also provided with an overview of the whole study process and instructed that they were allowed to ask questions at any time during the study.

After instruction, users were asked to complete a short questionnaire, after which they began the study with a practice simulation. In this simulation, the users were allowed to interact with the swarm using their assigned influence type, in the environment shown in Figure 4.3, with all agents shown, and with the assurance that the environment they were in was displayed correctly. After the practice simulation, each user performed the three simulations in their assigned information order, and then filled out a post-study questionnaire.

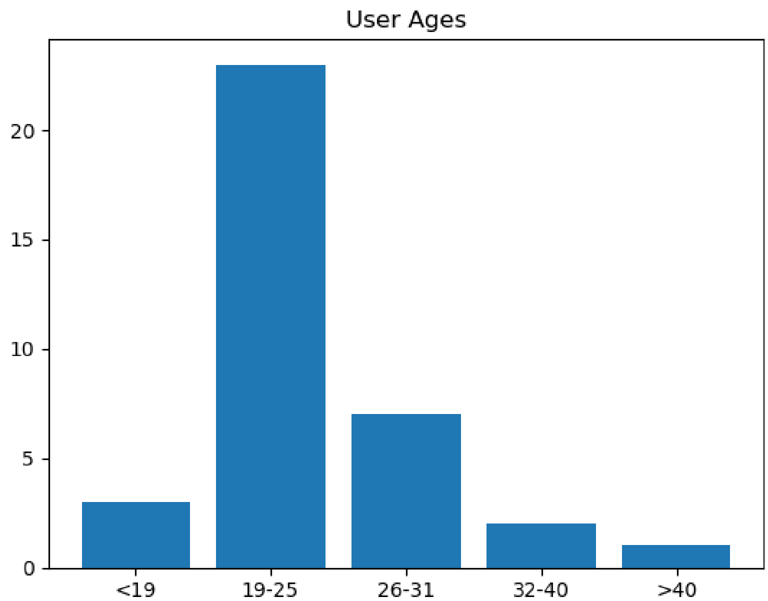


Figure 6.1: Distribution of user ages.

6.7 User Demographics

36 users were recruited via campus advertising and word of mouth. Most participants were students from the BYU community. Out of all participants, 30 were students, 20 were male, 16 were female. 10 of the students were in Computer Science or Computer Engineering, while the rest came from various other fields ranging from open major to Physiology and Developmental Biology. The age demographics are given in Figure 6.1.

Chapter 7

Analysis

In this chapter, we provide and discuss the results of the user study described in Chapter 6. First we provide the immediate results using the measure provided by Problem Holder as described in Chapters 4 and 6, then we provide a brief statistical analysis of those results and some qualitative analysis of user behavior. We finish the chapter by analyzing the results using decision trees, and considering some alternate measurements that are also mentioned in Chapter 6.

7.1 Primary Results

The direct results of the study are shown in Table 7.1. We observe that the IVAM was outperformed for each information type, and overall, compared to the other influence types, which contradicts our hypothesis. In the case of Perfect information, it seems that the IVAM was perhaps too restrictive, or too low influence, as Low influence outperformed it. For both imperfect information types, High influence performed the best. This suggests that High influence is the best choice for imperfect information, Low is for Perfect information, and IVAM should not be used at all. Although the IVAM influence dealt effectively with the challenges discussed in Chapter 5 and seemed to perform best with the simulated operators, it ultimately failed with human operators and the current user interface.

Infl\Info	Perfect	Missing	Mislabeled	Totals
High	6/12	8/12	11/12	25/36
Low	11/12	4/12	7/12	22/36
IVAM	9/12	4/12	5/12	18/36

Table 7.1: User study results.

Infl\Info	Perfect	Missing	Mislabeled	Totals
High	0.5	0.67	0.92	0.69
Low	0.92	0.33	0.58	0.61
IVAM	0.75	0.33	0.42	0.5

Table 7.2: User study results in decimal notation.

7.1.1 Comparison to Baseline and Simulation Results

Displaying the results in Table 7.1 in decimal notation, we can easily compare swarm performance with human operators with swarm performance alone (Table 6.3). Over all possible environments, we estimate that the frequency that the swarm converges to the best site is about 0.34. Table 7.2 shows that the human-swarm system chooses the best site greater than or equal to 33% of the time, suggesting that the human operators, on average, allowed the swarm to converge just as often or more so than the swarm would alone. The cases where the human-swarm system failed to perform better are those where Missing information was combined with Low and IVAM influence types. We speculate why this may be the case in Section 7.4.

We may repeat the above analysis with the simulated results in Chapter 6. The decimal notation results for the two simulated behaviors were already given in Tables 6.4 and 6.5, but are shown together in Table 7.3. For Perfect information, both simulated operators improve the frequency that the best site is chosen. For imperfect information, AI 1 almost never chooses the best site unless using the IVAM influence type. With Mislabeled information, the IVAM brings the frequency slightly above baseline, and with Missing, the IVAM frequency of convergence to the best site–though higher than those of High and Low influence–is only 23%, much less than baseline. In other words, AI 1's behavior is terrible for the swarm when

Infl\Info	Perfect	Missing	Mislabeled	Totals
		AI 1		
High	1.0	0.0	0.0	0.33
Low	1.0	0.0	0.03	0.34
IVAM	0.97	0.23	0.37	0.52
		AI 2		
High	0.67	0.37	0.33	0.46
Low	0.73	0.2	0.33	0.42
IVAM	0.93	0.37	0.6	0.63

Table 7.3: Simulated operator results in decimal notation.

Effect	Num DF	Den DF	F Value	Pr > F
Influence	2	66	1.26	0.2906
Information	2	66	3.55	0.0344
Infl*Info	4	66	3.02	0.0237

Table 7.4: Type III Test of Fixed Effects results.

information is poor, and AI 2's seems to improve or maintain performance over baseline–even with imperfect information–in every case but one; that of Missing information combined with Low influence.

Comparing these results with those of the human operators suggest that the human operators likely did not behave like AI 1 very often, if ever, and produced a somewhat opposite pattern of results as AI 2. The only similarity seems to be with Missing information and Low and IVAM influence, where both AI 2 and the human operators perform poorly.

7.2 Statistical Analysis

We use the GLIMMIX procedure from the SAS statistical software to examine the statistical significance of our results. The results for the Type III Test of Fixed Effects are provided in Table 7.4. From these results, we observe that influence was not statistically significant alone, but information was, and that there is an interaction effect between influence and information. Therefore, influence should not be considered independently from information, but we may consider influences within each information type or between influence-information pairs.

(Infl, Info) 1	(Infl, Info) 2	t	Pr > t
(H, ML)	(H, P)	2.04	0.0453
(H, ML)	(I, ML)	2.23	0.0289
(H, ML)	(I, MS)	2.50	0.0149
(H, ML)	(L, MS)	2.51	0.0145
(I, ML)	(L, P)	-2.26	0.0271
(I, MS)	(I, P)	-2.01	0.0486
(I, MS)	(L, P)	-2.53	0.0139
(L, MS)	(L, P)	-2.63	0.0107

Table 7.5: Statistically significant differences (using $p = 0.05$) between influence-information pairs. Note that H=High, L=Low, I=IVAM, P=Perfect, MS=Missing, and ML=Mislabeled.

Out of the 36 possible pairwise comparisons, only eight show statistically significant differences (i.e. $p \leq 0.05$). Those significant comparisons are given in Table 7.5, but most are easy to identify from the raw results simply by noting that High influence with Mislabeled information and Low influence with Perfect information scored very high compared to other influence-information pairs. The only slightly less obvious comparison is IVAM influence with Missing information and IVAM influence with Perfect information.

7.3 Non-convergence Analysis

Examining our data shows that a significant number of failures to choose the best site were caused by the swarm failing to converge in the allotted time. This is the case with both the simulated operators and the human operators. The exact results are given in Table 7.6. In the case of the simulated operators, the results are not particularly surprising, as the AIs react very quickly and are always placing beacons. For these cases, the IVAM appears to improve performance by allowing the swarm to more frequently converge to a site in the presence of excessive operator input. However, in the case of human operators, IVAM influence had practically the same number of failures from non-convergence as High influence. This further suggests that the IVAM is failing to appropriately balance influence, as we expect an appropriate balance to prevent failure due to non-convergence.

Infl\Info	Perfect	Missing	Mislabeled	Totals
		AI 1		
High	0/30	0/30	10/30	10/90
Low	0/30	0/30	12/30	12/90
IVAM	0/30	3/30	5/30	8/90
		AI 2		
High	10/30	7/30	8/30	25/90
Low	8/30	4/30	5/30	17/90
IVAM	1/30	3/30	0/30	4/90
		Human		
High	4/12	2/12	1/12	7/36
Low	0/12	0/12	1/12	1/36
IVAM	1/12	3/12	2/12	6/36

Table 7.6: Non-convergence counts for influence and information variables.

In order to better understand this, we reviewed the data saved from the simulations in which the failures occurred. As part of the study, we recorded agent locations, beacon placement, and best site indicator information at each time step, which we can play back to observe user behavior. In most cases of High influence, excessive input was indeed the culprit of the failure, whether by consistently placing beacons over the best site or by consistently sending agents to assess multiple sites. This was particularly a problem in Environment 2 (A.2) where centered beacon placement over the best site allows agents to assess the surrounding six sites as well as the best one, making it difficult for the swarm to meet their threshold value of assessors at a single site. However, while some of the non-convergence cases with the IVAM were caused by having the agents consistently assess too many sites, it appears that a few were the result of the user not having enough power to push the swarm to decide between two equal, or near equal, quality sites. This occurred when the best site found was equal, or near equal, in quality to the next best site. The swarm seemingly reached a type of equilibrium between the two sites, and would not reach the convergence criteria for either before the time limit. With the IVAM influence, even consistent beacon placement on one of the sites was insufficient to push the swarm to choose it.

Infl\Info	Perfect	Missing	Mislabeled	Totals
High	9/12	9/12	11/12	29/36
Low	12/12	4/12	7/12	23/36
IVAM	9/12	5/12	5/12	19/36

Table 7.7: Possible user study results if the users had been provided with additional training regarding when to discontinue input to the swarm.

From this we draw two conclusions. The first is while human operators are more likely to realize that too much input is a bad thing for this swarm, it is still a problem for humans that the IVAM could potentially solve. The second is that the IVAM provided too little influence to allow the human to push the swarm out of an equilibrium assessing two similar quality sites and converge to one or the other.

A final observation we noted from the non-convergence results was the number of cases where the user would likely have converged to the best site given that they knew reducing their input would allow the swarm to converge. We intentionally omitted that information from the instruction to see if the users would express this behavior, and some did. However, if that instruction had been provided, one plausible outcome would be the results in Table 7.7. This is based on whether the user found the best site in the environment, and if focus on it was maintained through the end of the simulation. The difference is most notable with High influence and Perfect information and in the totals, but the conclusion is the same; High influence performs best for imperfect information and Low for perfect information.

7.4 Qualitative Analysis

Something we noticed by examining the behavior of the users with different interaction schemes and comparing it with collected data was the utility of having multiple beacons. Users given High influence were much more liberal with their beacon use, and were more apt to explore the environment and confirm the displayed site information. Users with a single beacon (Low and IVAM) were quick to make their choice and attempt to convince the swarm to choose a site. This is particularly important because choosing the best site is strongly dependent on whether the swarm finds the best site during its initial exploration. If it does, between the agent reports and the best quality site indicator, users would notice it early in the simulation and attempt to drive the swarm there. If not, whether the best site was found was determined solely on users' exploration decisions. Thus with High influence, users were more likely to explore, and more likely to find the best site when not initially discovered by the swarm. In particular this explains the difference in performance between High influence and the other influence types with Missing information.

Another thing we observed was that most of the time taken by the users was usually spent trying to convince the swarm to make a choice, even when that choice was the best site. A legitimate question arises regarding the utility of letting the swarm make the final decision, instead of somehow expressing the confidence of the swarm's current choices and allowing the user to decide to have them converge before reaching their assessor threshold. From a practical standpoint, for most of the simulations in our study, users would have been been able to save a significant amount of time by being provided with this option, with minimal change to site choice performance.

7.5 Decision Tree Analysis

A brief but interesting analysis we performed was to train a decision tree model to our data and examine what it learned. We trained a `DecisionTreeClassifier` from `sklearn`'s [55] library with various parameters and achieved fairly consistent results. An example of the trees we trained is shown in Figure 7.1. Information is mapped to integers such that Perfect=1, Missing=2, and Mislabeled=3, while influence is mapped similarly such that High=1, Low=2, and IVAM=3. We see in Figure 7.1 that Perfect information is the top node, and that High influence is the next node in both branches. This reinforces the statistical analysis's claim that the amount of influence the Operator is given depends on the quality of information provided. It also suggests that the most important determining factors for success in our study were Perfect information and High influence. Another interesting feature from the

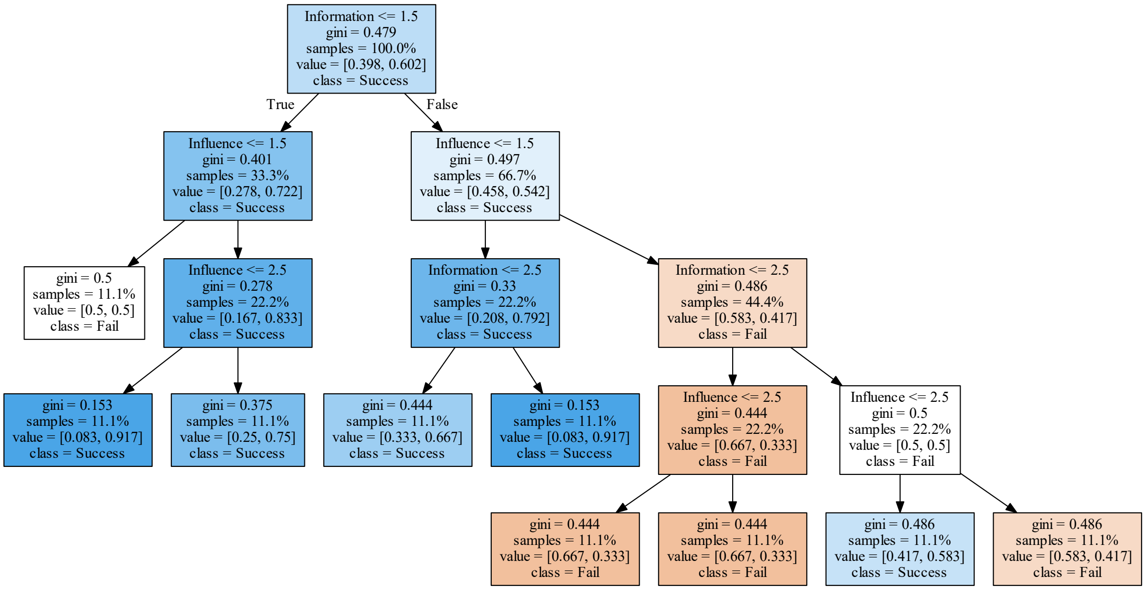


Figure 7.1: An example of a decision tree trained using our user study results.

`DecisionTreeClassifier` object is the ability to output feature importance. The `sklearn` website states that "The importance of a feature is computed as the (normalized) total reduction of the criterion brought by that feature. It is also known as the Gini importance." [56]. For the decision tree in Figure 7.1, the feature importance is roughly 0.7 and 0.3 for influence and information respectively. Using a `RandomForestClassifier` instead, with 100 trees and a max depth of four, gives a feature importance of roughly 0.57 and 0.43 for influence and information, respectively. This suggests that, while influence and information are highly related in our study, influence is possibly playing a more important role in determining the end performance outcome than information.

7.6 Other Measurements

While we assume that the Problem Holder's metric for performance is that the best site is chosen, we may also explore other reasonable measurements that indicate something regarding the human-swarm system's performance. The measurements we explore here are: scores for

Infl\Info	Perfect	Missing	Mislabeled	Totals
High	7/12	10/12	11/12	28/36
Low	12/12	9/12	9/12	30/36
IVAM	11/12	8/12	7/12	26/36

Table 7.8: User study results by counting "good enough" sites as successes.

Infl\Info	Perfect	Missing	Mislabeled	Overall
High	18.87	22.89	16.81	19.5
Low	4.92	5.09	4.80	4.94
IVAM	5.89	5.76	5.41	5.69

Table 7.9: Average beacon per minute usage for influence-information pairs.

choosing sites that are "good enough," average times for the swarm to converge, and beacon usage.

To consider near-best choice selection, we first must define what "near best" means. We do this based on the sites in each environment. For Environment 1, good enough is a site value ≥ 0.85, for Environment 2, ≥ 0.7, and for Environment 3, ≥ 0.8. With this metric, we get the results in Table 7.8. At first glance, it appears that Low influence is now the best performer, but we have also further reduced any statistical significance between influence, and influence-information pairs, so no viable conclusions can be made in that regard.

We may also consider resources used in terms of average beacons placed per minute. Table 7.9 shows the average number of beacons placed per minute for each influence-information pair. As anticipated, there is a significant difference between High influence and the other two influences since the latter were only allowed to place a single beacon at a time, while High influence was able to place 8 beacons consecutively. It may be worth noting, however, that the limit for High influence is around 67 beacons in a minute, while the limit for Low and IVAM is around 8. That the average for High is so much less than the limit suggests that users were often more conservative with their beacon placement than expected. Table 7.9 also shows that users' beacon use was higher for IVAM influence than for Low influence, suggesting that Low influence not only outperformed IVAM, but did so using fewer beacons.

Influence	Min	Max	Avg	StDev
High	2.7	14.5	6.8	3.1
Low	3.1	12.9	6.4	2.5
IVAM	3.1	13.3	6.8	2.5

Table 7.10: Convergence time results by influence type with non-convergence cases omitted. Units are in minutes.

Lastly, the average time to convergence for each influence type is given in Table 7.10, omitting the cases when the swarm does not converge. We see that there is surprisingly little difference in each of the values, perhaps with the exception of a higher standard deviation with High influence and a slightly lower average with Low influence. The only conclusion we draw from this is that users seemed to be able to converge slightly faster, on average, with Low influence than with High influence or IVAM influence. However, this also demonstrates that the IVAM is failing to outperform the other influence types.

Chapter 8

Conclusions and Future Work

As robot swarm systems become more of a reality, techniques for effective human-swarm interaction and efficient implementation of these techniques will become more important. In this work, we described a dynamic between three parties of interest in regards to the design and implementation of a robot swarm, the Problem Holder, the Designer, and the Operator, proposed a theory for cost-efficient human-swarm interaction design, and tested our theory using swarm simulation software we implemented. The theory is based on the idea that, given a performance metric from the Problem Holder, the Designer can increase or maintain performance by implementing an interaction scheme that balances the influence of the Operator according to the principle of *Feedback Based Dynamic Influence*. We demonstrated the need for balancing influence by implementing a simulated swarm an incorporating human interaction through beacon placement, and then showing how Operator input can be detrimental to swarm performance using a simulated operator. By incorporating the IVAM algorithm, an Operator influence balancing piece of software, into the the interaction mechanism, we were able to improve performance of the swarm using the same Operator behaviors.

To further test the effectiveness of the IVAM, we designed a user study where human operators would interact with the swarm and accomplish a task with different influence types corresponding to different levels of influence, including the IVAM, and different information types across difference environments. We ran the study with two simulated operators, one with exceptionally poor behavior and one with more reasonable behavior, and were encouraged by the results. Results from the user study show that the IVAM algorithm fails to appropriately balance influence with human operators, and performed worse overall than either of the two other influence levels considered. However, the results also demonstrate further the need to carefully allocate influence to each operator based on the skill of the operator and the reliability of the information provided. Further research would be required to support our theory for accomplishing this, or to discover other theories for how it should be accomplished.

8.1 Future Work

An obvious next step in this research would be to redesign the IVAM and re-test, possibly with a larger number of users, more environments, and more differences in information. We would want to further isolate influence as the independent variable, and see if the modified IVAM is maintaining or improving performance over other levels of influence. Simplifying the IVAM for our user study, as discussed in Section 6.4, also may have affected our results. Therefore, allowing more complex human interactions and additional IVAM behaviors may improve our results.

Another natural step would be to apply our theory to additional swarms and tasks, to see if we can find evidence to support it and see if it generalizes. We attempted to implement other swarm models to incorporate with our simulator for this purpose, but processing for simulations with a large number of agents performing a complex task is slow without special parallel processing techniques, and we were unable to do so successfully.

Finally, something that would be of great value to this research would be to find a working model for influence in general swarms. Having such a model, or perhaps some measurement that correlates with our intuitive notion of influence, would allow us model functions that map influence to performance in meaningful ways that could then be used in interaction scheme design. We also attempted this unsuccessfully, but advocate the utility of such a result, should it be accomplished.

References

[1] C. Hagel. (2014) "Defense Innovation Days" opening keynote. [Online]. Available: https://www.defense.gov/News/Speeches/Speech-View/Article/605602/

[2] P. Scharre, "Robotics on the battlefield part II," *Center for New American Security*, 2014.

[3] A. F. Winfield and J. Nembrini, "Safety in numbers: fault-tolerance in robot swarms," *International Journal of Modelling, Identification and Control*, vol. 1, no. 1, pp. 30–37, 2006.

[4] Z. Kira and M. A. Potter, "Exerting human control over decentralized robot swarms," in *Autonomous Robots and Agents, 2009. ICARA 2009. 4th International Conference on*. IEEE, 2009, pp. 566–571.

[5] A. Kolling, P. Walker, N. Chakraborty, K. Sycara, and M. Lewis, "Human interaction with robot swarms: A survey," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 1, pp. 9–26, 2016.

[6] D. R. Olsen Jr and S. B. Wood, "Fan-out: measuring human control of multiple robots," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2004, pp. 231–238.

[7] D. R. Olsen and M. A. Goodrich, "Metrics for evaluating human-robot interactions," in *Proceedings of Performance Metrics for Intelligent Systems*, vol. 2003, 2003, p. 4.

[8] J. W. Crandall, N. Anderson, C. Ashcraft, J. Grosh, J. Henderson, J. McClellan, A. Neupane, and M. A. Goodrich, "Human-swarm interaction as shared control: Achieving flexible fault-tolerant systems," in *International Conference on Engineering Psychology and Cognitive Ergonomics*. Springer, 2017, pp. 266–284.

[9] K. Giles and K. Giammarco, "Mission-based architecture for swarm composability (masc)," *Procedia Computer Science*, vol. 114, pp. 57–64, 2017.

[10] C. Miller, H. Funk, P. Wu, R. Goldman, J. Meisner, and M. Chapman, "The playbook approach to adaptive automation," in *Proceedings of the Human Factors and Ergonomics*

Society Annual Meeting, vol. 49, no. 1. SAGE Publications Sage CA: Los Angeles, CA, 2005, pp. 15–19.

[11] G. Kapellmann-Zafra, N. Salomons, A. Kolling, and R. Groß, "Human-robot swarm interaction with limited situational awareness," in *International Conference on Swarm Intelligence*. Springer, 2016, pp. 125–136.

[12] G. Beni, "From swarm intelligence to swarm robotics," in *International Workshop on Swarm Robotics*. Springer, 2004, pp. 1–9.

[13] C. Blum and X. Li, "Swarm intelligence in optimization," in *Swarm Intelligence*. Springer, 2008, pp. 43–85.

[14] E. Bonabeau, D. d. R. D. F. Marco, M. Dorigo, G. Théraulaz, G. Theraulaz *et al.*, *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, 1999, no. 1.

[15] D. Karaboga and B. Akay, "A survey: algorithms simulating bee swarm intelligence," *Artificial Intelligence Review*, vol. 31, no. 1-4, p. 61, 2009.

[16] J. Kennedy, "Swarm Intelligence," in *Handbook of Nature-Inspired And Innovative Computing*. Springer, 2006, pp. 187–219.

[17] I. D. Couzin, J. Krause, R. James, G. D. Ruxton, and N. R. Franks, "Collective memory and spatial sorting in animal groups," *Journal of Theoretical Biology*, vol. 218, no. 1, pp. 1–11, 2002.

[18] D. S. Brown, M. A. Goodrich, S.-Y. Jung, and S. C. Kerman, "Two invariants of human swarm interaction," *Journal of Human-Robot Interaction*, vol. 5, no. 1, pp. 1–31, 2015.

[19] M. A. Goodrich, S. Kerman, and S.-Y. Jun, "On leadership and influence in human-swarm interaction." in *AAAI Fall Symposium: Human Control of Bioinspired Swarms*, 2012.

[20] D. M. Gordon, *Ant Encounters: Interaction Networks and Colony Behavior*. Princeton University Press, 2010.

[21] M. Haque, E. Baker, C. Ren, D. Kirkpatrick, and J. A. Adams, "Analysis of biologically inspired swarm communication models," in *Advances in Hybridization of Intelligent Methods*. Springer, 2018, pp. 17–38.

[22] D. Sumpter and S. Pratt, “A modelling framework for understanding social insect foraging,” *Behavioral Ecology and Sociobiology*, vol. 53, no. 3, pp. 131–144, 2003.

[23] D. J. Sumpter, *Collective Animal Behavior*. Princeton University Press, 2010.

[24] R. Olfati-Saber, “Flocking for multi-agent dynamic systems: Algorithms and theory,” *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[25] R. Olfati-Saber and R. M. Murray, “Distributed structural stabilization and tracking for formations of dynamic multi-agents,” in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 1. IEEE, 2002, pp. 209–215.

[26] W. Ren and R. W. Beard, “Consensus seeking in multiagent systems under dynamically changing interaction topologies,” *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 655–661, 2005.

[27] C. Vasile, A. Pavel, and C. Buiu, “Integrating human swarm interaction in a distributed robotic control system,” in *Automation Science and Engineering (CASE), 2011 IEEE conference on*. IEEE, 2011, pp. 743–748.

[28] H. Abelson, D. Allen, D. Coore, C. Hanson, G. Homsy, T. F. Knight Jr, R. Nagpal, E. Rauch, G. J. Sussman, and R. Weiss, “Amorphous computing,” *Communications of the ACM*, vol. 43, no. 5, pp. 74–82, 2000.

[29] J. Bachrach, J. Beal, and J. McLurkin, “Composable continuous-space programs for robotic swarms,” *Neural Computing and Applications*, vol. 19, no. 6, pp. 825–847, 2010.

[30] J. Bachrach, J. McLurkin, and A. Grue, “Protoswarm: a language for programming multi-robot systems using the amorphous medium abstraction,” in *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 3*. International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 1175–1178.

[31] W. M. Spears and D. F. Spears, *Physicomimetics: Physics-Based Swarm Intelligence*. Springer Science & Business Media, 2012.

[32] W. M. Spears, D. F. Spears, R. Heil, W. Kerr, and S. Hettiarachchi, “An overview of physicomimetics,” in *International Workshop on Swarm Robotics*. Springer, 2004, pp. 84–97.

[33] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Phys. Rev. Lett.*, vol. 75, pp. 1226–1229, 8 1995. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.75.1226

[34] S. Nagavalli, L. Luo, N. Chakraborty, and K. Sycara, "Neglect benevolence in human control of robotic swarms," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6047–6053.

[35] T. B. Sheridan and W. L. Verplank, "Human and computer control of undersea teleoperators," Massachusetts Inst of Tech Cambridge Man-Machine Systems Lab, Tech. Rep., 1978.

[36] D. B. Kaber and M. R. Endsley, "Out-of-the-loop performance problems and the use of intermediate levels of automation for improved control system functioning and safety," *Process Safety Progress*, vol. 16, no. 3, pp. 126–131, 1997.

[37] V. Riley, "A general model of mixed-initiative human-machine systems," in *Proceedings of the Human Factors Society Annual Meeting*, vol. 33, no. 2. SAGE Publications Sage CA: Los Angeles, CA, 1989, pp. 124–128.

[38] H. A. Ruff, S. Narayanan, and M. H. Draper, "Human interaction with levels of automation and decision-aid fidelity in the supervisory control of multiple simulated unmanned air vehicles," *Presence: Teleoperators & Virtual Environments*, vol. 11, no. 4, pp. 335–351, 2002.

[39] G. Coppin and F. Legras, "Autonomy spectrum and performance perception issues in swarm supervisory control," *Proceedings of the IEEE*, vol. 100, no. 3, pp. 590–603, 2012.

[40] M. L. Cummings, "Human supervisory control of swarming networks," in *2nd Annual Swarming: Autonomous Intelligent Networked Systems Conference*. Citeseer, 2004, pp. 1–9.

[41] P. Walker, S. Nunnally, M. Lewis, N. Chakraborty, and K. Sycara, "Levels of automation for human influence of robot swarms," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 57, no. 1. SAGE Publications Sage CA: Los Angeles, CA, 2013, pp. 429–433.

[42] J. W. Crandall and M. A. Goodrich, "Characterizing efficiency of human robot interaction: A case study of shared-control teleoperation," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 2. IEEE, 2002, pp. 1290–1295.

[43] A. D. Dragan and S. S. Srinivasa, “A policy-blending formalism for shared control,” *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 790–805, 2013.

[44] D. S. Brown, S.-Y. Jung, and M. A. Goodrich, “Balancing human and inter-agent influences for shared control of bio-inspired collectives,” in *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4123–4128.

[45] N. Gilbert, *Agent-based models*. Sage, 2008, no. 153.

[46] M. J. North and C. M. Macal, *Managing business complexity: discovering strategic solutions with agent-based modeling and simulation*. Oxford University Press, 2007.

[47] L. Tesfatsion and K. L. Judd, *Handbook of Computational Economics: Agent-Based Computational Economics*. Elsevier, 2006, vol. 2.

[48] J. M. Epstein, *Generative Social Science: Studies in Agent-Based Computational Modeling*. Princeton University Press, 2006.

[49] A. L. Nevai, K. M. Passino, and P. Srinivasan, “Stability of choice in the honey bee nest-site selection process,” *Journal of Theoretical Biology*, vol. 263, no. 1, pp. 93–107, 2010.

[50] K. M. Passino and T. D. Seeley, “Modeling and analysis of nest-site selection by honeybee swarms: the speed and accuracy trade-off,” *Behavioral Ecology and Sociobiology*, vol. 59, no. 3, pp. 427–442, 2006.

[51] T. D. Seeley and S. C. Buhrman, “Nest-site selection in honey bees: how well do swarms implement the “best-of-N” decision rule?” *Behavioral Ecology and Sociobiology*, vol. 49, no. 5, pp. 416–427, 2001.

[52] T. D. Seeley and R. A. Morse, “Nest site selection by the honey bee, apis mellifera,” *Insectes Sociaux*, vol. 25, no. 4, pp. 323–337, 1978.

[53] A. Kolling, S. Nunnally, and M. Lewis, “Towards human control of robot swarms,” in *Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 2012, pp. 89–96.

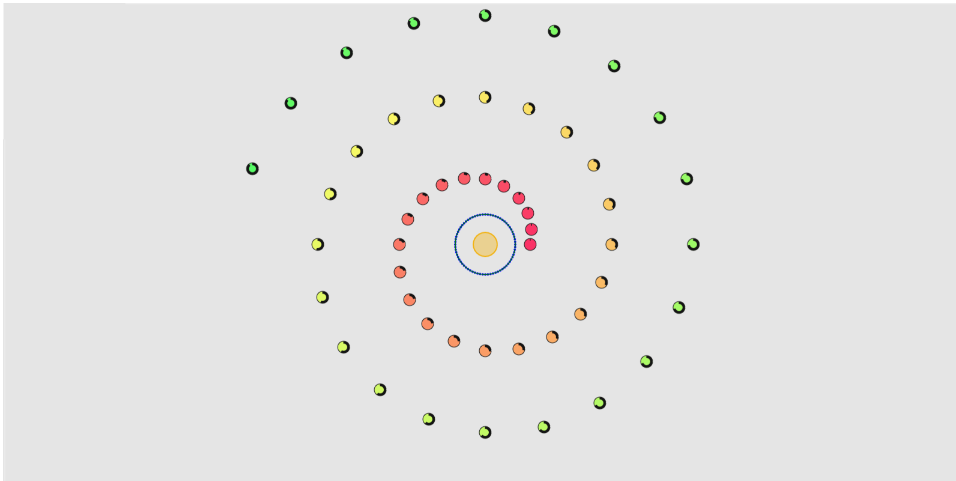
[54] A. Kolling, K. Sycara, S. Nunnally, and M. Lewis, “Human-swarm interaction: An experimental study of two types of interaction with foraging swarms,” *Journal of Human-Robot Interaction*, vol. 2, no. 2, pp. 103–129, 2013.

[55] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

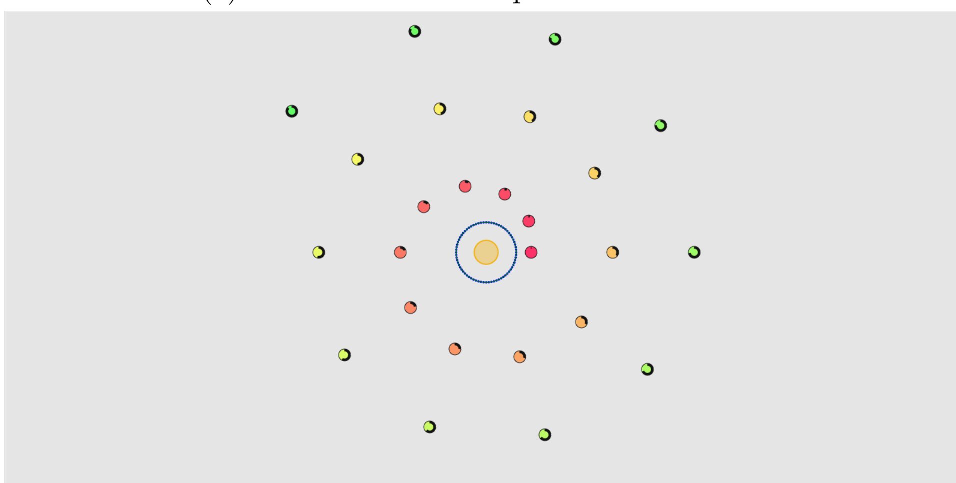
[56] (2018) sklearn.tree.decisiontreeclassifier. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

Appendix A

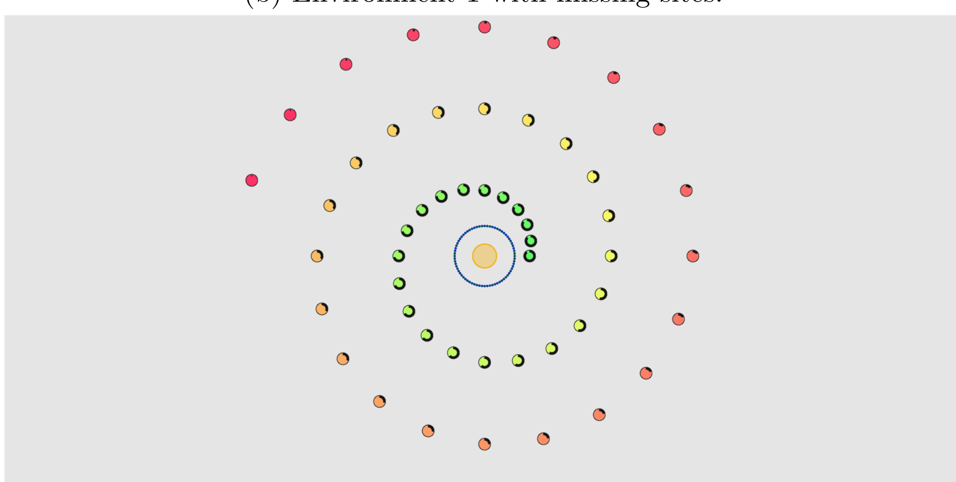
User Study Environments



(a) Environment 1 with perfect information.

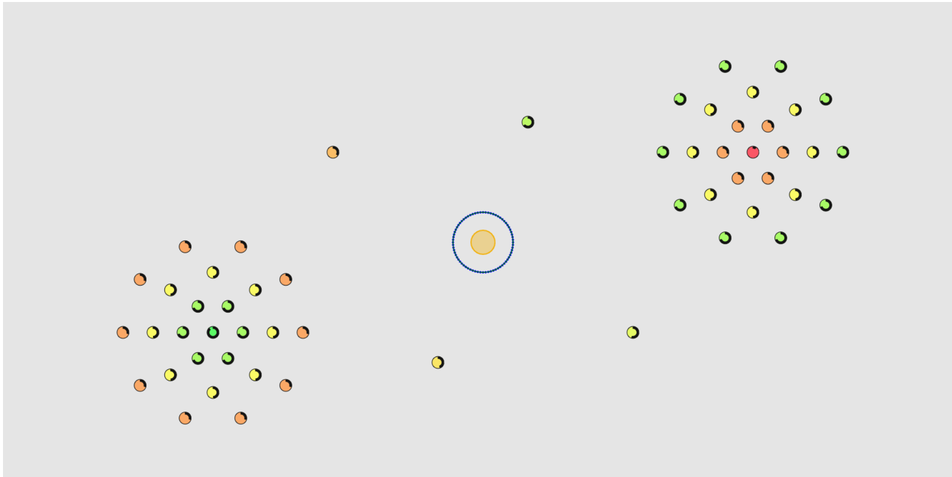


(b) Environment 1 with missing sites.

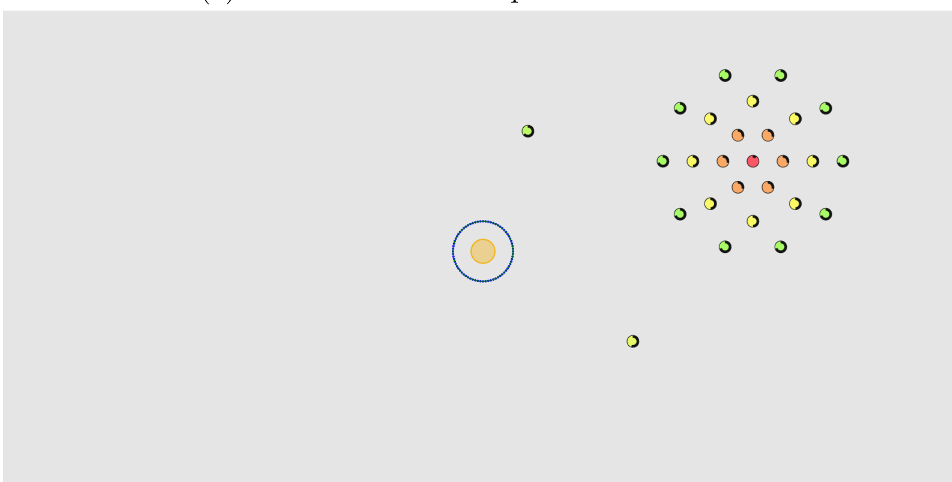


(c) Environment 1 with mislabeled sites.

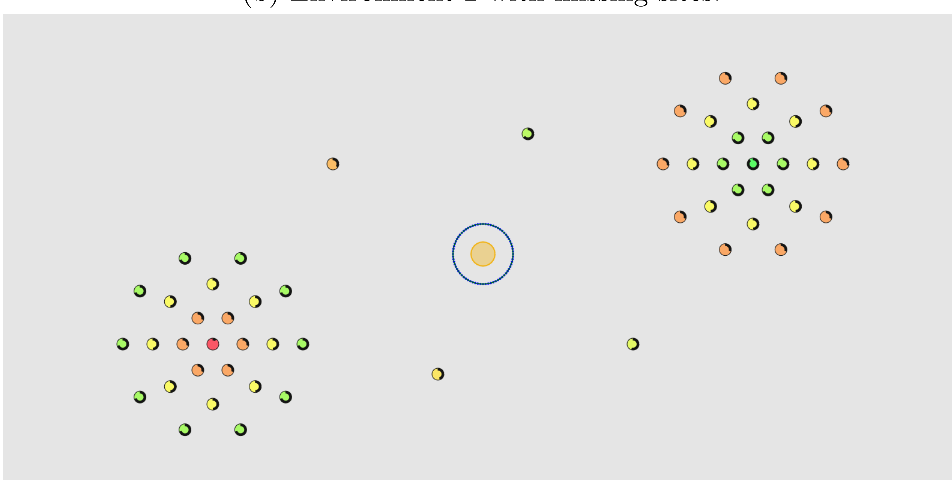
Figure A.1: The first environment seen in the user study, called “world_spiral”, with each type of information.



(a) Environment 2 with perfect information.

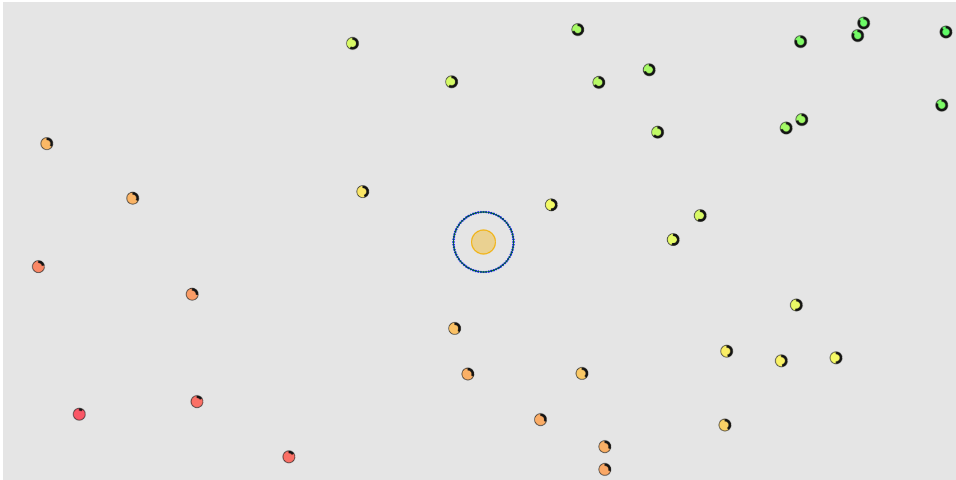


(b) Environment 2 with missing sites.

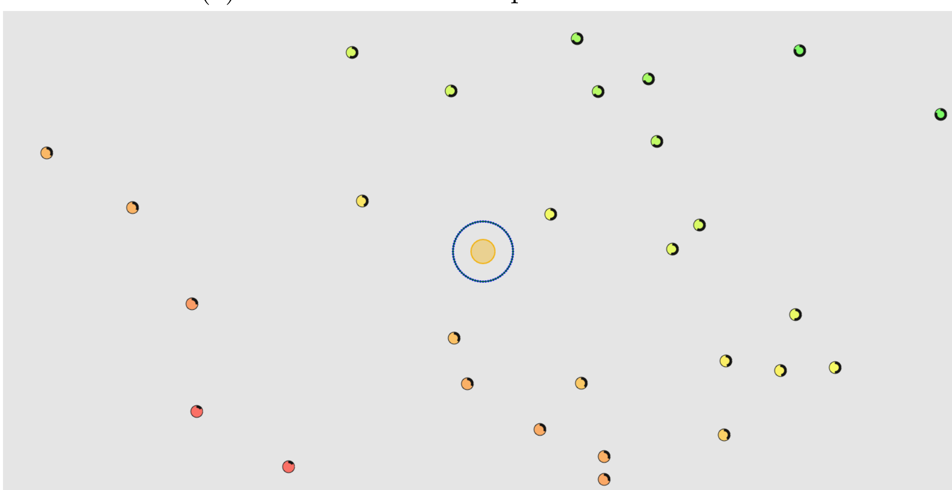


(c) Environment 2 with mislabeled sites.

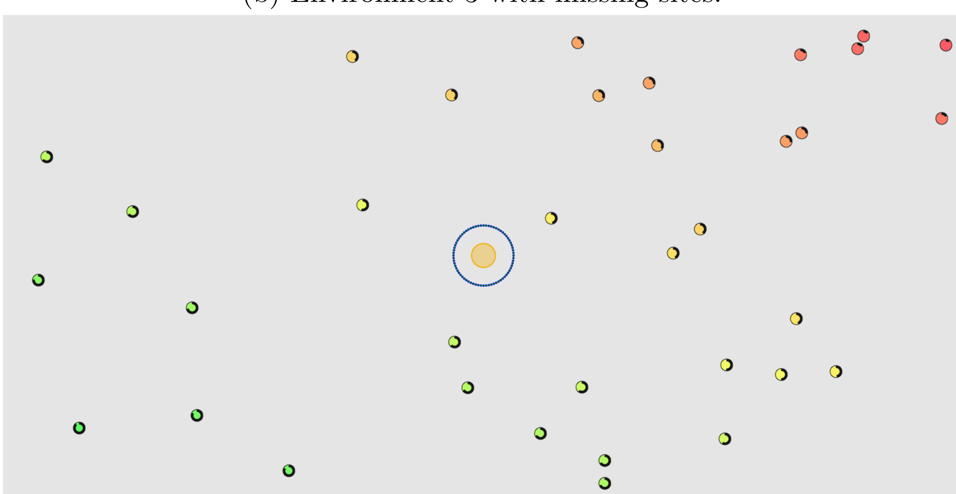
Figure A.2: The second environment seen in the user study, called "hill_vs_hole", with each type of information.



(a) Environment 3 with perfect information.



(b) Environment 3 with missing sites.



(c) Environment 3 with mislabeled sites.

Figure A.3: The third environment seen in the user study, called "grad_up_right", with each type of information.