

Decision making swarms

Sanza Kazadi*, George Jenó, Xinyu Guan, Nick Nussgart, Andriy Sheptunov
Swarm Engineering Laboratory
Illinois Mathematics and Science Academy
1500 Sullivan Rd.
Aurora, IL 60506

ABSTRACT

While swarms that execute decisions are well known in the swarm community, swarms that exhibit this capability *a priori* have never before been achieved. We demonstrate a methodology, based on the Hamiltonian method of swarm design, that enables the design and implementation of swarms that exhibit decision-making capability. We develop the theoretical structure of the method and apply it to the development of an ant algorithm and a swarm capable of deciding whether its density exceeds a specific predetermined value. The swarm designs are validated in simulation.

keywords. swarm engineering, distributed decision making, Hamiltonian method of swarm designs

1. INTRODUCTION

One of the most astounding properties of swarms is their ability to make decisions globally despite having only local knowledge, limited computational ability, and sometimes only reactionary behavior. Ant swarms have the seemingly innate ability to decide to reroute resources committed to collecting food from one source to another source in response to the second source being closer than the first [Dorigo et al., 1996, Schoonderwoerd et al., 1996]. Bee swarms have the ability to decide to settle in a specific location, even eschewing known locations in favor of secondary locations.

Despite being able to create artificial analogs and being the subject of ongoing work [Valentini et al., 2016] there are not yet any robust general engineering methodologies in place that may be used to engineer decision making swarms. As a result, engineering swarms that accomplish decision tasks *a priori* when no direct analog exists between the desired task and a known natural analog is still an area of active investigation. The goal of swarm engineering is to develop

*To whom correspondences should be addressed at skazadi@imsa.edu.

such a methodology, usable over a wide range of potential swarms.

In this paper, we address the development of a method of generating swarms with decision-making capabilities. The method is based on the *Hamiltonian method of swarm design* [Kazadi and Lee, 2007, Kazadi, 2009, Kazadi et al., 2015]. In this method, *global properties* are constructed that correspond to desired system outcomes. These properties can be used to generate goals and these goals can be used to generate agent-level behavior. The agent-level behavior is robust in the sense that if it satisfies the design requirements, the swarm will have the desired global properties. We will examine how to use this method to derive swarm-based decision making.

The remainder of the paper is laid out as follows. Section two describes the theoretical issues needed to address the design questions. Section three applies these properties to two different decision making swarms. Section four presents validating simulation results. Section five discusses these results and concludes.

2. LOCAL PROPERTIES, GLOBAL PROPERTIES, AND STATE

In the *Hamiltonian method*, we initially take stock of *local measurables*. These are quantities that can be measured or modified (or both) by the agents themselves. They are limited by the sensory and actuation capability of the agent. We denote these by $\{s_i\}_{i=1}^{N_s}$. Each of the s_i is limited and comes from a set S_i of all possible values for s_i . Clearly, the system can be described by the vector $\vec{s} = (s_1, \dots, s_{N_s})$ and rules for how \vec{s} is currently changing. The difficulty of using such a high-dimensional representation is that it very quickly becomes intractable.

Using the local measurables, we can develop *global variables* which are functions of local variables. I.e.,

$$P = f(\vec{s}). \quad (1)$$

Key among the properties of these functions are that they be real-valued and well-defined. Assuming that they are continuous and differentiable, we may write

$$\frac{dP}{dt} = \overrightarrow{\nabla} f \cdot \frac{d\vec{s}}{dt}. \quad (2)$$

The discrete version of this equation is

$$\Delta P = \overrightarrow{\nabla} f \cdot \Delta \vec{s}. \quad (3)$$

This indicates that the change in the global measurable depends on the nature of the function f and the changes in the local measurables, which are themselves controlled by the agents of the swarm. Therefore, we must understand the nature of f in order to determine what the appropriate behavior must be and then choose the right behaviors for the swarm members. Together, determining the right behavior, the value of P can be moved from one value to another.

3. STATE TRANSITIONS AND DECISION-MAKING

Suppose we have a system with a global variable P or even a set of global variables \vec{P} . We define the *state space* as the set of all possible values for \vec{P} . Let us suppose there exists a point \vec{P}_0 in the set of all possible values for P in which $|P_0 - P(t)| < \epsilon$ for all time t once the system has come to a state in which it is in the same ϵ -ball, as long as no change to the systems occurs. Then we say that \vec{P}_0 is an *attractor* of the system. The set of all points which, when the system enters, result in moving the system toward the attractor in phase space, is called the *basin of attraction*. We can define the *state* of a system as the attractor around which the system orbits in state space.

Let us suppose that a system is in orbit of an attractor in its phase space. The system is in a state defined by this attractor. Going forward in time, two options are possible: the system may remain in orbit of the attractor or it may leave the basin of attraction of the attractor. If the system remains out of any other basin of attraction, it is said to be in a *wandering state*. If it settles around another attractor, it is said to have made a *state change*. We are interested in engineering systems that make specific and well-defined state changes in response to changes in the system. While a clear anthropomorphism, we say that, when a system makes a reversible state change in response to an external situation, the system has made a *decision*. The hallmark of the decision is that some measurable global aspect of the swarm has changed in a predictable way in response to some change in the system.

3.1 Designing ant swarms

One of the first examples of swarms to be intensely studied were swarms of ants. One of the interesting facts about ants is the fact that they have the ability to reroute resources from one collection task to another collection task when the latter task involves a piece of food that is closer to the ant hill than the former task.

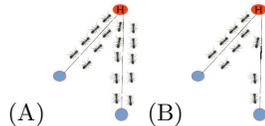


Figure 1.1: These figures illustrate the ways in which an ant swarm exploits two food sources at different distances when the closer food source is found first (A) and when it is found second (B).

It is interesting to ask how this is accomplished. We would like to determine the behavioral requirements of the system without resorting to behavioral studies of real ants.

Our methodology utilizes a global measurable for the system. First, let us position of the ant hill as the origin of the two-dimensional map making up the system's universe. Let $\lambda = \{\vec{x}_i\}_{i=1}^{N_f(t)}$ be the set of positions of all food items placed in the system. We define *trails* as the paths between the pieces of food and the ant hill. The agents then move in one of three different states: at the ant hill, on a trail, or freely moving off any trail. Let us suppose that we have a set of agents $\{a_j\}_{j=1}^{N_a}$ where N_a is the number of agents. For any trail τ , we define d_τ as the length of the trail. Then we define

$$d(a_j) = \begin{cases} d_\tau & \text{if } a_j \text{ is on the trail } \tau \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

and N_{a_p} as the number of agents actually on a path. We further define

$$P_1 = \frac{1}{N_{a_p}} \sum_{i=1}^{N_a} d(a_i) = \frac{1}{N_{a_p}} \sum_{i=1}^{N_f(t)} N_{\tau_i}(t) d_{\tau_i} \quad (5)$$

where $N_{\tau_i}(t)$ is the number of agents on trails i at time t and d_{τ_i} is the length of the trail τ_i . P_1 is the average path length of the paths which ants are traversing. This property has varies as

$$\frac{dP_1}{dt} = \frac{1}{N_{a_p}} \sum_{i=1}^{N_f(t)} \frac{dN_{\tau_i}(t)}{dt} d_{\tau_i} \quad (6)$$

Let $\vec{dN}_\tau = \left(\frac{dN_{\tau_1}(t)}{dt}, \dots, \frac{dN_{\tau_{N_{a_p}}}(t)}{dt} \right)$ and $\vec{d}_\tau = (d_{\tau_1}, \dots, d_{\tau_{N_{a_p}}})$.

We can also define P_2 as the rate at which food is acquired from the food source. We can write this as

$$P_2 = \sum_{i=1}^{N_f(t)} \frac{N_{\tau_i}(t) 2d_{\tau_i} a}{v} \quad (7)$$

where a is the quantity of food that is captured by each agent during each run and v is the speed of the agent. Interestingly, $\frac{2a}{v} P_1 = P_2$.

Suppose that the system consists of one path. Then, irrespective of the number of agents on the path, the value of P_1 is the same. Suppose now a new, shorter path becomes available. In order for the swarm to make the "right" decision, the ant system must transfer to the shorter path. It is not hard to see that

$$\frac{dP_1}{dt} < 0, \quad \frac{dP_2}{dt} < 0. \quad (8)$$

In order for this to happen, we must have $\vec{d}_\tau \cdot \vec{dN}_\tau < 0$.

One limitation of this condition is that we can satisfy the condition by making every element of \vec{dN}_τ negative. This is consistent with the agents coming off the trails. What we'd prefer is to have the agents generally stay on the trails. We can require this by stating that

$$P_3 \equiv \left\langle \left| \vec{dN}_\tau \right| \right\rangle \geq 0. \quad (9)$$

Together, equations (8) and (9) determine our swarm condition. As the vector of distances is all positive, \vec{dN}_τ must have negative components. Moreover, the product of the

negative components and the associated distances must exceed the product of the positive components and their associated distances. If the number of free agents is relatively unchanged, this means that the negative changes must be associated with longer distances than the positive changes. With exactly two paths, the condition becomes

$$d_{\tau_1} \Delta N_{\tau_1} < -d_{\tau_2} \Delta N_{\tau_2}. \quad (10)$$

In any swarm in which the number of active agents is conserved so that $\Delta N_{\tau_1} = -\Delta N_{\tau_2}$, this condition is satisfied whenever distance $d_{\tau_1} < d_{\tau_2}$. Since this is exactly the condition generally found in an ant colony, we see that the colony is able to achieve this “choice”. Therefore, we can guarantee the system’s choice with a behavior in which the number of agents is conserved, and the product is negative. Alternately, we can generate the opposite “choice” if we require the product to be positive.

3.2 Locust swarms

Locusts are a completely different kind of swarm, though their behaviors are no less astounding. Unlike ants, locusts maintain an internal state that only individual agents can experience. Experiments with locusts have demonstrated that the transition from grasshopper to locust is caused by serotonin levels which are triggered by excessive leg contact with objects. The serotonin builds up until it reaches a point that it initiates irreversible morphological changes to the grasshopper. The more aggressive locusts then interact with other grasshoppers, hastening their transition to locusts.

The entire transition can be viewed as a decision at the swarm level. The swarm is responding to the concentration of grasshoppers in a small space. When the number of grasshoppers in the space becomes small enough, the swarm initiates a change in overall behavior and moves the agents from that space. Since the swarm is undergoing a transition in which the agents undergo an irreversible change, the swarm itself is not capable of reacting to changes in the new space that agents find themselves. Therefore, when the swarm is not crowded any more, there is no change in behavior. However, this kind of swarm-based decision-making can be applied to a set of agents with reversible state changes.¹

We begin examining the way in which locust swarms make this decision by modeling the internal state s of the locust as having a numerical value and an update rule

$$\frac{ds}{dt} = -k_1 s + R(e_s) \quad (11)$$

where k_1 is some decay rate and $R(e_s)$ is the reaction to an external state e_s . We can define a global variable P as

$$P = \sum_{i=1}^{N_a} s_i. \quad (12)$$

The swarm is considered to have made a decision to move to a new state if P changes from one attractor to a second. Therefore, the behavior of the swarm must be such that it

¹This kind of decision making can be applied to bee hives making decisions about what food sources to exploit, the growth of ideas in a swarm of individuals, the collective decision to go to war, etc.

maintains an orbit around one attractor. Clearly,

$$\frac{dP}{dt} = \sum_{i=1}^{N_a} \{-k_1 s_i + R(e_{s_i})\} = -k_1 P + \sum_{i=1}^{N_a} R(e_{s_i}). \quad (13)$$

Therefore, if the magnitude of the sum of the agent reactions is low, this has an attractor at $P = 0$. In order to create a situation where a second attractor exists, we must have a case where a positive feedback occurs under the right conditions, overwhelming the decay. Note that actual locusts with irreversible changes have a mechanism which eliminates the decay mode, thereby enhancing the effect of the trigger on a single agent.

In order for the system to move rapidly to its new equilibrium state, we must have

$$\frac{dP}{dt} > 0$$

for longer than a spurious change. In such a case, this would require that

$$P < \frac{1}{k_1} \sum_{i=1}^{N_a} R(e_{s_i}). \quad (14)$$

If R is bounded, this would require that at least $N_{trigger}$ agents are state 1. This means that

$$\frac{k_1 P}{R_{max}} < N_{trigger}. \quad (15)$$

Therefore, such a system would be triggered when the number of agents exceeds that trigger amount.

In order for this swarm to make the decision we’re interested in, the minimal requirement is the condition given in (15). This is our swarm condition. Interestingly, if the external state being measured is the proximity of other agents or local density of agents, the swarm is, in totality, measuring deciding whether the space it inhabits is above a threshold size.

4. PERFORMANCE OF SIMULATED SWARMS

In this section, we present performance of simulated swarms. These swarms are designed according to the swarm requirements developed in Section 3. We then evaluate their performance on their ability to correctly execute the desired decisions according to their design.

4.1 Shortest Distance to Food

We are interested in testing the theory described in Section 3.1 using an embodied ant swarm simulation. The simulation is depicted in Figure 4.1 and contains a home, food sources, and agents passing between them.

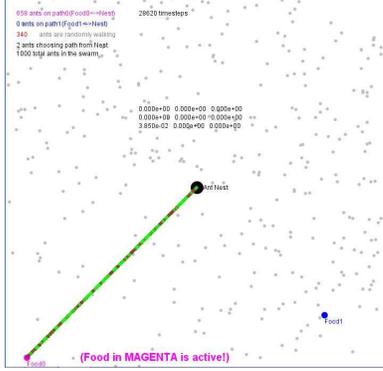


Figure 4.1: The simulated ant swarm. The gray circles are agents that are searching for food. Green circles are agents that are following a pheromone trail. Red circles are agents moving back toward home. The black circle is the home, the dark blue circle is an inactive food source (invisible to the agents), and the pink circle is an active (visible to the agents) food source.

Our simulation comprises a set of autonomous agents that move in a discrete 2-dimensional grid. The grid contains the ant home as well as two food locations. Each grid element is defined in part by a non-negative quantity of pheromone. Each iteration, agents may add to the pheromone at a given location. After each set of updates by the agents, the amount of pheromone on each location is updated. The pheromone evaporates according to an update rule

$$\rho(t+1) = \epsilon\rho(t) \quad (16)$$

where ϵ is a pre-determined evaporation rate.

The agents are defined by their position, direction of motion, active state, and their sensory state. The agents' active state is one of three possible states:

- *random wandering* in which they move forward in one direction and then randomly change their direction of travel,
- *food tracking* during which they move directly outward from their home, depositing pheromone as they move, and
- *homebound state* during which they move directly toward their home after picking up food.

The internal state defines the behavior of the agent. Agents begin in the random wandering state. Once they find a food source, they acquire a food item and transition to the homebound state. While in the homebound state, agents drop pheromone on the ground. The rate that pheromone is dropped is initially set to some maximum rate. As they move toward home, however, the rate falls off as the inverse of the distance traveled toward home.

If the agent is in the wandering state and it encounters a detectable pheromone trail they transition to the food tracking state and move directly away from their home.

When an agent makes it back to home, it deposits the food item and then decides which of possibly many paths it will follow. This entails measuring pheromones of all paths reaching home. A uniformly generated random number is used to decide which of the trails to take, with the probability of taking any specific trail linearly dependent on the

ratio of the pheromone at the end of the trail and the total amount of pheromone detectable at the home. It can be shown that this methodology satisfies the swarm condition given in equations (8) and (9).

Figure 4.2 illustrates the performance of the simulation over a run in which the swarm shifts from an initially longer path to a shorter path. Initially, an active food item (dark blue) and an inactive food item (light blue) are in the system. Only the active food can be detected by the agents, and a path forms between the food and the home. After the inactive food item becomes active, the swarm generates a path to it as well. Eventually, the swarm shifts its resources from the longer path to the shorter.

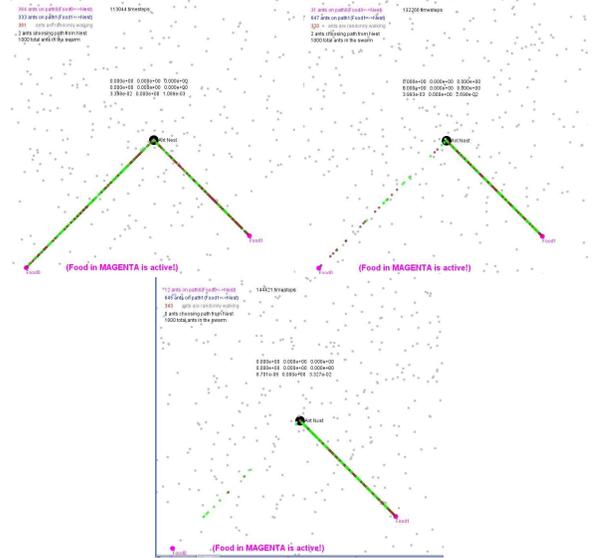


Figure 4.2 The simulation proceeds with an initial path which then develops into a shorter path when the second food item becomes active.

While the simulation proceeds, the value of P_1 changes. Initially it settles at a high value, but eventually it transitions to a lower value, as desired. The data can be seen in Figure 4.3 below.

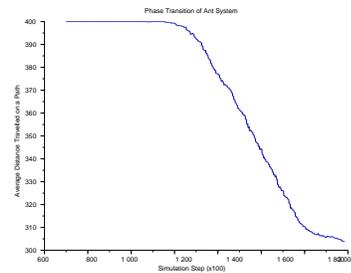


Figure 4.3: The evolution of the value of P_1 over the lifetime of the simulation.

4.2 Density determination

We now describe our second, quite different embodied simulation. Agents are represented by circles and move around in a two dimensional bounded space. The agents have a size and cannot overlap one another. The simulation is illustrated in Figure 4.4.

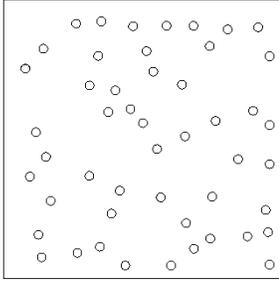


Figure 4.4: The simulated locust swarm, which can measure the size of the space.

The agents are characterized by a direction of motion, and internal state which is represented by a positive real number, and a position. The state is initialized at a value of zero. At each iteration, the state is updated according to

$$s(t+1) = \epsilon s(t) \quad (17)$$

where the ϵ is the decay rate of the state. When moving around, if they agents come into contact with a boundary, they change direction and begin moving in another direction. If the agents come into contact with another agent, they change direction but also increase the state by adding a quantity to it. If the state of the agent exceeds a given value, its color changes from white to red; dropping below the same value returns it to white. At this point the agent is said to be in the excited state.

A transition of the swarm state is considered to have happened if more than 90% of the swarm is in the excited state. Such a transition is illustrated in Figure 4.5. Figure 4.6 illustrates the change in the property given in (12). As can be seen, the property changes from one in which the state is completely off to one in which it is completely on as a distributed response to the global size of the swarm.

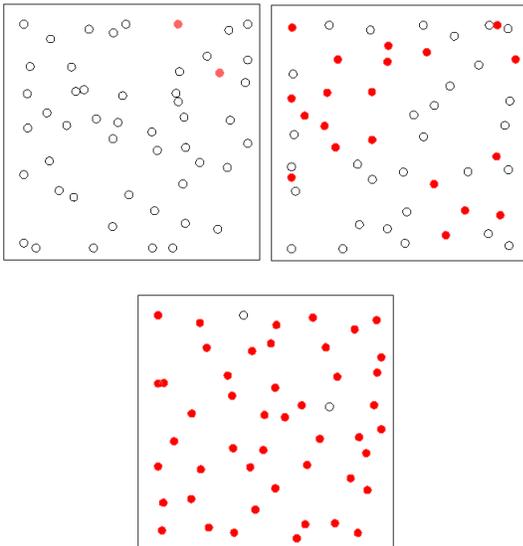


Figure 4.5: This set of screenshots illustrates the state transition of the swarm during a decision event.

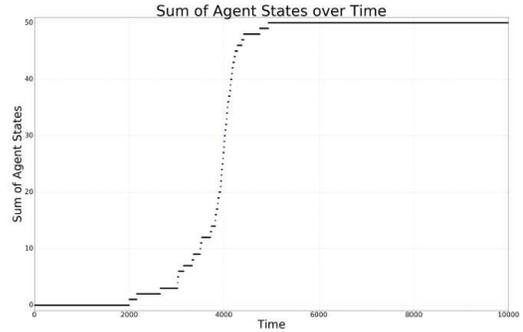


Figure 4.6: We provide the evolution of the property given in (12) as the swarm decides on the size of the space. This is the phase transition that is a hallmark of the decision.

We are interested in how well this swarm can discern the size of an area. Figure 4.7 provides the performance of the swarm in discerning the size of the area in which it is located. As can be seen, the swarm not only reliably determines the size of the space, but does so by reacting to the density of the swarm in the space. This approach to determining a space’s size might be useful in a variety of situations when rapid measurements of irregular spaces are required.

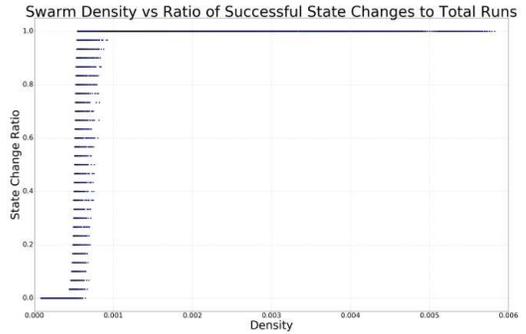


Figure 4.7: The performance of the swarm on the space size/density decision. The swarm not only reliably transitions state at and beyond a specific size, it also triggers when the swarm is beyond a specific density in spaces of different sizes.

5. DISCUSSION AND CONCLUDING REMARKS

In this study we’ve examined the design and implementation of swarms designed to carry out specific decisions. This extremely interesting area of swarm research represents one of the defining aspects of swarms that originally brought people to the field. How can it be that a group of simple agents can elicit complex behavior, generating group actions that are, to the untrained eye, complex and sometimes intelligent?

Our basic assertion is that the behaviors are basically the dynamics of the system, and it is that dynamics that we’re after. In the case of swarms generating behaviors that mimic the kinds of decisions that intelligent beings might make,

the question becomes how can the behaviors be generated? That is, rather than trying to understand the behavior as a model of intelligence, we are trying to develop a behavior that generates the “intelligent” behavior. In so doing, we are seeking to develop a tool that enables the subsequent generation of a variety of adaptive behaviors that can be implemented at the swarm level.

We have demonstrated that, using the Hamiltonian Method of Swarm Design, we are able to create, from basic principles, the very kinds of swarm systems that exhibit “intelligent” behavior. The swarms dynamically respond to the environment in ways that would seem to, if they were living systems, enhance their ability to survive. We have demonstrated that using one- and two-dimensional systems (number of defining global properties), we are able to create swarm requirements that lead to the development of swarms of the desired goals.

We have demonstrated this with the archetypal decision-making swarm modeled after biological ants. However, unlike other studies, we have motivated the design requirements by using a mathematical analysis rooted in the agents’ measurables and the global properties. We demonstrated that this is a two-dimensional swarm, as it is defined by two global properties. We found that, once the agents’ defining behavioral conditions were satisfied, the swarm behaved as it should. This right-out-of-the-box design is significant, short circuiting typical parameter setting which could take quite a bit of effort.

We have further used a swarm based on locusts, with entirely internal state. This swarm consists of circular agents moving in a bounded space and colliding with one another as they move around. Excessively packed in agents generate a transition from an internal unexcited state to a second, excited state. A positive feedback in the interactions drives, under the right conditions, a reliable transition of the whole swarm.

The key to the adaptive capability of the swarm seems to be the maintenance of the state of the system and the measurable transition in the system’s dynamics. The ant swarm maintains its state in the external state of the pheromones in the system. The locust-inspired swarm maintains state internally in each of the agents. In both cases, the state invokes behavioral changes in the agents and a positive feedback in the system. Both state changes can be tracked by monitoring the global properties that are used to define the system.

Future work will examine the role of state in generating distributed decision or adaptation. We will also examine additional swarms that are capable of making decisions between several potential outcomes.

6. REFERENCES

[Kazadi and Lee, 2007] S. Kazadi and J. Lee. *Artificial physics, swarm engineering, and the Hamiltonian Method*. **Proceedings of the World Congress on Engineering and Computer Science**, San Francisco, CA, 2007.

[Kazadi, 2009] S. Kazadi. *Model independence in swarm robotics*. **Journal of Intelligent Computing and Cybernetics, Special Issue on Swarm Robotics**, 2(4):672–694, 2009.

[Kazadi et al., 2015] S. Kazadi, D. Jin, and M. Li. *Utilizing*

abstract phase space in swarm design and validation. **Advances in Swarm and Computational Intelligence**, volume 9140, pages 14–29, New York, NY, USA. Springer, 2015.

[Dorigo et al., 1996] M. Dorigo, V. Maniezzo, and A. Coloni. *The ant system: optimization by a colony of cooperating agents*. **IEEE Transactions on Systems, Man, and Cybernetics - Part B**, 26 (1), pp. 1-13, 1996.

[Schoonderwoerd et al., 1996] R. Schoonderwoerd, O. Holland, J. Bruten, and L. Rothkrantz. *Ant-based load balancing in telecommunications networks*. **Adaptive Behavior**, 5 (2), pp. 169-207, 1996.

[Valentini et al., 2016] G. Valentini, E. Ferrante, H. Hamann, M. Dorigo. *Collective decision with 100 Kilobots: speed versus accuracy in binary discrimination problems*. **Autonomous Agents and Multi-Agent Systems**, 30 (3), pp. 553-580, 2016.

[Reina et al., 2015] A. Reina, G. Valentini, C. Fernández-Oto, M. Dorigo, V. Trianni. *A Design Pattern for Decentralised Decision Making*. **PLOS ONE**, 10(10), 2015: e0140950. doi: 10.1371/journal.pone.0140950

[Seeley and Buhrman, 1999] T. Seeley and S. Buhrman. *Group decision making in swarms of honey bees*. **Behavioral Ecology and Sociobiology**, 45, pp.19-31, 1999.

[Seeley and Visscher, 2004] T. Seeley and P. K. Visscher. *Group decision making in nest-site selection by honey bees*. **Apidologie**, 35 (2), pp.101-116, 2004. <10.1051/apido:2004004>. <hal- 00891879>