Artificial Intelligence for the Board Game Pandemic

Eugene Kramskoi, Mira Jacobs, Sean McCulloch

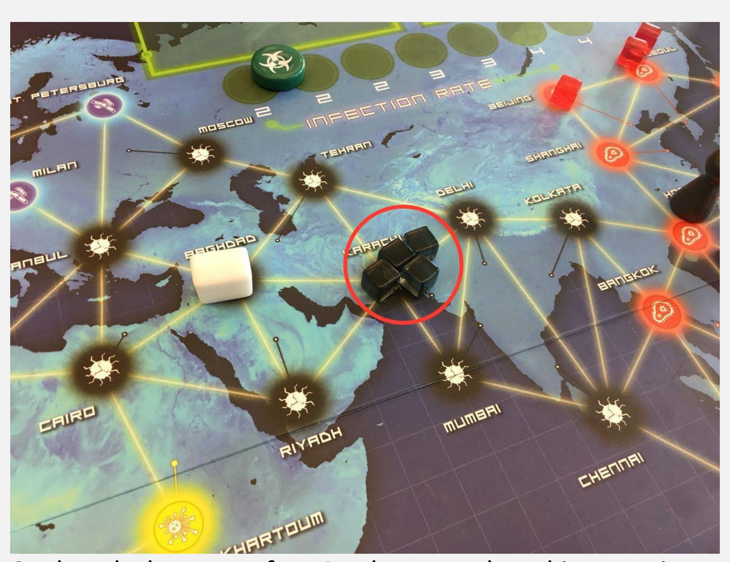
The Research

There has been a lot of work done on automatic computer players for games such as chess or checkers, but European-style games such as Pandemic have been largely ignored. These types of games generally emphasize player interactions and strategy and tend to involve a lot of information that is hidden from the players.

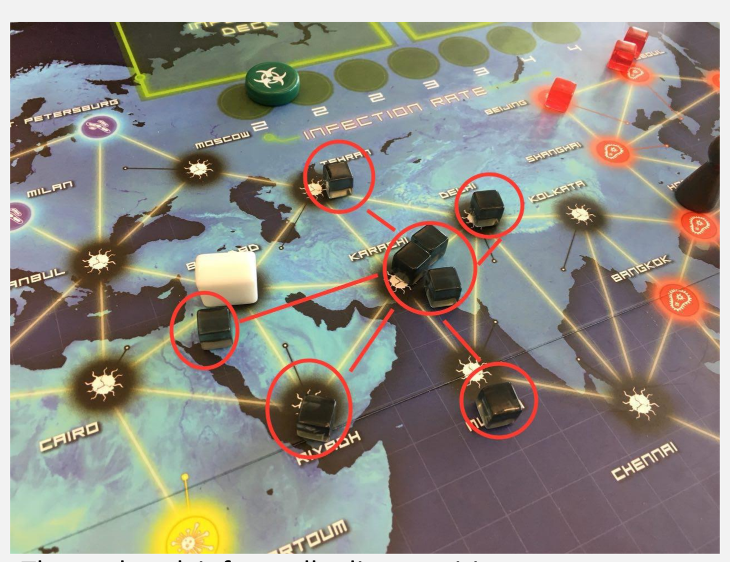
The goal of this research is to create a program that will intelligently play the board game Pandemic, and in the process learn about the algorithms and other challenges that an automatic player program for this type of game introduces.

The Game

Pandemic is a co-operative board game where two to four players are given the task of curing all four of the diseases that plague the forty-eight interconnected cities on the board's map. The players must work together to travel from city to city and prevent the diseases from spreading while simultaneously researching a cure.



Outbreaks happen after 4 cubes are placed in one city



The outbreak infects all adjacent cities

Epidemic Cards

- There are four epidemic cards and they add three cubes to any city that is drawn from the deck
- The deck reshuffles on top of the deck
- Cities in the discard pile have a higher chance of being drawn because of this
- The infection rate is increased (determines how many cards are pulled rom the deck)

Actions

Players can take four actions a turn. Actions can be viewed in two main categories:

- Actions that help players win such as:
  - Trading cards
  - curing diseases
- Actions that delay outbreaks such as:
  - Treating diseases
  - moving towards congested areas
- A player has to have five cards of one color to cure a disease
- During the game players can set up research stations at cities to make it easier to move around

Abstract Moves into Plans

A player has around 10,000 possible moves per turn. For a computer to be able to calculate all of these moves in reasonable amount of time, we combine several moves together into abstract <u>plans</u>.

Instead of having many simple moves such as

- "Drive from Madrid to Algiers" or
- "Give the Algiers card to Player 2"

the program instead thinks of larger, more abstract <u>plans</u>, such as

- "Get to Algiers and treat the disease there" or
- "Trade with all the players to get five red cards"

Evaluating Plans

Picking the Best Plan

The program needs to be able to pick the most effective plan from the list. Our program looks at each move in terms of <u>time to win</u> and <u>time to lose</u>.

To pick the best plan, it tries to find the one that:

- Minimize time to win
- Keep time to lose large

Time to Lose

Time to lose is an approximation of how long it would take for the players to lose the game if they didn't do anything. To calculate time to lose, we use simulation:

1. Create a copy of the current game state
2. Simulate new turns over and over until a game over
3. Count how many turns that took and save that to a list
4. Repeat this several hundred times and average out the result

It's also important to mention that when copying the game state, we reshuffle the decks so that the program can't 'cheat'.

Time to Win

Time to win is an approximation of how long it would take to win the game as fast as possible. To calculate time to win, we use probability. To win the game, all four diseases need to be cured. To do that, a player needs to:

1. Collect 5 cards of the same color - usually by meeting and exchanging cards with other players
2. Get to a research station
3. Spend the five cards to discover a cure

Each of these steps can be approximated by looking at the probability of drawing certain cards, the average travel distances, and other information.

Summary

To summarize, the automatic player program works by:

1. Generating all possible abstract plans
2. Rating each one in terms of time to win and time to lose
3. Use these metrics to chose the most effective plan from the list
4. Execute that plan
5. Repeat until a win or a game over

Current State

- We have created a program that enforces the rules of the game and allows a combination of both human and computer players to play.
- We made a tool that calculates the probability of drawing any specific card
- We made a tool that calculates the shortest path from A to B that takes into account travel costs
- The program can calculate time to win and time to lose
- The program will automatically figure out which cities are most effective to treat and do it
- The program will automatically attempt to cure diseases

Future Work

There is a lot more work to be done.

- The program currently only generates plans for treating cubes and curing diseases. More possible plans need to be implemented.
- The time to win can be more accurate
- Some features of the game, such as roles and event cards, are currently ignored by the program