Mass Programmed Agents for Simulating Human Strategies in Large Scale Systems

Michal Chalamish
Department of Computer Science
Bar Ilan University
Ramat Gan 52900, Israel
merlich@cs.biu.ac.il

David Sarne
School of Engineering and Applied Sciences
Harvard University, Cambridge
MA 02138 USA
sarned@eecs.harvard.edu

Sarit Kraus
Department of Computer Science
Bar-Ilan University
Ramat-Gan 52900, Israel
sarit@cs.biu.ac.il

Categories and Subject Descriptors

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

General Terms

Experimentation

Keywords

Human strategies, Self-interested agents, simulation

1. INTRODUCTION

Simulation is an important tool for studying systems' behavior under specific conditions. In recent years, agent technology has been recognized as a promising new approach for developing simulation systems, in particular, Multi-Agent Systems (MAS) based simulations.

One of the great challenges of simulation systems is modeling the behavior of self interested individuals [3] since people can not be trusted to exhibit the expected behavior.

This paper focuses on large-scale self-interested agents-based simulation systems. The parking space search problem [2], in which people entering a parking lot search for a free parking space, is a good example. Different people have different concepts of what a good parking space is. One person's behavior is affected by other people's behaviors since one cannot park in an already occupied space.

Another example is the fastpass mechanism recently instituted in the Disneyland amusement park. It is used to reserve rides in several of the park's rides. Predicting the fastpass impact on actual line lengths is very difficult. A person's decision of using her fastpass privilege depends on many factors and the behavior of any given visitor affects the behaviors exhibited by the other visitors. Having a simulation tool that can reliably capture visitors behaviors with regard to the fastpass, can help plan the proper implementation of such a mechanism in the park.

AAMAS'07 May 14–18 2007, Honolulu, Hawai'i, USA.

The following sections introduce our unique methodology of using mass programmed-based agents for generating a significant number of simulated individuals reliably representing self-interested behaviors. This allows the system to scale-up and test the influence of changing different system parameters. The main steps we use are:

- Define the simulated environment and the main settings alternatives that need to be compared.
- Develop a simulator server and client skeleton agents, lacking a strategy implementation.
- Elicit people's goals and utility functions.
- Have people implement their strategies as a programmed component of a skeleton agent, aiming at maximizing their benefit considering their utility functions.
- Change simulation settings and repeat the former step.

Analyzing the results of the different settings provide a good estimation of what to expect when implementing the proposed settings in the real world.

To Test our methodology a simulation tool was implemented. The simulation was used to test the effectiveness of different information models in the parking space search domain. We claim, that the methodology actually simulates real-people behavior and validate this claim by analyzing answers to questionnaires.

2. PARKING PROBLEM DOMAIN

Parking lot design is an important problem addressed by many traffic and transportation researchers [1]. The parking problem domain consists of a specific parking lot design and of cars constantly going into and out of the parking lot. Every driver entering the lot has to find a parking space and might implement a different strategy for doing so. This problem has several interesting characteristics as far as simulation systems are concerned.

1. It is a daily task executed in different environments.
2. One's behavior is directly influenced by the behavior of the surrounding people.
3. There are several possibilities one must choose from at every given time until actually parking.
4. A person driving a car has only his limited vision to relay on when making decisions.
5. Each parking attempt is unique. The parking lot and the people attempting to park change, as well as the occupied and free parking spaces.

Today, many different information models can be provided to the drivers, to help them find a satisfactory parking place more easily. Nevertheless, the more sophisticated the information is, the greater is the cost of supplying it and the complexity of producing it. Therefore, a parking lot designer/manager has a strong incentive to acquire tools for capturing the value of different information models.

One of the earliest simulated parking systems is PARKSIM, specified in [4]. All of it's driving agents share the same parking strategy, based on a matrix of knowledge which includes the perceived availability of every parking space, the minimum travel time to it and the perceived number of vehicles between the driver's current position and the parking space. The matrix is updated throughout the parking space search. The driver determines which parking space is 'best' by considering those not likely be taken by another vehicle.

Cassady and Kobza [2] developed a probabilistic approach to parking space search strategies. They assumed parking space availability probabilities for all parking spaces in the lot, defined two parking strategies and evaluated them using three performance measures. No actual data was collected or simulation run to verify the results.

To the best of our knowledge, no previous research simulated dynamic changes affecting the behavior of the drivers or evaluated the role of information. We used our innovative methodology to address these important questions.

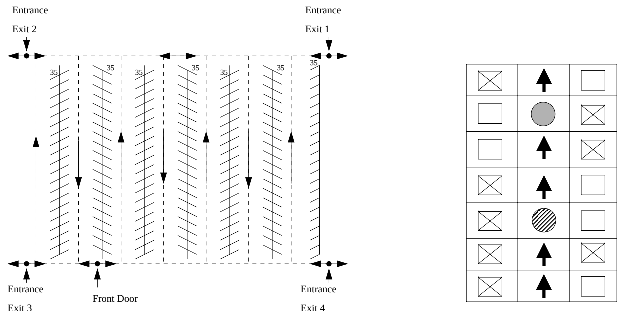
3. IMPLEMENTATION DESCRIPTION

Aiming at testing our proposed methodology, we created a simulation system for the parking search problem. The parking lot structure and the rate of cars arriving to the parking lot and exiting it at each time step are defined by the simulation manager (SM). Cars entering the parking lot are agents, operating on different clients which receive continuous information from the SM based on their locations in the lot and the information model defined. Each agent has access to a general map describing possible parking spaces and legal moving directions. This map, containing no real-time data, is based on coordination points of structure (x, y).

In every time step, an agent located at point (x, y) can either move to adjunctive point (x', y'), park at adjunctive point (x', y') or wait in place. The decision is reported to the SM and executed only if legal. The agent is also provided with a small matrix representing the portion of the parking lot (a "viewing frame") in its vicinity, containing real-time data, before deciding on the action to take.

New agents are instantiated based on the entry rates defined and their type is randomly selected from a pool of types, each associated with a different agent implementation. The agent searches for a suitable parking space and terminates its life cycle when it is found. The space it occupies will be vacant based on the exiting rates defined.

Each agent type has a different parking strategy set by a person, according to her perception of what a "good" parking space is. Our strategy developers (SDs) each defined her own cost function, using the same measures. Next, each SD had to implement her agent's strategy aiming at minimizing her cost. A skeleton agent handling all necessary communication with the SM and supplying an API for programming the parking strategy was provided to the SDs. The SDs were asked to update the agent's strategy layer for each of the tested information models.



(a) Parking lot structure

(b) Viewing frame example

Figure 1: Parking lot structure used in experiments and viewing frame example

4. EXPERIMENTAL DESIGN

We used the structure of an underground parking lot, similar to that used in [2]. It is sketched in figure 1(a) and has 4 Entrance-Exit points denoted as Entrance-Exit 1,2,3 and 4, as well as six rows of parking spaces, each with 70 aligned parking spaces, and one row aligned with 35. That is, a total of 455 available parking spaces. Traffic lanes are denoted by dashed lines and the arrows indicate the allowed traffic direction(s). The front door serves as a sole foot entrance/exit point. Foot traffic may proceed in either direction but both vehicle and foot traffic are restricted to the traffic lanes.

Figure 1(b) illustrates a viewing frame. The circle located at the lower part is the agent. The frame provides a view of 5 points forward, 2 backwards and 3 in width. In figure 1(b) we can identify: (a) another agent that entered the agent's viewing scope (filled circle); (b) 8 occupied parking spaces (X-ed squares); (c) 6 free parking spaces (non-X-ed squares); and (d) the traffic directions (arrows).

These are the measures used to evaluate the strategies:

- Walking distance from the parking space to the door.
- Parking space search time.
- Driving distance from the parking space to the exit.

Each SD had to choose a set of weights to create a single linear cost function. Then, the SDs had to implement their agent's strategy according to their chosen function. These functions were used to evaluate the system's performance but can also be used for analysis other performance measures. For example, a parking lot manager might want to know how much time drivers search for a parking space.

Following are the information models tested.

1. Nothing provided apart from the general map and viewing frame.
2. The number of free parking spaces provided on entry.
3. An updated map of the free parking spaces provided on entry.
4. An updated map of the free parking spaces provided at every time step.
5. An updated map of the free parking spaces and the locations of other agents provided at every time step.

34 students, all senior computer science undergraduates, acted as SDs, and were graded according to their agent's performances as evaluated by their own cost functions.

They were provided with a limited version of the simulation tool, always instantiating the same agent, for debugging and strategy testing. After the submission of all the agents of a single task, the full simulation tool was used by us in order to detect abnormal behaviors. SDs whose agent was found to behave abnormally (i.e., stand in place for a long time when there is an option of moving) were asked whether the abnormality was part of their strategy or a bug. In case of a bug, the agent was fixed and re-tested.

5. EXPERIMENTS AND RESULTS

Three initial capacities and three Entry/Exit rates were defined, the combination of which results in 9 simulation parameters settings. For each such setting, the simulator can produce a full randomized scenario. Further strengthening our results, three different scenarios were created for each parameters setting, resulting in 27 scenarios, used to test all information models. Simulations lasted 3000 time steps.

5.1 Improvements by Information Models

At the end of each simulation, we averaged the total cost of the different agents. For every information model and every scenario the total cost of each of the 34 agent types were averaged over all the agents of the same type.

We denote $info_i$ as the simulation result when using the i-th information model. The averaged costs are: $info_1$=35.51, $info_2$=34.72, $info_3$=33.71, $info_4$=32.16 and $info_5$=31.77. A lower bound for the overall cost calculated for this environment is: *bound*=20.93. This bound allows allocation of a parking space to multiple cars in parallel if the aggregated parking times for this space do not exceed the simulation duration. Using this assumption, the optimal allocation of cars to parking spaces, based only on the agents' cost functions, was produced. While this lower bound is significantly smaller[1] than the theoretical minimum cost, it still serves well to demonstrate the improvement obtained when using the different information models.

The results show that on average, $info_2$ lowers the total cost by 5.36%, $info_3$ by 12.30%, $info_4$, by 22.95 and $info_5$ by 25.63% relative to $info_1$. That is, each information model further lowers the total cost. A paired, single tailed t-test found all improvements to be significant ($p < 0.029$). The greatest improvement relative to $info_1$ is $info_5$, which is also the one to demonstrate the greatest improvement relative to former information model.

Drawing conclusions from these results, we can say that the different information models have great influence on the drivers' costs. They can help drivers find a better suited parking space, according to their own preferences.

5.2 Central Management Implication

Sometimes, the simulation designer is interested in different performance measures. For example, a parking lot manager may be solely focused on the overall time it takes the agents to look for the parking space. In this case $Cost_{mang} = \sum_{ag_i} WalkDist(ag_i) + SearchTime(ag_i)$ (assuming that it takes the same time to walk and to drive similar distances).

The results show that $Cost_{mang}$ improves considerably as the information given is more detailed. The results also show that the relative improvement from the parking lot designer perspective is significantly greater in comparison to the improvement measures in terms of social-welfare.

[1]Both because it uses more permissive assumption and because it uses a cooperative approach in which a central allocator is allocating cars to parking spaces.

6. VALIDITY OF RESULTS

Our methodology assumes people will use their own behavior when deciding on their agent's strategy. Following the completion of the entire task each SD was asked whether her strategy was specific to the given parking lot structure, whether she uses that strategy in real life, and does she consider it to be usable by a person.

Analyzing the answers revealed that 74% of the SDs considered their strategies to be parking lot structure independent. 82% reported on using approximately the same strategy themselves in the physical world. 88% considered their strategy to be usable by people, with the exception of non-accurate calculations. Apparently, even though asked to implement the best agent, almost all SDs chose to implement their own real-life strategy. We can also safely predict that similar results would have been reached if we would have used a different parking lot structure.

7. DISCUSSION AND FUTURE WORK

The encouraging feedback received from the SDs, indicates that the system successfully managed to capture real-life parking-related human behaviors. Settings in which agents programmed by people were proposed in recent years, however these have not been scaled to large simulations that can truly mimic a real-life domain. The validness of our parking search problem simulation derives from the incorporation of 34 different types of agents, each programmed by a different person trying to minimize a personal cost function. Furthermore, we can add as many additional types as needed to the simulation. This process can be performed incrementally, and requires no large group effort.

While our goal in this research was mainly to demonstrate the applicableness of our proposed methodology, the significant specific results obtained, are an important contribution to the parking search and parking lot design domains. The fact that most participants reported that their agent's strategy was not parking lot structure dependent, suggests using the system for designing new parking lots and test them with the information models that we used for our agents.

Acknowledgement: This research was supported in part by ISF grant #1211/04.

8. REFERENCES

[1] M. Brown, R. Evens, and I. Black. Evaluation of traffic managment and parking schemes: Whither now? *Traf. Eng. & Ctrl*, 32:110–119, 1991.

[2] C. R. Cassady and J. E. Kobza. A probablistic approach to evaluate strategies for selecting a parking space. *Transp. Science*, 32(1):30–42, February 1998.

[3] Y. Murakami, Y. Sugimoto, and T. Ishida. Modeling human behavior for virtual training systems. In *AAAI*, 2005.

[4] W. Young. Parksim/1 (1): a simulation model of driver behaviour in parking lots. *Traf. Eng. & Ctrl*, 1986.