Advances in Adding Influencing Agents to a Flock

Katie Genter and Peter Stone
Department of Computer Science, 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 to emerge from relatively simple local control rules utilized by the individuals 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, that are under our control, to a flock. Specifically, we advance previous work on adding influencing agents to 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.

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* (Stone et al. 2010). 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 (Vicsek et al. 1995), graphics (Reynolds 1987), biology (Couzin et al. 2005), and distributed control theory (Han, Li, and Guo 2006; Jadbabaie, Lin, and Morse 2003; Su, Wang, and Lin 2009) 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. These controllable agents — which we call *influencing agents* — are seen by the rest of the flock as just other agents in the flock; however, they behave according to our algorithms.

In our work, we follow a well-recognized flocking model (Reynolds 1987) 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 (Genter and Stone 2014; 2016) as well as where influencing agents should be placed in the flock if they were somehow able to join the flock instantaneously (Genter, Zhang, and Stone 2015). 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 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 (1987). 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 (Vicsek et al. 1995). 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 (2003). 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 (2009) and Celikkanat and Sahin used informed agents to lead a flock by their preference for a particular direction (2010). 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 (2005). 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 (2006). 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, in this section we specify (1) a model of the flock and (2) the performance objective.

3.1 Flocking Model

In order to model the flock, we use a simplified version of Reynolds' Boid algorithm for flocking (Reynolds 1987). 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.

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 $n_i(t) \le n$ 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($\theta_i(t), \theta_j(t)$)

if $((\theta_i(t) - \theta_j(t) \ge -\pi) \wedge (\theta_i(t) - \theta_j(t) \le \pi))$ **then**
    return $\theta_i(t) - \theta_j(t)$
else if $\theta_i(t) - \theta_j(t) < -\pi$ **then**
    return $2\pi + (\theta_i(t) - \theta_j(t))$
else
    return $(\theta_i(t) - \theta_j(t)) - 2\pi$

3.2 Performance Representation

In previous work we defined the *Agent Control and Placement Problem* (Genter, Zhang, and Stone 2015). 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 $\overline{t^*}$)

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

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 fewest agents but also try to minimize the chances of losing any agents.

4 Existing Behavior and Placement Methods

The research presented in this paper utilizes and compares against placement methods from previous work (Genter, Zhang, and Stone 2015). This previous work considered where to place influencing agents $\{a_0, \ldots, a_{k-1}\}$ into the flock at time $t = 0$. 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}.

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

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.

Border Approach The Border placement method places k influencing agents evenly along the borders of FA_{preset}. The placement of the influencing agents is not dependent on the positions of flocking agents and hence is determined in constant time.

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 (Luke et al. 2005) for our experiments in this paper. 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. 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.

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. The error bars in all of our graphs depict the standard error of the mean.

6 Placement in a Flock

Section 4 summarized various 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 as well as a new approach that mitigates the high computational complexity of the Graph placement method reviewed in Section 4.

6.1 Scaling Placement Area

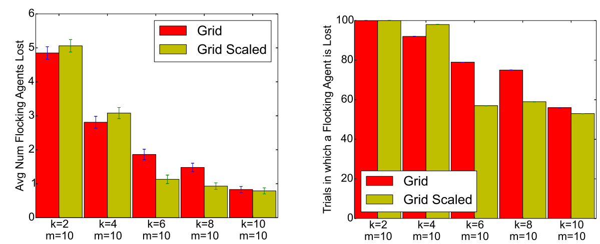
Three of the placement methods summarized in Section 4 do not place influencing agents based on where the flocking agents are placed. Instead, these methods place the influencing agents in predetermined locations based on (1) how many influencing agents are to be added and (2) FA_{preset}.

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}. 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. Experiments detailing how these three scaled approached fared in comparison to the approaches from previous work are provided next.

Experiments In this section we run *Placement* case experiments to compare the performance of the initial placement methods summarized in Section 4 against the scaled versions of these methods presented in Section 6.1.

Although we ran experiments for all three types of scaled placement algorithms, this section only shows results for Grid variants due to space limitations. Results from Border and Random variant experiments are similar.



(a) Average Lost (b) Number Lost

Figure 1: Selected results for the Scaling Placement Area experiments. (a) depicts the average number of flocking agents lost and (b) depicts the number of trials in which any flocking agents are lost.

Figure 1 compares the 'Grid' and 'Grid Scaled' placement approaches for six different placement methods when adding $k = 2$ to $k = 10$ influencing agents to a flock with $m = 10$ flocking agents.

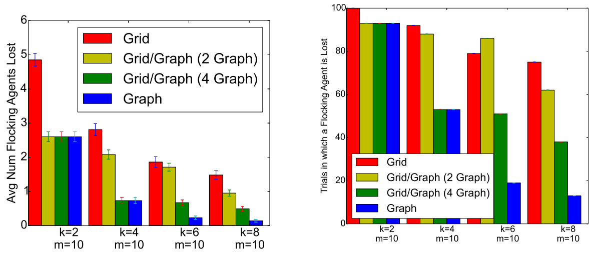
A few interesting trends arise in Figure 1. First, Grid Scaled loses more flocking agents on average than its non-scaled counterpart when $k = 2$ and $k = 4$ but loses fewer flocking agents when $k = 6$, $k = 8$, and $k = 10$ (significantly so for $k = 6$ and $k = 8$). Second, in terms of the number of trials in which no flocking agents are lost, Figure 1(b) 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 $k = 8$.

6.2 Combining Placement Methods for Better Scalability

The Graph placement approach summarized in Section 4 was shown in previous work (Genter, Zhang, and Stone 2015) to perform better than other approaches. However, the Graph placement method was not widely 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 efficient methods. In particular, in the next section we present experiments for 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.

Experiments We compared the four initial placement methods summarized in Section 4 against hybrid approaches that combine the Graph placement approach with more efficient approaches. These experiments were run under the *Placement* case settings described in Section 5.



(a) Average Lost (b) Number Lost

Figure 2: Results for the Grid hybrid approach 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 2(a) shows the average number of flocking agents lost when initially placing influencing agents according to the Grid, Graph, and hybrid placement approaches. The Grid/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 Grid placement approach.

Figure 2(b) 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 2(b) generally appear as we would expect, but there is one surprising result. Specifically, the data point showing that the Grid placement approach has fewer trials in which a flocking agent is lost than the hybrid approaches was 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 placement approaches.

For both of the graphs in Figure 2, 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 Grid/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 (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.

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 to be towards some desired heading θ^*. In this work, we consider the initial case in which the influencing agents are 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 two main aspects: where the influencing agents should *position* themselves ahead of the flock and how the influencing agents should *behave* as the flock arrives. We consider each of these aspects below and present experimental results in turn.

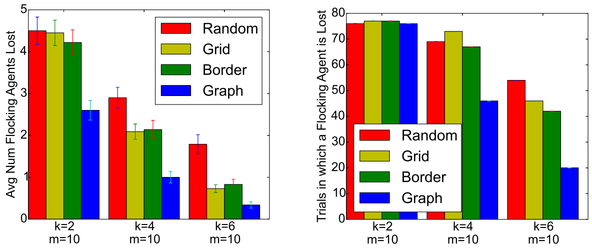
7.1 Positioning Ahead of the Flock

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 before the flock arrives.

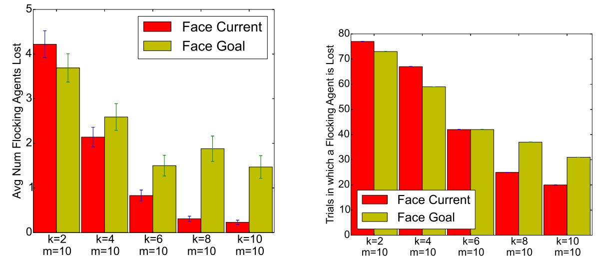
Experiments In this section, we assess how well the four initial placement methods summarized in Section 4 fare as desired positions in the *Joining* case.

As can be seen in Figure 3, 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 (Genter, Zhang, and Stone 2015).



(a) Average Lost (b) Number Lost

Figure 3: 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.



(a) Average Lost (b) Number Lost

Figure 4: 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.

Although the Graph placement approach was shown to perform best, it's complexity is more computationally intensive than the other methods. Hence, for the next experiment 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.

We present results for two possible behaviors for the influencing agents to 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.

Experiments In this section we present results for *Joining* experiments in which we compare the two proposed influencing agent behaviors for while the flock is arriving.

Figure 4 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 less 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.

8 Conclusion

In this paper we consider how influencing agents should initially be placed within a flock given computational limitations. We also we 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 issues as well as results from our 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 provides a good tradeoff between performance and computation time. Finally, our experiments 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.

Influencing agent placement in a flock is an important aspect to using influencing agents to influence a flock. There are many potential areas for future work. One particular area of interest is determining how to join a flock in motion when the influencing agents are not able to arrive and position themselves ahead of the flock.

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), AFRL (FA8750-14-1-0070), and AFOSR (FA9550-14-1-0087).

References

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

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

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

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

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

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

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

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

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

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

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

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