To appear in *Proceedings of the International Conference on Autonomous Agents & Multiagent Systems* (AAMAS-15), Istanbul, Turkey, May 2015.

Determining Placements of Influencing Agents in a Flock

Katie Genter, Shun Zhang and Peter Stone
Department of Computer Science, University of Texas at Austin
Austin, TX 78712 USA
{katie,menie482,pstone}@cs.utexas.edu

ABSTRACT

Flocking is a fascinating collective behavior exhibited by many different animals including birds and fish. As understood by biologists, the overall flocking behavior emerges from relatively simple local control rules by which each individual adjusts its own trajectory based on those of its closest neighbors. We consider the possibility of adding a small set of influencing agents, that are under our control, into a flock. Specifically, in this paper we consider where in the flock to place the influencing agents that we add to the flock. Following ad hoc teamwork methodology, we assume that we are given knowledge of, but no direct control over, the rest of the flock. We use the influencing agents to alter the flock's trajectory, for instance to avoid an obstacle. We define several methodologies for placing the influencing agents into the flock, and compare them via detailed experimental results.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Algorithms, Experimentation

Keywords

Ad Hoc Teamwork, Agent Placement, Flocking

1. INTRODUCTION

Flocking is an emergent swarm behavior found in various species in nature. Each animal in a flock follows a simple local behavior rule, but this simple behavior by individual agents often results in group behavior that appears well organized and stable. Flocking is often studied under the assumption that all of the agents are identical or represent a small set of well-defined behavior types. Indeed, various disciplines such as physics [12], graphics [9], biology [2], and distributed control theory [6, 7, 11] have studied flocking in order to characterize its emergent behavior. In our work, we instead consider how to lead a flock to adopt particular behaviors by adding a small amount of controllable agents to the flock.

As a motivating example, imagine that a flock of migrating birds is flying directly towards a dangerous area, such as an airport or wind farm. Our goal is to encourage the birds to avoid the dangerous area without significantly disturbing them. Since there is no way to directly control the flight path of the birds, we must instead alter the environment so as to encourage the flock to alter their flight path as desired. In this work, we choose to alter the environment by adding *influencing agents* to the flock. These influencing agents — which could be in the form of robotic birds[1], robotic bees [10], or ultralight aircraft[2] — follow our algorithms but are perceived by the rest of the flock to be one of their own.

Following a well-recognized flocking model [9], we assume that each bird in the flock dynamically adjusts its heading based on that of its immediate neighbors. Previous work has considered how *randomly* placed influencing agents should behave so as to influence the flock to face a particular direction or maneuver along a path so as to avoid an obstacle [4, 5]. In our work, we examine where the influencing agents should be placed in the flock. Specifically, our current research question is: *where should influencing agents be located within a flock to maximize their influence on the flock?*

The remainder of this paper is organized as follows. Section 2 situates our research in the literature and Section 3 introduces our problem and necessary terminology. Section 4 reviews past work on how the influencing agents decide how to behave. Section 5 introduces our graph-based method for deciding where to place the influencing agents and Section 6 introduces our approach for control of the influencing agents. We discuss our experiments in Section 7 and then Section 8 concludes.

2. RELATED WORK

Reynolds introduced the flocking model that we use in this work [9]. Reynolds focused on creating a flocking model that looked and behaved realistically. His model consisted of three simple steering behaviors that determine how each agent behaves based on the agents around it. Vicsek *et al.* considered only one aspect of Reynolds' model in physics work that studied the self-emergent nature of flocking [12]. However, neither of these lines of research considered how to influence the flock to adopt a particular behavior by introducing agents into the flock.

[1]www.mybionicbird.com
[2]www.operationmigration.org

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May, 4–8, 2015, Istanbul, Turkey.*

Jadbabaie *et al.* considered the impact of adding a controllable agent to a flock [7]. They also used just one aspect of Reynolds' model and showed that a flock with a controllable agent will always converge to the controllable agent's heading. Su *et al.* also presented work that used a controllable agent to make the flock converge [11]. Celikkanat and Sahin used informed agents to lead the flock by their preference for a particular direction [1]. Our work is different from these lines of research in that they all influence the flock to converge to a target heading eventually, while we influence the flock to converge quickly.

Couzin *et al.* considered how animals in groups make informed, unanimous decisions [2]. They showed that only a very small proportion of informed agents is required for such decisions, and that the larger the group, the smaller the proportion of informed individuals required. Ferrante *et al.* used communication to coordinate a flock to move towards a common goal [3]. These two lines of research are different from ours because they do not consider how to control agents by considering and accounting for how the other agents will react. Instead, in these lines of research, each agent behaves in a fixed manner that is pre-decided or solely based on its type.

Han *et al.* assume that an influencing agent can be placed at any position at any time step [6]. Because of this assumption, the authors place the influencing agent at the position of the 'worst' flocking agent, which is the one that deviates from the desired orientation the most. In our work, we consider the assumption that the influencing agents can be placed as we want in the first time step. However, we do not allow teleporting and hence we can not continually place an influencing agent at the 'worst' flocking agent.

To the best of our knowledge, the work presented in this paper is the first that considers how to place controllable agents that aim to influence the flock towards a particular behavior into a flock.

3. PROBLEM DEFINITION

To fully define our problem, we must specify (1) a model of the flock, (2) the possible options for placing influencing agents initially, (3) the actions available to the influencing agents, and (4) the performance objective. This section does so, and also describes the concrete simulation environment that we use in our experiments. Our proposed methodologies for addressing the defined problem are presented in Sections 5 and 6.

3.1 Flocking Model

To model the flock, we use a simplified version of Reynolds' Boid algorithm for flocking [9] that is similar to the model utilized in our previous work [4]. Specifically, we only use the flock centering aspect of Reynolds' model, and do not use the collision avoidance and velocity matching aspects of his model.

The flock is comprised of two types of agents. Specifically, the n agents that comprise the flock consist of k *influencing* agents and m *flocking* agents, where $k + m = n$. The influencing agents $\{a_0, \ldots, a_{k-1}\}$ are agents whose behavior we can control, while the flocking agents $\{a_k, \ldots, a_{N-1}\}$ are agents that we cannot directly control.

Each agent in the flock has a velocity, a position in the environment, and an orientation. Each agent a_i moves with velocity v_i. At each time step t, each agent a_i has a position $p_i(t) = (x_i(t), y_i(t))$ in the environment and an orientation $\theta_i(t)$. Each agent's position $p_i(t)$ at time t is updated after its orientation is updated, such that $x_i(t) = x_i(t-1) + v_i \cos(\theta_i(t))$ and $y_i(t) = y_i(t-1) - v_i \sin(\theta_i(t))$. Hence, the current state of agent a_i at time t can be represented by its ($x_i(t)$, $y_i(t)$,$\theta_i(t)$) pose.

Agents in a flock update their orientations based on the orientations of the other agents in their *neighborhood*. Let $N_i(t)$ be the set of $n_i(t) \leq n$ agents (not including agent a_i) at time t which are located within a *visibility radius* r of agent a_i. This visibility radius defines each agent's *neighborhood*. The global orientation of agent a_i at time step $t+1$, $\theta_i(t+1)$, is set to be the average orientation of all agents in $N_i(t)$ (not including itself) at time t. Formally,

$$\theta_i(t+1) = \theta_i(t) + \frac{1}{n_i(t)} \sum_{a_j \in N_i(t)} \text{calcDiff}(\theta_j(t), \theta_i(t)) \quad (1)$$

We use Equation 1 instead of taking the average orientation of all agents because of the special cases handled by Algorithm 1. For example, the mathematical average of 350° and 10° is 180°, but by Algorithm 1 it is 0°. Throughout this paper, we restrict $\theta_i(t)$ to be within $[0, 2\pi)$.

Algorithm 1 calcDiff($\theta_i(t), \theta_j(t)$)

```
1: if ((θ_i(t) − θ_j(t) ≥ −π) ∧ (θ_i(t) − θ_j(t) ≤ π)) then
2:     return θ_i(t) − θ_j(t)
3: else if θ_i(t) − θ_j(t) < −π then
4:     return 2π + (θ_i(t) − θ_j(t))
5: else
6:     return (θ_i(t) − θ_j(t)) − 2π
```

3.2 Influencing Agent Initial Placement

The influencing agents join the flock in order to influence the flock to behave in a particular way. Our previous work only considered random placement of the influencing agents [5]. However, in this work we consider cases in which $\{p_0(0), \ldots, p_{k-1}(0)\}$ is under our control (Section 7.2) and cases in which it is not (Section 7.3).

In the cases where $\{p_0(0), \ldots, p_{k-1}(0)\}$ is under our control we may place the agents $\{a_0, \ldots, a_{k-1}\}$ wherever we wish. In the cases where it is not, we assume the agents $\{a_0, \ldots, a_{k-1}\}$ begin in designated starting spots and must then actively move to locate themselves within the flock, perhaps as the flock passes by their designated starting spots.

3.3 Influencing Agent Control

Influencing agents $\{a_0, \ldots, a_{k-1}\}$ are agents whose behavior we can control. Specifically, we control what type of behavior these agents display at each given time step: (1) control to *adjust position* or (2) control to *influence neighbors*. Our previous work only focused on control to influence neighbors [5]. In this work, Section 6 presents reasoning about control to adjust position and introduces a method to arbitrate between control to adjust position and control to influence neighbors.

3.4 Performance Representation

We define the *Agent Control and Placement Problem* as follows: Given a target orientation θ^* and a team of n agents $\{a_0, \ldots, a_{n-1}\}$, where the m flocking agents $\{a_k, \ldots a_{n-1}\}$ have positions $\gamma_m(t) = \{p_k(t), \ldots p_{n-1}(t)\}$ at time t and calculate their orientation based on Equation 1, determine placement $\pi(0)$ of influencing agents $\{a_0, \ldots a_{k-1}\}$ at time 0 and control $\Phi = \phi(0), \ldots, \phi(t)$ at times $t >= 0$ such that loss $l(\pi(0), \Phi)$ is minimized.

A k-agent placement specifies the positions that each influencing agent $\{a_0, \ldots a_{k-1}\}$ will take at time 0. The k-agent placement is denoted by $\pi_k(0) = \{p_0(0), \ldots, p_{k-1}(0)\}$ where $\{p_0(0), \ldots, p_{k-1}(0)\}$ is the set of positions for influencing agents $\{a_0, \ldots a_{k-1}\}$ at time 0.

We denote t^* as the earliest time step at which flocking agents $\{a_k, \ldots, a_{N-1}\}$ are oriented such that, for all $t \geq t^*$, $\{\theta_k(t), \ldots, \theta_{N-1}(t)\}$ are all within ϵ of θ^*. However, in some cases this will never occur because some of the m flocking agents may become permanently separated from the flock — we say these agents are *lost*. An agent a_i is considered lost if there exists a subset of flocking agents with cardinality $m' < m$ and orientations within ϵ of θ^* for more than 200 time steps and $|\theta_i(t^*) - \theta^*| > \epsilon$, where t^* is the time step at which the subset converged to θ^*. Let $\gamma_{m'}(t)$ denote the set of positions of the non-lost flocking agents at time t. Additionally, let $\alpha = \frac{\sum_{j \in \gamma_{m'}(t)} \|j - \overline{\gamma_{m'}(t)}\|}{m'}$. In other words, let α represent the average distance of the non-lost flocking agents at time t from the center of the flock.

The *loss* $l(\pi(0), \Phi)$ of a k-agent placement $\pi_k(0)$ and control Φ is a weighted function of four terms:

- w_1 is a weight that emphasizes the importance of minimizing the number of lost agents (minimize $\overline{m - m'}$)
- w_2 is a weight that emphasizes the importance of minimizing the number of simulation experiments in which any agent is lost (minimize simulation experiments in which $m - m' > 0$)
- w_3 is a weight that emphasizes the importance of minimizing the number of time steps needed for convergence (minimize $\overline{t^*}$)
- w_4 is a weight that emphasizes the importance of the flock being compactly spaced at time t^* (minimize $\overline{\alpha}$)

$$l(\pi(0), \Phi) = w_1\overline{m - m'} + w_2\overline{p(m - m' > 0)} + w_3\overline{t^*} + w_4\overline{\alpha} \tag{2}$$

An optimal placement $\pi^*(0), \Phi^*$ is one with minimal loss $l(\pi^*(0), \Phi)$.

In this work, we set $w_1 > w_2 > w_3 > w_4$. With these preferences for w_1, w_2, w_3, and w_4 we select influencing agent placements that generally lose the least number of agents on average but that also attempt to minimize the chances of losing any agents.

3.5 Simulation Environment

We situate our research within the MASON simulator [8]. This simulator encodes all the flock dynamics as described in this section and we augment it to compute the performance metric discussed Section 3.4. Pictures of the Flockers domain are shown in Figure 1. Each agent points and moves in the direction of its current velocity vector. We describe our experimental setup in detail in Section 7.1.

4. CONTROL OF INFLUENCING AGENTS

Birds in a flock dynamically update their headings based on the headings of their neighbors. In Section 3 we presented the models that we assume birds use when determining which nearby birds are in their neighborhood and when updating their headings. Our previous work has focused on how randomly placed influencing agents should behave so as to influence the flock towards a particular behavior [4, 5]. We shortly review this work below.

(a) (b)

Figure 1: Images of (a) the start of a trial and (b) the end of a trial. The gray agents are influencing agents while the black agents are other members of the flock.

Our first paper in this area considered how influencing agents should behave from a theoretical perspective [4]. This work presented a formal definition of the flocking model that served as a base for the model we use in the work presented in this paper. We presented multiple general flocking theorems that apply across all flocking scenarios before considering theorems specific to cases in which all agents are stationary ($v_i = 0 \;\forall\; i$) and cases in which only the influencing agents are non-stationary ($v_i = 0 \;\forall\; i \geq k$). In the case where all of the agents are stationary, we presented and then empirically evaluated an algorithm for determining the necessary orientations for influencing agents at each time step as well as the number of steps needed for the flocking agents to orient towards θ^*. In the case of non-stationary influencing agents attempting to influence stationary flocking agents, we empirically evaluated two heuristics for how the influencing agents should behave when they are not currently within the neighborhood of any flocking agents.

In more recent work we considered how to influence a large, non-stationary flock to (1) quickly *orient* towards a target orientation and (2) *maneuver* through turns quickly but with minimal agents becoming lost as a result of these turns [5]. We introduced a 1-step lookahead algorithm for determining the individual behavior of each influencing agent $a_i \in \{a_0, \ldots, a_{k-1}\}$. This 1-step lookahead algorithm considered all of the influences on the neighbors $N_i(t)$ of the influencing agent a_i and allowed the influencing agent to determine the best orientation to adopt (where best is defined as the behavior that exerts the most influence on the next step). We used this algorithm to determine the behavior of each influencing agent in empirical experiments, and showed that the 1-step lookahead algorithm did better in terms of the number of steps required for the flock to converge to θ^* than the baseline algorithm in both cases.

In the work presented in this paper, we used the 1-step lookahead algorithm from our previous work [5] as the behavior of the influencing agents once they are in their chosen positions within the flock. In order to maintain consistency with this previous work, in our experiments we set two control and task parameters in the same manner as we did previously. First, we only consider *numAngles* discrete angle choices for each influencing agent when performing 1-step lookahead, where *numAngles*=50. Thus the unit circle is equally divided into 50 segments beginning at 0 radians and each of these orientations is considered as a possible orientation for each influencing agent during 1-step lookahead. Second, we conclude that the flock has converged to θ^* when every agent (that is not an influencing agent or *lost*) is facing within 0.1 radians of θ^*.

5. DETERMINING INITIAL POSITIONS OF INFLUENCING AGENTS

The main contribution of this paper is a consideration of where to place the influencing agents $\{a_0, \ldots, a_{k-1}\}$ into the flock, assuming that once there, they will follow the 1-step lookahead behavior described above. We consider two different cases when determining how to place $a_i \in \{a_0, \ldots, a_{k-1}\}$ into the flock. In the **Drop** case, we are able to drop each influencing agent a_i into the flock at whatever location $p_i(0)$ we desire at time $t = 0$. In the **Dispatch** case, each influencing agent begins at one or more stations outside the flock at time $t = 0$ and is directed to travel to a particular location in the flock. Note that in the **Dispatch** case, each influencing agent will take time to reach its assigned location and may influence flocking agents along the way.

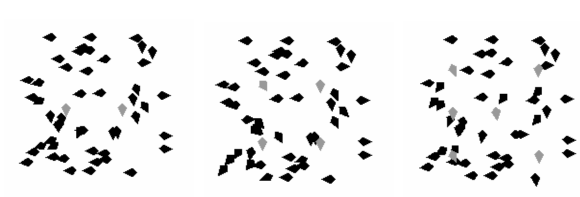
In the following subsections we discuss our approaches for placing influencing agents into the flock. Videos of these approaches are available on our web page.[3]

5.1 Random Approach

Our previous research randomly placed k influencing agents within the dimensions of the flock [5]. Hence, we use random placement as the base case for evaluating our placement approaches. Random placements are calculated for k influencing agents in constant time.

5.2 Grid Approach

Grid placement is another base case in which we place k influencing agents at predefined, well-spaced, gridded positions throughout flock. The placement of the influencing agents is dependent on the space covered by the flocking agents, and not on the positions of flocking agents. Hence, the placements of k influencing agents are determined in constant time. Grids are available that can fit at most x influencing agents, where we use the smallest grid in which $k \leq x$. Grids are available in which $x \in \{1, 2, 4, 9, 16, 25, 36, \ldots\}$. For each grid size, agents are spread out among the possible positions as much as possible. Examples of the grid approach for various values of k can be seen in Figure 2.



(a) $k = 2$ (b) $k = 4$ (c) $k = 6$

Figure 2: Images of influencing agent placement using the **grid approach** with (a) $k = 2$ (b) $k = 4$ and (c) $k = 6$. $n = 50$ in all three cases. The gray agents are influencing agents while the black agents are other members of the flock.

[3] `http://www.cs.utexas.edu/~larg/index.php/Placement_Into_a_Flock`

5.3 Border Approach

Our border approach works by placing k influencing agents as evenly as possible around the space covered by the flocking agents. As in the grid approach, the placement of the influencing agents is not dependent on the positions of flocking agents. Hence, the placements of k influencing agents are determined in constant time. We place influencing agents on the left side of the flock, right side of the flock, bottom of the flock, and top of the flock in order until all k influencing agents are placed. At most $\frac{k}{4}$ influencing agents are positioned on any particular side of the flock. If more than one influencing agent is placed on a particular side of the flock, the influencing agents spread out as much as possible on that side of the flock. Examples of the border approach for various values of k can be seen in Figure 3.

(a) $k = 2$ (b) $k = 4$ (c) $k = 6$

Figure 3: Images of influencing agent placement using the **border approach** with (a) $k = 2$ (b) $k = 4$ and (c) $k = 6$. $n = 50$ in all three cases. The gray agents are influencing agents while the black agents are other members of the flock.

5.4 Graph Approach

Our graph approach considers many possible k-sized sets of positions in which the k influencing agents could be placed, and then evaluates how well each of these sets connects the m flocking agents with the k influencing agents.

5.4.1 *Creating the Graph*

All $\{a_k, \ldots, a_{N-1}\}$ flocking agents are added to an initially empty graph G as nodes. Then, for each agent $a_i \in \{a_k, \ldots, a_{N-1}\}$, an undirected edge is added to G between a_i and each of its neighbors $a_b \in n_i(t)$ if such an edge does not already exist.

5.4.2 *Calculating Sets of Influencing Agent Positions*

Next we consider the positions at which we will consider adding influencing agents. For $a_i, a_j \in \{a_k, \ldots, a_{N-1}\}$, we consider adding an influencing agent at the mid-point $(\frac{x_i(t)+x_j(t)}{2}, \frac{y_i(t)+y_j(t)}{2})$ between $p_i(t)$ and $p_j(t)$ *only* if $p_i(t)$ and $p_j(t)$ are within $2r$ of each other. We consider this midpoint because placing an influencing agent here will allow the agent to influence both a_i and a_j. We also consider adding an influencing agent at $(x_i(t) + 0.1, y_i(t) + 0.1)$ for $a_i \in \{a_k, \ldots, a_{N-1}\}$ where $0.1 < r$. We consider this point, which is extremely close to a_i, since placing an influencing agent as this point will allow the agent to at least influence a_i. In cases where no or few flocking agents are within $2r$ of each other, placing influencing agents close to a flocking agent guaranteed that at least one flocking agent would be influenced by each influencing agent.

Once we have all of the positions at which we might add an influencing agents, we form all possible k-sized sets of these positions.

5.4.3 Evaluating Sets of Influencing Agent Positions

Finally, we take all possible k-sized sets and consider individually each set S of k influencing agent positions. In order to do this, we do the following for each S. Note that agents are 'directly' connected if they are neighbors and 'indirectly' connected if they have a neighbor, a neighbor of a neighbor, or a most distant agent in common.

- Add each influencing agent $a_i \in S$ to G
- For each agent $a_i \in S$, an edge is added to G between a_i and each of its neighbors $a_b \in n_i(t)$
- Run the Floyd Warshall shortest paths algorithm on G to obtain the following:
  - **numNoConn**: the number of flocking agents not connected to an influencing agent (directly or indirectly)
  - **numConn**: the number of connections between flocking agents and influencing agents (directly or indirectly)
  - **numDirectConn**: the number of direct connections between flocking agents and influencing agents
  - **numNoDirectConn**: the number of flocking agents not directly connected to an influencing agent
- Remove each influencing agent $a_i \in S$ from G

Once all possible k-sized sets T have been individually considered, we select a set based on the information we obtained. Specifically, we compare in order (lexicographically) criteria at four levels: (1) minimal **numNoConn**, (2) maximal **numConn**, (3) maximal **numDirectConn**, and (4) minimal **numNoDirectConn**. If only one set matches the criterion at a level, then we select it. Otherwise, all of the sets that matched the criterion at that level are considered at the next level. If multiple sets remain after the final level, we choose one of the remaining sets randomly.

In practice, we find that a set is usually selected using the first criterion. We have witnessed a few cases in which the fourth criterion has been used, but we have never witnessed a case in which the final criterion of selecting a remaining set randomly has been utilized.

The entire process of selecting placements for k influencing agents, given current placements of m flocking agents, has an algorithmic complexity of $O(n^3\binom{m^2+m}{k})$.

As can be seen in Figure 4, our graph approach places influencing agents in the areas in which their influence will be most impactful in decreasing loss of flocking agents.

(a) $k = 2$, $n = 10$ (b) $k = 4$, $n = 10$ (c) $k = 2$, $n = 20$

Figure 4: Images of influencing agent placement using the **graph approach** with (a) $k = 2$, $n = 10$ (b) $k = 4$, $n = 10$ and (c) $k = 2$, $n = 20$. The gray agents are influencing agents while the black agents are other members of the flock.

6. DETERMINING CONTROL OF INFLUENCING AGENTS

In the **Drop** case, the k influencing agents are at their desired positions at $t = 0$, so they can start influencing flocking agents directly via 1-step lookahead. In the **Dispatch** case, however, the influencing agents are initially positioned outside of the flock. They will only influence a limited number of flocking agents if they begin 1-step lookahead immediately. Hence, they must *reposition* themselves to their desired positions before they attempt to influence.

In the **Dispatch** case the influencing agents must approach and enter the flock in order to reach their desired positions and eventually influence more flocking agents. If they travel at the same speed as the flocking agents, they usually will not be able to catch up to the flocking agents. Instead, as the orientations of the flocking agents are affected by the influencing agents' directions, the flocking agents are likely to be driven away from the flock in the direction the influencing agents are traveling. Since the desired positions of the influencing agents are determined based on the current positions of the flocking agents, the influencing agents may never reach their desired positions if they travel at the same speed as the flocking agents. Hence, in the **Dispatch** case we assume the influencing agents are able to travel faster than the flocking agents. They do so when moving to their chosen position; when using 1-step lookahead to influence the flocking agents, they revert to the same speed as the flocking agents.

Influencing agents must always decide whether to attempt to move to their desired location *or* attempt to influence flocking agents in the **Dispatch** case. Once an influencing agent a_i reaches its desired position $p_i(t)$, it always enters the phase of influencing flocking agents via 1-step lookahead. However, behaving according to 1-step lookahead may cause a_i to leave its desired position (it could even leave the flock). As such, a_i should return to the repositioning phase if it strays too far from its desired position. However, if a_i switches between repositioning to its desired position and influencing flocking agents too frequently, it will oscillate between these two behaviors and not efficiently influence the flocking agents. To prevent excessive oscillation, we employ a hysteresis method to control the switch between these two phases. Specifically, when an influencing agent reaches a distance of h_1 away from its desired position, it returns to within h_2 of its desired position, where $h_1 > h_2$. In this way, the influencing agent will not reposition itself when it only deviates slightly from its desired position.

We examine the various types of desired positions described in Section 5 — random approach, grid approach, border approach, and graph approach. We use such placements as initial positions in the **Drop** case, and as desired positions in the **Dispatch** case. We report the empirical results in Section 7. However, we only report results for the graph approach in the **Drop** case. This is because as the desired positions need to be recalculated at each time step as the flocking agents update their positions, the graph approach was too computationally expensive to be evaluated in the **Dispatch** case. Moreover, the desired positions indicated by the graph approach are optimum points at one time step. They may change dramatically when the flock moves whereas the positions chosen by the border approach and grid approach change smoothly.

7. EXPERIMENTS

In this section we describe our experiments testing the various approaches for placing influencing agents into a flock in both the **Drop** case and the **Dispatch** case. We compare our novel approaches against baseline methods in both cases.

There are plenty of different metrics that can be used to access how 'good' a particular approach is — steps for the flock to converge, the number of trials in which any flocking agents were lost, the average number of flocking agents lost, and the average distance of the flocking agents from the center of flock at convergence are just some possible metrics. As discussed in Section 3.4, in this work we primarily focus on minimizing the average number of flocking agents lost. Our secondary focus is minimizing the number of trials in which any flocking agents are lost. Hence, for our experiments we set $w_1 = 0.45$, $w_2 = 0.4$, $w_3 = 0.1$, and $w_4 = 0.05$. However, our approach will be applicable as long as w_1 and w_2 are substantially greater than w_3 and w_4 — it is not sensitive to the exact values of these variables.

7.1 Experimental Setup

We utilize the MASON simulator [8] for our experiments in this paper. We introduced the MASON simulator in Section 3.5, but in this section we present the details of our experimental environment that are vital for completely understanding and replicating our experimental setup. We generally only discuss an experimental variable or control if we changed it from the default setting for the simulator.

The experimental settings for variables are given in Table 1 for both the **Drop** case and the **Dispatch** case.

Variable	**Drop Case**	**Dispatch Case**
toroidal domain	no	no
domain height	300	300
domain width	300	300
units moved by each flocking agent per time step ($v_k = \ldots = v_{N-1}$)	0.2	0.2
units moved by each influencing agent per time step ($v_0 = \ldots = v_{k-1}$)	0.2	0.2-1
number of agents in flock (n)	10-50	50
number of influencing agents (k)	1-5	5
neighborhood for each agent (radius)	*10*	*10*

Table 1: Experimental settings for variables in the **Drop** and **Dispatch** cases. Italicized values are default settings for the simulator.

Most of our experimental variables in Table 1, such as toroidal domain, domain height, domain width, and the units each agent moves per time step, are not set to the default settings for the MASON simulator. We chose to remove the toroidal nature of the domain in order to make the domain more realistic. Hence, if an agent moves off of one edge of our domain, it will not reappear. This is particularly important for *lost* agents remaining *lost*. We also increased the domain height and width, and decreased the units each agent moves per time step, in order to give agents a chance to converge with the flock before leaving the visible area. However, we have no reason to believe the exact values we chose for the experiments are of particular importance.

Flocking agents are randomly placed initially within a square in the top left of the domain, where this square occupies 4% of the domain. Agents are initially assigned random headings that are within 90 degrees of θ^*. If no influencing agents are added to the flock, some agents will likely be lost from the flock, the flock may separate into multiple smaller flocks, and the flock(s) will almost certainly not travel in the desired direction.

In all of our experiments, we run 100 trials for each experimental setting. We use the same set of 100 random seeds for each set of experiments for the purpose of variance reduction. The random seeds are used to determine the exact placement and orientation of all of the flocking agents at the start of a simulation experiment. In all of the graphs in the following sections, the error bars depict the sample standard deviation.

7.2 Drop Experimental Results

In the **Drop** case, we are able to drop each influencing agent a_i into the flock at whatever location $p_i(0)$ we desire.

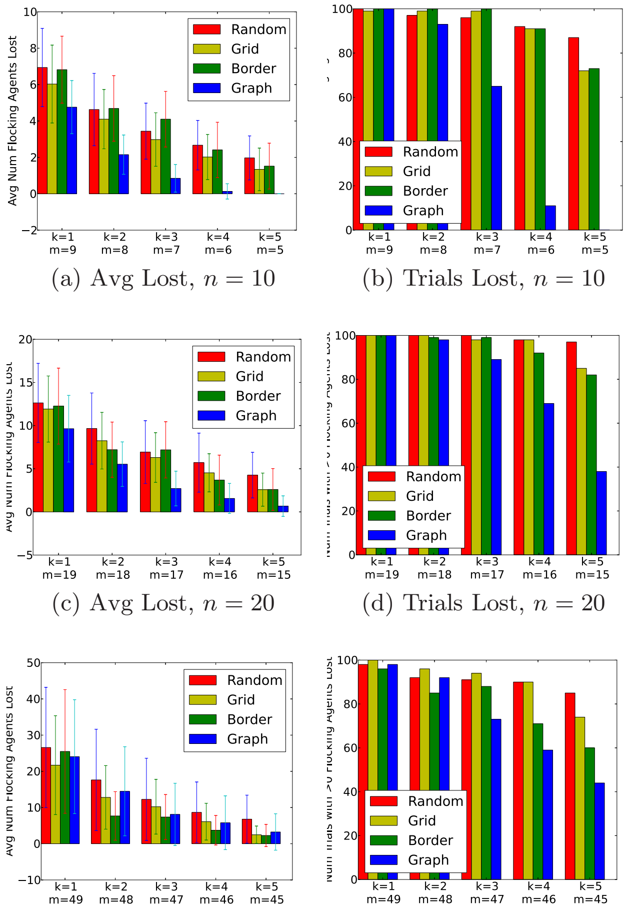
The left-hand side of Figure 5 shows graphs that depict the average number of flocking agents lost when $n = 10$, $n = 20$ and $n = 50$. We report results for $n = 50$ in order to allow comparison with results in Sections 7.2.1 and 7.3. However, results for flocks of $n = 10$ and $n = 20$ highlight the strength of the graph approach in the **Drop** case.

Note that the graph approach loses fewer flocking agents in comparison to the other approaches when the flock size is small ($n = 10$ and $n = 20$) because in these cases agents are more sparse in the environment and hence tend to have fewer neighbors. This is important because agents with fewer neighbors are less likely to be inadvertently influenced by distant influencing agents. Additionally, the graph approach also performs better than the other approaches when n is small and the percentage of influencing agents in the flock is high. This is because the graph approach focuses on minimizing the number of unconnected flocking agents, so as a higher percentage of the flock is composed of influencing agents, the number of unconnected — and hence uninfluenced — flocking agents will decrease quicker under the graph approach than under other the approaches. Finally, note that in Figure 5(a) no flocking agents are lost by the graph approach when 50% of the flock is composed of influencing agents. This is intuitively because the influencing agents will be placed such that each flocking agent is influenced by at least one influencing agent.

The right-hand side of Figure 5 shows how many trials out of 100 resulted in any flocking agents becoming lost. All of the methods begin to have more trials in which no flocking agents are lost as k increases. However, it is notable how stark the difference is between the number of trials in which no flocking agents are lost when using the graph approach versus any other approach. This supports our assertion that the graph approach places influencing agents in initial positions that minimize the number of disconnected — and hence uninfluenced — flocking agents.

7.2.1 *Drop + Reposition Experimental Results*

In the **Drop** case, the influencing agents are initially dropped at their desired positions. However, in this section we consider an extension to the **Drop** case which we call the **Drop + Reposition** case. In this case, we initially drop the influencing agents at their desired positions but also allow them to reposition as needed in order to stay near their desired positions. In order to reposition, the influencing agents employ the hysteresis method described in Section 6. To be consistent with Section 7.3, we set $h_1 = 5$ units and $h_2 = 3$ units for the border approach placements and $h_1 = 10$ units



(a) Avg Lost, $n = 10$ (b) Trials Lost, $n = 10$

(c) Avg Lost, $n = 20$ (d) Trials Lost, $n = 20$

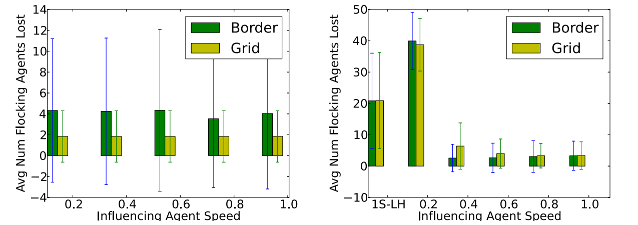
(e) Avg Lost, $n = 50$ (f) Trials Lost, $n = 50$

Figure 5: The average number of flocking agents lost (Avg Lost) when (a) $n = 10$ (c) $n = 20$ and (e) $n = 50$ and the number of trials in which any flocking agent was lost (Trials Lost) when (b) $n = 10$ (d) $n = 20$ and (f) $n = 50$. These results are obtained over 100 simulation experiments in the **Drop** case.

and $h_2 = 3$ for the grid approach placements

We present results using both the border approach and the grid approach for influencing agent placement in Figure 6(a). The influencing agents are initialized at their desired positions, and will return to these positions if they stray too far while attempting to influence the flock. Note that the difference in performance between different influencing agent speeds is small for both types of placement. This is because the influencing agents do not reposition very often, and since the speeds are only different while repositioning, the greater speeds do not have much affect on the results.

We can compare the results at speed **0.2** to the results for the border approach and the grid approach in Figure 5(e). These results look very similar, so we can conclude that, at least for these experimental settings, that the **Drop + Reposition** case does not provide any significant benefit over the **Drop** case. This is likely because there is not a significant need in this experimental setting for the influencing agents to reposition as they do not stray far from their desired positions. In the grid approach, the flock often converges before the influencing agents deviate from their desired positions by h_1 units. Hence, repositioning is rarely



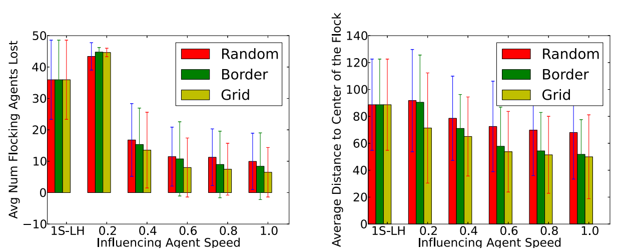
(a) **Drop** + **Reposition** case (b) **Dispatch** case

Figure 6: The average number of lost agents when $m = 45$ and $k = 5$. In (b), the influencing agents are initialized at two opposite corners.

triggered. In the border approach, however, the influencing agents sometimes travel away from their border positions, so repositioning is activated more frequently.

7.3 Dispatch Experimental Results

In the **Dispatch** case we initialize all of the influencing agents at one corner of the flock. The influencing agents first travel to their desired positions and then they focus on influencing the surrounding flocking agents in the **Dispatch** case. We use the hysteresis method described in Section 6 to control the switching of the influencing agents between repositioning to their desired position and attempting to influence nearby flocking agents. Based on a small amount of informal experimentation, we set $h_1 = 5$ units and $h_2 = 3$ units for the border approach and $h_1 = 10$ units and $h_2 = 3$ for the grid approach. In the border approach, the influencing agents are on the border and are more likely to leave the flock, so h_1 is smaller. In the grid approach, we are more tolerant on the distance that the influencing agents are away from their desired positions. In fact, a smaller h_1 in this case makes the influencing agents unnecessarily reposition inside the flock, which worsens performance. We do not use hysteresis for the random approach as there is no need to stay close to randomly chosen positions.



(a) Avg Lost, $n = 50$ (b) Avg Dist to Center

Figure 7: The average number of lost agents (Avg Lost) and the average distance to the center of the flock (Avg Dist to Center) when $m = 45$ and $k = 5$ in the **Dispatch** case.

In Figure 7 we show results in which the influencing agents are dispatched to three different types of positions: (1) random, (2) border and (3) grid. These positions are set as described in Section 5.

With regard to the x-axis labels of Figure 7, recall that in Section 6 we noted that in the **Dispatch** case, the influencing agents must move faster than the flocking agents when repositioning or otherwise the influencing agents may never reach their positions. In this graph we consider the effect of different influencing agent speeds on the number of lost flocking agents. **1S-LH** represents performance when, instead of repositioning, the influencing agents attempt to influence the flocking agents via performing 1-step lookahead from their initial positions. The flocking agents move 0.2 units per step, so **0.2** represents performance when the influencing agents move at the same speed as the flocking agents. Likewise, **0.4**, **0.6**, **0.8** and **1.0** represent the cases in which the influencing agents move two, three, four and five times as fast as the flocking agents, respectively.

A few trends stand out in the graphs of Figure 7; we discuss each of these trends in the following paragraphs.

We first consider Figure 7(a). These graphs show the average number of lost flocking agents in a flock where $m = 45$ and $k = 5$. As would be expected, **1S-LH** performs the same in each case since the influencing agents do not attempt to reposition. When $v_0 = \ldots = v_{n-1} = 0.2$, on average more flocking agents are lost than in **1S-LH** where the influencing agents do not attempt to reposition. This is because as the influencing agents enter the flock while traveling at the same velocity as the flocking agents, they push the flocking agents away and hence lose the entire flock in most cases. Across all of the positioning methods, faster speeds of the influencing agents results in fewer flocking agents becoming lost since the quicker speeds reduce the influence of the influencing agents on the flock while they are repositioning. However, the grid placement approach clearly loses fewer flocking agents across all of the cases in which the influencing agents are able to move quicker than the flocking agents.

Next we consider Figure 7(b). These graphs show the average distance of the flocking agents from the center of the flock at the point at which the flock converges. Faster influencing agents are able to reach their desired positions and begin influencing the flock towards θ^* much faster, which makes the flock more compact at convergence time.

When we consider both of the graphs from Figure 7 together, we conclude that the grid placement approach is best for this experimental setting in the **Dispatch** case. The grid approach loses fewer flocking agents than the border approach because the influencing agents in the border approach must travel longer distances across the flock in order to reach their desired positions. They must travel farther because some of the border positions are inherently located farther from the influencing agents' starting spot since they are located on the borders of the flock. This long distance traveling across the flock leads to lost flocking agents because it tends to push the flock to travel in the direction the influencing agents are traveling. This forces the influencing agents to work to reverse this effect once they are at their desired positions.

7.3.1 Multiple Initialization Points

We assume in the beginning of this section that the influencing agents are initialized at one point outside of the flock, which is, concretely, at one corner. Naturally, they can be initialized at multiple points instead of just one. In Figure 6(b), we examine the case in which the influencing agents are initialized at two opposite corners of the flock. In both the grid approach case and the border approach case, we initialize each influencing agent at the corner that is closest to its desired position. Initializing influencing agents in multiple corners generally has better performance than just initializing all agents at one corner. This is because the influencing agents do not have to travel as far to reach their desired positions. We leave a more thorough examination of multiple initialization points for future work.

7.4 Discussion

One impact of this work is to take a step towards the practical realization of influencing a flock by extending our previous positive results that showed that flocks could indeed be controlled if influencing agents are placed randomly in the flock [5]. In particular, the work presented in this paper contributes by determining that performance (1) can improve if we are able to control the initial positions of the influencing agents as in the **Drop** case and (2) is still generally positive even if influencing agents must be initialized outside the flock as in the **Dispatch** case.

The time that the influencing agents consume to travel to their desired destinations dominates in the **Dispatch** case. The border approach and grid approach have similar performance in the **Drop** case, as shown in Section 7.2. Their performance differs in the **Dispatch** case where the influencing agents take longer to reach their desired positions in the border approach. Our experiment in which influencing agents were dispatched from two initialization points shows that when less time is consumed traveling to desired positions, the border approach loses slightly fewer flocking agents on average than the grid approach when the influencing agents travel at faster speeds than the flocking agents.

Of course, we cannot completely compare the **Drop** case to the **Dispatch** case without being able to run the graph approach in the **Dispatch** case as well. The solid performance of the graph approach in the **Drop** case hints that it may perform well in the **Dispatch** case. We leave the task of implementing a more efficient or approximate graph approach for future work.

8. CONCLUSION

In this paper we consider where to place influencing agents that we add to a flock comprised of agents which we have no direct control over, but that we wish to influence towards a particular behavior. We present multiple methodologies as well as experimental results for placing influencing agents into a flock in two cases: (1) where we initially place the agents anywhere and (2) where the agents must travel to their desired positions after being initially placed outside the flock. Experimental results demonstrate that in (1), our graph approach performs better than other approaches in terms of the number of trials in which flocking agents are lost and the average number of flocking agents lost. In (2), the grid approach performs best in terms of the average number of lost flocking agents and flock compactness.

Positioning of influencing agents in a flock is a fertile research area with plenty of opportunities for future work. In our future work we plan to design a more efficient graph-based placement approach for use in the **Dispatch** case. Additionally, we intend to further consider the effect of utilizing multiple stations from which the influencing agents could emerge in the **Dispatch** case.

9. ACKNOWLEDGEMENTS

This work has taken place in the Learning Agents Research Group (LARG) at UT Austin. LARG research is supported in part by NSF (CNS-1330072, CNS-1305287), ONR (21C184-01), and AFOSR (FA8750-14-1-0070, FA9550-14-1-0087).

REFERENCES

[1] H. Celikkanat and E. Sahin. Steering self-organized robot flocks through externally guided individuals. *Neural Computing and Applications*, 19(6):849–865, September 2010.

[2] I. D. Couzin, J. Krause, N. R. Franks, and S. A. Levin. Effective leadership and decision-making in animal groups on the move. *Nature*, 433(7025):513–516, February 2005.

[3] E. Ferrante, A. E. Turgut, N. Mathews, M. Birattari, and M. Dorigo. Flocking in stationary and non-stationary environments: A novel communication strategy for heading alignment. In *Proceedings of the 11th International Conference on Parallel Problem Solving from Nature: Part II*, PPSN'10, pages 331–340. Springer-Verlag, 2010.

[4] K. Genter, N. Agmon, and P. Stone. Ad hoc teamwork for leading a flock. In *AAMAS'13*, May 2013.

[5] K. Genter and P. Stone. Influencing a flock via ad hoc teamwork. In *Ninth International Conference on Swarm Intelligence (ANTS'14)*, September 2014.

[6] J. Han, M. Li, and L. Guo. Soft control on collective behavior of a group of autonomous agents by a shill agent. *Systems Science and Complexity*, 19:54–62, 2006.

[7] A. Jadbabaie, J. Lin, and A. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988 – 1001, Jun 2003.

[8] S. Luke, C. Cioffi-Revilla, L. Panait, K. Sullivan, and G. Balan. Mason: A multi-agent simulation environment. *Simulation: Transactions of the Society for Modeling and Simulation International*, 81(7):517–527, 2005.

[9] C. W. Reynolds. Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH*, 21:25–34, August 1987.

[10] J. Shang, S. Combes, B. Finio, and R. Wood. Artificial insect wings of diverse morphology for flapping-wing micro air vehicles. *Bioinspiration & Biomimetics*, 4(3), 2009.

[11] H. Su, X. Wang, and Z. Lin. Flocking of multi-agents with a virtual leader. *IEEE Transactions on Automatic Control*, 54(2):293–307, Feb 2009.

[12] T. Vicsek, A. Czirok, E. Ben-Jacob, I. Cohen, and O. Sochet. Novel type of phase transition in a system of self-driven particles. *PHYS REV LETT.*, 75:1226, 1995.