To appear in *Proceedings of the Twelfth International Conference on Autonomous Agents and Multiagent Systems* (AAMAS-13), Saint Paul, MN, USA, May 2013.

Ad Hoc Teamwork for Leading a Flock

Katie Genter
Dept. of Computer Science
University of Texas at Austin
Austin, TX 78712 USA
katie@cs.utexas.edu

Noa Agmon
Dept. of Computer Science
Bar Ilan University
Ramat Gan, 52900, Israel
agmon@cs.biu.ac.il

Peter Stone
Dept. of Computer Science
University of Texas at Austin
Austin, TX 78712 USA
pstone@cs.utexas.edu

ABSTRACT

Designing agents that can cooperate with other agents as a team, without prior coordination or explicit communication, is becoming more desirable as autonomous agents become more prevalent. In this paper we examine an aspect of the problem of leading teammates in an ad hoc teamwork setting, where the designed ad hoc agents lead the other teammates to a desired behavior that maximizes team utility. Specifically, we consider the problem of leading a flock of agents to a desired orientation using a subset of ad hoc agents. We examine the problem theoretically, and set bounds on the extent of influence the ad hoc agents can have on the team when the agents are stationary. We use these results to examine the complicated problem of orienting a stationary team to a desired orientation using a set of non-stationary ad hoc agents. We then provide an empirical evaluation of the suggested solution using our custom-designed simulator FlockSim.

Categories and Subject Descriptors

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

General Terms

Algorithms, Experimentation

Keywords

Ad Hoc Teamwork, Agent Cooperation, Coordination

1. INTRODUCTION

The growing use of agents in various cooperative domains has emphasized the importance of designing agents capable of reasoning about *ad hoc* teamwork [10]. Such agents can cooperate within a team without using explicit communication or previously coordinating behaviors among teammates. One aspect of ad hoc teamwork involves *leading* teammates. Consider a case in which we want to influence a given team of agents to alter their actions in order to maximize the team utility. One way of doing so is by adding agents to the team in order to *lead* them to perform the desired actions. While previous research on leading teammates in ad hoc settings concentrated on a game theoretic analysis of this problem [1, 11], in this work we consider a more practical aspect of the problem by concentrating on *flocking agents*.

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013), Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.*

Flocking is an emergent behavior that can be found in different species in nature including flocks of birds, schools of fish, and swarms of insects. In each of these cases the animals follow a simple local behavior rule that results in a stable, well defined, group behavior. Research on flocking behavior can be found in various disciplines, for example in physics [14], graphics [9], biology [2, 3], and distributed control theory [6, 7, 12]. The main focus of each of these research directions is to characterize the emergent behavior. In this paper we introduce the problem of *leading* a team of flocking agents in ad hoc teamwork settings. In this case, we are given a team of flocking agents following a known, well defined rule characterizing their flocking behavior, and we wish to examine to what extent it is possible to influence the team. Specifically, the question addressed in this paper is: *is it possible for one or more agents to lead the team to a desired orientation, and if so - what is the most efficient way of doing so?*

The ad hoc teamwork perspective of this problem is highlighted by two facts. Firstly, we do not explicitly control the behavior of the flocking agents, thus we can only try to influence them implicitly using the behavior of the ad hoc agents (which appear to be identical to the flocking agents). Secondly, all agents — both flocking and ad hoc — act as one team, and their only desire is to optimize team utility. The ad hoc agents cannot communicate with the flocking agents, but they can coordinate their actions among themselves.

The challenge of designing ad hoc agents in such a dynamic system is twofold: the action space is continuous and the set of optimal solutions is not discrete. We therefore analyze the system in two steps. First, we examine the problem of leading a team of stationary flocking agents, i.e., the agents are aligned in a formation and the ad hoc agents attempt to influence only their *orientations* while all of the agents maintain their current positions. In the second stage, we use these results for establishing the desired behavior when the ad hoc agents can move.

One of the main contributions of this paper is the problem definition provided in Section 2, as it contributes a specification of the flocking problem as a new scenario for studying ad hoc teamwork. Another major contribution of this paper is an initial theoretical and empirical analysis of the problem. Some foundational results that apply to all flocking scenarios are presented in Section 3. The analysis for stationary ad hoc agents is presented in Section 4, while the analysis for non-stationary ad hoc agents is presented in Section 5. We also describe our empirical evaluation using our custom-designed simulator FlockSim[1] in Section 5. Section 6 situates this research in the literature, and Section 7 concludes.

2. PROBLEM DEFINITION

In this work we introduce a flocking model inspired by Vicsek *et al.* [14]. In our model, n homogeneous agents inhabit some environment where each agent a_i moves with some velocity v_i. Velocity v_i is not a function of time and hence remains constant for each agent over time. 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 $\theta_i(t)$. Note that $p_i(t)$ and $\theta_i(t)$ are dependent on the time t and can be different for each agent a_i. 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 $y_i(t) = y_i(t-1) - v_i \sin(\theta_i(t))$.

We let $N_i(t)$ be the set of $n_i(t) \leq n$ agents (including agent a_i) at time t which are visible to agent a_i. An agent is *visible* to agent a_i if its position is located within a *visibility cone* of angle α centered on orientation $\theta_i(t)$ and extending from agent a_i for an unlimited distance (see Figure 1 for an example). We say that angle α defines the visibility cone for each agent, and that this visibility cone defines each agent's *neighborhood* (i.e., the area in which the agent can see other agents). For example, assume an agent $a_j \in N_i(t)$ is in a_i's neighborhood and hence a *neighbor* of a_i.

Agent a_j's position $p_j(t) = (x_j(t), y_j(t))$ in the environment at time t is located at angle $\beta_j(i)$ with respect to agent a_i's position $p_i(t) = (x_i(t), y_i(t))$ and orientation $\theta_i(t)$. Agent a_j is in a_i's neighborhood at time t if angle $\beta_j(i)$ is less than or equal to $\frac{\alpha}{2}$.

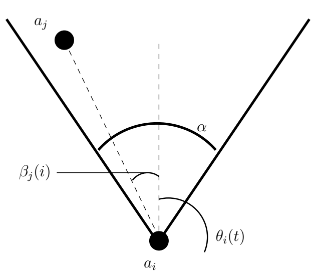

Figure 1: Angle α defines the visibility cone for agent a_i. Agent a_j is in a_i's neighborhood since angle $\beta_j(i) \leq \frac{\alpha}{2}$.

For simplicity, we use the visibility cone approach discussed above to define each agent's neighborhood. It should be noted, however, that this is not the exact approach real birds use to determine which neighboring birds influence their flight. For example, starlings are believed to consider the seven nearest birds in their flock as their neighborhood when performing orientation updates [2]. However, it is generally accepted that birds do have a 'blind' angle behind them such that any neighboring birds within this 'blind' area are not considered when performing orientation updates [3]. This 'blind' area was the motivation for our visibility cone approach.

[1]Videos of our FlockSim simulator are available at `http://aamas13.blogspot.com/`

Under our flocking model, 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 $N_i(t)$ (including itself) at time t. Formally,

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

We must use Equation 1 instead of merely taking the average orientation of all agents because of the special cases handled by Algorithm 1 (e.g. The average of 350° and 10° is 180°, but by Algorithm 1 it is 0°, as desired). Throughout this paper, we restrict $\theta_i(t)$ to be within $[0, 2\pi)$. Likewise, the difference between the orientations of two agents is always within $[-\pi, \pi]$ since a difference of $\pi + \epsilon$ is equivalent to a difference of $\epsilon - \pi$ in the opposite direction.

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 $(\theta_i(t) - \theta_j(t)) - 2\pi$

The n homogeneous agents that comprise the flock consist of k *ad hoc* agents and m *flocking* agents, where $k + m = n$. The ad hoc 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 but that we know calculate their orientation according to Equation 1. In this paper, we make the simplification that although we can have many flocking agents, they all must have the same position $p(t)$ in the environment. At each time step t, the number of ad hoc agents inside a_i's neighborhood is denoted by $k_i(t)$ and the number of flocking agents inside the neighborhood is denoted by $m_i(t)$, where $k_i(t) + m_i(t) = n_i(t)$ (the total number of agents in a_i's neighborhood).

Performance Representation and Objective

We define the Agent Flock Orientation Manipulation Problem as follows: Given a target orientation θ^* and a team of n homogeneous agents $\{a_0, \ldots, a_{n-1}\}$, where the flocking agents $\{a_k, \ldots a_{n-1}\}$ calculate their orientation based on Equation 1, determine whether the ad hoc agents can influence the flocking agents to align to θ^*, and if so, find the plan π that does so with minimum cost $c(\pi)$.

During each time step, the ad hoc agents first orient to their desired orientations based on some plan π. Next, the flocking agents update their orientations based on the orientations of all the agents in their neighborhoods (using Equation 1). Finally, the positions of all the agents are updated.

An x-step plan specifies the orientations that each ad hoc agent $\{a_0, a_1, \ldots a_{k-1}\}$ will align to at each time step when given exactly x time steps in which the agents have to act. The x-step plan is denoted by $\pi_x = (\pi_x^0, \pi_x^1, \ldots, \pi_x^{k-1})$, where $\pi_x^i = (\theta_i(0), \theta_i(1), \ldots, \theta_i(x-1))$ is the set of orientations for agent a_i. The *performance error* $E(\pi_x)$ of an x-step plan π_x is the sum of the differences between each flocking agent's final orientation after x steps and θ^*, formally

$$E(\pi_x) = \sum_{j=k}^{n-1} |\text{calcDiff}(\theta^*, \theta_j(x))| \tag{2}$$

For each plan π, performance error decreases when more time steps are available such that $E(\pi_0) \geq E(\pi_1) \geq E(\pi_2) \geq \ldots \geq E(\pi_\infty)$. Performance error never increases as more time steps are available because the optimal behavior given one additional time step is to either influence the same as with one fewer time step (and obtain the same performance error) or influence at least one flocking agent to orient itself closer to θ^* (and obtain lower performance error).

The *cost* of an x-step plan π_x is defined as

$$c(\pi_x) = w_1 x + w_2 E(\pi_x) \tag{3}$$

where w_1 is a weight that can be set to emphasize the importance of lesser time steps, x is a scalar representing the size of the plan π_x, and w_2 is a weight that can be set to emphasize the importance of lower performance error. At the extremes, setting $w_1 >> w_2$ encourages finding reasonably low performance error in as few steps as possible, while setting $w_2 >> w_1$ encourages minimizing performance error using as many steps as are needed.

An optimal plan π^* is one with minimal cost $c(\pi^*)$. The optimal number of time steps $|x|$ for a task is the x at which $c(\pi_x)$ is minimal. Likewise, the optimal cost $|c(\pi)|$ is equal to $c(\pi_{|x|})$.

In this work, we set w_1 to a moderate number and set w_2 to ∞. With these settings for w_1 and w_2 we obtain the least-step plan in which all flocking agents orient to θ^*, if such a plan exists. If such a plan does not exist, then we obtain a plan with low performance error that uses as few steps as possible.

3. GENERAL FLOCKING THEOREMS

In this paper we establish the building blocks towards examining a fully dynamic system of agents in which most agents base their behavior on the current behavior of neighboring agents and a few agents are controlled by our algorithms. However, the agents controlled by our algorithms do not appear any different to the other agents, and hence the other agents do not recognize that these agents are controlled by a different algorithm.

In this section, we present lemmas that are general in nature and will apply to both the stationary and non-stationary ad hoc agent cases examined in the later sections of this paper. In particular, in this section we consider the case in which there are $k_i(t)$ ad hoc agents and $m_i(t)$ flocking agents, where all $m_i(t)$ flocking agents are located at the same position $p_i(t)$ with identical orientations $\theta_i(t)$ (see Figure 2 for an example).

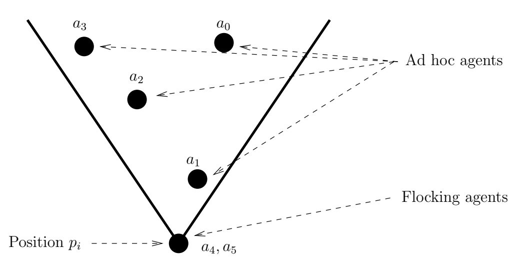

Figure 2: An example with two flocking agents located at the same position with identical initial orientations and four ad hoc agents located at different locations within the visibility cone of the flocking agents.

The first lemma we present in this section relates to the maximal amount the $k_i(t)$ ad hoc agents can influence the $m_i(t)$ flocking agents in a single time step.

LEMMA 1. *The $k_i(t)$ ad hoc agents can influence the $m_i(t)$ flocking agents to turn in a particular direction by any amount less than or equal to $\frac{k_i(t)\pi}{m_i(t)+k_i(t)}$ radians in one time step.*

PROOF. When the difference between $\theta_j(t)$ and $\theta_i(t)$ is less than π (or greater than π, in which case the difference is less than π in the opposite direction), then by Equation 1

$$\begin{aligned}
\theta_i(t+1) - \theta_i(t) &= \frac{1}{n_i(t)} \sum_{j \in N_i(t)} (\theta_j(t) - \theta_i(t)) \\
&\leq \frac{k_i(t)(\pi - \epsilon)}{m_i(t) + k_i(t)} \\
&\leq \frac{k_i(t)\pi}{m_i(t) + k_i(t)} - \frac{k_i(t)\epsilon}{m_i(t) + k_i(t)} \\
&< \frac{k_i(t)\pi}{m_i(t) + k_i(t)}
\end{aligned}$$

When the difference between $\theta_j(t)$ and $\theta_i(t)$ is equal to π, by Equation 1

$$\begin{aligned}
\theta_i(t+1) - \theta_i(t) &= \frac{1}{n_i(t)} \sum_{j \in N_i(t)} (\theta_j(t) - \theta_i(t)) \\
&= \frac{k_i(t)\pi}{m_i(t) + k_i(t)}
\end{aligned}$$

However, it is impossible to guarantee that the flocking agents turn in a particular direction when the difference between $\theta_j(t)$ and $\theta_i(t)$ is equal to π. Hence, in this case the ad hoc agents set $\theta_j(t)$ such that the difference between $\theta_j(t)$ and $\theta_i(t)$ is $\pi - \epsilon$ or $\pi + \epsilon$. When the ad hoc agents do this, directionality can be guaranteed and

$$\begin{aligned}
\theta_i(t+1) - \theta_i(t) &= \frac{k_i(t)(\pi - \epsilon)}{m_i(t) + k_i(t)} \\
&< \frac{k_i(t)\pi}{m_i(t) + k_i(t)}
\end{aligned}$$

□

The second lemma we present in this section states that all $k_i(t)$ ad hoc agents in a flocking agent's visibility cone can adopt the exact same orientation. We show that no extra influence can be obtained by some of the ad hoc agents adopting different orientations than the other ad hoc agents.

LEMMA 2. *When $k_i(t)$ ad hoc agents work together to influence $m_i(t)$ flocking agents to align the team to some θ, it suffices to consider only algorithms that choose at each time step just one orientation for all of the ad hoc agents to adopt.*

PROOF. Assume an algorithm makes the ad hoc agents adopt orientations $\theta_0(t), \ldots, \theta_{k_i(t)-1}(t)$, where some of these orientations may differ. Then, by Equation 1, the orientation of the flocking agents is

$$\theta_f(t+1) = \theta_f(t) + \frac{1}{n_f(t)} \sum_{j \in N_i(t)} \text{calcDiff}(\theta_j(t), \theta_f(t))$$

Now, assume the ad hoc agents adopt an angle σ that is the average of $\theta_0(t), \ldots, \theta_{k_i(t)-1}(t)$. Then by Equation 1, the new orientation is

$$\theta_f(t+1) = \theta_f(t) + \frac{k_i(t)}{n_f(t)}(\sigma)$$

Since σ is the average of $\theta_0(t), \ldots, \theta_{k_i(t)-1}(t)$,

$$\frac{1}{n_f(t)} \sum_{j \in N_i(t)} \text{calcDiff}(\theta_j(t), \theta_f(t)) = \frac{k_t(i)}{n_t(f)}(\sigma)$$

Therefore, for every algorithm assigning different orientations, there is some algorithm assigning the same orientation, which concludes the proof. $\square$

4. STATIONARY AGENTS

In this section we consider the case in which there are $m_i(t)$ flocking agents located at a single position p_i with identical initial orientations and $k_i(t)$ ad hoc agents located at various arbitrary locations. Each agent a_i has velocity $v_i = 0$. This means that although an agent's orientation may change, its position will remain constant. An example is provided in Figure 3.

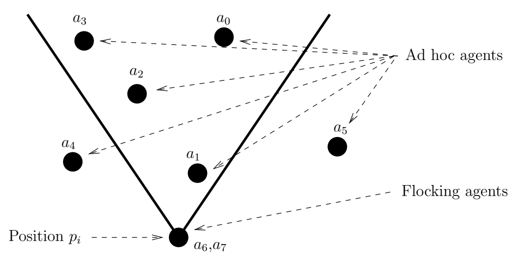

Figure 3: An example with two flocking agents (a_6 and a_7) located at the same position with identical initial orientations, four ad hoc agents (a_0, a_1, a_2, and a_3) located at different locations within the visibility cone of the flocking agents, and two ad hoc agents (a_4 and a_5) located at different locations outside the current visibility cone of the flocking agents.

As the flocking agents are influenced to turn towards θ^*, different ad hoc agents become available to influence the flocking agents. This is because some ad hoc agents may no longer be within the flocking agents' visibility cone, while other ad hoc agents may enter the visibility cone. Hence, at each time step the ad hoc agents must consider the trade-off between moving the flocking agents maximally towards θ^* and keeping (or moving) ad hoc agents within the flocking agents' visibility cone for future time steps.

In this section, we introduce some new terminology. A *border agent* is an ad hoc agent that is located within the visibility cone of the flocking agents, on the edge of the visibility cone that is farther away from the target. A *border influence orientation* is a flocking agent orientation at which an ad hoc agent is a border agent. Clearly for each ad hoc agent, there are exactly two possible border influence orientations — one in which the border agent is located on the left hand side of the flocking agents' visibility cone and one in which the border agent is located on the right hand side of the visibility cone. See Figure 4 for an example with a border agent.

Throughout this section, recall that α denotes the angle of the flocking agent a_i's visibility cone. Additionally, remember that an ad hoc agent a_j's location $p_j(t) = (x_j(t), y_j(t))$ in the environment at time t is located at angle $\beta_j(i)$ with

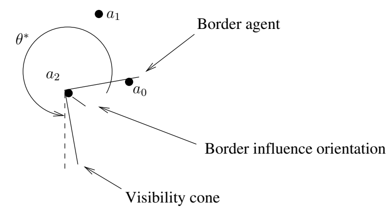

Figure 4: An example of a border agent (a_0) and the resulting border influence orientation of the flocking agent (a_2).

respect to agent a_i's position $p_i(t) = (x_i(t), y_i(t))$ and orientation $\theta_i(t)$.

The first lemma in this section puts a bound on the maximal amount the flocking agents can be influenced to turn and still have the same set of ad hoc agents and flocking agents within the flocking agents' visibility cone.

LEMMA 3. *$k_i(t)$ ad hoc agents within the neighborhood of $m_i(t)$ flocking agents can influence the $m_i(t)$ flocking agents to turn min$(\beta_j(i) + \frac{\alpha}{2}, \frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon)$ radians in one time step and still have the same $m_i(t)$ flocking agents and $k_i(t)$ ad hoc agents within the flocking agents' neighborhood.*

PROOF. In order for all the $k_i(t)$ agents to remain in the neighborhood of all $m_i(t)$ agents at time $t+1$, it is necessary for the amount the $m_i(t)$ flocking agents turn by $(\min(\beta_j(i) + \frac{\alpha}{2}, \frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon))$ plus the location of the current edge of the flocking agents' visibility cone $(\theta_i(t) - \frac{\alpha}{2})$ to be less than or equal to the orientation of the position of the ad hoc agents with respect to the position of the flocking agents $(\beta_j(i) + \theta_i(t))$. Hence,

$$\min(\beta_j(i) + \frac{\alpha}{2}, \frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon) + \theta_i(t) - \frac{\alpha}{2} \leq \beta_j(i) + \theta_i(t) \tag{4}$$

when $m_i(t)$ flocking agents and $k_i(t)$ ad hoc agents are in each flocking agents' neighborhood at time $t+1$.

If $\beta_j(i) + \frac{\alpha}{2} < \frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon$, then $\beta_j(i) + \frac{\alpha}{2} + \theta_i(t) - \frac{\alpha}{2} \leq \beta_j(i) + \theta_i(t)$. The left side of Equation 4 clearly equals the right side in this case.

Otherwise, if $\beta_j(i) + \frac{\alpha}{2} \geq \frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon$, then $\frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon + \theta_i(t) - \frac{\alpha}{2} \leq \beta_j(i) + \theta_i(t)$. Since, $\beta_j(i) + \frac{\alpha}{2} \geq \frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon$ in this case, the left side of Equation 4 is less than or equal to the right side. $\square$

The second lemma in this section sets a bound on the maximum number of time steps needed for the ad hoc agents to influence the flocking agents to reach θ^* when θ^* is reachable.

LEMMA 4. *The $k_i(t)$ ad hoc agents can influence the $m_i(t)$ flocking agents to align the team to θ^* within*

$$Z = 1 + \lceil \frac{min(\frac{\pi}{2}, \alpha)}{\frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon} \rceil$$

time steps when θ^* is reachable (i.e. the difference between $\theta_i(t)$ and θ^* is less than or equal to $(Z-1)\frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon + \beta_j(i) + \theta_i(t) - \theta_i(t+1) + \frac{\alpha}{2}$ and $\alpha > (Z-2)\frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon$).

PROOF. By Lemma 1, $k_i(t)$ ad hoc agents can influence $m_i(t)$ flocking agents to turn by $\frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon$ on each of the first $Z-2$ time steps. Additionally, by Lemma 3, $k_i(t)$ ad hoc agents can influence $m_i(t)$ flocking agents to turn by $\beta_j(i) + \theta_i(t) - \theta_i(t+1) + \frac{\alpha}{2}$ on the $Z-1$ time step and still have have $m_i(t)$ flocking agents and $k_i(t)$ ad hoc agents in each flocking agents' neighborhood. Finally, by Lemma 1 $k_i(t)$ ad hoc agents can influence $m_i(t)$ flocking agents to turn by any amount less than or equal to $\frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon$ on the last time step.

Influencing as described above must force the $m_i(t)$ flocking agents to align to θ^*; in other words, we must show that

$$(Z-2)\frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon + \beta_j(i) + \theta_i(t) - \theta_i(t+1) + \frac{\alpha}{2} + \frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon \geq \pi \tag{5}$$

By definition we know that $\alpha > (Z-2)\frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon$ and $\alpha \leq 2\pi$, so the left side of Equation 5 simplifies to $3\pi + \beta_j(i) + \theta_i(t) - \theta_i(t+1) + \frac{2\pi}{Z-2}$ such that the left side of Equation 5 is greater than or equal to the right side. □

The following theorem states that it is impossible to influence the flocking agents to orient themselves to θ^* (assuming it is reachable) in fewer than Z time steps. Remember that Lemma 4 showed that $k_i(t)$ ad hoc agents can influence $m_i(t)$ flocking agents to align the team to θ^* in Z time steps when θ^* is reachable.

THEOREM 5. *If alignment is possible,*

$$Z = 1 + \lceil \frac{min(\frac{\pi}{2}, \alpha)}{\frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon} \rceil$$

time steps are needed for the $k_i(t)$ ad hoc agents to influence the $m_i(t)$ flocking agents to align the team to θ^*.

PROOF. Assume, towards contradiction, that there exists an algorithm in which $k_i(t)$ ad hoc agents influence $m_i(t)$ flocking agents to align the team to θ^* (when alignment is possible) in $Z' < Z$ time steps.

By Lemmas 1, 3, and 4, $k_i(t)$ ad hoc agents can influence $m_i(t)$ flocking agents to turn $\frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon$ on each of the first $Z'-2$ time steps, by $\beta_j(i) + \theta_i(t) - \theta_i(t+1) + \frac{\alpha}{2}$ on the $Z'-1$ time step, and by at most $\frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon$ on time step Z'. Hence,

$$(Z'-1)\frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon + \beta_j(i) + \theta_i(t) - \theta_i(t+1) + \frac{\alpha}{2} \geq \pi \tag{6}$$

when alignment of the team to θ^* can be achieved in Z' time steps. By Lemmas 1 and 3, $\beta_j(i) + \theta_i(t) - \theta_i(t+1) + \frac{\alpha}{2} \leq \frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon$ so the left side of Equation 6 becomes $Z'(\frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon)$.

If $\frac{\pi}{2} > \alpha$, then

$$Z' \leq \frac{\alpha}{\frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon}$$

In this case, the left of Equation 6 becomes

$$(\frac{\alpha}{\frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon})(\frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon) = \alpha = \frac{\pi}{2}$$

Otherwise, if $\frac{\pi}{2} \leq \alpha$, then

$$Z' \leq \frac{\frac{\pi}{2}}{\frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon}$$

In this case, the left of Equation 6 becomes

$$(\frac{\frac{\pi}{2}}{\frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon})(\frac{k_i(t)\pi}{m_i(t)+k_i(t)} - \epsilon) = \frac{\pi}{2} < \pi$$

leading to a contradiction. □

When determining how the ad hoc agents should orient themselves to optimally influence the flocking agents, we use a *forward search* approach (see Figure 5). Specifically, beginning at the initial flocking orientation, we consider each possible border influence orientation. If the border influence orientation is reachable from the initial flocking orientation, then we consider each possible border influence orientation from this point. If the border influence orientation is not reachable from the initial flocking orientation, then we turn to the farthest reachable point and then determine if the border influence orientation is now reachable (and repeat this process until the border influence orientation is reachable). We repeat this process until the target is within reach, and we select the plan that reaches the target in the fewest number of steps.

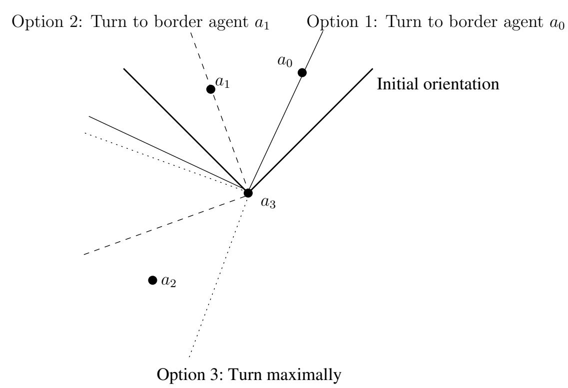

Figure 5: An example of the possible subsequent flocking agent orientations for a given initial orientation. The ad hoc agents are labelled with a_0, a_1, and a_2, while the flocking agent is labelled with a_3. Note that turning to border agent a_2 is not possible in the first time step.

A forward search such as this requires checking 2^k possible combinations of the number of ad hoc agents influencing the flocking agents at each time step. Consider the case where there are three ad hoc agents. The following eight combinations of targets covers all possible combinations: $[a_0, a_1, a_2]$, $[a_0, a_1]$, $[a_0, a_2]$, $[a_0]$, $[a_1, a_2]$, $[a_1]$, $[a_2]$, $[]$. By convention, agent a_0 will be oriented farther from the target than agent a_1, which will be oriented farther from the target than agent a_2, and so on. See Figure 6 for an example with two ad hoc agents and one flocking agent.

Algorithm 2 uses such a forward search approach to calculate and return the number of steps needed to reach θ^*

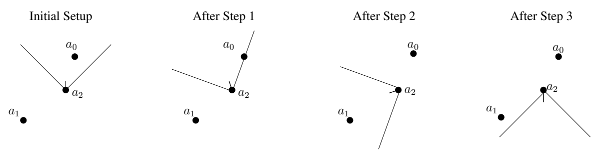

Figure 6: Example with two ad hoc agents (a_0 and a_1) and one flocking agent (a_2). On step 1, the flocking agent turns to have a_0 as a border agent. Then on step 2, the flocking agent turns as much as possible towards θ^*. Finally, on step 3 the flocking agent turns to θ^*.

and the necessary orientations for each of the ad hoc agents for each of these steps. Throughout the algorithm, element.get(x) returns the 0-indexed x item in element (where element is a list object), element.add(y) adds item y to the end of element, and element.size() returns the number of items contained in element. The variables used throughout Algorithm 2 are defined in Table 1. Remember that although the $k_i(t)$ ad hoc agents are located at many arbitrary locations, the $m_i(t)$ flocking agents are located at a single position p_i and begin with identical orientations.

Variable	Definition
adHocOrient	the orientation the ad hoc agents must adopt at this time step in order for the flocking agents to reach *target* from *current*
bestAHPlan	the ad hoc agent plan that uses the least number of time steps to reach θ^*
bestFSeq	the flocking sequence of orientations that uses the least number of time steps to reach θ^*
borderTarget	the border influence orientation needed to be a border agent
ccw	whether the flocking agents are rotating counter-clockwise
current	the orientation the flocking agents are currently oriented towards
currentAHPlan	the plan containing the orientations for each ad hoc agent at each time step so far
currentFSeq	the sequence of orientations for the flocking agents at each time step so far
initFOrient	the initial orientation of the flocking agents
maxSteps	the maximum number of steps a plan can be
numAH	the number of ad hoc agents within the flocking agents' visibility cone
numF	the number of flocking agents
target	the orientation the flocking agents should be oriented towards on the next time step
targetReachable	whether *target* is reachable from *current*

Table 1: Variables used in Algorithm 2.

In the worst case, line 1 will be executed 2^{numAH} times, line 4 will execute numAH + 1 times, and line 6 will execute maxSteps times. Hence, lines 7-24 are executed at most $(2^{\text{numAH}})(\text{numAH}+1)(\text{maxSteps})$ times.

Theorem 6. *Given θ^* and assuming the $m_i(t)$ flocking agents are influenced only by the $k_i(t)$ ad hoc agents and the $m_i(t)$ flocking agents at time t, then if θ^* is reachable, the ad hoc agents are guaranteed to lead the flocking agents to θ^* in the least number of time steps possible when the ad hoc agents determine their plan based on Algorithm 2 and the number of steps required is not larger than maxSteps.*

Proof. There are exactly 2^{numAH} possible ad hoc agent combinations. Hence, by line 1, Algorithm 2 is guaranteed to consider each possible ad hoc agent combination.

Each border influence orientation target is considered until it is reachable or the plan size becomes larger than maxSteps

Algorithm 2 plan, steps = calcPlan()

1: **for** each possible ad hoc agent combination **do**
2: currentFSeq, currentAHPlan ← empty list
3: current ← initFOrient
4: **for** each borderTarget in this ad hoc agent combination **do**
5: targetReachable ← false
6: **while** targetReachable == false ∧ currentFSeq.size() < maxSteps ∧ bestFSeq.size() > currentFSeq.size() **do**
7: target ← borderTarget
8: **if** ccw **then**
9: target ← target + $\frac{\alpha}{2}$
10: **else**
11: target ← target − $\frac{\alpha}{2}$
12: **if** |calcDiff(current, target)| > $\frac{\text{numAH}\pi}{\text{numAH+numF}}$ **then**
13: targetReachable ← false
14: **if** ccw **then**
15: target ← current + $\frac{\text{numAH}\pi}{\text{numAH+numF}} - \epsilon$
16: **else**
17: target ← current − $\frac{\text{numAH}\pi}{\text{numAH+numF}} + \epsilon$
18: **else**
19: targetReachable ← true
20: adHocOrient ← $\frac{|\text{calcDiff(target, current)}|*(\text{numF+numAH})}{\text{numAH}}$ + target
21: currentFSeq.add(target)
22: **for** each ad hoc agent x in the flocking agents' visibility cone when facing current **do**
23: currentAHPlan.get(x).add(adHocOrient)
24: current ← target
25: **if** currentFSeq is smaller than bestFSeq **then**
26: bestFSeq ← currentFSeq
27: bestAHPlan ← currentAHPlan
28: **return** bestAHPlan, bestFSeq.size()

(line 6), and necessary ad hoc orientations are added to the currentAHPlan and targets are added to the currentFSeq until it is reachable or the plan size becomes larger than maxSteps. The currentFSeq and currentAHPlan become the bestFSeq and bestAHPlan (lines 25-27) only if they use less steps to reach θ^* than the current bestFSeq and bestAHPlan. Line 25 will not be reached until all border orientation targets for a particular set of ad hoc agent combinations have been considered and θ^* has been reached or the plan size becomes larger than maxSteps. Hence, since the best possible ad hoc agent combinations are guaranteed to be considered and the number of steps required will not be larger than maxSteps, we are guaranteed that the bestAHPlan that is returned by Algorithm 2 is the least-step plan possible. □

Algorithm 2 has been implemented and tested in our custom-designed simulator FlockSim. Results from experiments using FlockSim are given in Section 5.

5. NON-STATIONARY AD HOC AGENTS

In this section we consider the case in which there are $m_i(t)$ flocking agents that are all located at position p_i with identical initial orientations. These flocking agents will remain at position p_i forever, but may change orientation if influenced by at least one ad hoc agent. In order to facilitate this, there are ad hoc agents located at arbitrary locations in the environment that can each travel with some constant velocity.

In the stationary ad hoc agents section, the main decision for each ad hoc agent was whether to influence the flocking agents to turn maximally towards θ^* or to influence the flocking agents to turn such that one of the ad hoc agents becomes a border agent. However, determining exactly how non-stationary ad hoc agents should behave is a more difficult problem. Hence, in this preliminary work we consider some heuristic approaches for how non-stationary ad hoc agents should behave.

In the stationary ad hoc agents case, it did not matter how the ad hoc agents that were not within any flocking agent visibility cones behaved because they had no influence over any flocking agents. However, non-stationary ad hoc agents travel in the direction they are facing, so it does matter what orientation they face even when they are not within the visibility cone of any flocking agents. Hence, in this work we present two heuristic behaviors for ad hoc agents that are *not* within the visibility cone of any flocking agents.

Towards Flocking Agent Orient towards the position of the flocking agent.

Towards Visibility Cone Orient towards the closest point on the flocking agents' visibility cone from the ad hoc agent's current position.

The performance of each of these behaviors is studied empirically later in this section and reported on in Figure 7. Although one of these behaviors must currently be chosen by the user for each trial, the optimal behavior likely consists of some combination of these behaviors and perhaps other behaviors. The exact situations in which each behavior should be utilized have not yet been determined. However, there are some situations where each behavior may be best. Moving towards the visibility cone may be ideal when no ad hoc agents are currently in the visibility cone to influence the flocking agent, as the flocking agent will not be able to be influenced until at least one ad hoc agent moves within its' visibility cone. On the other hand, moving towards the flocking agent may be ideal when there are ad hoc agents currently within the flocking agents' visibility cone, as moving closer to the flocking agent now will decrease the number of time steps required for the ad hoc agent to enter the flocking agents' visibility cone in future time steps.

The general behavior for non-stationary ad hoc agents that are inside a flocking agent's visibility cone is similar to the behavior of stationary ad hoc agents. Specifically, the non-stationary ad hoc agents will either influence the flocking agents to turn maximally or they will influence the flocking agents to turn such that an ad hoc agent is at the edge of the visibility cone (and hence a border agent). The main difference between the stationary ad hoc agents behavior and the non-stationary ad hoc agents behavior is that now the border agent must be on the edge of the visibility cone after updating its location. There are exactly two orientations at which a non-stationary ad hoc agent can orient and be a border agent. One of these orientations results in the ad hoc agent becoming a border agent on the left hand side of the visibility cone, while the other orientation results in the ad hoc agent becoming a border agent on the right hand side of the visibility cone. There must be exactly two orientations at which a non-stationary ad hoc agent can orient and be a border agent because any other orientations will result in either the flocking agents being influenced to turn farther and the ad hoc agent no longer moving enough to move into the visibility cone or in the flocking agents not being influenced to turn as much and the ad hoc agent being too far inside the visibility cone to be a border agent. We could find the exact orientation at which a non-stationary ad hoc agent can orient and be a border agent by performing a binary search for the exact orientation. However, we instead use a simpler, more efficient heuristic approach that finds an orientation close to the exact orientation that would be found by the binary search such that the ad hoc agent is still within the flocking agents' visibility cone after moving.

Non-stationary ad hoc agents clearly have more influence than stationary ad hoc agents. In some situations, convergence of the flocking agents to θ^* is able to occur quicker. In other situations, non-stationary ad hoc agents are able to lead the flocking agents to converge to θ^* in cases where stationary ad hoc agents would be unable to. There are situations in which an ad hoc agent can travel into the flocking agent's visibility cone and influence when it would have been unable to influence if it were stationary. In fact, stationary flocking agents can always be influenced to reach θ^* eventually when non-stationary ad hoc agents are utilized.

Empirical Evaluation

All of the heuristic behaviors discussed in this section have been implemented and tested in FlockSim. Earlier in this section we presented two heuristic behaviors for ad hoc agents that are located outside of the flocking agents' visibility cone. Now we examine each of these behaviors in FlockSim, and study (1) is there a significant difference in the number of steps required for the flocking agents to orient to θ^* with each heuristic behavior and (2) how well do our ad hoc agents perform when compared with the naive method used by others (e.g. [7, 12]) in which the controllable agents orient towards θ^* such that the flock slowly converges to θ^*?

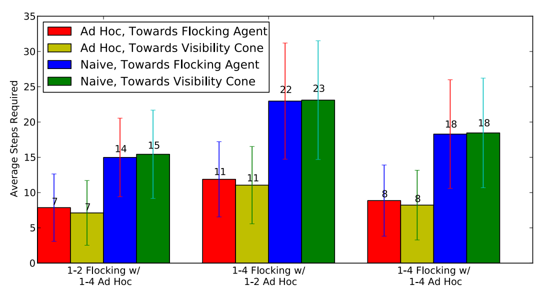

Figure 7: Results obtained using FlockSim on three different team configurations over 1000 trials.

The results of our experiments are presented in Figure 7. For these experiment, $v = 50$ for the ad hoc agents, $\alpha = 90^\circ$, $\theta^* = 270^\circ$, the initial flocking orientation was 90°, and the ad hoc agents and flocking agents were placed randomly in a 950 by 500 environment. Each of the runs within the three possible team configurations used the same randomization seed. maxSteps was set to 100, such that no trials stopped due to the plan size exceeding maxSteps. When run with teams composed of one to four non-stationary ad hoc agents and one to four stationary flocking agents on a Dell Precision-360 desktop computer, an optimal plan is found in 0.00368103 seconds on average.

As clearly seen in Figure 7, **Towards Visibility Cone** performs better than **Towards Flocking Agent** in all the configurations utilizing ad hoc agents. In order to determine whether there is a significant difference in the number of steps required for the flocking agents to orient to θ^*, we ran a Student's t-test on the step counts from the 1000 runs for each pair of heuristic behaviors. For each of the three pairs, we found the difference to be statistically significant at $p = 0.05$. **Towards Visibility Cone** likely performed better because getting into the visibility cone faster allows the ad hoc agent to influence the flocking agent sooner.

Figure 7 also clearly shows that our ad hoc agent algorithms perform significantly better than the naive method in which the controllable agents orient towards θ^*. Our ad hoc agent algorithms performed better because we purposely orient the ad hoc agents past θ^* in order to orient the flocking agents exactly to θ^* quickly. It is important to note that in this experiment we relaxed the definition of 'reaching' θ^* for the naive method. Due to the way in which the naive method slowly converges, under our strict definition of 'reaching', the naive methods would very rarely converge.

6. RELATED WORK

Although there has been much work in the field of multiagent teamwork, there has been relatively little work towards getting agents to collaborate towards a common goal without pre-coordination. Most prior multiagent teamwork research requires explicit coordination protocols or communication protocols (e.g. SharedPlans, STEAM, and GPGP) [5, 13, 4]. However, our work is different in that we do not assume that any protocol is known by all agents.

Han, Li and Guo study how one agent can influence the direction in which an entire flock of agents is moving [6]. Similarly to our work, in their work each agent follows a simple control rule based on its neighbors. However, unlike our work they only consider one ad hoc agent with unlimited, non-constant velocity. This allows their ad hoc agent to move to any position in the environment within one time step, which is unrealistic.

Reynolds introduced the original flocking model when he presented three flocking behaviors — collision avoidance, velocity matching, and flock centering [9]. His work was focused on creating graphical models that looked and behaved like real flocks, and hence did not consider adding controllable agents to the flock like we do.

Vicsek *et al.* considered just the flock centering aspect of Reynolds' model [14]. Hence, they use a model where all of the particles move at a constant velocity and adopt the average direction of the particles in their neighborhood. However, like Reynolds' work, they were only concerned with simulating flock behavior and not with adding controllable agents to the flock.

Jadbabaie, Lin, and Morse build on Vicsek *et al.*'s work [7]. They use a simpler direction update than Vicsek *et al.* and they show that a flock with a controllable agent will eventually converge to the controllable agent's heading. Like us, they show that a controllable agent can be used to influence the behavior of the other agents in a flock. However, they are only concerned with getting the flock to converge eventually, whereas we attempt to do so as quickly as possible. Su, Wang, and Lin also present work that is concerned with using a controllable agent to make the flock converge eventually [12].

Jones *et al.* perform an empirical study of dynamically formed teams of heterogeneous robots in a multirobot treasure hunt domain [8]. They assume that all of the robots know they are working as a team and that all of the robots can communicate with one another, whereas in our work we do not assume that the teammates realize they are working on a team with the ad hoc agents.

7. CONCLUSIONS

In this paper, we consider the problem of leading a flock of agents to a desired orientation using a subset of ad hoc agents. This paper's major contributions are (1) a specification for the flocking problem as a new scenario for studying ad hoc teamwork and (2) an initial theoretical and empirical analysis of this problem. We first set bounds on the extent of influence the ad hoc agents can have on the team when all the agents are stationary, and then we subsequently examine the more complicated problem of orienting a stationary team using a set of non-stationary ad hoc agents.

Although we begin to consider the non-stationary ad hoc agent case in this work, it is just an initial step towards solving the general case of non-stationary ad hoc and flocking agents. As such, we plan to extend the work presented here towards this general case in the near future. Additionally, as this paper introduces flocking from the ad hoc perspective, there are many exciting directions for future work, such as exploring different neighborhood models, and determining if there is an optimal behavior for non-stationary ad hoc agents that are outside the flocking agents' visibility cone.

8. ACKNOWLEDGEMENTS

This work has taken place in the Learning Agents Research Group (LARG) at UT Austin. LARG research is supported in part by NSF (IIS-0917122), ONR (N00014-09-1-0658), and the FHWA (DTFH61-07-H-00030).

9. REFERENCES

[1] N. Agmon and P. Stone. Leading ad hoc agents in joint action settings with multiple teammates. In *AAMAS'12*, June 2012.

[2] W. Bialeka, A. Cavagnab, I. Giardinab, T. Morad, E. Silvestrib, M. Vialeb, and A. Walczak. Statistical mechanics for natural flocks of birds. *Proceedings of the National Academy of Sciences*, 109(11), 2012.

[3] H. H. Charlotte K. Hemelrijk. Some causes of the variable shape of flocks of birds. *PLoS ONE*, 6(8), 2011.

[4] K. S. Decker and V. R. Lesser. Readings in agents. chapter Designing a family of coordination algorithms, pages 450–457. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.

[5] B. J. Grosz and S. Kraus. Collaborative plans for complex group action. *AIJ*, 86(2):269 – 357, 1996.

[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, June 2003.

[8] E. Jones, B. Browning, M. B. Dias, B. Argall, M. M. Veloso, and A. T. Stentz. Dynamically formed heterogeneous robot teams performing tightly-coordinated tasks. In *ICRA'06*, pages 570 – 575, 2006.

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

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

[11] P. Stone, G. A. Kaminka, and J. S. Rosenschein. Leading a best-response teammate in an ad hoc team. In *Agent-Mediated Electronic Commerce: Designing Trading Strategies and Mechanisms for Electronic Markets*, pages 132–146. 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] M. Tambe. Towards flexible teamwork. *JAIR*, 7:83–124, 1997.

[14] 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.