Placing Influencing Agents into 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 the 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.

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 typically studied under the assumption that all of the agents are identical or represent a small set of well-defined behavior types. Indeed, various disciplines have studied flocking in order to characterize its emergent behavior. In our work, we instead consider how to lead a flock to particular behaviors by adding some controllable agents to the flock.

For example, imagine that a flock of migrating birds is flying directly towards a dangerous area, such as a wind farm or an airport. 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 induce 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. The influencing agents — which could be in the form of robotic birds[1], robotic bees (Shang et al. 2009) or ultralight aircraft[2] — follow our algorithms but are perceived by the rest of the flock as one of their own.

Following well-recognized flocking models (Reynolds 1987), we assume that each bird in the flock dynamically adjusts its heading based on that of its immediate neighbors. Our 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 (Genter, Agmon, and Stone 2013; Genter and Stone 2014). In our work presented in this paper, we examine how the influencing agents should be placed in the flock. Specifically, our current research question is: *where should influencing agents be initially 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 6 reviews our 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 (1987). 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 (1995). However, neither of these lines of research considered how to influence the flock to adopt a particular behavior by introducing agents into the flock.

Jadbabaie *et al.* considered the impact of adding a controllable agent to a flock (2003). They 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 (2009). Celikkanat and Sahin used informed agents to lead the flock by their

[1] www.mybionicbird.com

[2] www.operationmigration.org

preference for a particular direction (2010). Our work is different from these lines of research in that they 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 (2005). Couzin *et al.* showed that only a very small proportion of informed agents is required for such decision, 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 (2010). 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 likely react.

Han *et al.* assume that an influencing agent can be placed at any desired position at any time step (2006). 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, unlike Han *et al.* we do not allow teleporting and hence we can not continuously 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 into a flock that aim to influence the flock towards a particular behavior.

3 Problem Definition

To fully specify our problem, we must 1) specify a model of the flock, 2) specify the possible options for placing influencing agents initially, 3) specify the actions available to the influencing agents, and 4) specify 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 (Reynolds 1987) that is similar to the model utilized in (Genter, Agmon, and Stone 2013).

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, 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. Throughout this paper, we restrict $\theta_i(t)$ to be within $[0, 2\pi)$.

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

if $((\theta_i(t) - \theta_j(t) \geq -\pi) \wedge (\theta_i(t) - \theta_j(t) \leq \pi))$ **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 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 (Genter and Stone 2014) only considered random placement of the influencing agents. 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 (Section 7.2) we may place the agents $\{a_0, \ldots, a_{k-1}\}$ wherever we wish. However, in the cases where $\{p_0(0), \ldots, p_{k-1}(0)\}$ is not under our control (Section 7.3), we are constrained to have the agents $\{a_0, \ldots, a_{k-1}\}$ begin in designated starting spots. These agents may then attempt to join the flock at a later time, perhaps as the flock passes by their spot.

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 (Genter and Stone 2014) only focused on control to influence neighbors. In this work, we add reasoning about control to adjust position in Section 6 and we introduce a method to arbitrate between control to adjust position and control to influence neighbors.

3.4 Performance Representation and Objective

We define the *Agent Control and Placement Problem* (ACAPP) as follows: Given a target orientation θ^* and a team of n agents $\{a_0, \ldots, a_{n-1}\}$, where the 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 $\{a_0, \dots a_{k-1}\}$ at time 0 and control $\phi(t)$ at time $t >= 0$ such that loss $l(\pi(0) + \phi(t))$ is minimized.

A k-agent placement specifies the positions that each influencing agent $\{a_0, a_1, \dots a_{k-1}\}$ will take at time 0. The k-agent placement is denoted by $\pi_k(0) = \{p_0(0), \dots, p_{k-1}(0)\}$ where $\{p_0(0), p_1(0), \dots, p_{k-1}(0)\}$ is the set of positions for influencing agents $\{a_0, a_1, \dots a_{k-1}\}$ at time 0.

We denote t^* as the time at which flocking agents $\{a_k, \dots, a_{N-1}\}$ are oriented such that $\{\theta_k(t^*), \dots, \theta_{N-1}(t^*)\}$ 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 $m' < m$ of flocking agents with orientations equal to θ^* for more than 200 time steps where a_i is not in m'. At this point, t^* is set to the time step at which the subset m' converged to θ^*. Let $\gamma_{m'}(t)$ denote the positions of the non-lost flocking agents at time t. 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* $l(\pi(0) + \phi(t))$ of a k-agent placement $\pi_k(0)$ and control $\phi(t)$ is a weighted function of four terms:

- w_1 is a weight that emphasizes the importance of minimizing the number of runs in which any agent is lost (minimize runs in which $m - m' > 0$)
- w_2 is a weight that emphasizes the importance of minimizing the number of lost agents (minimize $\overline{m - m'}$)
- 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 $\overline{\alpha}$)

$$\begin{aligned} l(\pi_k(0) + \phi_k(t)) = & w_1 \overline{p(m - m' > 0)} + w_2 \overline{m - m'} + \\ & w_3 \overline{t^*} + w_4 \overline{\alpha} \end{aligned} \tag{2}$$

An optimal placement $\pi^*(0) + \phi(t)$ is one with minimal loss $l(\pi^*(0) + \phi(t))$.

In this work, we set $w_2 > w_1 > w_3 > w_4$. 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 on flocking within the MASON simulator (Luke et al. 2005). This simulator encodes all the flock dynamics as described in this section and we augment it to compute the performance metric discussed in the previous section. Each agent points and moves in the direction of its current velocity vector.

Videos showing the simulator in action in both cases are available on our web page[3]. Our experimental setup is described in more detail in Section 7.1.

4 Control of the 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 expect birds to 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 to face a particular direction or maneuver along a path (Genter, Agmon, and Stone 2013; Genter and Stone 2014). We briefly review this work below.

Our first work in this area considered how influencing agents should behave from a more theoretical approach (Genter, Agmon, and Stone 2013). In this work we presented a formal definition of our flocking model, which served as a base for the model we use in our current work. 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 our 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 (Genter and Stone 2014). We introduced a 1-step lookahead algorithm for determining the individual behavior of each influencing agent $a_i \in \{a_0, \dots, a_{k-1}\}$. 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 the *orient* case and the *maneuver* case.

In the current work presented in this paper, we use the 1-step lookahead algorithm to determine the behavior of the influencing agents that we add to the flock.

5 Determining the Initial Positions of Influencing Agents

In previous work we considered how the influencing agents that we added to random locations within a flock should behave. In the work presented in this paper, we instead consider how to place the influencing agents $\{a_0, \dots, a_{k-1}\}$ into the flock. We consider two different cases when determining how to place $a_i \in \{a_0, \dots, a_{k-1}\}$ 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

[3] http://mipc15.blogspot.com/

at one of more stations outside the flock at time $t = 0$ and is directed to travel to a particular location in the flock.

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[4].

5.1 Random Placement

Our past research has randomly placed k influencing agents within the dimensions of the flock (Genter and Stone 2014). Hence, we use random placement as the base case for evaluating our other placement approaches.

5.2 Grid Placement

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. The grid size is dependent on k. 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.

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 placement approach, the placement of the influencing agents is not dependent on the positions of flocking agents. Instead, 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 will be 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.

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.

Creating the Graph Before any influencing agents are added to the graph, all $\{a_k, \ldots, a_{N-1}\}$ flocking agents are added to graph G as nodes. For each agent $a_i \in \{a_k, \ldots, a_{N-1}\}$, an edge is added to G between a_i and each of its neighbors $a_b \in n_i(t)$ if an edge does not already exist.

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
- $(x_i(t) + 0.1, y_i(t) + 0.1)$ where $r < 0.1$

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

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. Hence we do the following for each S:

- 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

Once all possible k-sized sets S have been individually considered, we select a set based on the information we obtained. Specifically, we consider (1) minimal **numNoConn**, (2) maximal **numConn**, (3) maximal **numDirectConn**, and (4) minimal **numNoDirectConn** in order. If only one set matches the description at a level, then we select it. Otherwise, all of the sets that matched the description at that level move onward to be 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 criteria.

6 Determining the Control of Influencing Agents

In the **Drop** case, the k influencing agents are at the desired positions at $t = 0$, so they can start 1-step lookahead directly. In the **Dispatch** case, however, the influencing agents are initialized outside of the flock. They would only influence a limited number of flocking agents if they start 1-step lookahead immediately. Hence they need to reposition themselves to their desired positions before they attempt to influence the flocking agents' orientations.

There are two issues we need to resolve for the **Dispatch** case. First, the influencing agents need to approach and enter the flock in order to influence more flocking agents. If they travel at the same speed as the flocking agents, they may not be able to catch up to the flocking agent. Instead, as the orientations of the flocking agents are affected by the influencing agents' directions, the flocking agents are more likely to be driven away. Therefore, we allow the influencing agents to travel faster than the flocking agents. Second, as we described earlier, once an influencing agent a_i reaches

[4] http://mipc15.blogspot.com/

its desired position $p_i(t)$, it enters the phase of 1-step lookahead. However, 1-step lookahead may cause a_i to leave its desired position (it may even leave the flock). a_i may wish to return to the repositioning phase. However, if a_i switches between repositioning and influencing too frequently, it will end up oscillating between these two behaviors and not efficiently influence the flocking agents. Therefore, we employ a hysteresis method to control the switch between these two phases.

We use the initial positions in the **Drop** case as the desired positions in the **Dispatch** case. We report the empirical results for random placement, grid placement and border approach as these approaches all run in constant time. As the desired positions need to be evaluated at each time step in the **Dispatch** case, we found the graph approach too computationally expensive to be used in the **Dispatch** case.

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.

7.1 Experimental Setup

We utilize the MASON simulator (Luke et al. 2005) 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 our experimental setup.

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

Variable	**Drop Default**	**Dispatch Default**
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.

Many of our experimental variables, 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. 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 viewable area.

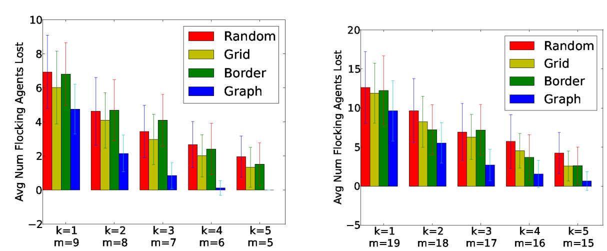
Flocking agents are randomly placed initially within a square in the top left of the domain, where this square occupies 4% of the domain. Agents are assigned random headings that are within 90 degrees of the initial θ^*. We conclude that the flock has converged when every agent (that is not an influencing agent or *lost*) is facing within 0.1 radians of θ^*.

In all of our experiments, we run 100 trials for each experimental setting. We use the same 100 random seeds for each set of experiments for the purpose of variance reduction. The random seeds are used to determine the initial placement and orientation of all of the flocking agents.

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 at time $t = 0$. There are many different metrics that can be used to assess 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.

Figure 1 shows graphs that depict the average number of flocking agents lost when $n = 10$ and when $n = 20$. Note that the graph approach does better in comparison to the other approaches when the flock size is small ($n = 10$) because in these cases agents are more sparse in the environment and hence tend to have fewer neighbors. Additionally, the graph approach also performs better than the other approaches when the percentage of influencing agents in the flock is high. This is likely 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 flocking agents will decrease quicker under the graph approach than under other approaches.



(a) $n = 10$ (b) $n = 20$

Figure 1: The average number of flocking agents lost. These results are obtained over 100 runs.

Figure 2 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 the number of influencing agents 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 flocking agents.

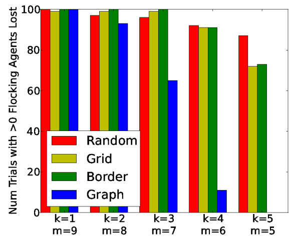


Figure 2: The number of trials (out of 100) in which any flocking agent was lost when $n = 10$.

7.3 Dispatch Experimental Results

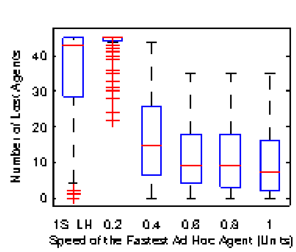
We initialize all the influencing agents at one corner of the flock. They start by traveling to their desired positions, and then do 1-step lookahead. We use a hysteresis method to control the switching between repositioning and influencing. Concretely, when an influencing agent reaches a distance of α away from its desired position, it returns to its desired position within distance β, where $\alpha > \beta$.

In Figure 3(a,b), we show a baseline case that the influencing agents are dispatched to random positions in the flock. We can observe that without considering placing them in the most influencing positions, the influencing agents end up with the worst performance compared to the following approaches. However, this result is still comparable with other approaches, as the influencing agents try to enter the flock, even though the positions in the flock are chosen randomly.

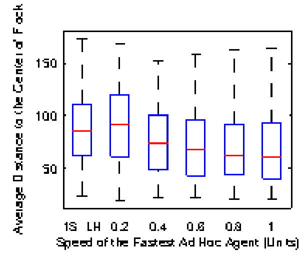
We can see that faster speed of the influencing agents yields better performance in general, as it reduces the influence on the flock when repositioning. On the contrary, when the speed of the influencing agents is same as that of the flocking agents, almost all of the flocking agents are lost. Finally, we compare to purely 1-step lookahead (the left bar) without positioning. It has a worse performance as the influencing agents remain at one corner.

In Figure 3(c,d), the influencing agents are dispatched to the border positions of the flock. This is slightly better than the random case. Although in the **Drop** case, influencing agents on the border can surround the flock, and thus reduce the number of lost agents. In the **Dispatch** case, however, this approach makes the influencing agents travel a longer distance to reach their destinations before they can start 1-step lookahead. This results in more flocking agents lost and the flock less compact.

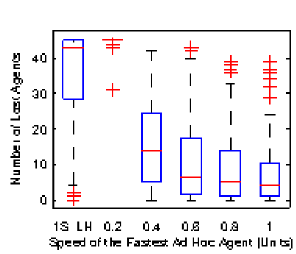
In Figure 3(e,f), the influencing agents are dispatched to the grid positions. Compared to the border case, the grid position case shows a better performance. Same as the reasoning above, the influencing agents travel a shorter distance to position themselves.



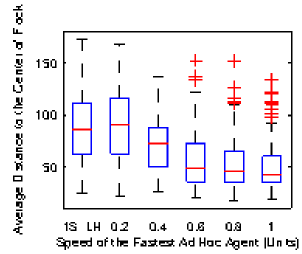
(a) Random, lost agents



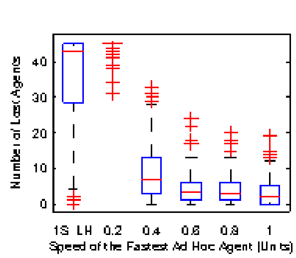
(b) Random, distance



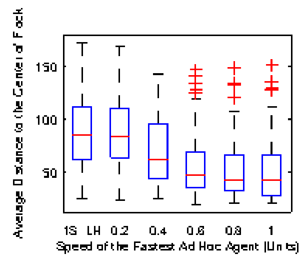
(c) Border, lost agents



(d) Border, distance



(e) Grid, lost agents



(f) Grid, distance

Figure 3: The average number of lost agents ($m = 45$) and the average distance to the center of the flock.

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 presented multiple methodologies as well as experimental results for placing influencing agents 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.

In future work we plan to also consider position adjustment control in the case in which we can place the influencing agents initially. We also plan to design a more efficient graph-based placement approach for the **Dispatch** case. Finally, it would be interesting to automatically determine the best formation for the influencing agents based on the current configuration 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), and AFOSR (FA8750-14-1-0070, 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.

Ferrante, E.; Turgut, A. E.; Mathews, N.; Birattari, M.; and Dorigo, M. 2010. 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, 331–340. Springer-Verlag.

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

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

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.

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

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.