Human-Swarm Interaction: An Experimental Study of Two Types of Interaction with Foraging Swarms

Andreas Kolling, Katia Sycara
Robotics Institute
Carnegie Mellon University
and
Steven Nunnally, Michael Lewis
School of Information Science
University of Pittsburgh

In this paper we present the first study of human-swarm interaction comparing two fundamental types of interaction, coined *intermittent* and *environmental*. These types are exemplified by two control methods, *selection* and *beacon* control, made available to a human operator to control a foraging swarm of robots. Selection and beacon control differ with respect to their temporal and spatial influence on the swarm and enable an operator to generate different strategies from the basic behaviors of the swarm. Selection control requires an active selection of groups of robots while beacon control exerts an influence on nearby robots within a set range. Both control methods are implemented in a testbed in which operators solve an information foraging problem by utilizing a set of swarm behaviors. The robotic swarm has only local communication and sensing capabilities. The number of robots in the swarm range from 50 to 200. Operator performance for each control method is compared in a series of missions in different environments with no obstacles up to cluttered and structured obstacles. In addition, performance is compared to simple and advanced autonomous swarms. Thirty-two participants were recruited for participation in the study. Autonomous swarm algorithms were tested in repeated simulations. Our results showed that selection control scales better to larger swarms and generally outperforms beacon control. Operators utilized different swarm behaviors with different frequency across control methods, suggesting an adaptation to different strategies induced by choice of control method. Simple autonomous swarms outperformed human operators in open environments, but operators adapted better to complex environments with obstacles. Human controlled swarms fell short of task-specific benchmarks under all conditions. Our results reinforce the importance of understanding and choosing appropriate types of human-swarm interaction when designing swarm systems, in addition to choosing appropriate swarm behaviors.

Keywords: Human-robot interaction, human-swarm interaction, swarms, multi-robot, operator interfaces, foraging

Due to a conflict of interest with the managing editor Michael Goodrich, the editor for this paper was guest editor Magnus Egerstedt.

Journal of Human-Robot Interaction, Vol. 2, No. 2, 2013, Pages 104–129. DOI 10.5898/JHRI.2.2.Kolling

1. Introduction

In recent decades the number of mobile robots deployed in the field has risen dramatically. Their usage in a wide range of applications offers the obvious advantages of reduced costs, removing humans from harms way, or enabling entirely new applications that were previous impossible. Especially the integration of very large teams of robots into comprehensive systems enables new tasks and missions ranging from search, exploration, rescue, surveillance, pursuit, up to deploying infrastructure. The domains of application are equally diverse and range from low-cost warehouse security, search & rescue, up to interplanetary exploration. New developments in commodity hardware which serve as low-cost replacements for otherwise expensive sensing or motion capabilities promise to further accelerate the trend towards deploying large teams of mobile robots. This, however, poses a challenge for the control of such systems, especially for human operators. Currently, most large robotic systems are controlled by multiple operators, often via remote control. For larger systems with more robots and low-cost hardware such an approach is not practical. While autonomy is already playing a vital role, even for powerful systems, in the form of tools such as mappers, path planners, and monitoring and detection systems, it is expected to play an even more important role for robotic systems with a very large number of robots, so-called swarms. An increased usage and reliance on autonomy, however, poses another challenge for the human operators, especially when distributed algorithms with complex dynamics are used. In short, enabling human operators to control robot swarms with hundreds of robots, or more, is still an open problem. Currently, multi-robot approaches generally scale to at most ten's of robots per operator even when using state of the art mapping, path planning, target detection, and coordination algorithms to alleviate the load on the operator (H. Wang et al., 2011; J. Wang & Lewis, 2007). In order to scale to the size of swarms and especially when dealing with robots with very simple sensors and actuators, we expect operators to rely on interacting with largely autonomous algorithms.

In addition to the large number of robots, swarms are also more difficult to control because the desired functioning of a swarm depends on the emergence of a useful property from the interaction of individual robots. This phenomenon of a non-apparent emergent behavior is a key characteristic of most swarms. For current robotic systems it is usually the complex interactions of a single powerful robot with its operator and surroundings that determine function. But for most swarms the interactions between a large number of robots determine function. Ideally, the emergent behavior is robust to changes in the environment, robot failure, and other unexpected events and does not depend on the proper functioning of every single robot. The difficulty is that in most cases the individual robot behavior that corresponds to a desired emergent behavior of the swarm is hard to determine. The emergent behavior or functionality arises on a higher level of abstraction and is difficult to predict or control. Consequently, many of the known algorithms, e.g., for flocking (Olfati-Saber, 2006), have been first observed in and inspired by biological systems but subsequently have been understood and studied in control theory. In fact, much of the recent work in swarm robotics has focused on autonomous algorithms for swarms, rather than methods to enable an operator to understand and interact with the swarm. While there are some examples of successful interactions of a human operator with a particular swarm system there has not been a systematic study or comparison of different methods of interaction. In addition, we have the obvious problem that swarms can be very varied and that the interaction depends on the tasks, hardware, and algorithms the swarm is capable of running.

In this paper we are concerned with more general questions regarding human-swarm interactions that are applicable across different swarms and domains. Prior work has often enabled successful interactions between human operators and swarms for particular system and purposes. A broader and systematic study of the human-swarm interaction itself, however, has not been undertaken. We provide the first starting point for such a systematic study by comparing fundamental types of human-swarm interactions. Our goal is to develop a basis for the study of types of human-swarm interaction that can inform system and swarm design. While it is clear that the design of swarm behavior is crucial for its performance we will show that swarms with an identical set of behaviors perform differently under different modes of interaction with operators. The wide range of possibilities for the design of swarms and human-swarm interactions makes such an effort worthwhile in order to determine the impact of design choices on the quality and properties of the human-swarm interaction.

1.1 Human-Swarm Interaction Types

The first type of human-swarm interaction is coined an *intermittent* interaction. Therein, an operator influences a selected subgroup of the swarm to switch from their current individual behavior to a new behavior. Once the individual robots switched behaviors the swarm will appear to generate a novel emergent behavior. This type of interaction is similar to the simple interfaces used in many computer applications, such as computer games, in which a set of objects are instructed to perform a particular function at a particular time. Consequently, our implementation of an instance of this type in form of the *selection control* method is a straightforward marquee selection tool. Once selected, an operator can influence a group of robots and set their mode to any of the available behaviors.

The second basic type of human-swarm interaction is coined an *environmental* interaction. Here an operator does not influence a selected subgroup, but rather manipulates the environment to which the robots in the swarm then react. Such environment manipulations are generally local and can influence all robots within a particular area and induce them to adopt a given behavior. Often, such as with artificial pheromones (Coppin & Legras, 2012), the environment manipulation is virtual and only visible to the robots themselves. In other cases gates or other parts of the environment are manipulated directly, for example with so-called wild robots (Bobadilla, Sanchez, Czarnowski, Gossman, & LaValle, 2011). Our implementation of environmental interactions is coined *beacon* control. Therein, an operator modifies the environment for the swarm by placing a beacon that influences nearby robots. A beacon has a location, a range, and an associated mode. All robots within its range switch to the associated mode. In contrast to selection control, which requires an active selection, robots are passively influenced by the closest beacons once they get into its range. Beacons are persistent until they are removed by an operator and multiple beacons can be present in an environment. Swarm phenomena such as leader or predator models can be simulated by placing beacons that attract or repel nearby robots. There is also a very practical difference between selection and beacon control. Selection control requires knowledge about the location of the robots to associate them to an area designated for selection. Beacon control, on the other hand, can be implemented by broadcasting a targeted signal within an area and thereby influencing all robots within that area without knowledge of their location.

In addition to intermittent and environmental interactions we have identified two additional general types of interactions with swarms. These are coined *persistent* and *parameter setting* interactions. With persistent interactions an operator provides a continuous control input for the swarm or members of the swarm. This type of interaction is found with haptic control of large robot formations, such as in (Lee, Franchi, Giordano, Son, & Bulthoff, 2011; Franchi, Robuffo Giordano, Secchi, Son, & Bulthoff, 2011; Franchi, Masone, Bulthoff, & Robuffo Giordano, 2011; Secchi, Franchi, Bulthoff, & Giordano, 2012). But also control via a predator or leader (Goodrich, Pendleton, Sujit, & Pinto, 2011) requires direct and persistent control of parts of the system. Here the considerable amount of work done in haptic and continuous control of centralized systems can be leveraged. In contrast to this, the fourth type of interaction, parameter setting, is very particular to swarms. Many swarm algorithms, e.g., the previously mentioned flocking algorithms (Olfati-Saber, 2006), are expressed as a distributed system with equations containing free parameters, such as the distance at which robots attract or repel each other. Changing these parameters can enable a wider or narrower range of possible emergent behaviors (Goodrich, Pendleton, Kerman, & Sujit, 2012). Parameter setting is arguable the most indirect form of influence on a swarm and the properties of such methods will depend even more heavily on type of autonomous behavior and swarm hardware. While these two types of interaction also warrant further study, our emphasis in this paper is solely on environmental and intermittent interactions.

Our contribution is the first study that compares the two fundamental types of interaction, *intermittent* and *environmental*, in an experimental setting with a variety of swarm missions and swarm behaviors. The implementation of intermittent and environmental interactions is achieved by using *selection* and *beacon* control, respectively. We show that operator performance can differ dramatically between these control methods, even when both have the same set of swarm behaviors available. From the perspective of the operator the main differences between selection and beacon controls are their spatial and temporal characteristics. More precisely, beacon control is spatially persistent and the robots affected by the change are generally within the area of the environment which is manipulated by the human operator. Selection control, on the other hand, keeps the same set of robots selected, even as they disperse in the swarm and move through the environment. The robot selection is hence persistent across time until a new selection is made. As a consequence both control approaches enable very different strategies with different degrees of effort and complexity. Consider a scenario in which robots have to perform a sequence of tasks when at a certain location, such as moving on a safe path around an obstacle. By using beacons any robot entering at any time will perform this sequence of tasks. Selection control requires the operator to select robots once they reach the location to give them the appropriate instructions. But if only one group of robots has to execute a sequence of task at a location, then selection control is expected to allow better control since all robots remain under control of the operator despite their continuous movement. In theory, both control approaches can in enable an operator to exert very precise control, even over individual robots by using very small and frequent selections or by placing many small-range beacons. The ideal operator should hence be able to utilize both methods with equal effectiveness; however, our experiments will show that these methods differ with regard to usability and performance.

Our experimental results also support the claim that autonomy is crucial to enable the control of robot swarms for complex missions such as foraging. We show that human operators are performing worse in foraging missions compared to even basic autonomous approaches, yet in complex structured environments human operators perform just as good as in simpler environment, suggesting that they adapt better to complex environments. In contrast, the simple autonomous algorithms deteriorate significantly in environments with complex structure. Different variants of entirely autonomous swarms also adapted differently to the tested environments. Hence, benchmarking the performance of swarm autonomy across a wide range of scenarios becomes crucial. As an open question we identify the integration of advanced autonomous swarm behavior that performs better at key aspects of the foraging mission with strategic operator intervention that enables better planning and adaption to structured environments and other environmental circumstances not considered a-priori in the design of the autonomy. The main open challenge, however, is to develop an entire field that enables a rigorous understanding of how to enable human-swarm interaction and categorize the different modalities and design choices for such systems.

We will briefly present related work in Section 2 and introduce our methods, testbed, and swarm behaviors in Section 3. This is followed by our results, a discussion in Section 4, and a conclusion in Section 5.

2. Related Work

The literature on human-swarm interaction is rather sparse. At the same time, literature on autonomous swarm behavior is too vast to be reviewed here and so we will focus only on a very small subset relating to the algorithms from control theoretic approaches that we utilized in our testbed. Surveys on swarm robotics are found in Bayindir and Sahin (2007), Dudek, Jenkin, and Milios (2002), and Mohan and Ponnambalam (2009). .

One of the earlier contributions on human control of large robot systems is made in Cummings (2004), where some of the main challenges for supervisory control of swarms are discussed. The author calls for further research to address the development of new swarming technology and the lack of understanding of supervisory control of such systems, particularly with respect to the interaction of autonomy with operators. One of the first researcher groups to consider practical challenges in controlling swarms of robots was McLurkin et al. (2006), who described experiences with a swarm of 112 robots, where the focus was kept on hardware and the development of software for the swarm. A similar approach was then taken in Li, Alvarez, De Pellegrini, Prabhakaran, and Chlamtac (2007), where the presented software tool ROBOTRAK addresses hardware and software problems, especially programmability, with regard to the control of robot swarms. Centered around another practical application, researchers Bashyal and Venayagamoorthy (2008) consider swarm control in the context of searching for a radiation source. This concept proposes yet another architecture based on a set of desirable features and presents a simple test system in which operators successfully aid a swarm in locating radiation sources. Also focused on a particular task, a team of Unmanned Aerial Vehicles (UAVs) was controlled by a single operator using behavior-based controls in Ding, Powers, Egerstedt, Young, and Balch (2009). These controls enable UAVs to perform a surveillance mission semi-autonomously while the operator generates a mission plan. Similarly as in Ding, Powers, Egerstedt, and Young (2009), much of the direct control is based on a leader-following approach and the operator can choose to teleoperate individual UAVs. This can be seen as an example of a *persistent* swarm interaction since persistent control inputs are given to at least one individual. Much of the above work is related to practical obstacles from a robotics perspective in the design, programming, and deployment of swarms rather than in the direct controllability of a swarm by an operator. No comprehensive user studies have yet been attempted.

Another (slightly more general approach) is taken in Kira and Potter (2009). Therein the authors use so-called *physicomimetics*, i.e., the simulation of physical forces to control a swarm. Two basic forms of control are distinguished. The first is a top-down control for which an operator sets global swarm parameters and the system learns to adjust the parameters of individual robots to achieve the desired global parameter. This is an example of a parameter setting interaction. The second is a bottom-up approach in which virtual agents are used to modify the behavior of existing agents through interaction instead of directly setting their parameters. This can be viewed as an example of an environmental interaction since the placement of virtual agents changes the perceived environment of all other agents. A learning method for both approaches is proposed to either set the parameters or determine the placement of virtual agents. The application considered is the defense of a resource against an attacker. The physicomimetic approach promises to be an intuitive control paradigm due to the force metaphors borrowed from physics with which operators should be familiar. Since no comprehensive user studies have been attempted this claim remains to be validated.

A recent study on the so-called *autonomy spectrum* (Coppin & Legras, 2012) uses a swarm system with a pheromone-based human-swarm interaction. The emphasis therein is on the assignment of a level of autonomy, as originally was proposed in Sheridan and Parasuraman (2005) to the information acquisition, analysis, decision selection, and up to action implementation. The work in Coppin and Legras (2012) on the problem of determining appropriate levels of autonomy for the subtasks in a complex system complements our study, but they were not concerned with a systematic comparison of human-swarm interactions. They do, however, utilize environmental and direct swarm interaction types in their system (Coppin & Legras, 2012).

The first study that investigated the degree of control a human operator can exert on a bio-inspired swarm is found in Goodrich et al. (2012). Leader and predator-based approaches were compared with regard to their influence on a swarm behaving according to Couzin's model (Couzin, Krause, James, Ruxton, & Franks, 2002). In these approaches, a human operator controls the leader or predator continuously and thereby influences the swarm locally. While leaders were more effective in influencing coherent flocks, an operator could use predators to divide flocking swarms into sub-groups. This work is another example of a persistent interaction in which the persistent control of predators or leaders influence the remaining swarm.

While the above considerations are worthwhile, our focus is rather on general principles of human-swarm interaction that enable a variety of missions with the operator injecting mission-specific knowledge into the system. The main goal is the integration of simple as well as complex robot behaviors, which may be autonomous algorithms, into a system controlled by a human operator. For this purpose we now shortly introduce some related work on the autonomous control of robotic networks. More precisely, we are considering the problems of rendezvous, coverage and deployment, and connectivity maintenance. An algorithm for the rendezvous problem for distributed robotic networks was introduced in Cortés, Martínez, and Bullo (2004), which assumed open and uncluttered environments where every robot can obtain the location of its neighbors. An algorithm for optimally distributing a network of robots in an open environment is given in Cortés, Martínez, Karatasand, and Bullo (2004). An additional algorithm that deploys robots in non-convex and simply-connected environments is presented in Ganguli, Cortés, and Bullo (2008). Although therein the goal was to simply cover the entire space, Cortés et al. (2004) computed an optimal cover given a sensor deprecation with respect to distance. A rigorous formalization and unifying framework of much of the above is presented in Bullo, Cortés, and Martínez (2009). While this work is rigorous in terms of the theory the working assumptions are rather strict and often violated in practical application; however, these algorithms make ideal candidates for our swarm behaviors so we are going to use the connectivity maintenance, rendezvous, and deployment algorithms. Connectivity maintenance allows us to guarantee that all robots form a connected communication network at all times and compute a set of admissible control inputs that satisfy this constraint. This is crucial for swarms with only local communication. Different communication networks can be chosen that lead to different communication constraints. Rendezvous algorithms guarantee that a set of robots can find agreement over a location where they get together, which could provide a useful functionality for users in the presence of obstacles. The more complex algorithm is the optimal deployment that relies on the robots to compute a Voronoi Diagram and move toward their respective centroid. Using this algorithm, a user can achieve optimal deployment in open spaces without manually dispersing robots.

The mission we developed for our experimental testbed is based on the problem of robotic foraging. The surveys in Ostergaard, Sukhatme, and Matari (2001) and Winfield (2009) provide an overview and a taxonomy of some of the variations of the problem. Most of this work, however, focuses on foraging problems in which robot performance is primarily influenced by the coordination to bring foraged objects back to a designated area. Such foraging problems are considered in Labella, Dorigo, and Deneubourg (2006), Liu and Winfield (2010), Liu, Winfield, and Sa (2007), Panait and Luke (2004), and Shell and Mataric (2006). The focus in much of this work is the planning of efficient paths for the repeated collection of objects for a source to a sink or the avoidance of collisions near a sink. Other considerations include the energy level of the swarm that can be replenished with foraged objects. In our information collection scenario, the foraged objects are not directly usable for the swarm and do not need to be transported. The search part of the foraging problem in the above papers was mostly solved by random motion, similar to the biological systems that inspired this work.

A more closely-related problem to information foraging is that of deploying robots to cover a large area. This problem is addressed in Howard, Mataric, and Sukhatme (2002) with a potential field-based approach. Robots compute a local force field that repels them from each other and from obstacles in order to distribute the robot swarm in an environment. This approach requires only local computation, and some experimental results show swarms of robots successfully dispersing in complex and structured environments. This algorithm is the basis for two of our autonomous swarm variants presented in more detail below. In another deployment and coverage algorithm as presented in Bullo et al. (2009) and Cortés et al. (2004), control laws distribute a robot swarm in order to obtain good sensor coverage, which is also optimal under certain conditions. Guarantees of optimality do not hold up in the presence of obstacles. In open space and with all parts of the environment equally important for the coverage, the robot swarm is guaranteed to converge to an optimal Voronoi configuration, where each robot is at the centroid of its Voronoi face. Due to these properties, we used this algorithm as a feature for human operators that can use it to distribute robots in open parts of the environment. Another related problem is that of connectivity maintenance, also addressed in Bullo et al. (2009). We use the presented algorithms to guarantee that the motion of robots in our swarm does not break the connected communication network since robots can only communicate locally and need to transmit their information back to a base station, as described in further detail below. Another algorithm from Bullo et al. (2009) that we employ enables a swarm of robots to rendezvous at a location, providing yet other basic capability that the human operator can utilize.

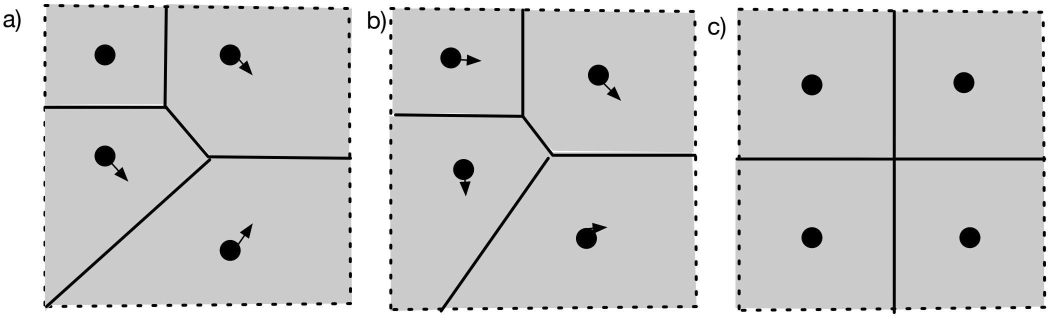


Figure 1. An example of the deployment algorithm in which four robots converge to an optimal configuration from starting positions in a) to the final configuration in c).

3. Methods

In order to study human-swarm interaction in a controlled setting we developed a testbed in which a human operator can control a large swarm and solve complex missions. The missions are based on the problem of information foraging which we describe in more detail in Section 3.1. Operators solve variants of the information foraging problem in different environments and with different robot swarms. These are described in further detail in Section 3.2. Details on swarm behaviors and the operator interface and controls are provided in Section 3.3. The missions used for the experiments are described in 3.4. Autonomous swarms that are used as performance benchmarks can be found in Section 3.5.

3.1 Information Foraging

The information foraging problem is defined as follows: Let $\mathcal{E} \subset \mathbb{R}^2$ be the bounded free space of an environment. The swarm consists of m robots. Each robot $i = 1, \ldots, m$ has a configuration $q_i \in \mathcal{E}$, in our case this is simply a position in the planar environment. We write $q \in \mathcal{E}^m$ for the swarm configuration. Robots can communicate locally within communication range r_c when within line-of-sight in $\mathcal{E}$. Each robot is equipped with a sensor with a footprint $V(q_i) \subset \mathcal{E}$ which includes the area visible from the robot up to the sensing range r_s. We write $V(q) = \bigcup_{i \in I} V(q_i)$ for the area visible by all robots. Within this area robots can detect information objects, written as o, which represent events of interest. They appear randomly through the environment according to some probability density p_E on $\mathcal{E}$ as time progresses from $t \in [0, T]$. A mission ends at time T. Information objects also have an informational value. The higher this value the more information can be collected from the target. Each robot collects information for an information object at the rate of c units per time step. Hence, more valuable targets will have to be observed for longer times to forage their value, simulating the gathering of information on more complex and interesting events. Let us write $q(o) \in \mathcal{E}$ for the configuration and $v(o)$ for the value of an information object o.

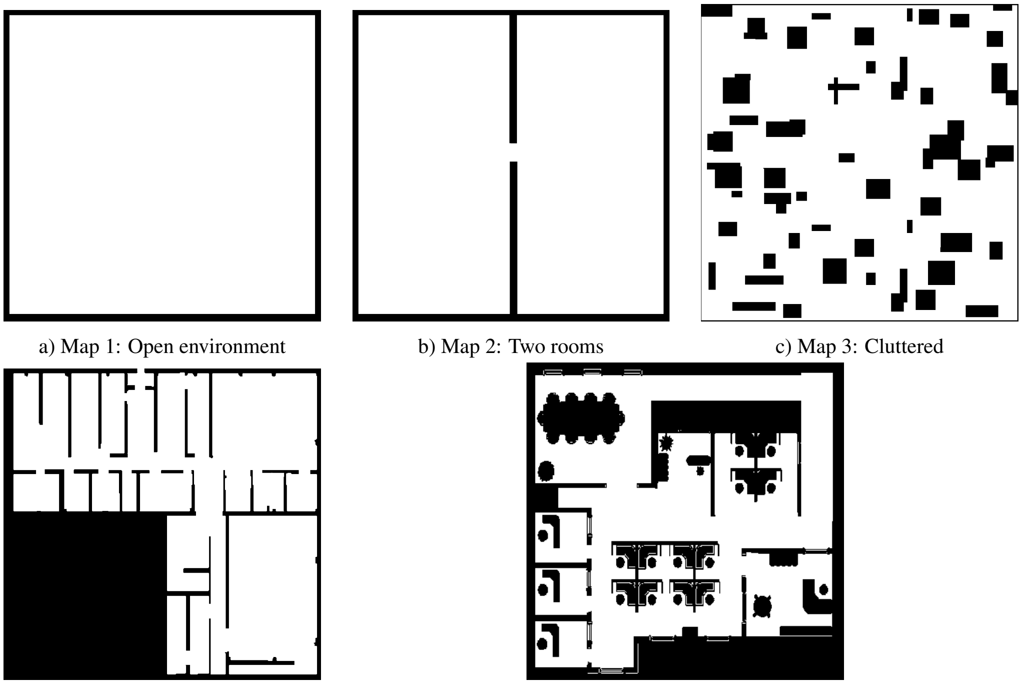
The goal of the swarm is to collect information from information objects and transmit it to a base station $b \in \mathcal{E}$. Let $G_c = (V, E)$ with $V = (b, q_1, \ldots, q_m)$ and $E \subset V \times V$ be the communication graph of the swarm with an edge between any two robots, including the base station, within line of sight and range r_c. Let $I \subseteq \{1, \ldots, m\}$ be all robot indices so that there exists a path in G_c for all q_i with $i \in I$ to the base station b, i.e., simply the connected communication network of the base station. Every robot q_i with $i \in I$ can transmit information to the base station.

In addition to detection sensors robots also have a collection footprint $C(q_i)$ which includes all points visible from q_i within range r_o. Let $C(q) = \bigcup_{i \in I} C(q_i)$ be the joint collection footprint. A robot $q_i, i \in I$, collects information only from information objects o so that $q(o) \in C(q_i)$. Information is collected at a constant rate of c units per time step for every information object in $C(q_i)$. When information is collected by a robot it is removed from the information object, i.e., for every information object o we have $\dot{v}(o) = -c \cdot |\{i | q(o) \in C(q_i), i \in I\}|$. All information objects with $q(o) \notin C(q)$, i.e., not in collection range of any robot, deteriorate at rate d, i.e., $\dot{v}(o) = -d$. This simulates the loss associated with not observing an interesting event as it unfolds. Finally, the overall performance of the robot swarm, called the score, is determined by the total information value collected and transmitted to the base station. Evidently, to reach the highest possible score, the entire environment needs to be covered with the detection sensors to avoid deterioration of information. A sufficient number of robots need to be moved toward high value targets to collect all information before time T.

3.2 Testbed, Environments, and Swarm Configurations

The testbed was developed in NetLogo (Wilensky, n.d.), a simulation platform suitable for modeling interactions between a large number of agents. In addition to NetLogo, we developed Java extensions for computing Voronoi Diagrams and Delaunay graphs. Five different environments, as shown in Figure 2, were created and used. The size of each environment is 400 by 400 NetLogo patches. Each NetLogo patch in the user interface shown in Figure 3 has a width and a height of two pixels.

For our missions based on the information foraging problem, we used the five maps from Figure 2 with uniform p_E to spawn information objects at a rate of $\frac{1}{4}$ per second with $v(o)$ sampled uniformly from $[1, 50] \subset \mathbb{N}$. For the robot swarm we set up four configurations, as seen in Table 1, with a disk of range r_o and a disk of range r_s centered at q_i as the footprints for information collection $C(q_i)$ and sensing $V(q_i)$ respectively. The decay rate for information is set to $d = 0.5$ per second. The overall collection rate of the robot swarm is $\frac{20}{m}$ units per second for all swarm configurations, i.e., swarms with more robots are assumed to have weaker collection sensors on every individual



a) Map 1: Open environment

b) Map 2: Two rooms

c) Map 3: Cluttered

d) Map 4: Structured

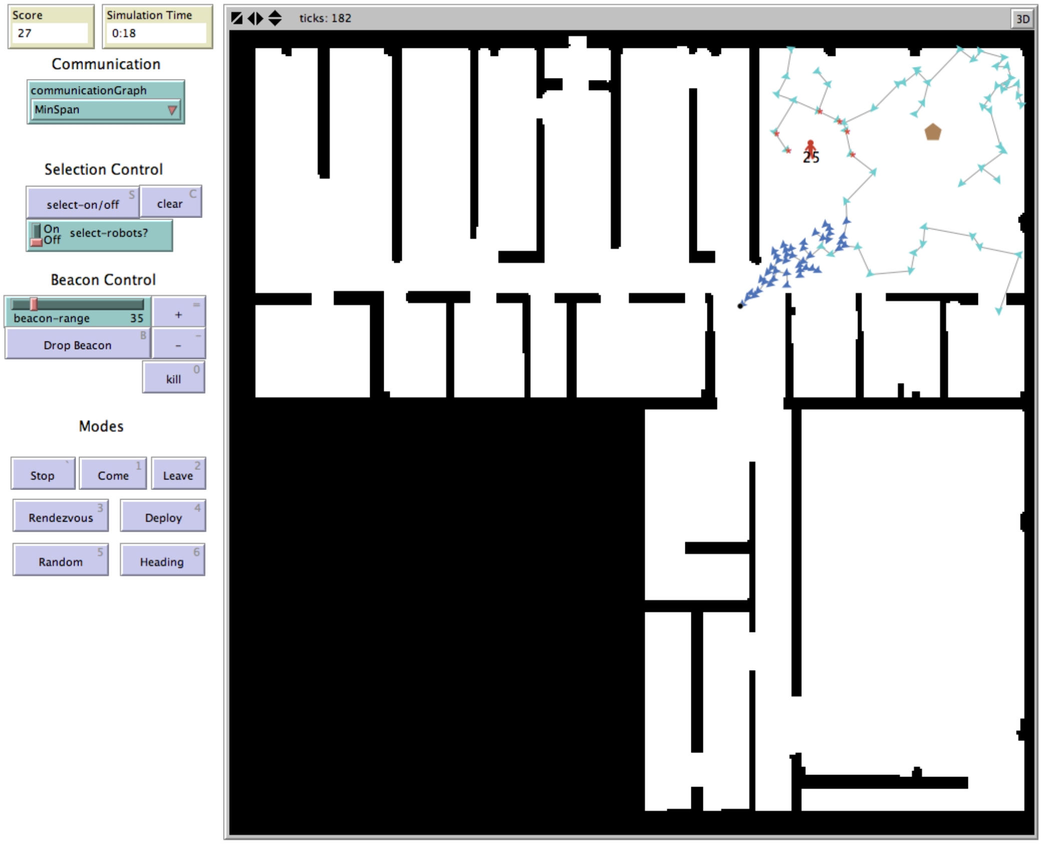
(e) Map 5: Cluttered and structured

Figure 2. Five test environments a), b), c), d), and e). Obstacles are black and free space is white.

robot. The overall spawn rate of information that can be collected is $\frac{25}{4}$. To match this information spawn rate all information objects have to be observed by $\frac{5}{16}$ of the robot swarm, i.e., approximately a third of all robots have to collect information while the others can serve as communication infrastructure and cover $\mathcal{E}$ to monitor for newly spawned information objects. The time horizon is set to $T = 300$ seconds. Robots collide at a distance of two patches. At the beginning of a mission they are placed on random locations in the top right corner of each environment around the base station. All robots have the same maximum speed of five patches per second. The swarm configurations in Table 1 are chosen so that all configurations, despite the varying number of robots, have a similar overall capability. Here capability refers to the ability to cover an environment and collect information. As a simple heuristic, doubling the number of robots and reducing the collection and communication rates and ranges by half provides a similar overall capability. It is worth noting that the swarm with more but less capable robots is more flexible with regard to the potential spatial configurations it can be in. Larger swarms are also expected to be more difficult to control for human operators. Our results with benchmark autonomous algorithms, presented in Section 4.1, show that all four configurations obtain a similar performance in these benchmarks. This indicates that these configurations, given appropriate control inputs, can reach the same levels of performance.

3.3 Swarm Behaviors and Operator Controls

The operator is assumed to be connected to the base station $b \in \mathcal{E}$. In order for the operator to see the location of a robot and send it instructions it has to be connected to the communication network rooted at the base station. In addition to obstacles, the communication links of a robot also constrain



Score 27

Simulation Time 0:18

ticks: 182

communicationGraph MinSpan

beacon-range 35

Figure 3. The user interface of the swarm testbed. Robots are small arrows, communication links are gray edges, the base station is brown, and information objects are marked as persons (information objects) in red with their information value displayed in black. Robots currently within information range r_i of an information object receive a red ”*”. Operators in the beacon and in the selection control conditions only see their respective control panel.

Table 1: The settings for each swarm configuration.

Configuration	m Robots	r_c	r_s	r_o	c
1	50	60	60	30	0.4
2	100	40	40	20	0.2
3	150	30	30	15	0.1333
4	200	25	25	15	0.1

its motion. To maintain connectivity, only motion that does not break an existing communication link is permitted. Robots can communicate with each other when within communication range r_c and when no obstacles are blocking their line of sight. The operator can choose whether robots have to maintain all communication links or a subset of these by choosing one of the following communication graphs: r_c-disk graph, r_c-limited Delaunay graph, r_c-limited Gabriel graph, or minimum spanning tree (see (Bullo, Cortés, & Martínez, 2009) for formal definitions). The r_c-disk graph is a graph given by embedding all robots into the environment and connecting all that are within line-of-sight and range r_c. All other graphs are subgraphs of the r_c-disk graph. More precisely, the r_c-limited Delaunay graph is the intersection between the Delaunay graph of all robots and the r_c-disk graph. Similarly, the r_c-limited Gabriel graph is the Gabriel graph of all robots intersected with the r_c-disk graph. The minimum spanning tree is computed from the r_c-disk graph by considering Euclidean distance as weights. A simple illustration of these graphs is provided in Figure 4.

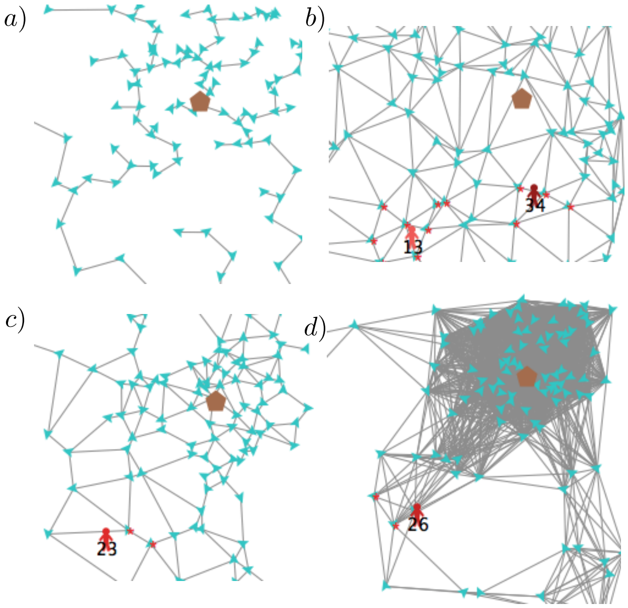


Figure 4. An illustration of the communication graphs with connections as straight gray lines between robots: a) minimum spanning tree, b) Delaunay graph, c) r_c-limited Gabriel graph, and d) r_c-disk graph.

The minimum spanning tree imposes the fewest constraints on motion since it contains the fewest possible communication links while the disk graph contains all possible links. Robots that cannot communicate with the base station are invisible to the operator and wander around the environment choosing a new random direction upon collision. They are, however, still subject to communication constraints due to their local communication links to other robots and they form a separate local connected network. The base station imposes an additional motion constraint on its closest robot to ensure that at least one robot stays within its range. In our missions all robots are initially in a connected network and are expected to remain connected. In more realistic settings than our simulations, however, noise and other factors can still lead to communication loss.

For maps 1 to 4, the map is known by the operator and drawn in the user interface. For map 5, no map is given in advance and the operator has to explore the environment. A trail for each robot is drawn to facilitate exploration. The basic swarm behaviors that are made available to the operator are represented by the following modes:

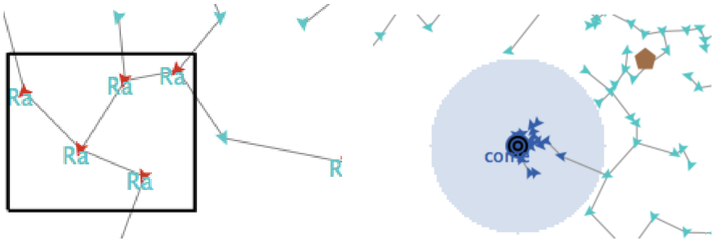
1. Stop: robots stop at their current position;
2. Come: robots move toward a target location;
3. Rendezvous: robots execute the rendezvous algorithm from Bullo et al., 2009;
4. Deploy: robots execute the deployment algorithm from Bullo et al., 2009;

5. Random: robots move with a new random heading after every collision;
6. Heading: robots synchronize their heading and move into the same direction;
7. Leave: robots move away from a target location

Robots in the above modes receive the following colors respectively: gray, blue, yellow, green, turquoise, purple, and pink. All robots start in the random mode. In all modes, except deploy and random, robots slide along obstacles when these obstruct the desired direction of motion. Collisions between robots occur within a distance of two patches.

To set the respective modes of a subgroup of robots an operator uses either only the selection or only the beacon controls, depending on the experimental condition. For the selection control the operator can select a group of robots with a rectangular marquee, clear the current selection, and set the mode of all robots in the current selection. Selected robots are marked in red, with the current mode written in abbreviated form in its respective color. Robots that are not selected are drawn in the color of the mode. The come, leave, and heading modes require an additional click to determine the target location or direction. A selection is active until it is cleared and the same set of robots can receive repeated mode instructions. Figure 5 shows a screenshot of a selection of robots.

For beacon control the operator can place, move, set the mode of, change the range of, and remove beacons. A beacon is displayed in the map as a circular object. Its range of influence is marked by drawing the surrounding environment in the color of the respective mode, as seen in Figure 5. The heading mode requires an additional mouse click to determine the heading. Any number of beacons can be placed in the environment. Every robot will assume the mode of the closest beacon that covers its area. In the come and leave modes robots either approach the beacon or move away from it. Beacons can also be moved while they exert an influence on nearby robots, simulating a virtual leader that nearby robots follow. The rendezvous mode, in contrast to the come mode, has robots meet at a location other than the beacon. This is usually a location close to where most robots were to begin with.



a)

b)

Figure 5. a) A rectangular marquee selection of robots for selection control. Robots are currently in the random mode. b) A beacon placed in an environment in the blue come mode attracting all nearby robots.

3.4 Missions

Based on the five maps from Figure 2 we presented the operator with one training scenario and seven missions. Participants were instructed that their robots had to collect information from persons (information objects) that appear randomly throughout the map. Each information object had a different amount of information to be collected and each robot close enough to an information object collects at a standard rate. The amount of information an information object had was displayed in the user interface. Once all information was collected from an information object it disappeared. Information objects that appeared with no robots within sensing range r_s were not visible on the display until a robot came within sensing range. As described above, a new information object appeared with a probability of $\frac{1}{4}$ at a random location and with an information value sampled uniformly from $[1, 50] \subset \mathbb{N}$, i.e., an average spawn rate of 6.25 units per second. Information decayed at a rate of 0.5 units per second. Information collected by robots was added to the operator's score. The four different robot swarm configurations from Table 1 determined the capabilities of individual robots. The training scenario took place in map 3 with robot configuration 2 and lasted 25 minutes and it preceded the seven missions that each lasted five minutes. On average, 1,875 information units spawned for every mission. In order to collect all this information and reach the maximum possible score robots needed to cover the entire environment to find every new information object and exactly $\frac{5}{16}$ of all robots needed to collect information at all times.

Table 2 shows the map and swarm configuration for each mission. Participants in missions 3 to 6 were controlling a variable number of robots from 50 to 200. The assignment of robot configurations in these missions was balanced across both maps (3 and 4) and with respect to whether a participant first had a larger number of robots in a map. All other missions had the standard configuration 2 with 100 robots.

Table 2: The configuration and map used for every mission. Missions 3 to 6 have four possible and counter-balanced sequences of configurations to which participants were assigned.

Mission	Map	Robot Configuration
1	1	2
2	2	2
3	3	1|3|2|4
4	3	3|1|4|2
5	4	2|4|1|3
6	4	4|2|3|1
7	5	2

We recruited 32 participants from the campus of the University of Pittsburgh, most of whom were graduate and undergraduate students. We also tested the system with two experienced operators who contributed to the design of the interface to obtain a baseline for the scores human operators could achieve. The results of these experiments are presented in Section 4.

3.5 Performance Benchmarks

In addition to the algorithms that implemented the communication maintenance and swarm behaviors made available to the human operators, we also implemented algorithms to provide problem-specific benchmarks. A common problem with simulated robots operating in a controlled setting, which is necessary for conducting reproducible experiments with human subjects, is that the mission specifications are well-defined and known. In simulation all uncertainty and unexpected events are pre-determined, at least probabilistically with known distributions. Even with real robots, most experiments are carried in more controlled environments than will be encountered in real environments. This is especially true for systems that are deployed in harsh environments, e.g., for search & rescue. As a consequence, designing algorithms tailored to a specific experimental scenario is not nearly as challenging as for real missions in real environments. Under real circumstances even mission specifications may change unexpectedly. In fact, this is one of the main reasons why human operators still have great value to add to otherwise increasingly autonomous systems. In order to study human-swarm interaction we purposefully provided the operator only with algorithms that solve subtasks of the foraging problem, i.e., basic motion primitives, rendezvous, and deployment. This emulates the difficulty of anticipating the actual mission a robot system will have to perform once it is deployed. All of this does not, however, prevent us from comparing the performance of human operator to an algorithm customized for the information foraging problem. We do not expect the human operator to outperform an algorithm geared specifically to solve a well-defined problem in simulation. Such a benchmark is important, however, to judge the quality of human-swarm system performance and assess the potential for improvement.

We implemented five variations of autonomous swarms. Every variant uses the same connectivity maintenance available to the operator, for which the communication network is set to the minimum spanning tree. The first autonomous benchmark (A.1) only uses random motion and robots turn into a new random direction each time they collide with another robot, a wall, or a communication constraint. The second variant (A.2) uses the same random motion except when a robot is within sensing range, r_s, of an information object. In this case the robot moves toward the closest information object until it is within collection range, r_o, and then stops until all information is collected. The robot then continues with random motion.

The third variant (A.3) uses a potential field-based algorithm similar to that used in Howard et al. (2002), in which robots are repulsed by each other and obstacles in order to disperse and deploy in a complex environment with obstacles. Here we add another force that attracts them to information objects they sense. The potential field for the force acting on a robots is given by:

$$\mathbf{F} = \mathbf{F}_b + \mathbf{F}_r + \mathbf{F}_o, \tag{1}$$

with the terms being due to obstacles, robots, and information objects respectively. More precisely, let $b \in V(q_i)$ be all obstacles in range of q_i and $r_b = |q_i - q(b)|$ and $\mathbf{r}_b = q_i - q(b)$, then:

$$\mathbf{F_b} = -k_b \sum_{b \in V(q_i)} \frac{1}{r_b^2} \cdot \frac{\mathbf{r}_b}{r_b}. \tag{2}$$

Similarly, let $q \in V(q_i)$ be all robots in range of q_i and $r_r = |q_i - q|$ and $\mathbf{r}_r = q_i - q$, then:

$$\mathbf{F_r} = -k_r \sum_{q \in V(q_i)} \frac{1}{r_r^2} \cdot \frac{\mathbf{r}_r}{r_r}. \tag{3}$$

Finally, let $o \in V(q_i)$ be all information objects in range of q_i and $r_o = |q_i - q(o)|$ and $\mathbf{r}_o = q_i - q(o)$, then:

$$\mathbf{F_o} = +k_o \sum_{o \in V(q_i)} \frac{1}{r_o^2} \cdot \frac{\mathbf{r}_o}{r_o}. \tag{4}$$

The resulting motion of the robots is computed as in Howard et al. (2002), considering the limitations in terms of the maximum speed of the robots. The values for the relative strength of the forces are chosen as $k_b = 2000$, $k_r = 50000$, and $k_o = 500000$. A robot senses grid cells with obstacles within a field of view of 360 degrees and at an angular resolution of 4 degrees, i.e., at most 90 obstacle grid cells. The relationship between the information object and robot forces leads to at most ten robots being attracted by a target.

The fourth variant (A.4) uses the potential field-based approach with $\mathbf{F} = \mathbf{F}_b + \mathbf{F}_r$. Robots that sense information objects ignore the potential field and move toward the information object. In addition, they transmit a message to their neighbors. Robots that receive such a message move toward the closest neighbor from whom they received the message. Once ten robots are collecting information about an object, then no more messages are transmitted and only these ten closest robots move toward the information object. Hence, an information object can attract up to ten robots within a one-hop neighborhood in G_c.

The fifth variant (A.5) implements a bio-inspired pheromone-based approach. Since in the information foraging task no objects need to be returned to a base but only transmitted, the main motivation for pheromones to establish trails that are used repeatedly is not applicable. We are instead using pheromones to coordinate the exploration along obstacle boundaries. Robots generally move just as in the random mode while also being attracted to information objects they sense, just as for algorithm A.2. In addition, once a robot hits an obstacle boundary it follows it to the right, placing a pheromone trail that lasts for eight seconds and prevents additional robots from following the same trail. This prevents too many robots from getting stuck to the obstacle boundaries but allows part of the swarm to explore the environment using simple boundary exploration patterns. Note that robots following a boundary are passively pulling other robots via the communication constraints.

Variant A.1. serves as the baseline benchmark for human and autonomous performance since it represents the idle operator not doing anything after the initial deployment. Variant A.2 is difficult for an operator to replicate manually due to the large number of robots and information objects that spawn during a mission. On average, 75 information objects spawn during our 5 minute missions in the experiment. To emulate only the autonomous stopping and seeking behavior using beacon control an operator would have to place a beacon at every target they see and set the range to r_i and the mode to stop, then set the beacon to random once the information is collected and remove the beacon. This leads to an average total of 225 actions. For selection control all robots that get into information range of a target would need to be selected immediately and then stopped. This could lead to an even larger number of actions since robots can enter the range of an information object at different times. In this case the number of actions also increases with the number of robots rather than with the number of information objects. The issue of replicating the autonomous behavior, customized for the specific foraging mission, already illustrates a basic difference between these two types of interaction.

Variant A.2 can be thought of as a custom-designed autonomous algorithm for solving the information foraging problem in open environments. The random motion in combination with connectivity maintenance leads to swift exploration of the open spaces and the autonomous information seeking behavior ensures that no visible information is deteriorating. Yet, once this algorithm is deployed in more challenging environments with complex obstacles we would expect it to deteriorate. It can hence serves as a comparison benchmark for the adaptation of the human operators to more complex environments, giving us a bound for the performance impediment in complex environments. In colloquial terms, from the perspective of the A.2 algorithm being placed in a complex environment is an unexpected circumstance that was not anticipated at the design stage. For this reason we rely on A.2 to compare with human operators. Similarly, A.3 and A.4 are autonomous algorithms that work particularly well for exploring structured environments.

While it may seem obvious to have autonomous seeking of all detected information objects, it is worth noting that in most realistic applications humans operators are still responsible for determining whether a target of interest is present, especially when the sensor data is complex, e.g., video in a search & rescue mission (H. Wang et al., 2011).

4. Results and Discussion

In this section we present and discuss the results of our experiments. Our emphasis is on the analysis of our experiments with the 32 human operators. In particular we address the following questions:

1. Do selection, beacon, and simple autonomous control (A.2) perform differently?
2. What impact do more complex environments have on performance?

3. How do participants make use of the available modes and is there a difference induced by the interaction type?

4. How do the control methods scale to larger swarms?

5. How does human performance compare to the benchmark autonomous control?

The average scores for participants and the autonomous swarm using A.2 for all maps and missions in robot configuration 2 (100 robots) are shown in Figure 6[1]. As a reference point a score of 1000 required the collection of all information from, on average, 40 information objects within the 5 minutes of the mission.

A two-way analysis of variance (ANOVA) of the scores across maps and conditions revealed a significant interaction between maps and control method ($p < 0.001$***), a significant effect of control method ($p < 0.001$***), and a significant effect of maps ($p < 0.001$***). In a separate ANOVA, comparing only the beacon and selection control conditions, there was no significant interaction between maps and control method ($p = 0.7944$) but the effect of control method and maps remained. This suggests that human operators maintained performance across maps. In Figure 6 this becomes apparent when looking at the steep drop of the autonomous swarm from map 1 to maps with obstacles. Excluding map 1 led to no significant interaction between all control methods and maps 2 to 5 ($p = 0.3907$) as well as no significant effect of maps ($p = 0.5761$). The effect of the control method remained significant ($p < 0.001$***). On maps with obstacles the average scores for control methods selection, beacon, and autonomous were 719, 590, and 695 respectively. Here selection control performs best. On all maps these averages became 779, 627, and 847, and autonomous swarms performed best overall due to the high scores in map 1. This suggests that human operators are generally poor at solving foraging tasks with swarms, not beating the simplest form of autonomy, but can adapt to complex environments. Table 3 shows results from running the experiment with two experienced operators who contributed to programming the system. These provided a rough indicator for the scores that are achievable by human operators with some experience. Note that despite the added experience the high performance of autonomous swarm is difficult to replicate with beacon control. The experienced operator with selection control achieves scores close to the autonomy in map 1 and can also maintain high scores in environments with obstacles (see Table 3, missions 3 to 7).

Table 3: Scores from two experienced operators using selection (S) and beacon control (B). Missions 3, 4, 5 and 6 were tested with 50, 150, 100, and 200 robots respectively. Normalized scores indicate the percentage of points collected of the actually spawned information and allows for a better comparison of this small sample.

Mission	1	2	3	4	5	6	7
Expert (S)	1317	1075	1075	888	1140	991	822
-normalized	72%	64%	53%	39%	60%	48%	50%
Expert (B)	1078	884	519	903	664	654	556
-normalized	55%	44%	35%	45%	38%	40%	30%

Participants using different control methods also utilized different robot modes with different frequencies, as seen in Figure 7. This suggests a different strategic adaptation to the interaction types. An explanation for this difference can be found in the average impact that a single operator issued instruction has on the swarm. A mode instruction here is either a switch of mode for a selected set of robots or a beacon, affecting all nearby robots. Let us call the mean number of robots

[1] The scores reported are the actual scores participants achieved and saw on their interface. Normalized scores that show the collected fraction of all information spawned is presented in the next section.

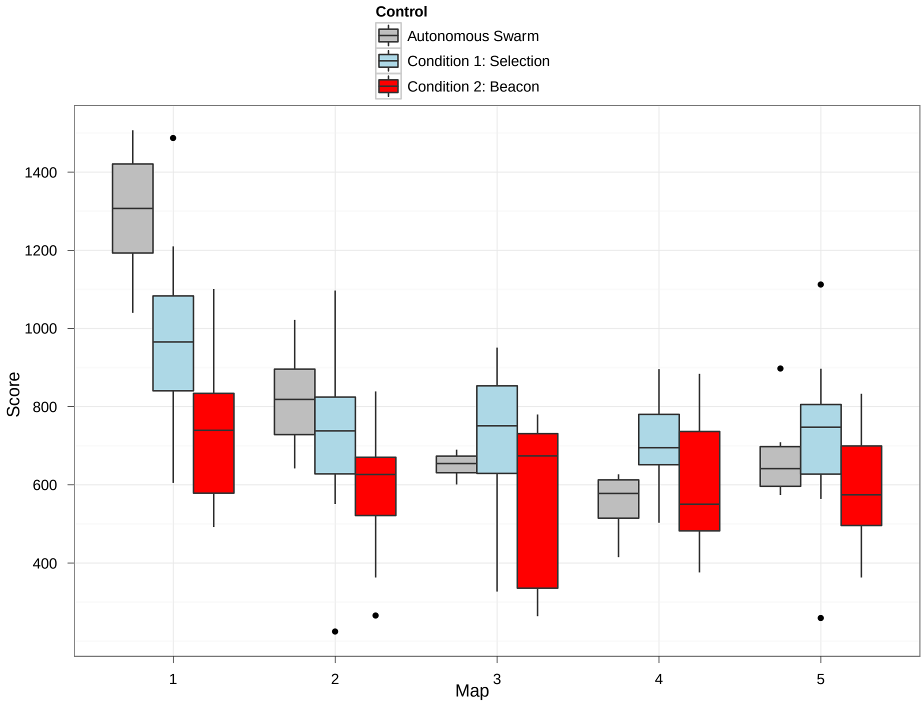


Figure 6. A box plot of the scores across participants for maps 1 to 5. Note that these scores only include missions with 100 robots where participants have either only map 3 or only map 4 with 100 robots, i.e., the sample size is reduced for map 3 and 4.

influenced by a mode instruction *mode impact*. For selection control, mode impact relates to the size of the selections and for beacon control it relates to the number of robots influenced by a beacon. Selection and beacon control differ significantly with respect to mode impact with 10 robots per instruction for selection control and 30 robots for beacon control (as seen in Table 4). The number of instructions given differs as well with 56 for selection control and 39 for beacon control. This leads to an overall higher number of robot mode switches for beacon control at 1027 robot mode switches per mission due to the higher mode impact. Some of these switches can be attributed to robots in the random mode getting close to a beacon, explaining some of the differences seen in Figure 7. For beacon control, the large mode impact has a statistically significant ($p < 0.001$***) negative effect on score while for selection control mode impact does not have any effect. The correlation between score and mode impact for beacon control is -0.45. Conversely, the number of instructions has a marginally significant ($p = 0.0897$) positive impact on score for selection control and no impact for beacon control. The correlation between the number of instructions and score is 0.414 for selection control. This is an indication that increased activity of the user, i.e., more mode instructions, helps more for selection control and that in beacon control many of the induced robot mode switches actually impede performance. On another note, operators seem not to exploit the rendezvous algorithm and rather adapt to the presence of obstacles manually and achieve rendezvous with the come mode.

Table 4: A comparison of selection and beacon control across all missions with 100 robots. All differences between selection and beacon control are significant (t-test,$df = 1$,$***p < .001$).

	Selection	Beacon
Score***	779	627
Operator mode switches***	56	39
Robot mode switches***	428	1027
Mode impact***	10	30

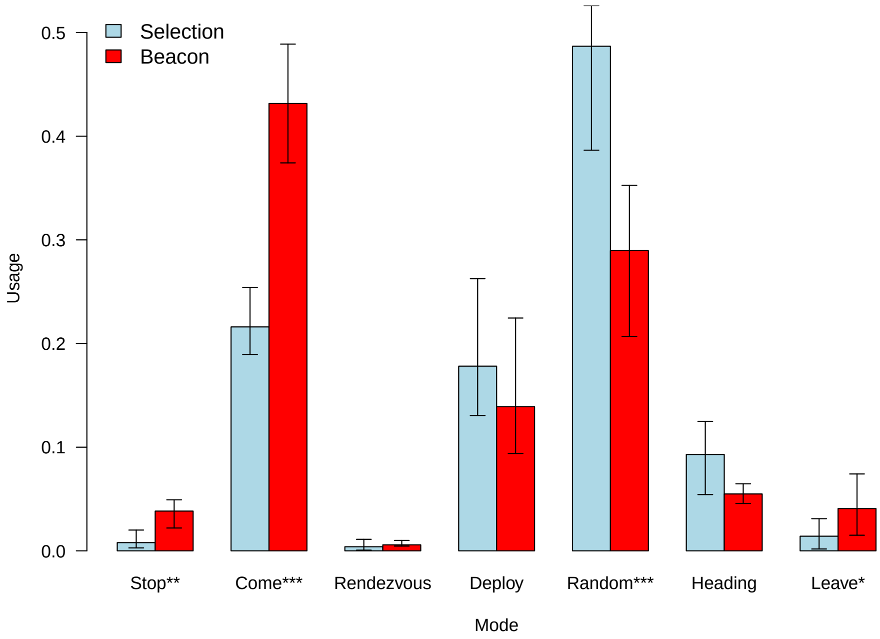


Figure 7. The figure shows the usage of each mode as the proportion of robots in that mode across the entire mission. The mean usage across all missions is shown. Arrows indicate the smallest and largest mean usage across all missions.

The above results all refer to missions with 100 robots. In the following we shall investigate differences with regard to changing robot configurations. These are available for maps 3 and 4. The main questions here relate to scalability, i.e., the ability to control larger swarms of robots of similar overall capability but with individual robots being far less capable. The scores across different robot swarm sizes are shown in Figure 8. A multiple ANOVA of scores across control condition (autonomous, selection, beacon), maps (3,4), and number of robots (50, 100, 150, and 200) revealed a significant impact in the number of robots ($p < 0.001$***) but no significant interactions nor effects of control condition or maps.

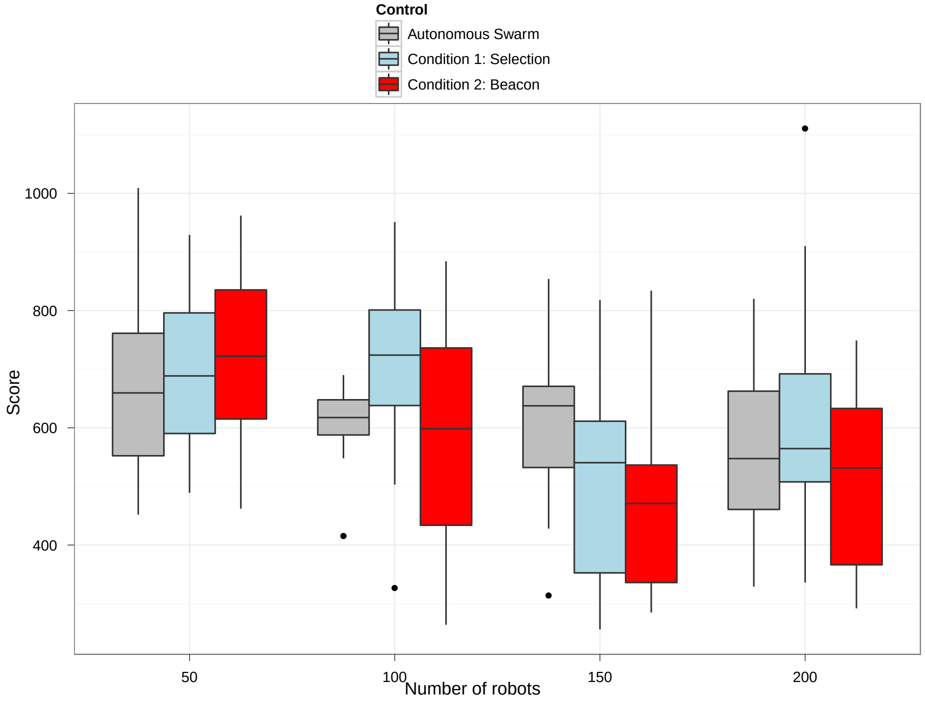


Figure 8. Score of maps 3 and 4 across different robot configurations with 50,100,150, to 200 robots across control conditions.

One would expect that the autonomous swarm is not affected by increasing the number of robots if the overall capabilities were in fact similar. In fact, for the autonomous swarm there is no significant difference across the number of robots for the score ($p = 0.5457$). This supports the claim that all four configurations do have similar capabilities. For selection and beacon control we do, however, have a significant effect of the number of robots ($p < 0.01$** and $p < 0.001$*** respectively). A simple linear regression for selection and for beacon control gives $b = -0.8144$ ($t(62) = -2.073$, $p < 0.05$*) with an intercept of $a = 733.6875$ for selection control and $b = -1.4689$ ($t(62) = -3.894$, $p < 0.001$***) with an intercept of a=759.4062 for beacon control. Both are shown in Figure 9, with a downward trend in score correlating to increasing number of robots, but less so for selection control. Considering the relatively stable performance of the autonomous swarm we can conclude that the increased difficulty of instructing the robots in a larger swarm impedes performance.

While it is expected that larger swarms are more challenging to control, we can also investigate whether and how operators adapt to these larger swarms, e.g., by increasing the frequency of mode instructions, number of beacons, or size of selections, i.e., increasing the mode impact. Figure 10 shows that only the mode impact but not the number of instructions scales with the number of robots. Hence, operators are not adapting directly to the larger swarm with increased activity but affect more robots with each mode instruction. Hence, each selection and each beacon influences more robots. In principle, this would suggest that the control methods both scale to the larger swarms and that the detriment in performance is due to a reduction in the per-robot precision of the operator's control.

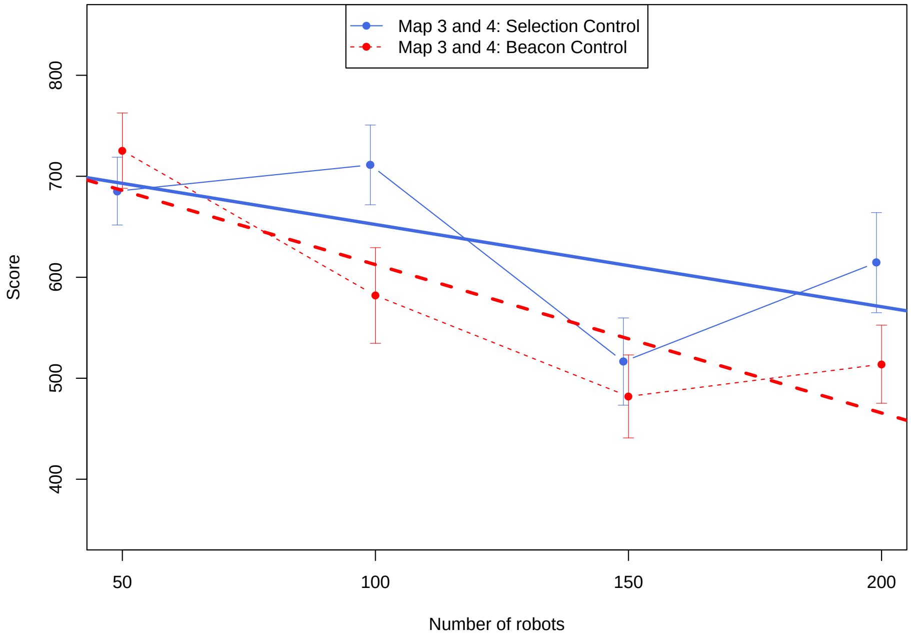
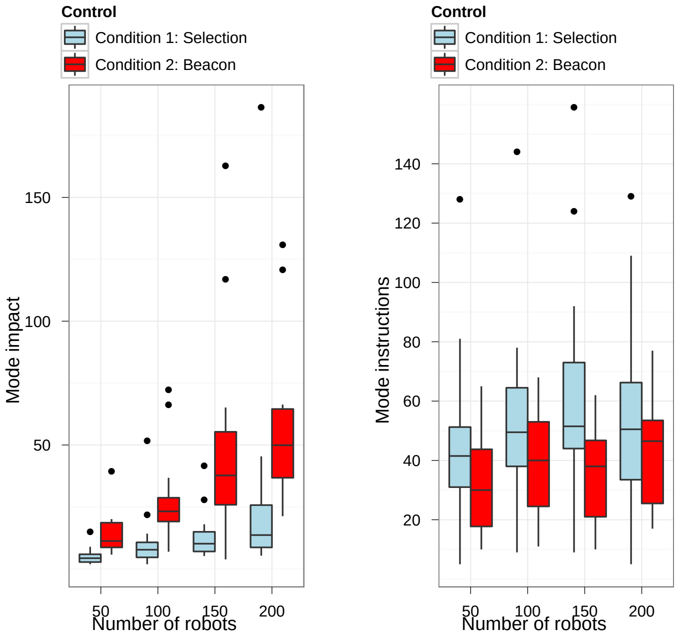


Figure 9. The mean scores across different robot configurations with 50,100,150, and 200 robots for selection and beacon control. The standard error is shown with whiskers and the regression line as a thicker line.

Looking more closely at the correlation between activity and score gives us a more detailed picture. For selection control, the number of mode instructions correlates with score by 0.1141, 0.4676, 0.3700, and 0.7897 for 50, 100, 150, and 200 robots respectively. This suggests that, for larger swarms, increased activity is rewarded with better scores. For beacon control, the correlations with scores are 0.2271, 0.1523, 0.3297, and 0.0891 for 50, 100, 150, and 200 robots, respectively. Beacon control shows no clear tendency for increased rewards for increased activity. This suggests that the two types of control do indeed scale differently and the performance impediment in beacon control is not mitigated by increased activity. In addition to the observation that different swarm modes are utilized at different frequency, this further strengthens the claim that the two interaction types lead to different strategic interactions between the swarm and the operator.

4.1 Autonomous Benchmarks

In addition to the above analysis of the differences between selection and beacon control, we can compare the performance of human operators to that of the autonomous algorithms A.1-5. The scores reported here are normalized, i.e., they are the amount of information collected as a proportion of the overall amount spawned during the mission. A normalized score of one corresponds to a perfect performance and the collection of all spawned information. But even an optimal algorithm that distributes the robots as fast as possible would still need at least 115 seconds to move robots from the edges of the initial area to the furthest part of the environment. Hence, even with the best possible performance some information is likely to deteriorate. The best observed normalized



(a)

(b)

Figure 10. (a) Mode impact for varying numbers of robots. (b) Number of mode instructions for varying numbers of robots.

scores across all trials was 0.8, i.e., 80 percent of all spawned information collected, in the open environment.

We compared the normalized scores on all maps with 100 robots (Figure 11 and Table 5) analogue to the analysis in the previous section. A two-way ANOVA of the normalized scores across maps and control method showed a significant interaction between maps and controls ($p < 0.001$***), a significant effect of control condition ($p < 0.001$***) and a significant effect of maps ($p < 0.001$***). All autonomous swarms in map 1 performed significantly better than human controls ($p < 0.001$***). All control methods dropped in performance for maps with more complex obstacles, i.e., map 2 with two rooms, map 3 with cluttered obstacles, map 4 with structured obstacles, and map 5 with cluttered and structured obstacles. Autonomous control methods dropped in performance by a similar amount from map 1 to 2 for each autonomous control method but all still maintained superior performance to human operators. This decrease was expected due to the difficulty to distribute the robot swarm onto both sides of the environment. Map 3, however, affected the autonomous control methods differently and the potential field-based approaches, A.3 and A.4, outperformed all others by a large margin ($p < 0.001$***). They also outperform human controls and the other autonomous controls in map 4. Human controls and A.1 and A.2 do not perform significantly differently ($p > 0.1$) in maps 3, 4, and 5. These results suggest that the potential field approaches manage to solve the foraging problem best, yet they do suffer a performance penalty in structured environments with multiple narrow corridors and entrances to rooms, i.e., map 4. The one-hop attraction of A.5 with neighbors improved the results further, especially in maps 3 and 5 that had fewer narrow structures but more cluttered obstacles than in map 4.

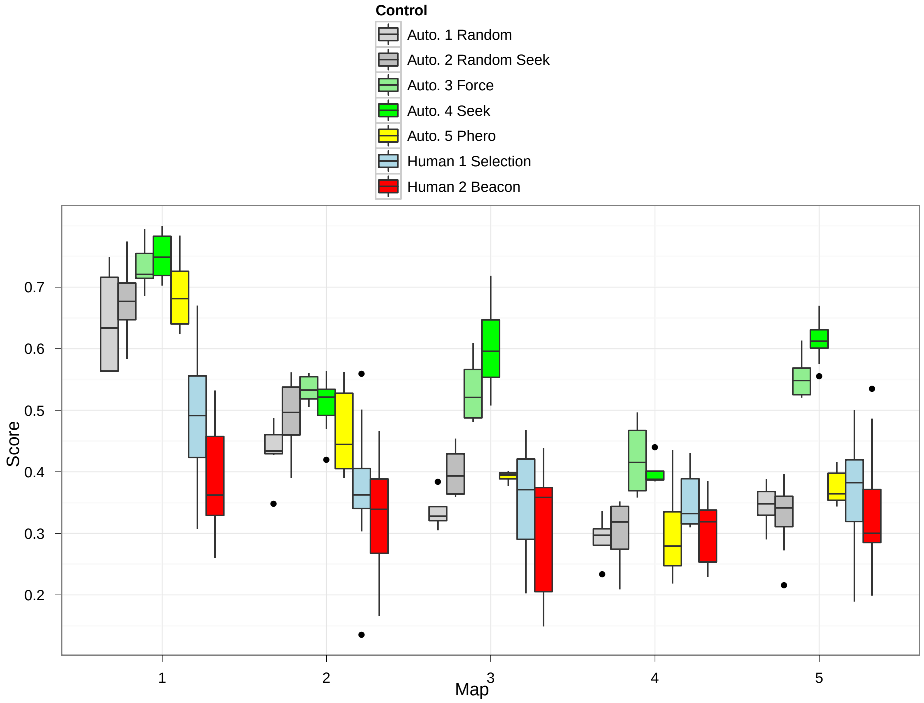


Figure 11. A box plot of the normalized scores across participants for maps 1 to 5. Note that these scores only include missions with 100 robots and participants have either only map 3 or only map 4 with 100 robots, i.e., the sample size is reduced for map 3 and 4.

Similar to the previous section, we also tested the performance with different configurations of robot swarms, i.e., 50, 100, 150, and 200 robots, while maintaining a similar overall swarm capability, effectively comparing scalability to larger robot swarms. The results are shown in Figure 12 and Table 6. An ANOVA of the normalized scores for each autonomous control method across different robot swarm sizes shows no significant differences in performance. All autonomous control methods scale to a similar performance with a similarly capable swarm regardless of swarm size. This further supports the claim that the increased difficulty of instructing the robots in a larger swarm impedes performance and that this is not an artifact of the swarms being less capable.

The above results suggest that human operators provide a reasonable adaptation to complex environments as evidenced by their performance in maps 3 and 4. Especially in map 4, human operators suffered far less performance loss than the autonomous algorithms.

5. Conclusion

We presented an investigation of two basic types of human-swarm interaction to enable operator control of robot swarms. These types were implemented as a selection and a beacon control method. We compared their performance when used by a human operator on a set of foraging missions in environments with different complexity. Key differences between these two control methods are their spatial and temporal persistence and the resulting active or passive influence on the robots swarm,

Table 5: An overview of the mean normalized score for the two human controls (selection and beacon), abbreviated H. 1 and H. 2, and the five variants of autonomous control (random, random with seeking, potential fields, potential field with seeking and one-hop sensing, and the pheromone-based approach), abbreviated A.1 to A.5.

Map	H. 1	H. 2	A. 1	A. 2	A. 3	A. 4	A. 5
1	0.490	0.384	0.643	0.679	0.733	0.750	0.690
2	0.375	0.328	0.436	0.493	0.534	0.510	0.464
3	0.356	0.306	0.336	0.400	0.533	0.604	0.392
4	0.354	0.303	0.291	0.300	0.421	0.400	0.303
5	0.365	0.336	0.346	0.327	0.556	0.612	0.374

Table 6: An overview of the mean normalized score for all control methods across different robot swarm sizes. The abbreviations are the same as in Figure 11.

Robots	H. 1	H. 2	A. 1	A. 2	A. 3	A. 4	A. 5
50	0.370	0.392	0.367	0.359	0.494	0.498	0.402
100	0.355	0.304	0.314	0.350	0.477	0.502	0.348
150	0.281	0.255	0.316	0.334	0.518	0.490	0.333
200	0.312	0.271	0.321	0.354	0.514	0.508	0.382

enabling different control strategies. Our results showed that novice human operators perform better with selection control, but that both types of control enable human operators to adapt to environments with complex obstacles and their drop in performance is less than that of a simple autonomous swarm performing better than human operators in open environments. In fact, the different types of maps (two rooms, cluttered, structured, or blind with cluttered and structured obstacles) impede human performance similarly. Variations of autonomous swarms, however, are affected differently across environments and show larger variability to changing circumstances, especially those for which the algorithms were not specially designed. Overall, the influence of the operator to adapt to complex environments was successful despite human operators being generally worse at controlling large swarms for foraging tasks. Supporting the capabilities of human operators to adapt to complex environments with improved autonomy could combine the best of both worlds.

In addition, we observed different operator behaviors for both types of interactions. With an environmental interaction operators were less able to adapt to larger swarms by increasing their activity and utilized different swarm behaviors. We conjecture that environmental interaction requires more operator training, especially since erroneous instructions that impede performance are temporally more persistent while chains of instructions are easier to set up by placing many beacons. So even though our studies indicate worse performance for foraging missions when using an environmental interaction type, there is more work to be done to disassociate the respective qualitative properties of interaction types. The required operator training to successfully utilize each interaction and cognitive effort to do so in missions with longer durations are expected to be among the many dimensions upon which the interaction types can differ. One of the main problems to be tackled to enable human control of swarms is scaling the controls to larger number of robots and hence larger environments and tasks. We observed a stronger correlation between activity and scores in larger swarms for selection control. In beacon there was no such increased correlation. We conjecture that

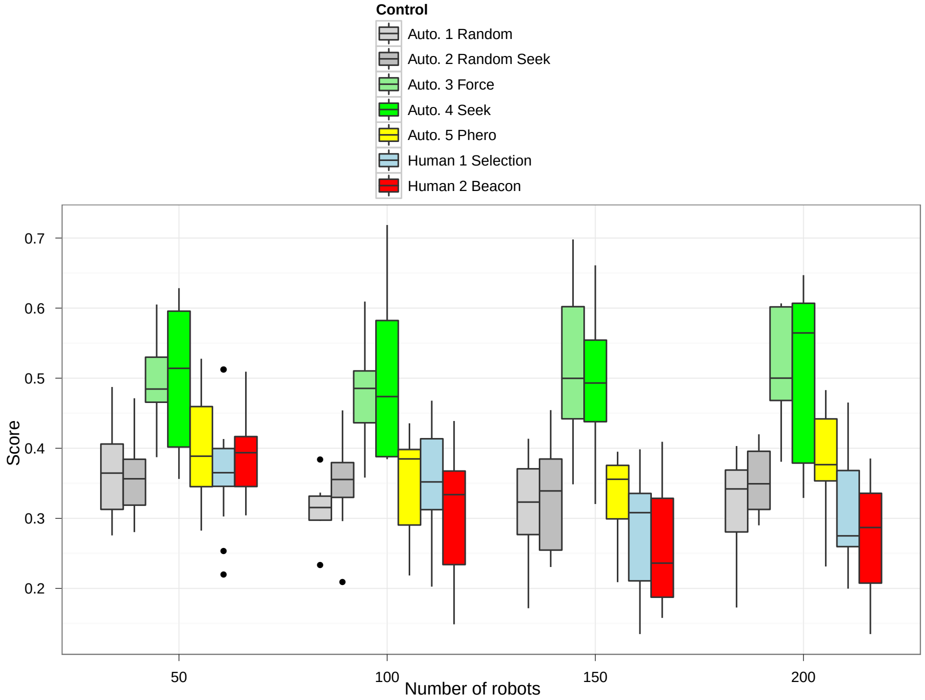


Figure 12. Score of maps 3 and 4 across different robot configurations with 50,100,150, to 200 robots across control conditions.

this is due to the fact that beacon control is in fact more scalable if used in its full potential. The strategic placement of beacons becomes more important as the swarm gets larger and mere activity alone does not help. While it is more difficult to employ and learn, beacon control seems to have a reduced dependence on activity, which is crucial for scaling to larger swarms. Further work, possibly with extensive training of participants, might well show that beacon control can perform well and scale, despite our current results showing the opposite. Swarms are complex systems and in for future work studies that train experts offer a promising direction.

Our setup and testbed are significantly simplified from what one would expect in a real scenario. While we expect many of the difficulties in controlling swarms to translate rather well to these simplified and game-like scenarios, there are certainly some issue worth investigating that relate to the limitations expected for real swarms. These include errors in sensing, localization, motors, and communication. Recent work on human-swarm interaction (Nunnally et al., 2012; Walker et al., 2012) has begun to investigate intermittent interactions with swarms in the presence of latency, bandwidth, sensing, and localization limitations.

Even though the missions in this paper are based on the foraging problem they also relate to other tasks that require the distribution of a swarm in an environment, exploiting information from many sensors, and coordinating the motion of a large number of robots. Some examples of such tasks are the establishment of an ad-hoc network infrastructure, the transport of assets to target locations, exploration, and mapping. But the two principles of control should, in addition, be investigated under different conditions with different types of swarms and tasks. Nonetheless, the principle differences between beacon and selection control that we have studied should already provide an insight into how they would perform in other missions and contexts with regard to difficulty of use and scalability.

In addition, further autonomous methods and modes should be considered. It may well be that beacon control approaches work better for particular sets of modes and tasks while selection control performs better for others. Our study provides a starting point for further similar investigations. Together with further studies investigating issues regarding the hardware limitations of swarms and other fundamental interaction types, such as parameter setting and persistent control, this research will provide a sound basis for the robust design of human-swarm interactions.

Acknowledgements

Some results presented in this article were also reported in Kolling, Nunnally, Lewis, and Sycara (2012). This work was funded by ONR Science of Autonomy Grant N0001409-10680.

References

Bashyal, S., & Venayagamoorthy, G. (2008). Human swarm interaction for radiation source search and localization. In *Proceedings of the IEEE swarm intelligence symposium* (pp. 1–8). St. Louis, MO. http://dx.doi.org/10.1109/SIS.2008.4668287.

Bayindir, L., & Sahin, E. (2007). A review of studies in swarm robotics. *Turkish Journal of Electrical Engineering*, *15*(2), 115–147.

Bobadilla, L., Sanchez, O., Czarnowski, J., Gossman, K., & LaValle, S. (2011). Controlling wild bodies using linear temporal logic. In *Proceedings of robotics: Science and systems* (pp. 17–24). Sydney, Australia.

Bullo, F., Cortés, J., & Martínez, S. (2009). *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*. Princeton University Press. (Electronically available at http://coordinationbook.info)

Coppin, G., & Legras, F. (2012). Autonomy spectrum and performance perception issues in swarm supervisory control. *Proceedings of the IEEE*, *100*(3), 590–603. http://dx.doi.org/10.1109/JPROC.2011.2174103.

Cortés, J., Martínez, S., & Bullo, F. (2004). Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. *IEEE Transactions on Automatic Control*, *51*(8), 1289–1298. http://dx.doi.org/10.1109/TAC.2006.878713.

Cortés, J., Martínez, S., Karatasand, T., & Bullo, F. (2004). Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, *20*, 243–255. http://dx.doi.org/10.1109/TRA.2004.824698.

Couzin, I., Krause, J., James, R., Ruxton, G., & Franks, N. (2002). Collective memory and spatial sorting in animal groups. *Journal of Theoretical Biology*, *218*(1), 1–11. http://dx.doi.org/10.1006/jtbi.2002.3065.

Cummings, M. L. (2004). Human supervisory control of swarming networks. In *Proceedings of the 2nd annual swarming: Autonomous intelligent networked systems conference* (pp. 1–9).

Ding, X., Powers, M., Egerstedt, M., & Young, R. (2009). An optimal timing approach to controlling multiple UAVs. In *Proceedings of the american control conference* (pp. 5374–5379). St. Louis, MO. http://dx.doi.org/10.1109/ACC.2009.5159857.

Ding, X., Powers, M., Egerstedt, M., Young, S., & Balch, T. (2009). Executive decision support. *IEEE Robotics & Automation Magazine*, *16*(2), 73–81. http://dx.doi.org/10.1109/MRA.2009.932526.

Dudek, G., Jenkin, M., & Milios, E. (2002). A taxonomy of multirobot systems. In T. Balch & L. Parker (Eds.), *Robot teams* (pp. 3–22).

Franchi, A., Masone, C., Bulthoff, H., & Robuffo Giordano, P. (2011). Bilateral teleoperation of multiple UAVs with decentralized bearing-only formation control. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems* (pp. 2215–2222). San Francisco, CA. http://dx.doi.org/10.1109/IROS.2011.6094525.

Franchi, A., Robuffo Giordano, P., Secchi, C., Son, H. I., & Bulthoff, H. H. (2011, May). A passivity-based decentralized approach for the bilateral teleoperation of a group of UAVs with switching topology. In

Proceedings of the IEEE international conference on robotics and automation (pp. 898–905). Shanghai, China. http://dx.doi.org/10.1109/ICRA.2011.5980368.

Ganguli, A., Cortés, J., & Bullo, F. (2008). Distributed coverage of nonconvex environments. In V. Saligrama (Ed.), *Networked sensing information and control* (p. 289-305. http://dx.doi.org/10.1007/978-0-387-68845-9_12). Springer US.

Goodrich, M. A., Pendleton, B., Kerman, S., & Sujit, P. (2012, July). What types of interactions to bio-inspired robot swarms and flocks afford a human? In *Proceedings of robotics: Science and systems.* Sydney, Australia.

Goodrich, M. A., Pendleton, B., Sujit, P. B., & Pinto, J. (2011). Toward human interaction with bio-inspired robot teams. In *Proceedings of the IEEE international conference on systems, man, and cybernetics* (pp. 2859–2864). Anchorage, AK. http://dx.doi.org/10.1109/ICSMC.2011.6084115.

Howard, A., Mataric, M., & Sukhatme, G. (2002). Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In H. Asama, T. Arai, T. Fukuda, & T. Hasegawa (Eds.), *Distributed autonomous robotic systems 5* (Vol. 5, pp. 299–308. http://dx.doi.org/10.1007/978-4-431-65941-9_30). Fukuoka, Japan: Springer Japan.

Kira, Z., & Potter, M. (2009). Exerting human control over decentralized robot swarms. In *Proceedings of the international conference on autonomous robots and agents* (p. 566-571). Wellington, USA. http://dx.doi.org/10.1109/ICARA.2000.4803934.

Kolling, A., Nunnally, S., Lewis, M., & Sycara, K. (2012). Towards human control of robots swarms. In *Proceedings of the ACM/IEEE international conference on Human-Robot interaction* (pp. 89–96). Boston, MA, USA. http://dx.doi.org/10.1145/2157689.2157704.

Labella, T., Dorigo, M., & Deneubourg, J. (2006). Division of labor in a group of robots inspired by ants' foraging behavior. *ACM Transactions on Autonomous and Adaptive Systems*, *1*(1), 4–25. http://dx.doi.org/10.1145/1152934.1152936.

Lee, D., Franchi, A., Giordano, P., Son, H., & Bulthoff, H. (2011). Haptic teleoperation of multiple unmanned aerial vehicles over the internet. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 1341–1347). Shanghai, China. http://dx.doi.org/10.1109/ICRA.2011.5979993.

Li, M., Alvarez, A., De Pellegrini, F., Prabhakaran, B., & Chlamtac, I. (2007). ROBOTRAK: a centralized real-time monitoring, control, and coordination system for robot swarms. In *Proceedings of the 1st international conference on robot communication and coordination* (pp. 37:1–37:4). Piscataway, NJ, USA.

Liu, W., & Winfield, A. (2010). Modeling and optimization of adaptive foraging in swarm robotic systems. *The International Journal of Robotics Research*, *29*(14), 1743–1760. http://dx.doi.org/10.1177/0278364910375139.

Liu, W., Winfield, A., & Sa, J. (2007). Modelling swarm robotic systems: A case study in collective foraging. In *Proceedings of the conference towards autonomous robotic systems* (pp. 25–32). Aberystwyth, UK.

McLurkin, J., Smith, J., Frankel, J., Sotkowitz, D., Blau, D., & Schmidt, B. (2006). Speaking swarmish: Human-robot interface design for large swarms of autonomous mobile robots. In *Proceedings of the AAAI spring symposium* (pp. 72–75). Palo Alto, California.

Mohan, Y., & Ponnambalam, S. (2009). An extensive review of research in swarm robotics. In *Proceedings of the world congress on nature & biologically inspired computing* (pp. 140–145). Coimbatore, India. http://dx.doi.org/10.1109/NABIC.2009.5393617.

Nunnally, S., Walker, P., Kolling, A., Chakroborty, N., Lewis, M., Sycara, K., et al. (2012). Human influence of robotic swarms with bandwidth and localization issues. In *Proceedings of the IEEE international conference on systems, man, and cybernetics* (pp. 333–338). Seoul, South Korea. http://dx.doi.org/10.1109/ICSMC.2012.6377723.

Olfati-Saber, R. (2006). Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, *51*(3), 401–420, http://dx.doi.org/10.1109/TAC.2005.864190.

Ostergaard, E., Sukhatme, G., & Matari, M. (2001). Emergent bucket brigading: a simple mechanisms for improving performance in multi-robot constrained-space foraging tasks. In *Proceedings of the international conference on autonomous agents* (pp. 29–30). Montreal, QC, Canada. http://dx.doi.org/10.1145/375735.375825.

Panait, L., & Luke, S. (2004). A pheromone-based utility model for collaborative foraging. In *Proceedings of the international joint conference on autonomous agents and multiagent systems* (pp. 36–43). New York, NY, USA. http://dx.doi.org/10.1109/AAMAS.2004.25.

Secchi, C., Franchi, A., Bulthoff, H. H., & Giordano, P. (2012). Bilateral teleoperation of a group of UAVs with communication delays and switching topology. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 4307–4314). Saint Paul, MN. http://dx.doi.org/10.1109/ICRA.2012.6225304.

Shell, D., & Mataric, M. (2006). On foraging strategies for large-scale multi-robot systems. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems* (pp. 2717–2723). Beijing, China. http://dx.doi.org/10.1109/IROS.2006.281996.

Sheridan, T., & Parasuraman, R. (2005). Human-automation interaction. *Reviews of Human Factors and Ergonomics*, *1*(1), 89–129. http://dx.doi.org/10.1518/155723405783703082.

Walker, P., Kolling, A., Nunnally, S., Chakroborty, N., Lewis, M., & Sycara, K. (2012). Neglect benevolence in human control of swarms in the presence of latency. In *Proceedings of the IEEE international conference on systems, man, and cybernetics* (pp. 3009–3014). Seoul, South Korea. http://dx.doi.org/10.1109/ICSMC.2012.6378253.

Wang, H., Kolling, A., Brooks, N., Owens, S., Abedin, S., Scerri, P., et al. (2011). Scalable target detection for large robot teams. In *Proceedings of the ACM/IEEE international conference on Human-Robot interaction* (pp. 363–370). Lausanne, Switzerland. http://dx.doi.org/10.1145/1957656.1957792.

Wang, J., & Lewis, M. (2007). Assessing coordination overhead in control of robot teams. In *Proceedings of the IEEE international conference on systems, man, and cybernetics* (p. 2645-2649). Montreal, Que. http://dx.doi.org/10.1109/ICSMC.2007.4414055.

Wilensky, U. (n.d.). *Netlogo.* Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL, http://ccl.northwestern.edu/netlogo/.

Winfield, A. (2009). Towards an engineering science of robot foraging. In H. Asama, H. Kurokawa, J. Ota, & K. Sekiyama (Eds.), *Distributed autonomous robotic system 8* (pp. 185–192. http://dx.doi.org/10.1007/978-3-642-00644-9_16). Springer Berlin Heidelberg.

Andreas Kolling, Robotics Institute, Carnegie Mellon University, Pittsburgh, USA. Email: andreas.kolling@gmail.com. Steven Nunnally, School of Information Science, University of Pittsburgh, Pittsburgh, USA. Email: smn34@pitt.edu Michael Lewis, School of Information Science, University of Pittsburgh, Pittsburgh, USA. Email: ml@sis.pitt.edu. Katia Sycara, Robotics Institute, Carnegie Mellon University, Pittsburgh, USA. Email: katia@cs.cmu.edu.