Adding Influencing Agents to a Flock

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

ABSTRACT

Many different animals, including birds and fish, exhibit a collective behavior known as flocking. Flocking behavior is believed by biologists to emerge from relatively simple local control rules utilized by each individual in a flock. Specifically, each individual adjusts its behavior based on the behaviors of its closest neighbors. In our work we consider the possibility of adding a small set of influencing agents, which are under our control, to a flock. Specifically, we advance existing work on adding influencing agents into a flock and begin to consider the case in which influencing agents must join a flock in motion. Following ad hoc teamwork methodology, we assume that we are given knowledge of, but no direct control over, the rest of the flock. As such, we use the influencing agents to alter the flock's behavior — for example by encouraging all of the individuals to face the same direction or by altering the trajectory of the flock. In this paper we define several new methods for adding influencing agents into the flock and compare them against existing methods.

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

As teams of robots become useful in real-world settings, it becomes increasingly likely that robots that are not programmed to explicitly be part of these teams will need to join or otherwise assist these teams. This need could arise because some members of the team become damaged or lost, and there is neither the time nor the expertise to program new robots to add to the team. In this case, an ideal solution would be to add general agents to the team that can perform well as part of the team with no pre-coordination. Such agents are referred to in literature as *ad hoc agents* [11]. In this paper, we consider how ad hoc agents can be used in a flocking setting.

Flocking is an emergent swarm behavior found in nature. Each animal in a flock follows a simple local behavior rule, but these simple behaviors often result in group behavior that appears well organized and stable. Flocking has been studied in various disciplines including physics [13], graphics [9], biology [2], and distributed control theory [6, 7, 12] with the goal of characterizing its emergent behavior. In our work, we instead consider how to influence a flock to adopt particular behaviors by adding a small number of controllable agents to the flock.

As a motivating example, imagine that a flock of 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 flock's flight path, 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.

In our work, we follow a well-recognized flocking model [9] when we assume that each bird in the flock updates its heading at each time step based on the headings of its neighbors. In previous work we have considered how influencing agents should behave so as to influence the flock [3, 4] as well as where influencing agents should be placed in the flock if they were somehow able to join the flock instantaneously [5]. In this paper, we (1) extend our work on determining where influencing agents should be placed in the flock (henceforth called the *Placement* case) and (2) consider how influencing agents should behave in order to join a flock in motion (henceforth called the *Joining* case). As such, the research questions addressed by this paper are: *Given computational limitations, how should influencing agents be placed within a flock?* and *How should influencing agents join a flock in motion if they are able to arrive ahead of the flock?*

The remainder of this paper is organized as follows. Section 2 situates our research in the literature. Section 3 introduces the problem of adding influencing agents to a flock. Section 4 overviews existing methods. Section 5 details the

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

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

experimental set-up. Section 6 introduces and evaluates our improved methods for deciding where to place the influencing agents. Section 7 introduces and evaluates our work on joining a flock in motion. Finally, Section 8 concludes.

2. RELATED WORK

To the best of our knowledge, our work is the only work so far that has considered how to add controllable agents to a flock with the goal of using these controllable agents to influence the flock towards a particular behavior. This section highlights the work most related to our own.

In a more complete version of the flocking model used in this work, Reynolds focused on creating a flocking model that behaved realistically [9]. Reynolds' model consists of three simple steering behaviors that determine how each agent maneuvers based on the behavior of the agents around it (henceforth called *neighbors*): *Collision Avoidance* steers the agent such that it avoids collisions with neighbors, *Velocity Matching* moves the agents at a velocity similar to nearby neighbors, and *Flock Centering* steers the agent towards the average heading and position of its neighbors. Vicsek *et al.* considered just part of the *Flock Centering* aspect of Reynolds' model [13]. Hence, like in our work, Vicsek *et al.* use a model where all of the particles move at a constant velocity and adopt the average direction neighboring particles. However, like Reynolds' work, Vicsek *et al.* were only concerned with simulating flock behavior and not with adding controllable agents to the flock.

Jadbabaie *et al.* considered the impact of adding a controllable agent to a flock [7]. Like Vicsek *et al.* , they also used part of the *Flock Centering* aspect of Reynolds' model. Their work showed that a flock with a controllable agent will always converge to the controllable agent's heading. Su *et al.* presented work that used a controllable agent to make a flock converge [12] and Celikkanat and Sahin used informed agents to lead a flock by their preference for a particular direction [1]. However, our work is different from all of these lines of research in that they all influence a flock to converge to a target heading *eventually*, while in our work we influence a 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. This line of research is different from ours because they do not study how to control agents by accounting for how the other agents will react. Instead, in this line of research, each agent behaves in a fixed manner that is pre-decided or solely based on its type.

Finally, 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 repeatedly 'teleport' the influencing agent to the position of the 'worst' flocking agent, which is the one that deviates from the desired orientation the most. In our work, we do not allow teleporting and hence we cannot continually place an influencing agent at the 'worst' flocking agent.

3. PROBLEM DEFINITION

To precisely introduce and define the problem of adding influencing agents to a flock, we must specify (1) a model of the flock, (2) the possible options for adding influencing agents to a flock, (3) the actions available to the influencing agents, and (4) the performance objective. This section outlines these specifications and introduces the simulation environment we use in our experiments. Previous methodologies for addressing the defined problem are reviewed in Section 4 while our new methods are presented in Sections 6 and 7.

3.1 Flocking Model

In order to model the flock, we use a simplified version of Reynolds' Boid algorithm for flocking [9]. Specifically, the simplified version only utilizes part of the *Flock Centering* aspect of Reynolds' model and does not use the *Collision Avoidance* and *Velocity Matching* aspects. Although this model is similar to the model utilized in previous work [3, 5], we present the important details of the model to both include new aspects of our model as well as to help this paper stand on its own.

We assume that the flock is comprised of two types of agents: k *influencing* agents and m *flocking* agents. The flock thus contains $n = k+m$ total agents. The k influencing agents $\{a_0, \ldots, a_{k-1}\}$ are agents whose behavior we control via the algorithms presented in our work. The m flocking agents $\{a_k, \ldots, a_{N-1}\}$ are agents that we cannot directly control but that instead behave according to the simplified version of Reynolds' flocking algorithm.

Every agent a_i in the flock has a velocity $v_i(t)$, a position in the environment $p_i(t)$, and an orientation $\theta_i(t)$ at time t. Each agent's position $p_i(t)$ at time t is updated after its orientation is updated, such that $x_i(t) = x_i(t-1) + v_i(t)\cos(\theta_i(t))$ and $y_i(t) = y_i(t-1) - v_i(t)\sin(\theta_i(t))$.

As is commonly accepted in most flocking models, we assume that the agents in a flock are only influenced by the other agents in their *neighborhood*. We use a *visibility radius* to define each agent's *neighborhood*. Specifically we let $N_i(t)$ be the set of $n_i(t) \leq n$ agents at time t which are located within the *visibility radius* r of agent a_i. All agents in $N_i(t)$ are considered to be neighbors of a_i.

Under the simplified version of Reynolds' model that we employ, we assume that agents in a flock update their orientations based on the orientations of the other agents in their neighborhood. Hence, 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 $N_i(t)$ 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 ((θi(t) − θj(t) ≥ −π) ∧ (θi(t) − θj(t) ≤ π)) then
2:     return θi(t) − θj(t)
3: else if θi(t) − θj(t) < −π then
4:     return 2π + (θi(t) − θj(t))
5: else
6:     return (θi(t) − θj(t)) − 2π
```

3.2 Adding Influencing Agents

Influencing agents are added to the flock in order to influence the flock to behave in a particular way. Our previous work considered four methods for initially placing influencing agents directly into a flock by setting $\{p_0(0), \ldots, p_{k-1}(0)\}$ [5]. We overview these methods in Section 4 before considering improvements to these methods in Section 6. Additionally, in Section 7 we consider the case where $\{p_0(0), \ldots, p_{k-1}(0)\}$ is not under our control.

In the *Placement* case where $\{p_0(0), \ldots, p_{k-1}(0)\}$ is under our control we place the influencing agents $\{a_0, \ldots, a_{k-1}\}$ wherever we wish at $t = 0$. In the *Joining* case where $\{p_0(0), \ldots, p_{k-1}(0)\}$ is not under our control, we assume the influencing agents $\{a_0, \ldots, a_{k-1}\}$ begin from a designated station and must then actively move to join the flock.

3.3 Influencing Agent Behavior

We can control the behavior of influencing agents $\{a_0, \ldots, a_{k-1}\}$. Specifically, influencing agents can behave to either (1) influence neighbors or (2) position to a desired location. Hence, at each time step our algorithms for the influencing agents must decide whether to influence or position, as well as how to best execute the chosen behavior.

In this paper, Section 4 reviews the algorithm the influencing agents use in this work to influence neighbors. Then, Section 7 presents some new methods for the influencing agents to utilize while attempting to join a flock in motion.

3.4 Performance Representation

In previous work we defined the *Agent Control and Placement Problem* [5]. In this work we present generalizations that allow influencing agents to travel to their desired positions instead of being placed in these positions. Specifically, the generalized *Agent Control and Placement Problem* is stated 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 the desired influencing position $\pi(t_i)$ of influencing agents $\{a_0, \ldots a_{k-1}\}$ at time t_i and control $\Phi = \phi(0), \ldots, \phi(t)$ at times $t >= 0$ such that loss $l(\pi(t_i), \Phi)$ is minimized.

A k-agent placement specifies the positions that each influencing agent $\{a_0, \ldots a_{k-1}\}$ will adopt at time t_i, where time t_i is the time at which the influencing agents begin attempting to influence their neighbors. The k-agent placement is denoted by $\pi_k(t_i) = \{p_0(t_i), \ldots, p_{k-1}(t_i)\}$ where $\{p_0(t_i), \ldots, p_{k-1}(t_i)\}$ is the set of positions for influencing agents $\{a_0, \ldots a_{k-1}\}$ at time t_i.

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 can not occur because some 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^*.

The *loss* $l(\pi(t_i), \Phi)$ of a k-agent placement $\pi_k(t_i)$ and control Φ is a weighted function of three 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 $\overline{t^*}$)

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

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

In this work, we set $w_1 > w_2 > w_3$. With these preferences for w_1, w_2, and w_3 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 use the MASON simulator [8] to run the experiments described in this paper. We altered this simulator to encode the flock dynamics described above as well as to compute the performance metric also described above. Pictures of our altered version of the MASON simulator's Flockers domain are shown in Figure 1. In the simulator, each agent orients and moves in the direction of its current velocity vector. We describe our experimental setup in detail in Section 5.

(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.

4. EXISTING BEHAVIOR AND PLACEMENT METHODS

The research presented in this paper utilizes and compares against multiple behavior and placement methods from previous work [3, 5]. In this section, we present high-level overviews of these methods with the goal of making our own methods and experiments that rely on these methods approachable.

4.1 1-Step Lookahead Behavior

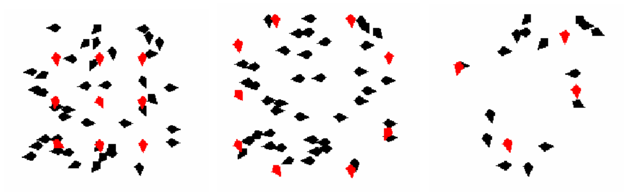
The 1-Step Lookahead behavior specifies how an influencing agent should behave in order to influence its neighbors to orient towards a particular heading [3]. Specifically, the

1-Step Lookahead behavior considers all of the influences on neighbors of the influencing agent at a particular point in time. By considering the influences on neighbors, the influencing agent can determine the orientation to adopt that will exert the most influence on the flock during the next time step. The 1-Step Lookahead behavior was found in previous work to be the best behavior in terms of the trade-off between performance and algorithmic complexity. Hence, this is why we chose it to be used by influencing agents in the research described in this paper.

4.2 Placement Methods

Previous work considered where to place influencing agents $\{a_0, \ldots, a_{k-1}\}$ into the flock at time $t = 0$ [5]. We assume that once the influencing agents are placed into the flock, they will immediately follow the 1-Step Lookahead behavior described in Section 4.1 to influence the flock.

In this section, we review the four initial placement methods presented in previous work. For each of these placement methods, we assume that the m flocking agents are initially placed within a pre-set area that is formed by the area in which the flocking agents could initially be placed at time $t = 0$ — we refer to this pre-set area as FA_{preset}.



(a) Grid (b) Border (c) Graph

Figure 2: Examples of three of the influencing agent placement methods from previous work.

4.2.1 Random Approach

The Random placement method serves as a simple baseline approach. Specifically, it randomly places the k influencing agents within FA_{preset}. These influencing agent placements are calculated in constant time.

4.2.2 Grid Approach

The Grid placement method places k influencing agents at predefined, well-spaced, gridded positions within FA_{preset}. The placement of the influencing agents is dependent on FA_{preset} and not on the actual positions of the flocking agents. Hence, the placement of k influencing agents is determined in constant time. Grids are available that can fit at most x influencing agents, where the smallest grid in which $k \leq x$ is used. 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.

4.2.3 Border Approach

The Border placement method places k influencing agents as evenly as possible along the borders of FA_{preset}. 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. The Border placement method places 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 $\lceil \frac{k}{4} \rceil$ influencing agents are placed 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.

4.2.4 Graph Approach

The 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. Specifically, it considers adding influencing agents at two different types of positions: (1) mid-points between flocking agents that are within two neighborhood radii of each other and (2) extremely near each flocking agent. Each possible k-sized set of positions is then evaluated in terms of three criteria: (1) minimize the number of flocking agents to which any influencing agent's influence will not spread, (2) maximize the flocking agents that are influenced (both immediately and over time) by influencing agents, and (3) maximize the number of flocking agents that have an influencing agent as a neighbor. The complexity of placing k influencing agents using the Graph placement method is $O(n^3 \binom{m^2+m}{k})$.

5. 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 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. We introduce our experimental setup at this point in the paper so that we can present experimental results throughout the remainder of the paper.

The relevant experimental variables for both the **Placement** case and the **Joining** case are given in Table 1 .

Variable	Placement Case	Joining Case
toroidal domain	no	no
domain height	300	600
domain width	300	600
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	0-0.2
neighborhood for each agent (radius)	*10*	*10*

Table 1: Experimental variables for the **Placement** and **Joining** 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 removed the toroidal nature of the domain in order to make the domain more realistic. Hence, if an agent moves off of an 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 experimental settings we chose for the experiments are of particular importance.

All of the experiments reported in this paper used $m = 10$ flocking agents and $k = 2$ to $k = 10$ influencing agents. During our initial research, we ran smaller scale experiments with as many as $m = 50$ flocking agents and $k = 25$ influencing agents. We did not run full scale experiments for flocks with more than $m = 10$ flocking agents mainly due to the high computation time required for the Graph placement approach — this computation issue was the main motivation behind the hybrid approach presented in Section 6.2. As such, we decided to maintain consistency by utilizing $m = 10$ flocking agents for all our experiments in this paper. However, our limited experiments did indicate that results from smaller flocks generally do scale to larger flocks.

In all of our experiments, the flocking agents are initially randomly placed within FA_{preset}, which is a small square at the top left of the environment. Agents are initially assigned random headings that are within 90 degrees of θ^* for the *Placement* case. For the *Joining* case, the flocking agents all begin facing the same orientation (not equal to θ^*) — in our experiments reported in this paper, this particular orientation was 90 degrees away from θ^*.

Experiments and experimental results will be presented throughout the following sections. In all of our experiments, we run 100 trials for each experimental setting and we use the same set of 100 random seeds for each set of experiments. The random seeds are used to determine the exact placement (and orientation for *Placement* experiments) of all of the flocking agents at the start of a simulation experiment. The error bars in all of our graphs depict the standard error of the mean.

6. PLACEMENT IN A FLOCK

Section 4 summarized various published methods for placing influencing agents into a flock at $t = 0$, with the goal of placement being for the influencing agents to influence the flocking agents to orient towards a particular orientation. In this section, we introduce an effective extension to the methods discussed in Section 4.2 as well as a new approach that mitigates the high computational complexity of the Graph placement method reviewed in Section 4.2.4.

6.1 Scaling Placement Area

Three of the placement methods summarized in Section 4.2 — the Random placement approach, the Grid placement approach, and the Border placement approach — do not place influencing agents based on where the flocking agents are initially placed. Instead, these methods all place the influencing agents in predetermined locations based on (1) how many influencing agents are to be added and (2) FA_{preset}.

In order to improve upon these three placement approaches, we decided to scale the area in which the influencing agents are placed based on the area actually occupied by the flocking agents. Specifically, we search all of the locations of the flocking agents and save the highest and lowest x and y values at which flocking agents are located (x_{low}, x_{high}, y_{low}, and y_{high}). We then extend these x and y values by r (the neighborhood radii) and then use the rectangular box formed by $x_{low} - r$, $x_{high} + r$, $y_{low} - r$, and $y_{high} + r$ as the area in which influencing agents can be placed. We call this area FA_{scaled}.

For the *Random Scaled Approach*, we randomly place influencing agents within FA_{scaled}. Similarly, for the *Border Scaled Approach* we spread influencing agents along the borders of FA_{scaled}. For the *Grid Scaled Approach*, we place the k influencing agents at predefined, well-spaced, gridded positions within FA_{scaled}. Within FA_{scaled} we follow the technique for the Grid placement approach from Section 4.2.2. Specifically, we spread out the influencing agents among the possible grid positions as much as possible. Note that there is not a *Graph Scaled Approach* because the Graph Placement Approach already places influencing agents based on where the flocking agents are placed.

In this section we run *Placement* case experiments to compare the performance of the initial placement methods summarized in Section 4.2 against the scaled versions of these methods presented in Section 6.1.

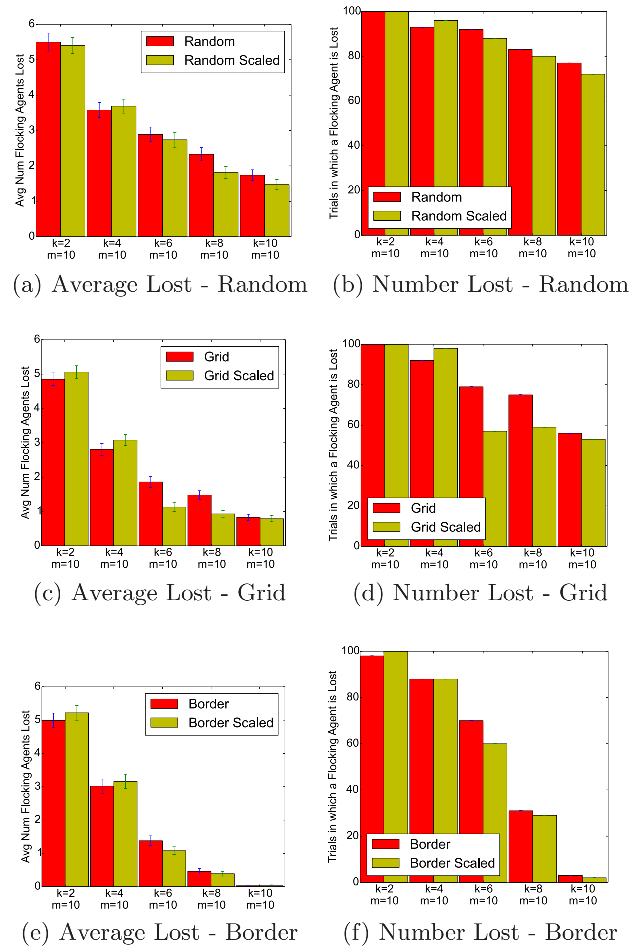


Figure 3: Results for the Scaling Placement Area experiments. These graphs compare (a,c,e) the average number of flocking agents lost and (b,d,f) the number of trials in which any flocking agents are lost.

Figure 3(a,c,e) shows the average number of flocking agents lost and Figure 3(b,d,f) shows the total number of trials in which at least one flocking agent is lost. The graphs show these data for six different placement methods when adding $k = 2$ to $k = 10$ influencing agents to a flock with $m = 10$ flocking agents. In the graphs, 'Random' should be compared to ‘Random Scaled’, ‘Grid’ should be compared to ‘Grid Scaled’, and ‘Border’ should be compared to ‘Border Scaled’.

A few interesting trends arise in Figure 3. First, the Grid Scaled and Border Scaled lose more flocking agents on average than their non-scaled counterparts when $k = 2$ and $k = 4$ but lose fewer flocking agents on average than their non-scaled counterparts when $k = 6$, $k = 8$, and $k = 10$ (significantly so for $k = 6$ and $k = 8$ of the Grid placement approach). However, the difference between the scaled and non-scaled approaches are not as large for any of the three placement approaches when $k = 2$, $k = 4$, and $k = 10$. Second, in terms of the number of trials in which no flocking agents are lost, Figure 3(d) shows that the Grid Scaled approach has many more trials in which no flocking agents are lost than its non-scaled counterpart when $k = 6$ and $k = 8$. Additionally — outside of $k = 6$ for the Border placement approach — this difference between the scaled and non-scaled methods does not appear for the other approaches. Both of these trends lead us to believe that influencing agent placement may be less important when there are either few or many influencing agents added to a flock. Of course, the actual k and m values where the scaled approaches perform much better than the non-scaled approaches will vary based on the exact experimental settings.

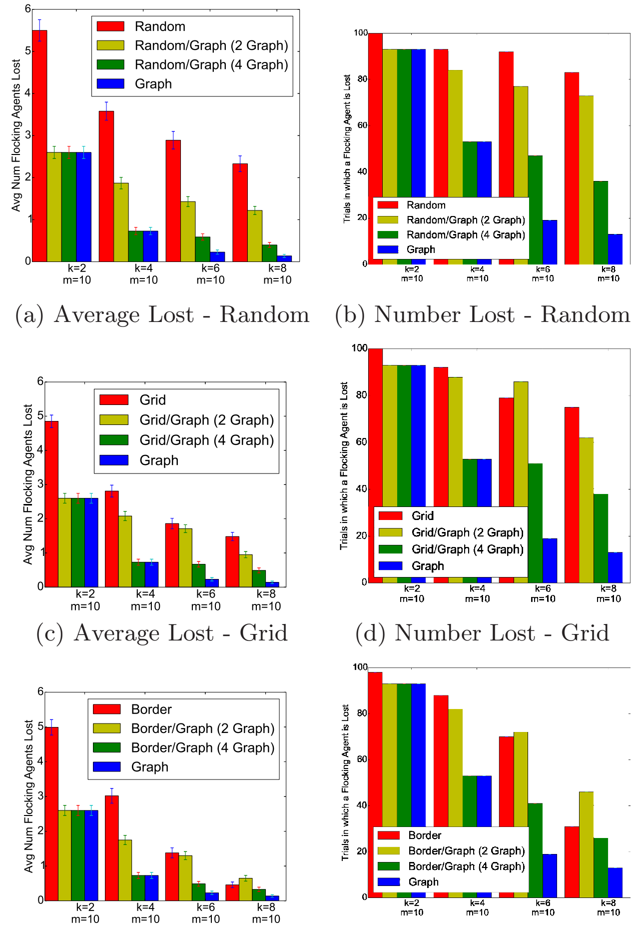
6.2 Combining Placement Methods for Better Scalability

The Graph placement approach summarized in Section 4.2.4 was shown in previous work [5] to perform better than other approaches in terms of agents lost and trials in which any flocking agents were lost. However, the Graph placement method was not widely and generally useful because the $O(n^3\binom{m^2+m}{k})$ computational complexity limited the sizes of the flocks in which it could be quickly applied.

With this computational complexity issue in mind, in this paper we evaluate hybrid approaches that utilize the Graph placement method to pick the first k_g influencing agent positions and then assign the remaining $k - k_g$ positions based on more computationally efficient methods. The remaining $k - k_g$ positions are randomly chosen from the possible k placements of the more computationally efficient method.

In our experiments, we compare multiple values of k_g as well as multiple placement methods to assign the $k - k_g$ placements not assigned by the Graph placement method. We compared the four initial placement methods summarized in Section 4.2 against hybrid approaches that combine the Graph placement approach with more computationally efficient approaches. These experiments were run under the *Placement* case settings described in Section 5.

Figure 4(a,c,e) shows the average number of flocking agents lost when initially placing influencing agents according to the Baseline, Graph, and hybrid placement approaches. ‘Baseline’ refers to the initial placement method that is used, chosen from the Random placement approach, the Grid placement approach, and the Border placement approach. The Baseline/Graph (2 Graph) hybrid placement approach places two influencing agents according to the Graph placement approach and then places any remaining influencing agents based on the Baseline placement approach. Likewise, the Baseline/Graph (4 Graph) hybrid placement approach places four influencing agents according to the Graph placement approach and then places any remaining influencing agents based on the Baseline placement approach.



(a) Average Lost - Random (b) Number Lost - Random

(c) Average Lost - Grid (d) Number Lost - Grid

(e) Average Lost - Border (f) Number Lost - Border

Figure 4: Results for the hybrid approach experiments. These graphs compare (a,c,e) the average number of flocking agents lost and (b,d,f) the number of trials in which any flocking agents are lost.

Figure 4(b,d,f) shows the number of trials in which at least one flocking agent was lost. We strive to minimize the number of trials in which any flocking agents are lost. The results in Figure 4(b,d,f) generally appear as we would expect, but there are a few surprising results. Specifically, the data points showing that the Grid and Border placement approaches have fewer trials in which a flocking agent is lost than the hybrid approaches were initially surprising. However, this could be because the hybrid placement approach may cover some areas twice while leaving other areas open that would be covered by the Grid and Border placement approaches. This hypothesis is supported by the fact that the Random placement graph in Figure 4(b) shows that for all k values, the Random placement approach loses at least one flocking agent in more trials than the hybrid placement approaches.

For all of the graphs in Figure 4, for $k = 2$ the results for all of the placement approaches except the Random placement approach are the same — this is expected because both hybrid placement approaches should use the Graph placement approach for both influencing agent placements. Likewise, for $k = 4$, the results of both the Graph placement approach and the Baseline/Graph (4 Graph) approach should be the same since both approaches use the Graph placement approach for all four influencing agent placements.

Finally, although the computation complexity is better for the hybrid placement approach ($\mathrm{O}(n^3\binom{m^2+m}{k_g})$) than for the Graph placement approach, the complexity is still dominated by the general flock size. Even so, we still obtained the expected result of the hybrid placement approaches being a better trade-off in terms of performance and complexity. To make this tradeoff more concrete, consider processor CPU timings across 10 different placements with $k = 5$ and $m = 15$ — the average placement times were 3.11 minutes for the graph placement approach, 19.4 seconds for the Grid/Graph (4 Graph) approach, and 2.51 seconds for the Grid/Graph (2 Graph) approach.

7. JOINING A FLOCK

So far, all of the work in this paper has considered where influencing agents should be placed into a flock if we assume they can somehow be placed into the flock instantaneously. However, a more realistic scenario would require influencing agents to *join* a flock in motion. Specifically, the influencing agents would have to leave stations positioned throughout the environment and join the flock. In this section, we introduce some initial work on behaviors for influencing agents to join a flock.

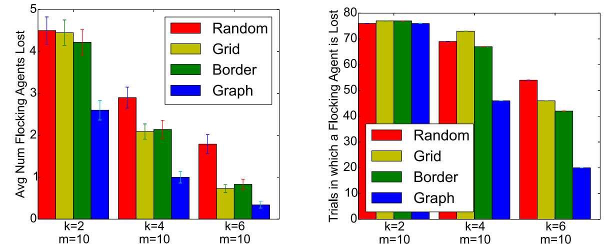
The scenario we consider is as follows: flocking agents begin flocking together in a specific direction and the influencing agents must join the flock and influence it to change its current direction of travel to be towards some desired heading θ^*. In this work, we consider the initial case in which the influencing agents are always able to arrive ahead of the flock's expected path of travel. We make the important assumption that although the influencing agents can move slower than the flock, they are unable to move quicker than the flock.

The questions we ask in this work involve three main aspects: where the influencing agents should position themselves ahead of the flock, how the influencing agents should behave as the flock arrives, and how the influencing agents should attempt to influence the flock. We consider each of these aspects below and present experimental results in turn.

7.1 Positioning Ahead of the Flock

In this work we assume that the influencing agents are able to fly ahead of the flock and position themselves ahead of the flock's expected path. By positioning themselves as they wish before the flock arrives and then remaining stationary (i.e. adopting a 0 velocity), the influencing agents are able to reach their desired positions before any flocking agents enter their neighborhood. This scenario differs from the previous one because once the influencing agents are within the flocking agents' neighborhoods, the flocking agents will be influenced. As such, the important question addressed in this section is: *How should the influencing agents position themselves if they are able to arrive ahead of the flock?*

To answer this question, we run experiments in which the influencing agents move ahead of the flock's arrival to adopt the positions suggested by methods in Section 4.2 before the flock arrives.



(a) Average Lost (b) Number Lost

Figure 5: Graphs for the positioning ahead of the flock experiments. These graphs compare (a) the average number of flocking agents lost and (b) the number of trials in which any flocking agents are lost.

Figure 5 shows how well the four initial placement methods summarized in Section 4.2 fare as desired positions in the *Joining* case. As can be seen in Figure 5, our experimental results show that these placement approaches do well in the *Joining* case. Specifically, the Graph placement approach does consistently much better than the other approaches in terms of both the average number of flocking agents lost and the number of trials in which at least one flocking agent is lost. Additionally, the Grid placement approach and the Border placement approach do better than the Random placement approach for larger values of k. As expected, these general trends are similar to those seen in *Placement* experimental results reported in previous work [5].

Although the Graph placement approach was shown to perform best, its complexity is much more computationally intensive than the other methods. Hence, for the remainder of the paper we will use the Border placement method for selecting the desired positions of the influencing agents within the flock.

7.2 Behavior as the Flock Arrives

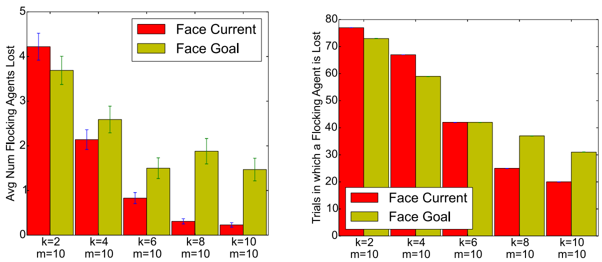
As the flock approaches the waiting influencing agents, the influencing agents will begin to be within the flocking agents' neighborhoods and hence will begin to influence the flocking agents. In this section, we consider how the influencing agents should behave in the short time span between when they are first able to influence the flocking agents and when the flock has reached the point at which they are in their desired positions with respect to the flock's current position.

We present results for two possible behaviors the influencing agents may adopt as the flock is arriving. However, we tested many other behaviors, most of which were unsuccessful for one of two reasons: (1) they influenced the flock before the flock reached the point at which the influencing agents are in their desired positions with respect to the flock's current position or (2) they splintered the flock.

The two influencing agent behaviors that worked best as the flock was arriving were some of the simplest approaches we attempted. Under the first behavior, called 'Face Current', the influencing agents orient towards the heading at which the flock is currently traveling and stay in place until it is time to influence the flock at time t_i. The second behavior, called 'Face Goal', is very similar to the first. Under the 'Face Goal' behavior, the influencing agents orient towards

θ^* and stay in place until it is time to influence the flock at time t_i. In both cases, the influencing agents stay in place by setting $(v_0(t_{start}) \ldots v_{k-1}(t_i)) = 0$. t_{start} denotes the time at which the flocking agents begin to remain stationary after they originally adopt their desired position.

Finally, at time t_i the influencing agents begin influencing the flocking agents with the goal of orienting the flock as a whole towards θ^*. Previous work [3] showed that the flocking agents can become lost if the flock is influenced too quickly. Hence, in addition to considering influencing immediately via the 1-Step Lookahead algorithm we also consider influencing the flock to alter its orientation over a longer set of time (specifically, 100 and 200 time steps in our experiments).



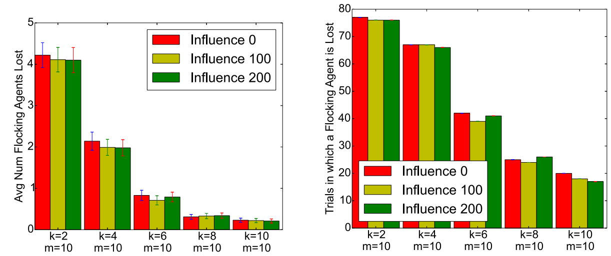
(a) Average Lost (b) Number Lost

Figure 6: Results for the arrival experiments. These graphs compare (a) the average number of flocking agents lost and (b) the number of trials in which any flocking agents are lost.

Figure 6 compares two different behaviors for the influencing agents while the flock is arriving. Although the 'Face Goal' behavior does better in terms of both the average number of flocking agents lost and the number of trials in which any flocking agents are lost when $k = 2$, in general the 'Face Current' behavior does much better. When $k = 8$ and $k = 10$, the 'Face Current' behavior loses significantly fewer flocking agents on average than the 'Face Goal' behavior. This is likely because the 'Face Current' behavior waits until the influencing agents are in their desired positions before influencing, whereas the 'Face Goal' behavior influences flocking agents towards θ^* as the influencing agents enter the flocking agents' neighborhood. Although this type of coincidental influence sometimes influences flocking agents to orient towards θ^*, it is much more likely to result in the flocking agents straying away from both the flock and θ^*.

There are cases in which the 'Face Goal' behavior performs much better than the 'Face Current' behavior. One such case we found in our extensive experiments was when $k = 2$ or $k = 4$ with $m = 10$ and a neighborhood radius $r = 5$. Hence, it seems that there are cases in which each behavior would be appropriate, but based on the experiments we ran, the 'Face Current' behavior usually performs better.

Figure 7 shows results from taking 0 (default), 100, and 200 time steps to slowly influence the flock to orient towards θ^*. The experiments show that although 'Influence 100' and 'Influence 200' perform slightly better than 'Influence 0' in most cases, in general the performance is effectively the same. Despite this, we expect that influencing the flock to orient slowly may be more effective than influencing the flock to orient quickly under particular experimental settings. In particular, one case that showed promise was attempting to influencing a dense flock using a very limited number of influencing agents.



(a) Average Lost (b) Number Lost

Figure 7: Results for the influencing experiments. These graphs compare (a) the average number of flocking agents lost and (b) the number of trials in which any flocking agents are lost.

8. CONCLUSION

In this paper we consider how influencing agents should initially be placed within a flock given computational limitations. We also begin to consider how influencing agents should join a flock in motion if they can arrive ahead of the flock. We present multiple methods towards addressing both of these areas, and we present selected results from our extensive experiments. Experimental results show that for larger numbers of influencing agents, scaling the area in which influencing agents are placed to match the area where flocking agents are initially placed fares well. Additionally, hybrid methods that combine simple methods with the computationally complex Graph placement approach provide a decent trade-off between performance and computation time when initially placing influencing agents into a flock. Finally, we also showed that the influencing agent placement methods that worked well for initial placement also work well if the influencing agents are able to position themselves within the flight path of an incoming flock.

As was seen throughout this paper, influencing agent placement in a flock is an important aspect to utilizing influencing agents to influence a flock. As such, there are many potential areas for future work. One natural extension to this work is to determine how to join a flock in motion when the influencing agents are not able to arrive and position themselves ahead of the flock. Additionally, a related question of interest regards how the influencing agents will be able to detach themselves from the flock once they are finished influencing the flock. Finally, although the 1-Step Lookahead algorithm is effective at influencing flocking agents, it would be interesting to develop an algorithm that leveraged coordinated behaviors between multiple influencing agents.

9. ACKNOWLEDGEMENTS

This work has taken place in the Learning Agents Research Group (LARG) at the Artificial Intelligence Laboratory, The University of Texas at Austin. LARG research is supported in part by grants from the National Science Foundation (CNS-1330072, CNS-1305287), ONR (21C184-01), AFRL (FA8750-14-1-0070), AFOSR (FA9550-14-1-0087), and Yujin Robot.

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

[4] K. Genter and P. Stone. Ad hoc teamwork behaviors for influencing a flock. *Acta Polytechnica*, 2016.

[5] K. Genter, S. Zhang, and P. Stone. Determining placements of influencing agents in a flock. In *AAMAS'15*, May 2015.

[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] P. Stone, G. A. Kaminka, S. Kraus, and J. S. Rosenschein. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *AAAI'10*, 2010.

[12] 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.

[13] 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.