Roadmap-Based Group Behaviors: Generation and Evaluation

Samuel Rodri guez sor8786@cs.tamu.edu
Robert Salazar rms1454@cs.tamu.edu
Troy McMahon tmcmahon@cs.tamu.edu
Nancy M. Amato amato@cs.tamu.edu

Technical Report TR07-004
Parasol Lab.
Department of Computer Science
Texas A&M University

September 25, 2007

Abstract

We explore the benefits of integrating roadmap-based path planning methods with agents performing group behaviors to achieve different objectives. We show how a wide range of group behaviors can be facilitated by using dynamic roadmaps.

1 Introduction

Simulating the coordinated behavior of multiple agents has been studied in many fields including robotics, computer animation and games. Creating complex behaviors for a group of agents in order to achieve some behavior can be a difficult and time consuming task. Automated approaches for motion generation typically involve explicitly defining a set of possible agent behaviors, associating appropriate behaviors with all environmental events, and setting the priorities among various behaviors in every possible situation.

Generally, such approaches are pre-tuned to particular situations and are difficult to adapt for other scenarios or for different sets of behaviors. An adaptive approach to automatic behavior generation will typically involve the steps of creating initial behaviors, selecting potential behaviors to use, and adaptively selecting between an approapriate behavior depending on the performance of a given behavior.

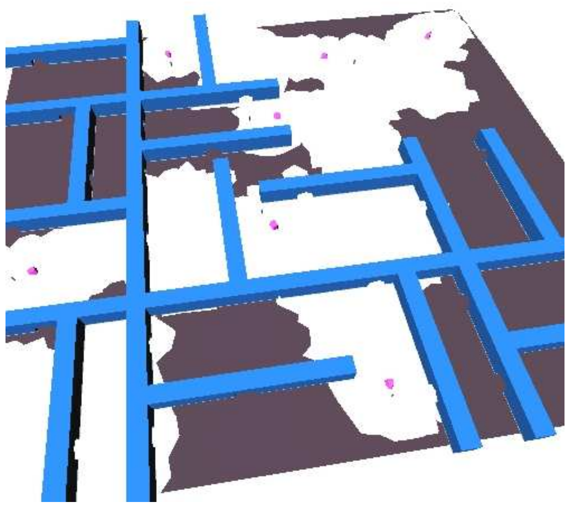


Figure 1: Coverage shown of agents searching through an environment.

2 Framework

In this paper we focus on two aspects of the adaptive approach to behavior generation. First, we propose a variety of seaching and hiding group behaviors. The behaviors we propose use an underlying roadmap which will be described more later. Although not all of these behaviors can be considered the most effective, there can be applications where each behavior can be useful. In addition, in a complete framework, the searching and hiding behaviors could be combined adaptively with their respective variants to produce more effective searching and hiding behaviors. Even though we focus on searching and hiding behaviors, the type of analysis we propose here is general enough to work for many kinds of behaviors.

Secondly, in this paper we focus on how these behaviors can be evaluated. The evaluation of a behavior that a group of agents are executing can depend on the objective the agents are trying to achieve. For example, the evaluation of agents searching through an environment would differ than that of agents trying to spread out through the environment to occupy the most space depending of the evaluation function. The coverage of agents searching through an environment can be seen in Figure 1. The evaluation can also depend on the type of behavior being evaluated (i.e., a behavior that can be evaluated on its own versus one that needs to be evaluated while considering another group of agents).

In a complete framework of automatically generating behaviors, the evaluation function could be used in order to adaptively select between behaviors. Although we do not consider the aspect of selecting between behaviors, we will compare behaviors under varying conditions in order to evaluate the effectiveness of each behavior. The kind of behavior testing, evaluation and comparisons that we propose in this work can be used when equiping a group of agents with potential behaviors. This can be done for actual robots or agents in a virtual reality environment.

3 Related Work

There has been much research in the area of allocating tasks to multiple robots. An algorithm for multi-task allocation which is based on the behavior patterns of insects is described in [1]. Several algorithms to select for a desired distribution of behaviors are presented in [2]. In [3] several more strategies for multi-robot task allocation are explored.

Specific factors can also be considered when assigning tasks. The algorithm developed in [4] explores the problem of task allocation for multiple robots, each with a predetermined set of tools. Events are assigned to robots such that the robots have all the tools necessary to handle that event. In [5], a psychological model is used to select between various behaviors. Some of the factors considered are emotion, stress, and motivation. In [6] a model is presented in which the location in the environment can provide access to additional behaviors for agents, suchas crossing a street.

Shepherding techniques have been developed where one group of agents (the shepherds) try to control the movement of another group of agents (the flock). In [7, 8] a variety of behaviors were developed that utilize a roadmap in order to perform shepherding tasks. Some of these tasks include herding a flock, pushing the flock in order to cover an environment or patrol a given area. These behaviors have been developed for a single and multiple shepherds.

In [9], a Multi-agent Navigational Graph is constructed using first- and second-order Voronoi diagrams and then used for local collision avoidance and global path planning. This sort of data structure can be an alternate to using a roadmap to describe an environment. The use of roadmaps and related techniques are discussed in the following section.

4 Roadmap-Based Group Behaviors

In [10–12], we explore the benefits of integrating roadmap-based path planning techniques with flocking techniques. We extend ideas from cognitive modeling [13], and embed behavior rules in individual flock members and in the nodes and edges of thae roadmap. We find that the global information provided by our rule-based roadmaps improves the behavior of autonomous characters, and in particular, enables more sophisticated group behaviors than are possible using traditional (local) flocking methods [14].

4.1 Aspects of Integrating Roadmaps with Group Behaviors

Some key features of integrating roadmaps with basic group behavior include:

- The roadmap provides a convenient abstract representation of global information in complex environments.
- Adaptive roadmaps (e.g., modifying node and edge weights) enable communication between agents.
- Associating rules with roadmap nodes and edges enables local customization of behaviors.

4.2 Generating and Maintaining Roadmaps

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 [15]. 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 robot). These points are then 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.

4.3 Utilizing Roadmaps

Our primary utilization of the roadmap is to generate paths between an agent and its destination. We than generate forces between the agent and the successive nodes in this path until it reaches its destination. The roadmap is also used to coordinate the actions of multiple agents. We also use the roadmap to store information about the environment, and to facilitate communication between agents. Agents that pass a node in the roadmap can store information in that node. This information can be anything from observations about the local environment to instructions for other agents. Subsequent agents that pass this node can access this information and take the appropriate action based on this information. Agents can also perform searches on the roadmap in order to determine what actions they should take. The roadmap can also be used to associate agents with regions in the roadmap. Agents can be assigned to a region of the environment by being assigned to nodes within that region.

5 Behaviors

In this section we describe details related to the searching and hiding behaviors used in this study.

5.1 Searching Behaviors

5.1.1 Covering – Basic

This covering behavior was developed to show how a group of agents can utilize a roadmap in order to effectively cover an environment [11]. The goal of this behavior is for the agents in the group to visit as much of the environment as possible. For this behavior, agents use the roadmap to obtain pathways to free areas of the environment. The effectiveness of this covering behavior has an underlying dependence on the quality of the roadmap used by the agents.

In this behavior, the goal is to have some agent in the group visit as many edges and vertices of the roadmap. As agents in the group traverse a roadmap, they select roadmap edges to follow based on the edge weights of edges connected to the nearest roadmap node. All edges are initially weighted the same. The edge selected is randomly selected considering the weight associated with each of the edges connected to a node. As the agents traverse a roadmap edge the weight is increased to indicate that the edge has been traversed. Since the goal is to cover the environment, the individual agents are biased toward relatively uncovered areas of the roadmap. This is achieved by having them select roadmap edges with smaller weights.

5.1.2 Covering – Rendezvous

The objective of this behavior is for the agents to traverse the roadmap to a randomly selected goal node. Furthermore, the agents are to to select different paths on the roadmap if possible. In order to accomplish this, the agents must have access to a global roadmap with adaptive edges, preferably with a high level of connectivity to provide many alternate routes from a given node to any other node.

As the agents determine their paths to a goal, they will reweight the edges connected to the nodes on the path. Subsequent agents will be biased away from these reweighted edges, and will seek other paths.

Upon reaching the goal node, an agent will wait at that node until the other agents have also arrived. At this point, the original weights on the roadmap will be restored, and a new goal will be randomly chosen.

5.1.3 Covering – Scanning

The objective of this behavior is to determine locations that are currently unobserved, and move to a position where those locations are observable. All points in list of unobserved locations will meet one of two conditions. Firstly, the point is out of the sensory range of the agent. Second, the point is within sensory range of the agent, but there is an obstacle between the agent and the point, thus preventing detection of the point.

The agent utilizes the roadmap to navigate to the currently selected point in the list. Once that point is observed, it is removed from the list. Furthermore, any other points in the list that are observed on the way to the currently selected point will also be removed from the list.

The agents are able to communicate when in direct sensory range, provided that there are no obstacles between the agents. When communicating, any points that one agent can observe that the other is seeking will be marked as covered. Similarly, when obtaining a new list of locations to explore, an agent will discard points that are detectable by agents with whom it can communicate.

5.1.4 Covering – Territorial

Territorial agents attempt to claim regions while avoiding regions claimed by other agents. Whenever they encounter a region that is claimed by another agent they leave that region. Whenever they encounter an unclaimed region they have the option of claiming it. If the agent claims a region then that region will remain claimed for a fixed amount of time.

There are 3 variations of this behavior. The first variation selects its destination at random; this was intended to use this as a base line to compare the other variations as well as to model the behavior simpler organisms. The second variation gives preference to nodes that it has visited. The third variation utilizes an A* search where regions are evaluated based on the agents previous observations of the region. If the region is already claimed by the agent then the value of that region will be based on the time remaining on that claim

5.2 Evading Behaviors

5.2.1 Hiding

This objective of this behavior is for the agents to find a location in the roadmap that is unobserved by opposing agents. An agent will remain at a hiding location until it is detected by an opposing agent, and it will then determine its next hiding location.

The criteria for selecting a hiding location is primarily the number of opposing agents that are currently able to detect that position. The agents are biased towards locations that have the fewest agents able to detect a location, with the preference being zero detecting agents. In the case that several locations of equal rank are selected, the distance to the current location is used to choose between them. The new location may be discarded if the agent determines that there are more agents able to detect the new location that had previously been estimated.

5.2.2 Zone Avoid

This behavior follows the same strategy as the hiding behavior. However, it keeps more information about the environment in memory. In particular, the agents will store the last observed locations of the opposing agents, and will use that to estimate the areas which those agents could potentially observe. The agents will then attempt to stay out of these areas when chosing a hiding location.

Agents of this behavior are able to communicate, and thus share information about last known locations of other agents. In the case that both agents have differing information about the same opposing agent, both of the agents will accept the information that has a more current location for the opposing agent.

6 Evaluation Functions

In order to determine the effectiveness of behaviors that have been developed, we have implemented evaluation functions to evaluate the behaviors. These evaluation functions should return a value that indicates some performance metric of a given behavior periodically as the behavior is running. The evaluation functions we discuss here are independent functions, which can evaluate a behavior on its own, and behavior-behavior functions which require complementary behaviors being run in order to evaluate a given behavior.

6.1 Independent Functions

We are considering an evaluation function independent if it can evaluate a behavior being executed by a group of agents in an environment without needing any other information. This kind of evaluation can be specifically useful because a behavior can be developed, tested and evaluated independently of any other behaviors.

It is important to note that an evaluation function should not necessarily be made for a single behavior but for a class of behaviors. In this way, multiple behaviors with similar objectives can be evaluated with the same function. In some cases, using the same evaluation function can effect the resulting performance value of a behavior if a behavior is not well suited for a given function. We present details and experimental results for the independent evaluation function of covering an environment.

Covering. This evaluation function is used to measure how well the environment is covered. The function is initialized with a specified number of "coverage" points generated uniformly at random in the free-space of the environment. As a group of agents are executing a behavior in the environment, the covering evaluation function tests whether any of the "coverage" point is visible to any of the agents. If a "coverage" point is visible to one of the agents the point is marked as covered and the performance value is updated. The performance value generated is based on the percentage of the "coverage" points that have been covered.

A variant to this coverage evaluation function allows for the coverage points to become uncovered after a period of time. The time until the points become uncovered is based off of the last time that the point has been covered and not the first time it was covered. This will prevent those points from becoming uncovered unncecessarily if they have been recently covered.

6.2 Behavior-Behavior Comparison

We consider an evaluation function a behavior-behavior comparison if it requires another group of agents to be present. While it is not necessary for the opposing group of agents to be performing a task that directly opposes the agents under evaluation, it is important that the task be similar enough that the general purpose is filled (i.e a hiding behavior should not be evaluated against another hiding behavior).

It is important to note that when one of the behaviors is not well-suited for the behavior that it is being tested against, the performance values of the opposing group of agents will will not be indicative of actual performance. For this reason our experimental results are based on comparisons between the hiding behaviors and the behaviors that performed well in coverage over time.

Agent Search. This evaluation measures the number of opposing agents that the searching agents have encountered. The performance value is based on the percentage of the total number of other agents that have been detected. As such, the evaluation function has access to the total number of agents in the environment, even though the agents being evaluated will not have this additional information.

Similar to the covering evaluation function, it is possible for agents to return to a state of being unencountered. This will only occur if the agent is undetected for a designated period of time from the last time that it was detected.

Agent Hiding. This evaluation measures the number of times the agents have been discovered. A discovery is considered to be a change from a state where the opposing agent is not able to detect the hiding agent to a state where such a detection occurs. Thus, if an opposing agent keeps track of a hiding agent for an extended period of time, it is considered a single discovery.

The performance is reported in three parts. The first is the average number of times that the agents in the group have been discovered. The second is the minimum number of times that a single agent was discovered. The third is the maximum number of times that a single agent was discovered.

Time Hidden. This evaluation meausres the time that an agent has been undetected by all opposing agents. The performance value is based on the percentage of the total time of the simulation, with an initial period of time discarded to compensate for starting positions where the agent is already discovered.

7 Experimental Setup and Results

In our experiments, we use two different environments to test the performance of the behaviors. The first set of experiments tests searching behaviors (Covering: Basic, Random, Rendezvous, Scan and Territorial) with the covering evaluation function. These experiments will show the effectiveness of individual behaviors.

For the second set of experiments, the hiding and zone avoidance behaviors are evaluated using the agent hiding and time hidden evaluation functions. The covering, scanning, and random behaviors use the covering and agent search evaluation functions. The behaviors are categorized as either searching or evading. Each trial

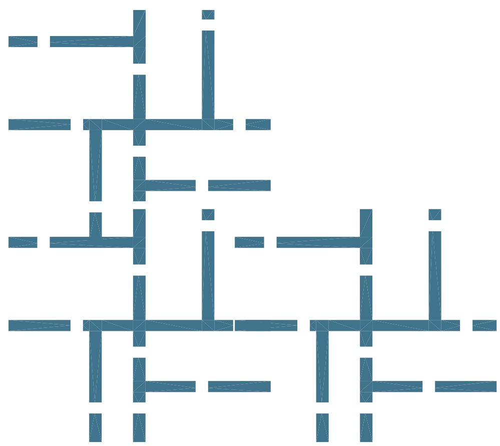


Figure 2: Environment 1 with a two-dimensional rendering of the obstacles. Passages connect various chambers throughout the environment.

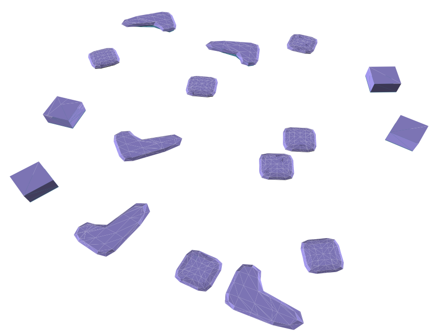


Figure 3: Environment 2 consisting of obstacles randomly placed throughout the environment.

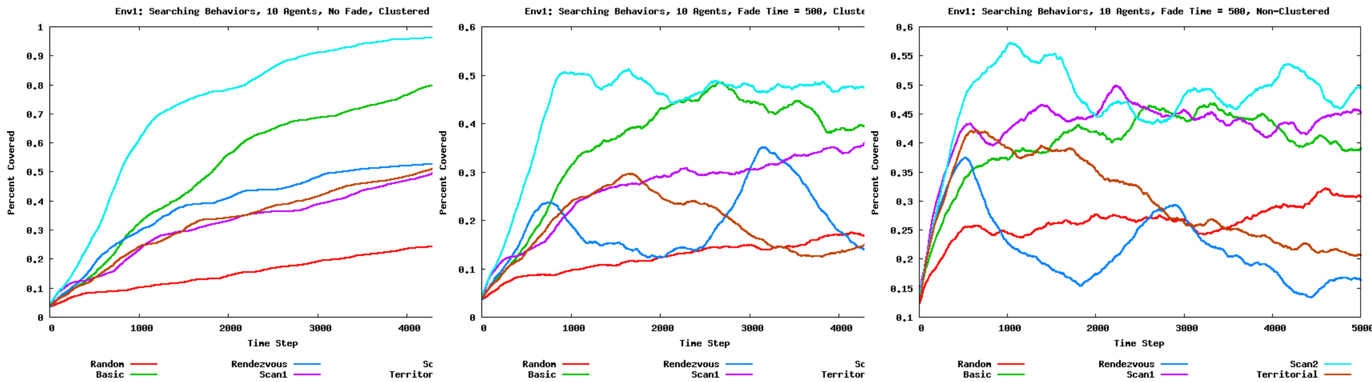
runs an evading behavior against a searching behavior. In both environments, each evasion behavior was tested against each of the evasion behaviors.

7.1 Searching Behavioral Comparisons

The covering evaluation function, described earlier, is used to test the searching behaviors. The searching behaviors tested are Covering: Random, Basic, Rendezvous, two variants of Scanning and Territorial. The random behavior is the most basic of covering behaviors in that the agents will move in a random direction for a given number of time steps, after which they will change direction. The two variants of the Scanning behavior tested vary the range in which scanning points are generated with Scan1 generating points within two times the agent's view radius and Scan2 generating points throught the entire environment.

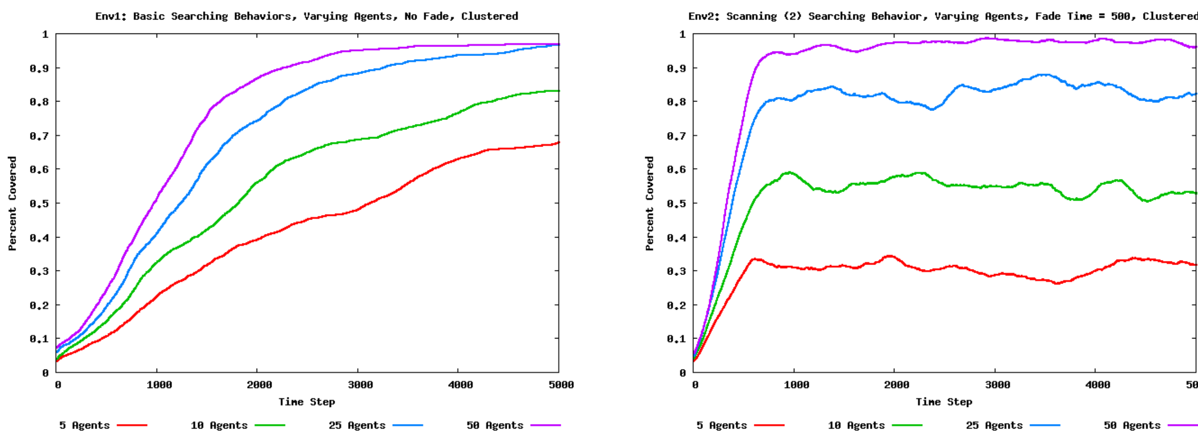
Simulation results are shown for various covering behaviors in Figure 4 run in Environment 1, shown in 2. In Figure 4(a) the agents start clustered at an initial location and the evalution function will consider a point covered if it has ever been visible to an agent (No Fade). It can be seen in Figure 4(a) that the Scan2 behavior performs the best, coming the closest to completely covering the environment. The next best performing behavior is the Basic behavior which steadily approaches full coverage of the environment. The other behaviors perform about the same with the exception of the Random behavior which does not perform very well.

Figure 4(b) shows the results of 10 agents starting with an initial clustered location in Environemnt 1 but for this evaluation function a "coverage" point is considered uncovered if 500 time steps have passed since it was



(a) (b) (c)

Figure 4: Performance of searching behaviors for 10 agents in environment 1 with differing parameters for evaluation function and behavior conditions.



(a) (b)

Figure 5: (a) Covering and (b) Scanning (2) behaviors compared when varying the number of agents in the environment.

last covered (Fade Time = 500). The Scan2 behavior performs the best followed by Basic and Scan2 behaviors which both have similar performance. Something interesting to note for the Rendezvous behavior is the periodic coverage performance. This can be attributed to the behaviors execution where all agents attempt to meet at a location in the environment and wait until all agents have rendezvoused. The territorial behavior has good initial coverage but slowly decreases it's coverage effectiveness when territories have been found and the agents settle into their territory.

In Figure 4(c) results are shown for 10 agents that start more dispersed in the environment (Non-Clustered) and also have a Fade Time of 500 time steps. It can be seen again that the Scan2 behavior performs the best followed by Basic and Scan1, with these behaviors averaging a coverage of close to 50 percent, in a large environment. The Rendezvous behavior again shows the periodic behavior indicative of the agents waiting for all agents to rendezvous. The territorial behavior again has good initial coverage followed by decreased coverage as the agents generate more set territories. The Random behavior actually performs better than Rendezvous and Territorial which can partially be attributed to the initial non-clustered starting locations.

In Figure 5(a) results are shown for the Basic covering behavior searching through environment 1. The coverage evaluation function does not return a coverage point to being uncovered (No Fade). It can be seen that as the number of agents increases the amount of the environment that is covered over time also increases. With 50 agents, using the Basic covering behavior, the agents are able to get close to full coverage much faster than with fewer agents although after 5000 time steps, 25 agents can approach full coverage.

In the next experiment, agents search through environment 2 using the scanning behavior with coverage points returning to uncovered after 500 time steps. The results, shown in Figure 5(b), vary the number of agents searching through the environment. In general, after a fast exploration period, the amount of coverage levels off. The coverage that is able to be obtained also increases with the number of agents searching through the environment. For a virtual reality environment or gaming system, this type of behavior and coverage analysis could be used to determine levels of difficulty, behaviors required and number of agents needed depending on the application or objective.

7.2 Searching Behavior – Hiding Behavior Comparisons

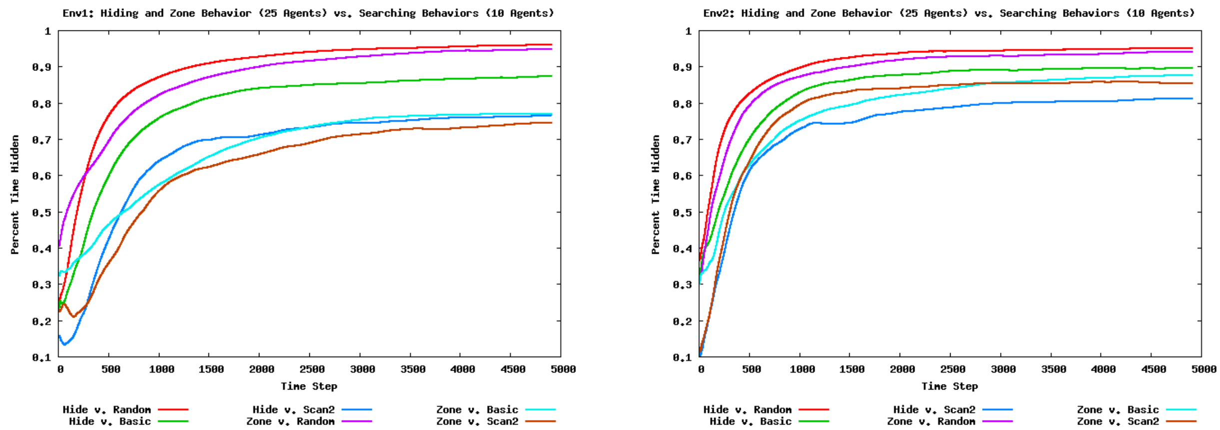
The time hidden evaluation function and agent hiding evaluation function are used to test the hiding behaviors against some of the searching behaviors. We do not discuss the agent search evaluation function, since the searching behaviors are designed for covering the environment rather than actively seeking agents. The hiding behaviors tested are the Hiding and Zone Avoid behaviors. The searching behaviors against which they are tested are Covering: Basic, Random, and Scanning. The random behavior was selected because it is the most naive approach to searching, and thus was intended to be easy for the hiding behaviors to avoid. The scanning behavior utilizes the Scan2 parameters, as this setup demonstrated the best performance in covering the environment over time. The basic covering behavior was included as an average performing behavior.

Simulation results for the hiding behaviors under the time hidden evaluation function are shown in Figure 6. In both cases, the hiding agents and the searching agents began clustered in the same area. This is the cause of the initial low percentage of time hidden, as the hiding agents are attempting to move away from the searching agents. The simulations included ten searching agents and twenty-five hiding agents, and the hiding agents have a slightly larger view radius than the searching agents.

Figure 6(a) shows the results in Environment 1, shown in Figure 2. The random searching permitted the hiding agents to remain undetected for the most time, with both zone avoidance and the hiding behaviors having similar performances. When run against the basic behavior, the two hiding behaviors performed somewhat worse, with Zone Avoid doing worse than Hiding. Against Scan2, both behaviors did worse still. The difference in the performance is less, but Hiding still performed better. The environment was disadvantageous for the Zone Avoid behavior, since the estimates of the locations of the searching agents do not take obstacles into account. This causes the Zone Avoid behavior to believe that it could become visible when it is in fact safe from detection due to an obstacle. It will seek a new hiding location, and will be more vulnerable to detection since neither behavior seeks to avoid detection while moving to a new location. A nearby hiding location may be behind a wall, so the agent would have to move a greater distance to get to this new location.

In Figure 6(b), the results are shown for the same experimental setup, but in Environment 2, illustrated in Figure 3. Again, the agents did best against random covering, followed by basic covering, followed by Scan2. Against Basic and Scan2, both Hiding and Zone Avoid performed better in this environment. Zone Avoid had a significant increase in the percentage of time hidden, and only had a small difference in performance against basic covering and Scan2. In this more open environment, the estimates of agent location were more accurate, and thus the Zone Avoid behavior was better able to avoid detection. Furthermore, the open environment allowed for more communication between the Zone Avoid agents, so other agents could avoid potential detections. Finally, fewer hiding locations require a long path to reach, so the time when the agents most vulnerable to detection is reduced, which accounts for the increase in performace of both types of agents.

Figure 7 shows the average number of detections incurred by the Hiding and Zone Avoid agents in Environment 2 against the Basic, Random, and Scan2 searching behaviors. In the beginning, there is a sharp increase in the number of detections, because the hiding agents are still in the process of moving away from the starting cluster of hiding and searching agents. For each of the searching behaviors, the rate of increase in the number of detections for the Hiding is similar to that of the Zone Avoid behavior. The covering results are reflected here, as the random covering has the lowest rate of increase in discoveries, while Scan2 provides the highest rate of increase. The random covering behavior does not do well at discovering the agents, and the two hiding behaviors have a similar result. When matched with the Scan2 behavior, the Zone Avoid behavior is able to escape the initial clustering sooner, resulting in a lower number of detections. Conversely, the Zone Avoid is unable to escape from the initial clustering quickly when matched with the Basic covering behavior, so it has a poor performance.



(a) (b)

Figure 6: Comparing Hiding and ZoneAvoid behaviors against ten covering behavior agents in (a) environment 1 and (b) environment 2.

It may seem surprising that the Hiding behavior performs better than the Zone Avoid behavior. A key factor to this is the relative sizes of the view radius when compared to the searching behaviors. With a larger view radius, both the Zone Avoid and the Hiding behaviors are able to detect an incoming searching agent, and take action to avoid detection. In this case, the zones lose the benefit of anticipating the location of searching agents, and only provide information for estimating which hiding locations are potentially unsafe. With a view radius that is equal to or less than that of the searching agents, the Zone Avoid behavior gains the advantage of being able to anticipate where a previously observed agent might be located, thus avoiding subsequent discoveries, while the Hiding behavior would only be able to react after being detected. Furthermore, the communication would allow other Zone Avoid behavior agents to avoid even an initial discovery from an agent that had been recently observed.

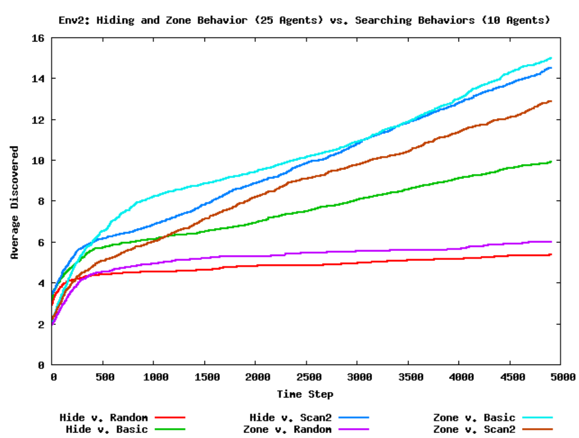


Figure 7:

8 Conclusion

We have shown that complex group behaviors can be generated using a roadmap providing an abstract representation of global environment information. The information the roadmap contains, such as topological information and adaptive edge weights, enables the group to achieve behaviors that cannot be modeled with local information alone. Moreover, there are many applications where agents can use roadmaps to effectively perform group behaviors. We have also described a framework in which behaviors can be developed, tested and evaluated for a group of agents.

References

[1] S. Nouyan, "Agent-based approach to dynamic task allocation," *ANTS 2002 (Lec. Notes in CS 2463)*, 2002.

[2] J. McLurkin and D. Yamins, "Dynamic task assignment in robot swarms," in *Proc. Robotics: Science and Systems*, June 2005.

[3] O. Shehory and S. Kraus, "Methods for task allocation via agent coalition formation," *Artificial Intelligence*, vol. 101, 1998.

[4] M. Mataric, G. Sukhatme, and E. Ostergaard, "Distributed multi-robot task allocation for emergency handling," *Autonomous Robots*, vol. 14, 2003.

[5] N. Pelechano, K. O'Brien, B. Silverman, and N. Badler, "Crowd simulation incorporating agent psychological models, roles and communication," *First International Workshop on Crowd Simulation*, 2005.

[6] M. Sung, M. Gleicher, and S. Chenney, "Scalable behaviors for crowd simulation," *Computer Graphics Forum*, vol. 23, pp. 519–528, September 2004.

[7] J.-M. Lien, O. B. Bayazit, R.-T. Sowell, S. Rodriguez, and N. M. Amato, "Shepherding behaviors," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, April 2004, pp. 4159–4164.

[8] J.-M. Lien, S. Rodriguez, J.-P. Malric, and N. M. Amato, "Shepherding behaviors with multiple shepherds," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, April 2005, pp. 3413–3418.

[9] A. Sud, E. Andersen, S. Curtis, L. Ming, and D. Manocha, "Real-time path planning for virtual agents in dynamic environments," in *Virtual Reality Conference*, March 2007.

[10] O. B. Bayazit, J.-M. Lien, and N. M. Amato, "Roadmap-based flocking for complex environments," in *Proc. Pacific Graphics*, Oct 2002, pp. 104–113.

[11] ——, "Better group behaviors in complex environments using global roadmaps," in *Artif. Life*, Dec 2002, pp. 362–370.

[12] ——, "Better flocking behaviors using rule-based roadmaps," in *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, Dec 2002, pp. 95–111.

[13] J. Funge, X. Tu, and D. Terzopoulos, "Cognitive modeling: Knowledge, reasoning and planning for interlligent characters," in *Computer Graphics*, 1999, pp. 29–38.

[14] C. W. Reynolds, "Flocks, herds, and schools: A distributed behaviroal model," in *Computer Graphics*, 1987, pp. 25–34.

[15] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Automat.*, vol. 12, no. 4, pp. 566–580, August 1996.