

Agent Based Simulation, Negotiation, and Strategy Optimization of Monopoly

Nicholas Loffredo

1/23/08

Abstract

Computers have a difficult time performing common human tasks, such as learning a language well enough to be able to "talk" intelligently with someone/something. Monopoly, one of the most well known and understood board games in the United States, if not the world, provides a good environment to see whether or not a computer can "learn" to negotiate through a number of strategies. It is much simpler than negotiating in the real world, due to the simplified environment, yet complex enough that it may be useful as an example of computer negotiation. By creating a Monopoly simulation with computer agents playing the game, it can be used as a test bed for these computer negotiations. The methods used in this test bed, if it works, could then be applied to more complex computer negotiation. The agents can be given aggressiveness values for different negotiation techniques, such as price "stubbornness" when selling or buying properties from other agents. The results from running the simulation hundreds of times can then be graphed to show which strategies are the optimal strategies for agents.

1 Introduction

Computers currently are unable to perform common human tasks such as understanding a language well enough to speak it and effectively communicate. A good example of this is negotiation. Most humans are able to negotiate with one another for various goods. Computers, on the other hand, cant.

If computer were able to negotiate effectively, they could be used in many situations that currently require people such as in diplomacy, selling/buying goods, trading goods, or just negotiating with other people in general. More importantly, it would allow people to instruct a robot/computer to negotiate using certain items and to meet certain goals, instead of hiring people to do it. These computers would be resistant to common human flaws, such as anger or impatience.

Making a computer than can negotiate effectively in a limited environment is a first step towards being able to negotiate in a more complex one. The game of Monopoly is simple enough that negotiation should be able to be implemented within a year, yet complex enough that the method used to achieve the results may be able to be applied towards real negotiation. By making a working simulation of Monopoly, a negotiation capability can be implemented for computer agents that will play the game.

Fundamentally, the system must simulate all the rules of Monopoly. Agents must be able to move around the board based on the dice roll, be able to buy titles they land on, and buy houses on monopolies they own. Additionally, they should be able to sell houses and mortgage properties. When an agent lands on a Chance or Community Chest square, they should receive the top card from a deck which was randomly sorted before the game, and do whatever the card says. Furthermore, to explore the research areas contained herein, agents should also be able to negotiate with players. In particular, these negotiations will be based on aggressiveness levels. For example, how far an agent is willing to drop/raise his initial price in order to complete a negotiation.

2 Background

Surprisingly, not much research has been done into making agents for Monopoly that can learn the optimal strategy for negotiation. One of the few existing simulations is one that determines the probability of landing on each square in Monopoly. This can be used to see if my Monopoly simulation results correlate to theirs, and determine whether or not the simulation works correctly. Fortunately, there has been research done into reinforcement learning, which is effectively how the agents will learn. Reinforcement learning is when agents take a number of actions over a course of a game, and then are basically told that they did well (when they won) or they did poorly (when they

lost). Based on this feedback, each agent will try to change its aggressiveness values (which may involve different strategies) to find the winning values. Also, agents may learn from the strategies other agents used, and whether they won or lost, to determine how their aggressiveness levels should change, which varies from the traditional approach slightly. There is not a state-of-the-art reinforcement learning program yet, however. Everything I Need to Know About Business I Learned from Monopoly, which discusses various strategies for Monopoly, can be used to see if agents develop the strategies the book discusses.

According to "On Verifying Game Designs and Playing Strategies using Reinforcement Learning", neural nets could also be used for my reinforcement learning strategy. However, as there is not enough time for a neural net to be implemented, reinforcement learning with aggressive sets will be used. Also, my method of using the aggressiveness levels is general enough to have the potential to be used for many other applications.

3 Procedures

3.1 Preliminary Testing and Results

The simulation was tested by adding features that allow the program to play multiple games in one run, display the number of wins of each agent, and display the aggressiveness levels of each agent. Both agents initially had a 50 percent chance to take actions. One agent, however, was allowed to learn, whereas the other would not change. Doing this would show whether or not the agent was actually evolving to beat the other agent - i.e., if his win ratio over a large number of games was greater than 50 percent. After running the simulation about 100 times, the learning agent had roughly a 60 percent chance of winning. This shows that the agent was learning. The likely reason that the chance of winning was not higher is that the agent was not able to trade properties - therefore, whether or not the agent wins is highly influenced by whether or not he lands on enough properties first to be able to buy a monopoly.

After the first implementation of trading, my learning agent has a 58 percent chance of winning after about 500 games. I think the reason it is not as high as expected is because my trading logic is flawed, so the agents trade every time they can until they are out of cash.

3.2 Software

Java is the programming language being used.

4 Schedule

In the first quarter of the research project, the focus will be on designing the Monopoly environment with all the correct rules.

The second quarter will focus on the researching, planning for, and initial designing of the negotiation capability of the agents. (e.g., based on an array of aggressiveness values for different items).

In the third quarter, the design will be completed, as well as the implementation and testing. At this point, experiments will be designed and executed, and the results will be analyzed. Conclusions will be drawn, and recommendations will be made. Graphs will be used to display the results of how often agents win with various strategies.

5 Development

In the first quarter of the project, I developed the monopoly simulation program, finishing all features of monopoly (agents 'roll' dice, buy properties, etc.) except for auctions, trading properties, buying and selling houses, and mortgaging properties. I also created an interface to display the monopoly board.

In the second quarter of the project, I finished implementing buying and selling houses, and mortgaging/unmortgaging properties. After that I created and implemented aggressiveness levels which are used to determine how often an agent will take certain actions for each property group - i.e. buying properties, buying/selling houses, or mortgaging/unmortgaging properties. When that was finished, the learning algorithm for the agents was implemented so that agents became able to learn over time which values are optimal for different actions on each property. A display was then created to show the aggressiveness levels and wins of each agent.

In the third quarter I added trading to my game.

6 Bibliography

Axelrod, Alan. Everything I Know About Business I Learned from MONOPOLY. Pennsylvania: Running Press Book Publishers, 2002.

Ghory, Imrin. Reinforcement learning in board games. Publications. 4 May 2004. University of Bristol, Department of Computer Science. <http://www.cs.bris.ac.uk/Pu>

Johansson, Stefan J. "On using Multi-agent Systems in Playing Board Games." 2006. portal.acm.org

Kalles, Dimitrios. "On Verifying Game Designs and Playing Strategies using Reinforcement Learning." 2001. portal.acm.org

Lazaric, Alessandro. "Reinforcement Learning in Extensive Form Games with Incomplete Information: the Bargaining Case Study." 2007. portal.acm.org

Lin, Jon. Jons Monopoly Simulation. 1 Nov. 2007 <http://www.tesuji.org/monopoly.html>

Matsuno, Yoichiro. "A Multi-Agent Reinforcement Learning Method for a Partially-Observable Competitive Game." May 28, 2001. portal.acm.org