Swarming Behavior Using Probabilistic Roadmap Techniques

O. Burçhan Bayazıt[1], Jyh-Ming Lien[2], and Nancy M. Amato[2]

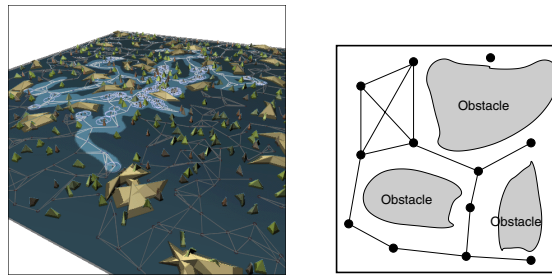
[1] Washington University, St. Louis, MO 63130, USA
bayazit@wustl.edu
[2] Texas A&M University, College Station, TX 77843, USA
{neilien,amato}@cs.tamu.edu

Abstract. While techniques exist for simulating swarming behaviors, these methods usually provide only simplistic navigation and planning capabilities. In this review, we explore the benefits of integrating roadmap-based path planning methods with flocking techniques to achieve different behaviors. We show how group behaviors such as exploring can be facilitated by using dynamic roadmaps (e.g., modifying edge weights) as an implicit means of communication between flock members. Extending ideas from cognitive modeling, we embed behavior rules in individual flock members and in the roadmap. These behavior rules enable the flock members to modify their actions based on their current location and state. We propose new techniques for several distinct group behaviors: homing, exploring (covering and goal searching), passing through narrow areas and shepherding. We present results that show that our methods provide significant improvement over methods that utilize purely local knowledge and moreover, that we achieve performance approaching that which could be obtained by an ideal method that has complete global knowledge. Animations of these behaviors can be viewed on our webpages.

1 Vision

Coordinating the movement of a swarm of robots plays an important role in robotics. Although techniques to achieve coordinated movements have attracted the attention of many researchers, most research has focused on techniques for modeling individual behaviors of flock members inspired by Reynolds' *boids* [1]. Boids exhibit so-called *emergent behavior* in which characters only react to immediate events. Although, they can be coupled with simple methods for guiding global flock movement [2], existing methods have difficulty if complex navigation is required, such as in cities, through crowded rooms, or over rough terrain. In contrast, path planning algorithms developed in the robotics community are capable of navigation in complex environments [3]. In particular we note the *roadmap*-based methods which construct, usually during preprocessing, a network of representative feasible paths in the environment. While roadmap methods can efficiently support complex navigation, they have generally not been customized to support coordinated group behavior.

E. Şahin and W.M. Spears (Eds.): Swarm Robotics WS 2004, LNCS 3342, pp. 112–125, 2005.



(a) (b)

Fig. 1. Roadmaps in navigation: (a) global navigation information can assist coordinated group behaviors, such as flocking or mine sweeping (shown here), in complex environments, (b) a PRM roadmap (C-space).

In this review, we present the benefits of integrating flocking techniques with roadmap-based path planning methods to achieve different swarming behaviors. The details of our approach including the related work can be found in [4–7]. We find that the global navigation information provided by the roadmaps can also be exploited to support more sophisticated group behaviors than possible using traditional (local) flocking methods. In particular, we consider several different behaviors: homing, goal searching, covering, passing through narrow passages and shepherding. Our new techniques can be applied to an entire flock, to individual flock members, or to an external agent that may influence the flock (e.g., a sheep dog).

2 The Swarm Robotic Environment/Methodology

In this section, we briefly describe probabilistic roadmap techniques and then discuss how these techniques can be used for swarming behaviors.

2.1 Probabilistic Roadmap Methods and Flocking Systems

Given a description of the environment and a movable object (the 'robot'), the motion planning problem is to find a feasible path that takes the movable object from a given start to a given goal configuration [3]. Since there is strong evidence that any complete planner (one that is guaranteed to find a solution, or determine that none exists) requires time exponential in the number of degrees of freedom (DOF) of the movable object [3], attention has focused on randomized or probabilistic methods.

As mentioned in Section 1, our approach utilizes a roadmap encoding representative feasible paths in the environment. While noting that our techniques could use any roadmap, our current implementation is based on the probabilistic roadmap (PRM) approach to motion planning [8]. Briefly, PRMs work by sampling points 'randomly' from the robot's configuration space (C-space), and retaining those that satisfy certain feasibility requirements (e.g., they must correspond to collision-free configurations of the movable object). Then, these points are connected to form a graph, or roadmap, using some simple planning method

to connect 'nearby' points. During query processing, the start and goal are connected to the roadmap and a path connecting their connection points is extracted from the roadmap using standard graph search techniques (see Figure 1(b)).

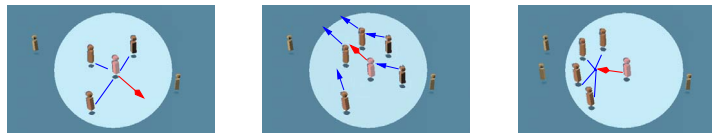
We use a particular variant of the PRM called the Medial-Axis PRM, or MAPRM [9]. In MAPRM, instead of generating the nodes uniformly at random in C-space, they are generated on or near the medial-axis of C-Space. MAPRM is particularly well suited to flocking behavior since roadmap nodes tend to maximize clearance from obstacles. Note that although the initial roadmap is found for the static environment, the roadmap, or a path extracted from it, can be modified according to dynamic changes in the environment, e.g., a new roadmap could be built from scratch [10], the existing roadmap can be modified [11–13], or a path containing collisions (an approximate path) can be modified to fit the new requirements [14].

Basic flocking systems [1] model simple group behavior by providing individual members with simple rules that implement separation (to avoid collision with nearby neighbors), alignment (to move in the same direction as its neighbors) and coherence (to stay close to neighbors) maneuvers based on the positions and velocities of the flockmates inside the sensing range. Constraints are satisfied by generating forces for each rule and applying an integrated force to change the state of the flock member, e.g., the flock member's velocity vector updated by finding the acceleration resulted from the integrated force using the Newtonian equation $F = ma$. Our implementation of this is based on particle systems [15]. In the presence of obstacles, force is also generated to push the flock member away. This basic system can be seen in Figure 2. More complicated behavior is usually simulated by adding other forces.

2.2 Roadmap-Based Group Behavior

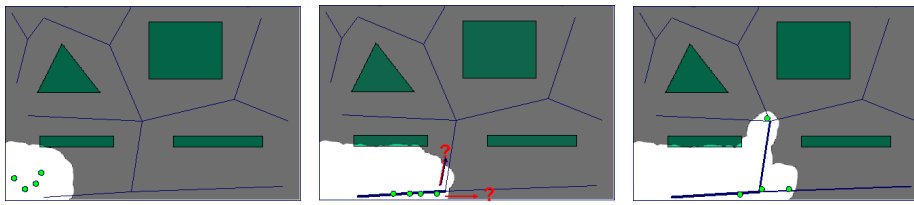
In this section, we show how roadmap-based techniques can be used to achieve different behaviors. We consider several behaviors: *homing*, *exploring (covering and goal searching)*, *traversing narrow passages* and *shepherding*. The first two behaviors influence *where* the flock goes – reaching a pre-defined goal (homing), attempting to cover (visit) all reachable areas of the environment (covering) or search for a goal whose location is not known. The narrow passage behavior influences *how* the flock members position themselves relative to each other when they move through the passage. In the shepherding, an external agent controls the movement of the flock.

Homing Behavior. Homing behavior is usually simulated by adding an attractive force toward the goal [16]. However, this method may easily be trapped in a local minimum even in a simple environment. A method commonly used in computer games requiring motion of a group of objects is a grid-based A^* search [17]. In this approach, the environment is discretized to small grid cells and the search for the flock's path is based on expanding toward the most promising neighbor of already visited positions. Although A^* search finds shortest paths and it is usually fairly fast, it does have some drawbacks. Of particular note here



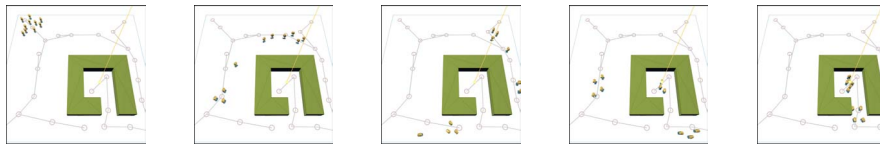
(a) (b) (c)

Fig. 2. Individual member behavior for flocks. (a) Separation: avoid crowding neighbors. (b) Alignment: match velocity of neighbors. (c) Cohesion: stay close to neighbors. The arrow represents steering direction.



(a) (b) (c)

Fig. 3. Covering an environment. (a) A roadmap is built. (b–c) Robots move to the roadmap and increase the weights as they move along the edges. At an intersection the robots select their destination by a probability function based on the edge weights (edges with small weights are preferred).



(a) (b) (c) (d) (e)

Fig. 4. Ten flock members are searching for an unknown goal. (a) The flock faces a branch point. (b) Since both edges have the same weight, the flock splits into two groups. (c) After dead ends are encountered in the lower left and upper right, edge weights leading to them are decreased. (d) As some members find the goal, edge weights leading to it are increased. (e) The remaining members reach the goal.

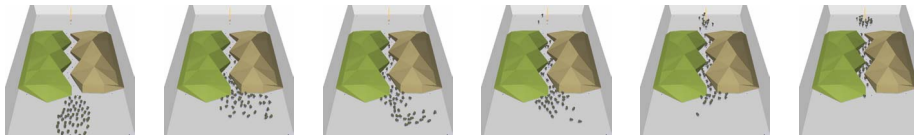


Fig. 5. Passing through a narrow passage using the FOLLOW THE LEADER behavior (Algorithm 2.4).

is the necessity of finding a completely new path for each new goal which reduces efficiency and increases the computation time for complex environments.

In contrast, roadmap-based path planning methods work on a global scale and once the roadmap is generated, finding new paths is fast and efficient. Once a

path is found, individual flock members follow the path. The path is partitioned into a set of subpaths (identified by subgoals) based on the individual flock member's sensor range. Each member keeps track of subgoals and as soon as a subgoal comes within sensory range the next subgoal becomes the steering direction for the global goal.

With other interacting forces from neighboring flock members and obstacles, steering toward the subgoal has the lowest priority, so individual members still move together while moving toward the subgoal. Since the subgoals are usually away from the obstacles, due to global roadmap, this approach results in a flocking toward the goal and avoids getting trapped in local minima. The homing behavior is shown in Algorithm 2.1.

Covering the Environment. In this behavior we want some member of our flock to have covered every location in the environment. We assume we start with a roadmap covering all relevant portions of the environment and the roadmap has adaptive edge weights. In this approach, each individual member uses the roadmap to wander around. Specifically, the flock members follow roadmap edges and there are no predefined paths. The goal is to have some flock member visit every edge and vertex of the roadmap (see Figure 3). The edge weights represent how relevant the edge is to the current task, in this case exploring the environment. Initially, edges all have weight one. As the flock members traverse a roadmap edge they increase its weight. This is similar to ant pheromones which increase as more ants follow the same path. Since our goal is to cover the environment, the individual flock members are biased toward relatively uncovered areas of the roadmap. This is achieved by having them select roadmap edges with smaller weights with some higher probability at the intersections (roadmap nodes). This algorithm is shown in Algorithm 2.2.

Goal Searching. Our goal searching behavior is similar to *ant colony optimization (ACO)*. Although the individual flock members know the environment, they don't know the location of the goal. If an individual reaches a location where the goal is within sensor range, all other members try to reach the goal. Like the previous case, we implemented this behavior using adaptive roadmap edge weights. The weight of an edge shows how promising a path segment is. Again, the member chooses an edge to leave a roadmap node with some probability based on the edge's weight. As an individual traverses a path in the roadmap, it remembers the route it has taken. Then, when it reaches a goal, it increases the weight of the edges on the route it took. If the individual reaches a roadmap node without any outgoing connections (i.e., with only one edge) or a node already contained in the current path (i.e., a cycle), the weight of the edges it followed will be decreased. This approach is summarized in Algorithm 2.3 and illustrated in Figure 4.

Narrow Passage Behavior. Sometimes the flock's behavior depends on the surrounding environment. For example, different group formations may be used in relatively open areas than are used when passing through narrow regions. One

Algorithm 2.1 Homing

```
if (goal is in view range) then
    set goal as target.
else if (target is in view range) then
    set next subgoal as the target.
end if
steer toward the target.
```

Algorithm 2.2 Covering the Env.

```
for (each flock member) do
    while (not all nodes visited) do
        if (not in the roadmap) then
            move to closest roadmap node
        end if
        if (current node has no outgoing edge) then
            pop stack until a new branch is found
        else
            probabilistically pick a lower-weight edge
            increase edge weight
            push this node onto the stack
        end if
    end while
end for
```

Algorithm 2.3 Goal Searching

```
for (each flock member) do
    if ( goal found) then
        increase edge weights on path to goal
    else if (dead end found) then
        pop stack until a new branch is found
        decrease weight of edge corr. to popped node
    else
        select a neighboring node of the current node
        push this node onto the stack
    end if
end for
```

Algorithm 2.4 Narrow Passage

```
while (not all flock members in gathering area) do
    set individual members' goal to gathering area
end while
set leader to NIL
while (there are flock members outside passage) do
    select the closest unselected member as Current
    if (Leader is NIL) then
        set Leader to Current and set Leader's goal to next step in the path
    else
        set Current's goal to Previous
    end if
    set Previous to Current
    increase neighbor avoidance threshold
end while
```

Algorithm 2.5 Shepherding (for dog)

```
Find a path on roadmap
while (goal not reached) do
    Select the next node on the path as subgoal
    while (subgoal not reached) do
        Move to rear of flock on the far side of subgoal
        if (flock separates) then
            Move the subgroup that is farthest from subgoal toward other subgroups
        end if
    end while
end while
```

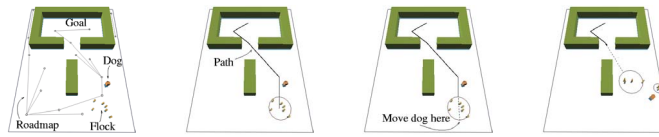


Fig. 6. Shepherding: sheep are represented by large circles and the dog by a small dark circle. (a) Roadmap, (b) path selected by dog, (c) dog's steering location, (d) flock is separated.

nice property of roadmaps is that, the roadmap nodes can be the representatives of different regions of the environment. Hence, different rules or navigation strategies can be assigned to different nodes and if a flock member reaches a node, it follows the rules associated with that node.

We employ different rules to pass through a narrow passage. A naive way to achieve narrow passage traversal by the flock is to use the homing behavior and to select two nodes as goals, first a node in front of the entrance to the passage and then a node outside the exit from the passage. One drawback of this approach is that flock members may bunch up and conflict with each other as they try to move through the passage.

A *follow-the-leader* strategy may avoid the congestion problems of the naive strategy (see Figure 5). In this strategy, we first assemble the flock in front of the narrow passage, and then select the closest agent to the narrow passage entrance as the leader. Then, the remaining flock members are arranged into a queue that follows the leader. Their position in the queue depends on their distance to the entrance of the narrow corridor. They can be kept from crowding each other by selecting appropriate values for the repulsive force from other flock members.

Note that different behaviors can be achieved by using a different criterion to select the next flock member in line 6 of Algorithm 2.4. For example, instead of selecting the next closest flock member to the narrow passage, one might select the farthest, which would create a 'milling around' effect at the entrance to the passage.

Shepherding Behavior. In the previous sections we have observed two distinct class of flocking behaviors. In the first case, the flock members were moving toward a goal together, i.e., as a flock. The motion was planned for the flock. In the second case, the flock members were exploring and planning their motions individually. In a sense, the flock had control of the motion in the first case and individual flock members had control in the second case. In our third scenario, neither the flock nor the individuals have control of the motion. Instead, an outside agent guides or shepherds them. In the simulation shown in Figure 6(a), the external agent is a dog whose objective is to move the flock of sheep toward the goal. The only motion control for the flock is to move away from the dog. A similar implementation has been done in [18] where a robot was programmed to move geese toward a goal position. We would like to implement a similar algorithm where a subgoal will be a roadmap node found in the path. Until the

subgoal is reached, the robot will move toward that goal and then will choose the next roadmap node on the path as the next subgoal (see Figure 6(b)).

To move the flock toward the goal, the dog steers the flock from behind (Figure 6(c)). If any subgroup separates from the flock, it is the dog's job to move the subgroup back to the flock (Figure 6(d)). Our approach is presented in Algorithm 2.5. We present an improved shepherding algorithm in [7].

3 Results

In this section we evaluate our roadmap-based techniques for the homing, exploring, and shepherding behaviors that were described in Section 2. Movies illustrating the experiments as well as the behaviors in three-dimensional space with rigid or deformable objects can be found on our webpage (`http://www.cse.wustl.edu/~bayazit`).

Our experiments are designed to compare our roadmap-based techniques with more traditional approaches for simulating flocking behavior and to study the improvements possible by incorporating global information about the environment as encoded in a roadmap.

To study the efficiency of our covering and goal searching techniques, we also compare our roadmap-based techniques with 'ideal' variants which have complete knowledge of the environment and the current status of the search. For example, in the goal searching behavior, the location of the goal is known at all times in the ideal variant. A more through evaluation of our approach can be found in [4–7] where we presented additional results for narrow passage and shepherding.

All of our experiments were run on a Linux system with Athlon 1.33 processor and 256MB memory.

3.1 Homing Behavior

For the homing behavior, our roadmap-based technique is compared with a basic flocking behavior using a potential field [16] and a grid-based A^* search behavior.

The environment is a square with sides measuring 420 meters (see Figure 7). It contains a total of 301 randomly placed obstacles (six types of obstacles are used). At any given time there is one goal, and when all flock members reach it, a new goal is randomly generated; this process continues until eight goals have been generated and reached. The experiment involves 40 flock members, which are initially placed according to a Gaussian distribution around the center of the square environment. The simulation is updated every 100 ms.

For the flocking behavior using a potential field, flock members are attracted towards the current goal. For the grid-based A^* behavior, a bitmap of the environment of 914×914 cells is constructed; the length of a side of each square cell is equal to the diameter of a flock member. Cells are classified as free cells and collision cells. Path to the current goal is found in this bitmap using A^* search. For the roadmap-based behavior, the roadmap is built using the MAPRMmethod (Section 2.1) to generate 400 roadmap nodes and we attempt to connect each

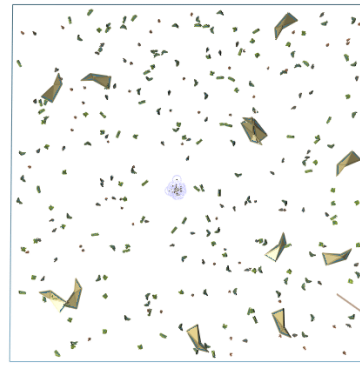


Fig. 7. Environment for homing experiments.

node to its 4 nearest neighbors. Path to the current goal is found using this roadmap.

Table 1 shows that, without global information, only a few flock members reach the last goal and most are trapped in local minima. On the other hand, when global navigation information is utilized, either with the grid-based A^* method or our roadmap-based method, all flock members reach the goal.

In Table 2 we show the time spent searching for paths, the number of local minima encountered along all paths, and the total time spent escaping from local minima. This offers some insight into the methods studied, as can be seen more clearly in Figure 8. Although the flock takes a shorter path with the grid-based A^* search than with the roadmap-based method (Figure 8(a)), the flock reaches the final goal faster with the roadmap-based method (Figure 8(b)). As A^* search

Table 1. Homing behavior. This table shows how many of the 40 flock members reach the last goal (8th) within 30 seconds using the basic flocking behavior, the grid-based A^* behavior, and the roadmap-based behavior.

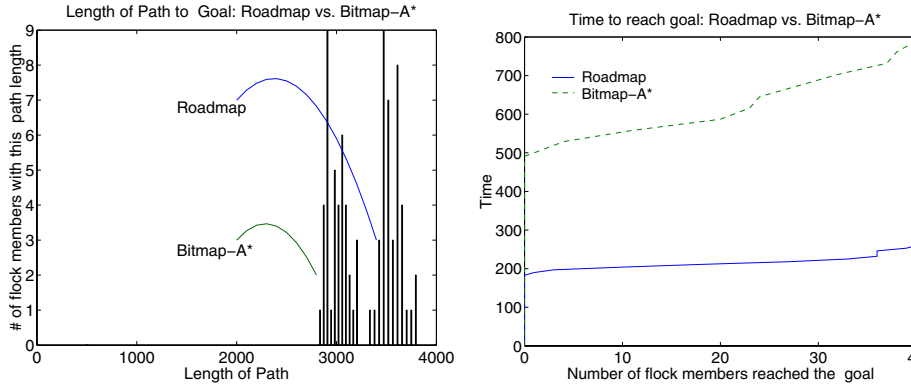
Homing behavior: Basic v.s. Roadmap

Method	#flockmates reaching the goal
Basic	10
grid-based A^*	40
roadmap-based	40

Table 2. Homing behavior. This table shows the time for initialization, the average time to find a path, and the total time spent by all flockmates escaping local minima if they stuck in a place due to quickly changing forces.

Homing behavior: Roadmap v.s. grid-based A^*

Behavior / Method	init time	find path time	local minima #	local minima escape (s)
roadmap-based	0.88	0.652	255	22.99
grid-based A^*	6.02	5.757	2005	1035.43



(a) (b)

Fig. 8. Homing behavior: (a) The number of flock members reaching goals with respect to the length of the paths they took. (b) The number of flock members reaching goals over time. Although the grid-based A^* behavior finds shorter paths, the flock spends less time to reach the goals with the roadmap-based behavior.

is known to be fast and to find shortest paths, this example illustrates that our roadmap-based method indeed is a competitor for grid-based A^* methods – while the paths found are a bit longer, they are found faster.

3.2 Covering the Environment

Space covering is tested on the environment shown in Figure 9, which requires flock members to pass through narrow passages to access undiscovered areas. In this experiment, we compare basic flocking behavior, roadmap-based behavior, and an ideal variant of the roadmap-based behavior that has dynamic knowledge of the undiscovered regions.

The environment (80×100) is populated with 16 obstacles (6 types of obstacles) and in total 24% of the environment is occupied by obstacles. 50 flock members are simulated and states are updated every 100ms. A bitmap is built to record discovered/undiscovered information. A bitmap cell is discovered when it is inside the sensory range of any flock member. We set the radius of the sensory circle as 5m. For the roadmap, 120 nodes are sampled and connections are attempted to each node's 4 nearest neighbors.

The roadmap-based covering behavior is described in Section 2.2.

The basic behavior uses only local information, and is essentially a random walk through the environment. It shows that the lack of global knowledge results in some areas never being discovered, especially those nearly surrounded by obstacles.

The behavior with perfect knowledge of the undiscovered locations uses the roadmap to find paths from a flockmate's current position to the closest unexplored spot. Although such knowledge would not be available in this covering application, this variant gives us an idea of how fast an environment can be covered in the best case.

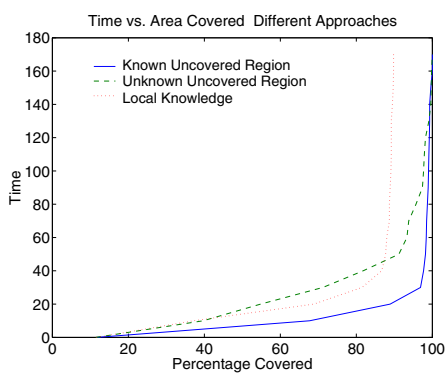


Fig. 9. Covering behavior: the percentage of the environment covered in terms of time (seconds).

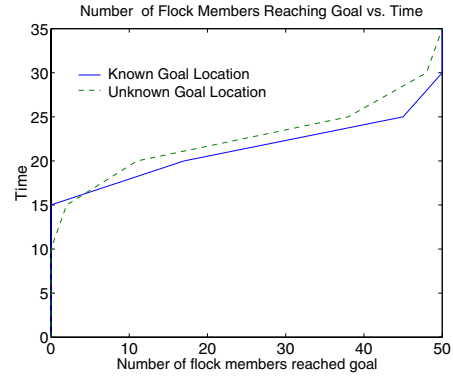


Fig. 10. Goal searching behavior: the number of flock members reaching the goal area in terms of time (seconds).

As seen in Figure 9, the perfect behavior rapidly covered almost 91% of the environment in the first 30 seconds. The roadmap-based behavior, using indirect communication (adaptive edge weights) takes about three times as long (90 seconds), to reach a similar coverage point of 91.6%. Nevertheless, like the perfect behavior, the roadmap-based behavior found most reachable areas. In contrast, the basic flocking behavior had difficulty covering more than 80% of the environment. However, it is interesting to note that the basic flocking behavior found more undiscovered areas than the roadmap-based approach in the first 40 seconds; this is due to the basic behavior which tends to bounce around and discover 'easy' areas very quickly.

3.3 Searching for a Goal

In this experiment, the roadmap-based behavior is compared with a simple flocking behavior that has only local information about the environment and no knowledge of the goal position, and with an ideal variant of the roadmap-based behavior that has *a priori* knowledge of the position of the goal. The environment is the same as that used in the covering experiment.

We are interested in how many flock members reach the goal and how fast they get there. As previously mentioned, the behavior with complete knowledge is used to establish a best case (lowerbound) for the simulation efficiency, and the basic behavior using only local information is used to illustrate the importance of global knowledge. The results of some experiments are shown in Figure 10. The flocks using the basic behavior do not discover any goals within 35 seconds, and in particular, none of the flock members discover the narrow passage out of the confined region in which they start. Overall, the roadmap-based behavior is competitive with the ideal roadmap-based behavior – only taking 5 seconds longer than the method in which the position of the goal is known *a priori*. In addition, it is surprising to note that two of the flock members in the roadmap-

based method reach the goal earlier than any of their flockmates in the ideal roadmap-based behavior. While we expect the roadmap-based method to continue to perform well in more complex environments, we expect its efficiency relative to the ideal method to decline somewhat.

4 Discussion and Outlook

In this paper, we have shown that complex group behaviors can be generated if some global information of the environment is available. The global knowledge used is a roadmap of the environment. The information it contains, such as topological information and adaptive edge weights, enables the flock to achieve behaviors that cannot be modeled with local information alone. Moreover, since in many cases global knowledge involves high communication costs between individuals, indirect communication though dynamic updates of the roadmap's edge weights provides a less expensive means of obtaining global information.

Our simulation results for the types of behaviors studied show that the performance of the roadmap-based behavior is very close to an ideal behavior that has complete knowledge. Our future work will focus on shepherding with multiple external agents and searching for moving goals, as in pursuit/evasion games. We are also working on exploration strategies if the environment is not known a priori.

5 Related Work

Reynolds' influential flocking simulation [1] showed that flocking is a dramatic example of emergent behavior in which global behavior arises from the interaction of simple local rules. Each individual member of the flock (boid), has a simple rule set stating that it should move with its neighbors. This concept has been used successfully by researchers both in computer graphics and robotics. Tu and Terzopoulos [19] used flocking behaviors with intention generators to simulate a school of fish. They also demonstrated shepherding behavior in which a T-Rex herds raptors out of its territory.

A number of related methods for achieving group behaviors have been proposed. Nishimura and Ikegami [20] used flocking dynamics to investigate collective strategies in a "prey-predator" game model. Ward et al. [21] studied an evolving sensory controller for producing schooling behavior based on "boids". Brogan and Hodgins [22] investigated group behavior with significant dynamics, such as human-like bicycle riders. Sun et al. [23] achieve swarm behaviors based on a biological immune system. Balch and Hybinette [24] propose a behavior-based solution to the robot formation-keeping problem. Fukuda et al. [25] describe group behavior for a Micro Autonomous Robotics System. Mataric [26] classifies a basic set of group behaviors which can be used to create more complex behaviors including flocking and herding. Saiwaki et al. [27] use a chaos model to simulate a moving crowd. An interesting approach by Vaughan et al. [18] used a robotic external agent to steer a flock of real geese.

Although there is little research on path planning for flocks, many methods have been proposed for planning for multiple robots. These methods can be characterized as centralized or decoupled. Centralized methods consider all robots as one entity, while decoupled methods first find a path for each robot independently and then resolve conflicts. In work from Li et al. [28], each group of crowds is guided by a leader and the paths of the leaders are generated using a decoupled approach.

The observation of the behavior of ant colonies has inspired the ant colony optimization (ACO) meta-heuristic for discrete optimization. Dorigo et al. [29] exploit this ant-like behavior to optimize solutions for several NP-Complete problems. In our work, the flock's ability to explore comes from using an ACO-like approach to adaptively adjust roadmap edge weights.

References

1. Reynolds, C.W.: Flocks, herds, and schools: A distributed behavioral model. In: Computer Graphics. (1987) 25–34
2. Reynolds, C.W.: Steering behaviors for autonomous characters. In: Game Developers Conference. (1999)
3. Latombe, J.C.: Robot Motion Planning. Kluwer Academic Publishers, Boston (1991)
4. Bayazit, O.B., Lien, J.M., Amato, N.M.: Better flocking behaviors using rule-based roadmaps. In: Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR). (2002)
5. Bayazit, O.B., Lien, J.M., Amato, N.M.: Better group behaviors in complex environments using global roadmaps. In: Artif. Life. (2002)
6. Bayazit, O.B., Lien, J.M., Amato, N.M.: Roadmap-based flocking for complex environments. In: Proc. Pacific Graphics. (2002) 104–113
7. Lien, J.M., Bayazit, O.B., Sowell, R.T., S. Rodrigues, L., Amato, N.M.: Shepherding behaviors. In: Proc. IEEE Int. Conf. Robot. Autom. (ICRA). (2004) 4159–4164
8. Kavraki, L., Svestka, P., Latombe, J.C., Overmars, M.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Trans. Robot. Automat. **12** (1996) 566–580
9. Amato, N.M., Bayazit, O.B., Dale, L.K., Jones, C.V., Vallejo, D.: OBPRM: An obstacle-based PRM for 3D workspaces. In: Robotics: The Algorithmic Perspective, Natick, MA, A.K. Peters (1998) 155–168 Proceedings of the Third Workshop on the Algorithmic Foundations of Robotics (WAFR), Houston, TX, 1998.
10. Hsu, D., Kindel, R., Latombe, J.C., Rock, S.: Randomized Kinodynamic Motion Planning with Moving Obstacles. In: Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR). (2000) SA1–SA18
11. Bohlin, R., Kavraki, L.E.: Path planning using Lazy PRM. In: Proc. IEEE Int. Conf. Robot. Autom. (ICRA). (2000) 521–528
12. Nielsen, C.L., Kavraki, L.E.: A two level fuzzy prm for manipulation planning. IEEE/RSJ International Conference on Intelligent Robotics and Systems (2000)
13. Song, G., Miller, S.L., Amato, N.M.: Customizing PRM roadmaps at query time. In: Proc. IEEE Int. Conf. Robot. Autom. (ICRA). (2001) 1500–1505
14. Bayazit, O.B., Song, G., Amato, N.M.: Enhancing randomized motion planners: Exploring with haptic hints. Autonomous Robots, Special Issue on Personal Robotics **10** (2001) 163–174 Preliminary version appeared in *ICRA 2000*, pp. 529–536.

15. Witkin, A., Baraff, D.: Physically Based Modeling: Principles and Practice. SIGGRAPH'97 Course Notes #19, SIGGRAPH-ACM publication (1997)
16. Khatib, O.: Real–time obstacle avoidance for manipulators and mobile robots. Int. J. Robot. Res. **5** (1986) 90–98
17. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. 1th edn. Prentice Hall (1994)
18. Vaughan, R.T., Sumpter, N., Henderson, J., Frost, A., Cameron, S.: Experiments in automatic flock control. J. Robot. and Autonom. Sys. **31** (2000) 109–117
19. Tu, X., Terzopoulos, D.: Artificial fishes: Physics, locomotion, perception, behavior. In: Computer Graphics. (1994) 24–29
20. Nishimura, S., Ikegami, T.: Emergence of collective strategies in prey-predator game model. Artif. Life **3** (1997) 243–260
21. Ward, C., Gobet, F., Kendall, G.: Evolving collective behavior in an artificial ecology. Artif. Life **7** (2001) 191–209
22. Brogan, D.C., Hodgins, J.K.: Group behaviors for systems with significant dynamics. In: Autonomous Robots. (1997) 137–153
23. Sun, S.J., Sim, D.W.L.K.B.: Artificial immune-based swarm behaviors of distributed autonomous robotic systems. In: Proc. IEEE Int. Conf. Robot. Autom. (ICRA). (2001) 3993–3998
24. Balch, T., Hybinette, M.: Social potentials for scalable multirobot formations. In: Proc. IEEE Int. Conf. Robot. Autom. (ICRA). (2000) 73–80
25. Fukuda, T., Mizoguchi, H., Sekiyama, K., Arai, F.: Group behavior control for MARS (micro autonomous robotic system). In: Proc. IEEE Int. Conf. Robot. Autom. (ICRA). (1999) 1550–1555
26. Mataric, M.J.: Interaction and Intelligent Behavior. PhD thesis, MIT EECS (1994)
27. Saiwaki, N., Komatsu, T., Yoshida, T., Nishida, S.: Automatic generation of moving crowd using chaos model. In: IEEE Int. Conference on System, Man and Cybernetics. (1997) 3715–3721
28. Li, T.Y., Jeng, Y.J., Chang, S.I.: Simulating virtual human crowds with a leader-follower model. In: Proceedings of 2001 Computer Animation Conference. (2001)
29. Dorigo, M., Caro, G.D., Gambardella, L.M.: Ant algorithms for discrete optimization. In: Artificial Life. Volume 5. (1999) 137–172