AI as Evaluator: Search Driven Playtesting in Game Design

Fernando Silva, Scott Lee and Nicholas Ng
New York University,
New York, US
{fernandomsilva,sl3998,nn733}@nyu.edu
N14662208, N14931088, N15058306

***Abstract*— This paper presents a demonstration of how AI can be useful in the game design and development process. By using an artificial intelligence algorithm to play a substantial amount of matches of the Ticket to Ride board game and collecting data we can analyze several features of the gameplay as well as of its board. Preliminary results revealed loopholes in the game's rules and pointed towards trends in how the game is played. The results lead to the conclusion that large scale simulation utilizing artificial intelligence can offer valuable information regarding games and their designs that would ordinarily be prohibitively expensive or time-consuming to discover manually.**

I. INTRODUCTION

Testing and experimentation are essential aspects of game design. Designers can only test for scenarios that they can think of, and design flaws are inevitable in any system of considerable complexity. Ideas and mechanics that work in theory or on paper can fall apart once the playerbase finds a weakness it can exploit. Such scenarios are often so obscure and ultimately impractical that they are harmless to the state of the game, but they can occasionally give way to optimal strategies or undesirable gameplay. It is often in a designer's best interest to minimize the number and severity of these weaknesses or loopholes in the game's logic.

A common metric in evaluating multiplayer games is competitive balance. Competitive games that reward highly skilled play result in players feeling more fulfilled, while simultaneously motivating players to improve further [1]. Additionally, a lack of balance is problematic. Games with widely known optimal strategies tend to have its playerbase converging on a single playstyle or emphasizing a single game mechanic, condemning divergent playstyles as taboo or extraneous mechanics as worthless.

There is an established relationship between balance and player enjoyment. Kraaijenbrink et al. found in a study that when players are told that a game is balanced, they are more likely to feel a stronger sense of effort and challenge. Additionally, the study found that players felt more successful after playing a game they believed was balanced, regardless of whether they won or lost. [1] Therefore, there is ample reason for game designers and developers to maximize the degree of balance in their projects.

It is important to note that as the complexity of a game increases, the task of maintaining competitive balance and game integrity quickly becomes non-trivial. Each variable in a game needs to be inspected for its impact on the system as a whole. Ensuring that a game is balanced, or its logic is sound, therefore requires a great deal of mathematics and testing, and even then, individually accounting for each and every possible scenario is a herculean effort. As such, any method to expedite the testing process and verify results is useful in the field of game design.

This paper is interested primarily in exploring the potential uses of several articifial intelligence algorithms as well as artificial intelligence as a whole in testing for game integrity and balance. Because AI can be used to perform a large number of simulations in a relatively fast and cost-effective way, it stands to reason that AI could be useful in identifying potential loopholes or imbalances in a game. To that end, several experiments were performed in which AI played the Ticket to Ride board game in an attempt to identify imperfections and unaccounted scenarios which could arise during gameplay.

II. RELATED WORK

There exists a body of work relating to the assessment of balance in games. In previous work, a set of balance metrics was utilized to some success. This set attempted to find the importance of various factors: the ability to react to the current game state, the importance of long-term strategy, and the fairness of starting conditions in the context of an educational card game [2]. In this instance, the set of balance metrics worked well at finding which cards were used how often, how accessible the game was to new players, and the efficacy of several specialized mechanics and playstyles. However, this method of assessment via a set of metrics is limited, as this form of play analysis was only put to use on games with perfect information, and the metrics were constrained by the preexisting notions of optimality in the game.

Much like restrictive play, the Information Theory based concept of Relevant Information has also been appplied to the evaluation of game mechanics [3]. This theory of Relevant Information is interested in discovering the minimum amount of information a game agent needs to realize one of its optimum strategies. In Salge and Mahlmann's paper, Relevant Information is represented as a numerical value, which can be obtained via gathering statistical information from an adaptive, genetic algorithm driven neural network. This data is then used in different game scenarios to detect game play flaws, and change mechanics appropriately. Relevant

Information is found to be a useful method of evaluating the quality of game rules, especially for strategically inclined games, and as the authors put it, is a strong first step towards automating evaluation of game mechanics in the pursuit of game balance.

There has also been reasearch in algorithmically optimizing balance in board games. Genetic algorithms have been used to great effect in the analysis and general game playing of games of various board sizes, piece types, and win conditions based upon the base games of Tic-Tac-Toe, Reversi, and Checkers [4]. As a result, these genetic algorithms optimally find games that are intrinsically more interesting. Most important to note is that having large scale AI play games is an effective way of figuring out niche cases in games, that humans would have a difficult time thinking of. In this case, genetic algorithms were especially effective at finding divergent or aberrant methods of winning games. However, the elements of the game that the genetic algorithm in this case was able to change were rather limited, as they only included the board size, piece types, and win conditions based upon three simple games. Other research has been conducted with AI that has the ability to alter more interesting and complicated elements of the game, as shown in Krucher's paper on algorithmically balancing a collectible card game (CCG) [5]. Krucher balanced his CCG implementation via rating and modifying cards every 100 iterations of his AI playing the game. These ratings in turn effected consequent gameplay, as the AI accounted for the ratings of cards when deciding what action to perform. However, these experiments resulted in cards that were not properly balanced, most likely due to the limited nature of the AI and a randomized card generator and modifier. For example, the AI did not take into account the possibility of enemy moves, or implement card counting to determine what cards the enemy might have in its hand.

AI has also been used before to evaluate balance in the card game Dominion [6]. In this paper, Mahlmann, Togelius, and Yannakakis utilized three different AI agents each utilizing different fitness functions and skill levels to determine a balanced card set. Interestingly, artificial neural networks (ANN) were used for evaluating the state of nodes in the MCTS, as well as the general state of the game board. From this work, it was found out that certain cards were present in the winning sets of all three AI agents, and thus it was concluded that these cards made the game more balanced independently of gameplay style. Furthermore, it can be seen that this method of game balance evaluation has credence in designing and balancing other games.

III. TICKET TO RIDE

Ticket to Ride is a 2 to 5 player competitive board game designed by Alan R. Moon, published by Days of Wonder in 2004. The game has won multiple awards and sold over 3 million copies by 2014 [7]. Multiple versions of the game have been released since. When we talk about the rules of the game, we will be referring to the original rule set, for the first version of the game, as well as its board.

In Ticket to Ride, players are trying to score the most amount of points. The game is composed of 4 elements: Train Cards, Destination Cards, Train Tokens and the Board. To score points, players will collect Train Cards that allow them to use their Train Tokens to connect different cities on the Board.

The components are comprised of a deck of 110 Train Cards: 12 for each of the basic colors, Red, Blue, Green, Yellow, Pink, Green, White and Black, and 14 locomotive, or wild, cards. The wild cards are unique in that they can take the place of any other color of Train Card. A deck of 30 unique Destination Cards, 45 Train Tokens for each player, and 1 Board are also shared by all players. Each player has their own hands which are hidden from other players, where they keep the Train Cards and Destination Cards they collect, and their own Train Token pool.

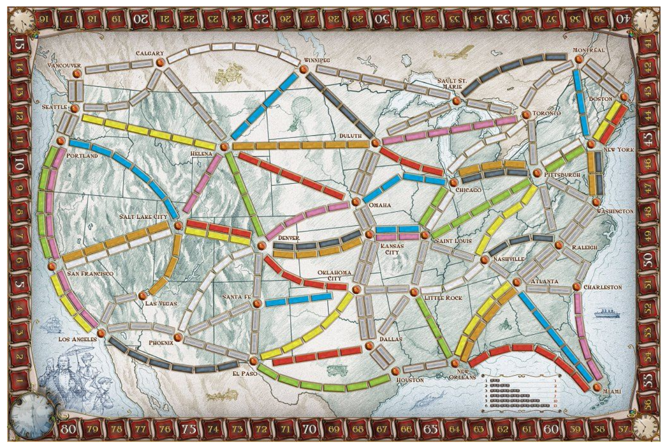

Fig. 1. The USA map which portrays the board used in the original game. Each city is represented by a circle. Each route is a sequence of rectangles. The number of rectangles is equal to the route size and the color of the rectangles portray the route's color.

The board is represented as a graph, where each node is a city and each edge is called a route. Two cities connected by a route are called adjacent cities. There can be either 1 or 2 routes connecting the same 2 cities. Each route has two attributes: color and size. The route's color can be any of 9 different possible, either one of the 8 basic colors of the game or a special color: Grey. The basic colors are represented by the different Train Cards, so to claim a route of a basic color, Train Cards of said color are needed. Meanwhile, a Grey colored route can be claimed with any color of Train Card, as long as all the cards used are of the same color. The size of route determines how many of a specific Train Card a player must have to claim that route. Figure 1 shows the board used to simulate games in this paper.

Claiming a route between two cities will score the player points, although the main form of scoring comes from connecting two specific non-adjacent cities required by a Destination Card by chaining multiple routes in a row. The Destination Cards each player has are kept hidden for the others until the game has ended, and when this state is reached they are revealed and their points are tallied. A

Destination Card then has 2 attributes: the name of the two cities that the player has to connect, and how many points it is worth if the player succeeds in achieving that goal. There is one caveat, however: if the players are not able to connect the two cities in their Destination Cards, not only are they not rewarded with that amount of points, they lose points equal to the number of points printed in the card. For two non-adjacent cities to be considered connected by a player, it must be possible to start in either one of the cities and arrive at the other by traveling only through routes claimed by that player.

One player is selected at random to be the first player. From them on, each player will then execute one of the 3 possible moves when their turn comes. Once they are done, the play proceeds to next player in clockwise fashion. The possible moves a player can take on their turn are: Draw Train Cards, Draw Destination Cards or Claim a route.

When drawing Train Cards, a player can choose to pick one of the cards currently face up or to draw the card at the top of the deck to add to his/her hand. If they choose a face up card, that card gets immediately replaced by revealing the next top card of the Train deck before proceeding. If at any time, 3 or more of the 5 face up Train Cards are wild cards, all face up Train Cards are discarded and 5 new ones replace them immediately. When making the move of drawing Train Cards, the player will take 2 cards on their turn, with wild cards being an exception. If the player chooses a face up wild card to draw as their first card, they will get only that card for the turn. If the player has already drawn 1 card, whether it was face up or not, they cannot take a face up wild card as their second card. If a card would have to be drawn, but the Train deck is depleted, the Train Card discard pile is shuffled and becomes the new Train deck.

For a player to be able to draw more Destination Cards, the Destination Card deck must not be depleted. Differently than with the Train Card deck, this deck never gets replenished during a match. When drawing Destination Cards the player gets the top 3 cards of the destination deck, or all if there are 3 or less cards left. The player then looks at the cards they drew and chooses to keep 1, 2 or all the cards drawn, which are added to their hand. The cards they did not decide to keep are then discarded.

To claim a route, a player must have Train Tokens equal to or higher than the size of the route they want to claim, and that route must be available, meaning it has not been claimed by anyone. To claim a route they must then discard from their hand a number of Train Cards equal to the size of the route whose color matches the color of the route they want to claim. They then move on to place their tokens on top of such route to mark it as being claimed by them. The player also immediately scores points proportional to the size of the route claimed: A route of size 1 rewards 1 point, size 2 rewards 2 points, size 3 rewards 4 points, size 4 rewards 7 points, size 5 rewards 10 points and size 6 rewards 15 points.

If after claiming a route a player is left with 2 or less Train Tokens in his pool, he announces the last round of the game. Then, every player, including them, gets to make one more move. After this player's next turn, the game ends, every player reveals their Destination Cards, and the points are tallied.

The points obtained, or lost, from the Destination Cards are added to the ones gotten from claiming routes, for every player. In addition to those, all players then count the number of their Train Tokens that make up for the longest continuous route, where longest means the one with the most Tokens, that connect any two cities in the board, adjacent or not. The player with the longest route overall gains an additional 10 points. After adding all points, the player that achieved the highest amount of points wins the match.

IV. PLAYER AGENTS

To evaluate different aspects of Ticket to Ride, we decided to implement multiple player agents that would approach game strategies differently. Out of the agents we have, we will be discussing the 2 that standout.

A. A* Agent

The A* agent implementation is straight forward. For a game state, the algorithm extracts all possible moves, e.g., one for each card it can draw, and one for each route it can claim. The moves are evaluated following a heuristic and are put in a priority queue, ordered by their heuristic evaluation. The algorithm proceeds to remove the highest priority element from the queue and simulate the changes in game state of executing that move. It then queries all possible moves it can make after simulating such and adds those to the queue. When it finds a moves that reach the end of the game, it then returns the very first move that it used to originate it.

Our A* agent plays Ticket to Ride as if it were the only player, and as such it does not simulate the moves for the adversary. Instead, the heuristic we designed uses the amount of points that the player has at any given point in time as a key aspect, so maximizing the heuristic would be maximizing the amount of points scored by the agent.

With the intent of making the A* Agent as competitive as possible, we derived a heuristic that would reflect some of the principles used by more experienced players. The heuristic we used to evaluate each game state is as follows:

$$h = \begin{cases} P_{pts} + (20 - H_{size}) + D_{score} & \text{if } H_{size} > 20 \\ P_{pts} + H_{size} + D_{score} & \text{if } H_{size} \leq 20 \end{cases}$$

where P_{pts} is the number of points the agent currently has. H_{size} is number of train cards the player currently is holding. D_{score} is the relative points the player scored in regards to their destination cards. For each destination card they currently own, if they claimed routes that connect the cities on that card, the amount of points of such are added to D_{score}, and if not, the amount of points of that Destination Card are subtracted from D_{score}.

The heuristic is built such that the agent is rewarded for drawing Train Cards until it has a reasonable number of options to claim routes with and punished when it starts drawing too many cards without using them.

B. Route Focus Agent

This agent is built to mimic some of the strategies used by more experienced players. The main focus of the agent is to complete the Destination Cards. At the start of the game, the agent keeps all of 3 Destinations it gets during the setup of the match. Then, in every subsequent turn, it evaluates all Destinations Cards it has not completed yet. For each Destination, it looks for the shortest path between the two cities, considering the routes it has already claimed. It then puts the routes that make up that path into a priority queue, where their priority is defined by the function:

$$P = R_{value} + 2 * DR_{score}$$

where R_{value} is the amount of points the player scores just for claiming that route. DR_{score} is the point value of the Destination Card that route belongs to.

If all of its Destination Cards have been completed, it looks at all of the free routes and puts them in the queue using the only R_{value} as their priority.

After the agent fills the priority queue, it goes to every route in the queue and checks if it has the Train Cards to claim it, and if it does, it returns that as a move. If it can't take any of the routes, it checks the color of the highest priority route. It chooses to draw the face up Train Card that match that color, and if that color is not available, it draws the top card of the deck or a face up wild card.

C. AI Match-up

In order to evaluate how our agents fare when playing the game, we simulated 240 2-player Ticket to Ride matches where A* Agent played against Route Focus Agent. We counted ties as a win for both players, and there were 3 ties in the simulations. We compiled the most interesting aspects in Table I.

From looking through the data we can evaluate some of the aspects of how the two Agents play the game. While the Route Focus Agent is more efficient in scoring points, it is also slower in playing the game. The A* Agent reached the end condition of the game first in 71% of the games played. However, when the Route Focus Agent actually depletes its Train Tokens first it wins the game 94% of the time, against the 45% success ratio A* has. It is also clear that the Route Focus Agent strategies achieves higher amounts of points, when successful. The Route Focus Agent scores, on average, 21 more points than A* and its high score exceeds that of A* by 50 points. On the other hand, when A* loses, it is able to score a higher point average than when Route Focus loses. We believe that this is due to the fact that A* is able to adjust better to the moves made by the adversary during the game

Another factor in running the A* Agent and Route Focus Agent is the efficiency of the two algorithms. The Route Focus Agent completes in an insignificant amount of time, while A* can take from 2 to 10 minutes per game.

Since the Route Focus Agent can win more consistently, it is able to score higher point averages and employs an easier to analyze and more straight-forward approach to playing the game, we believe that such agent seems to be closer to achieving results that is more expressive of how non-novice human players play the game. That, together with being able to generate higher volumes of data, due to the faster running time of the agent, is why we decided to do an analysis of the game by simulating multiple matches between Route Focus Agents.

V. ANALYSIS OF THE GAME

The analysis of the game was performed by having the Route Focus Agent play against itself in 10,000 games. Additionally, randomly playing agents were used in experiments to reveal niche cases and increase coverage of the game space.

A. Scenarios not covered by the rules

Over the course of the experiments, the Agents encountered three scenarios that are not accounted for by the rules of the game. All three scenarios created situations in which one or both players could not make any moves.

1) No Train Cards: This scenario arises when one player exclusively draws Train Cards, and the other player fails to finish the game before all Train Cards are depleted. If all of the Train Cards have been drawn, but the player lacks the necessary Train Cards to claim any routes, then they are forced to draw Destination Cards, forcing them to decrease their own score. Once the Destination Cards have been depleted, the player is no longer able to draw any more cards, nor is he able to claim any routes.

2) Not Enough Trains: This scenario is much more common than the scenario discussed above. This scenario arises when both players play greedily, aggresively claiming routes of cost 1 and 2. There are 5 paths of cost 1, and 25 paths of cost 2, so all of them can be claimed by spending 55 Train Tokens. Between 2 players, there are 90 Train Tokens, making this a plausible scenario for inexperienced players. The last turn is announced once a player has 2 or less Train Tokens. If all of the Train and Destination Cards have been drawn at this point, a player cannot draw an additional card because there are none to draw, and they simply lack the number of trains necessary to claim any route.

3) 3 Wild Loop: This scenario involves a rule involving face up Train Cards. When 3 or more of the face up Train Cards are wild cards, the face up cards are returned to the deck, and 5 new cards are added. If both players exclusively draw non-wild Train Cards, then at some point, the deck will consist exclusively of wild Train Cards. At some point, the wild cards will appear in the face up set, which will prompt a reshuffle. However, if there are no other cards in the Train Card deck, then the wild cards will reappear in the face up set, prompting another reshuffle. The reshuffle cannot end because there are no other cards in the deck.

TABLE I

DATA FROM THE A* STAR AGENT VERSUS ROUTE FOCUS AGENT GAMES

	# Wins	Win Ratio	Avg. Score	Win Avg. Score	Lost Avg. Score	Win Highest Score	# last turns	Win Ratio w/ last turn
A*	83	0.35	51.42	63.63	45.08	94	171	0.45
Route	160	0.65	72.62	89.58	39.92	144	69	0.94

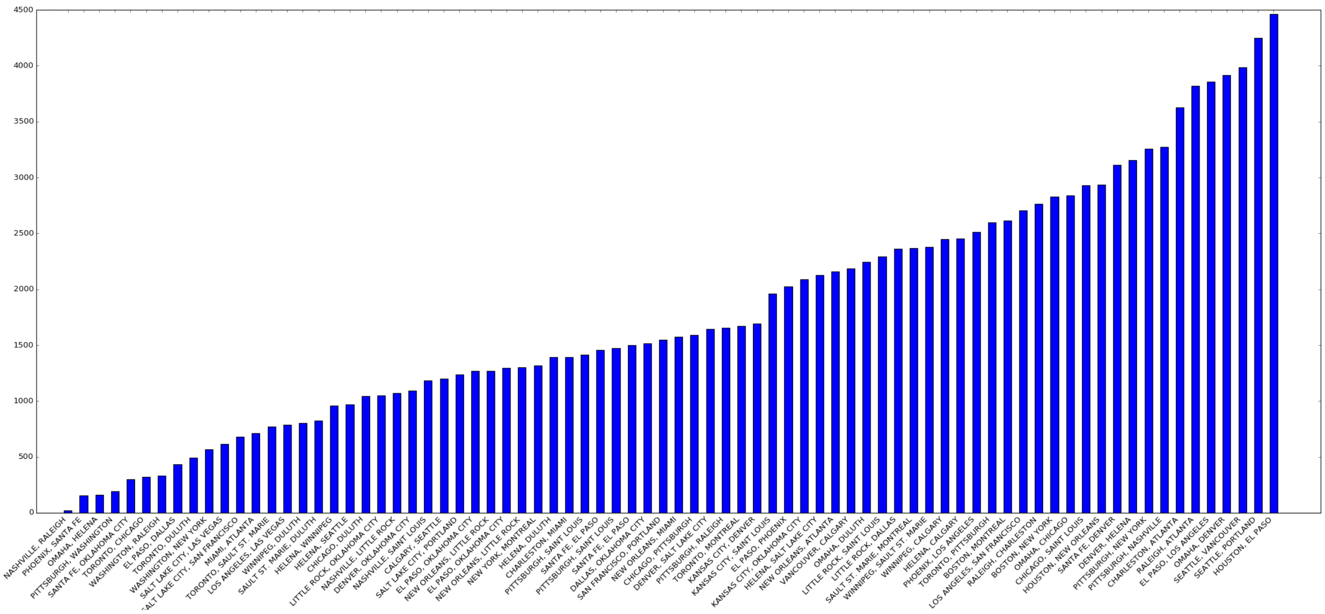

Fig. 2. The bar chart for the frequency in which the winner of a match takes a specific route. On the y-axis are the number of games in which the winner took a route, out of 10,000 games. The labels on the x-axis show the routes, named after the two cities it connects.

B. Winning Routes

Fig. 2 shows the distribution of routes claimed by winning players over the course of the simulations. As can be seen, there is substantial variation between the routes, indicating that as a whole, winning agents favored certain routes over others. Several plateaus are also visible in the distribution, implying a tiered nature to the value of connections in the game board. There are two interesting features to note about the 7 connections that make up the highest plateau. First, routes of length 1 and 6 are disproportionately represented in this cluster relative to its quantity on the game board. Roughly 6 percent of all routes on the board are cost 1, and 15 percent of routes are cost 6, yet both make up 28 percent of the highest tier of routes. While there are 2 routes of cost 2, they are the most common type of route on the board, and so the representation is approximately proportionate.

Also of note in this cluster is the set of cities the routes connect. El Paso, Atlanta, and Seattle each have two connections in the top cluster. These three cities are notable for having a substantial number of connections. However, Helena, Toronto, and Pittsburgh all have more connections than Seattle, yet no connections to them appear in the top cluster. The reason for this is currently unknown and bears further investigation, but the data indicates that these locations hold some particular value in gameplay.

C. Route Costs

Fig. 3 depicts how many routes of each cost exist in the game board, as well as how many of each route is claimed by a winning player over the course of the experiment. As a whole, the distribution of winning route costs is consistent with the distribution of route costs in the game. However, there are points in which the two distributions diverge substantially. For example, connections that cost 1 Train Token are claimed fairly often by winning players, although there are very few of them on the game board. This over-representation indicates that winning play disproportionately favors connections that cost 1 Train Token. The opposite relation can be seen when considering routes that cost 3 Train Tokens. Winning players claim very few of these routes, relative to how many there are on the game board. This underrepresentation implies a certain deficit in this category of connections. One possible explanation for this deficit is that these routes are too expensive to be desirable for greedy play, but are also too inefficient to be desirable in long-term play.

D. Undesirable Cities

Fig. 4 is a heatmap displaying the frequency with which cities were ignored over the course of the experiment. For a city to be counted as ignored, all of its connections must

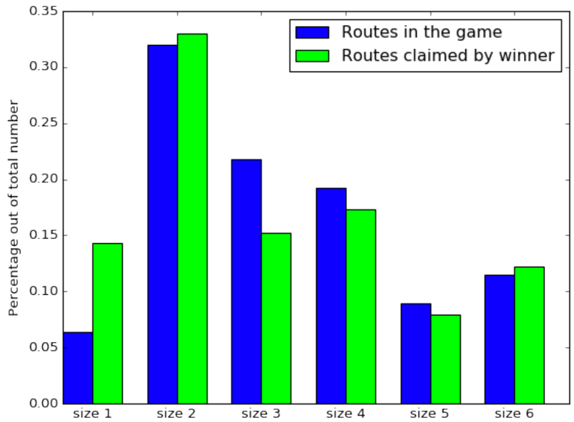

Fig. 3. A bar chart showing the difference in prevalence between how many of which size route there is in the game and how frequent they are taken by the winning player. The x-axis shows the different sizes of route. The y-axis shows how much of the total each one represents. The values for the routes in the game are in blue and the value for the routes claimed by the winner are in green.

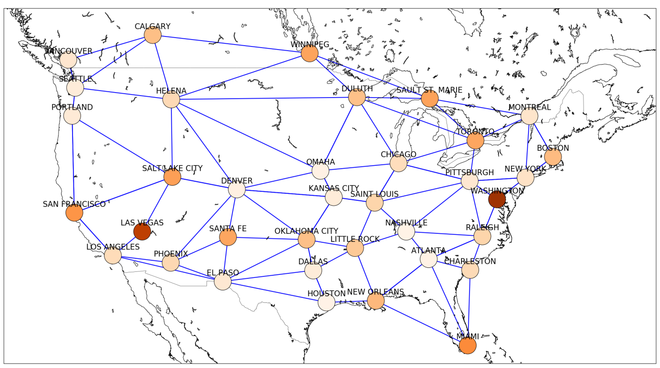

Fig. 4. A color map of how often a city is claimed by players. A claimed city is one that has at least one route connecting it to an adjacent city claimed by the end of the game. On the map, the darker the city color is, the less desirable it is, meaning, there were more games where no routes connecting to it were claimed.

be unclaimed. The implication behind an unclaimed city is that it is neither a destination nor a city through which a destination can be reached optimally. While unclaimed cities can change based on Destination Cards, a large number of simulations can reveal trends in the overall value of the city and its connections.

Particularly surprising is the frequency with which Washington is ignored over the course of the experiment. The values for other undesirable cities such as Las Vegas and Miami are understandable, because the former has only 2 connections and the latter is expensive to reach. However, Washington has 3 connections, all of which are inexpensive methods to reach key cities such as New York and Pittsburgh. One possible explanation is the fact that Washington is not on any of the Destinaion Cards, meaning that at most, it will be used as a passage to other cities. It is also worth noting that Pittsburgh is a more connected alternative to Washington for players attempting to connect Raleigh to New York. Under this assumption, it makes sense that Washington is undesirable, because Pittsburgh is simply a more attractive alternative.

VI. Discussion and Conclusion

The results demonstrate that large scale AI-based testing and exploration has the potential to be useful in analyzing both concrete loopholes in game rules as well more abstract strategic biases. Tree-search and evoutionary algorithms are relatively easy to modify to meet certain optimization requirements, and so designers can create AIs that modify various playstyles. This is potentially useful because it gives the designers the ability to promote certain strategies by testing to see how well it performs in the game they are creating, and tuning the game accordingly. This provides a great deal of freedom to designers who have a specific vision for their game.

So far, we were only able to generate sizable amounts of data for the random and Route Focus Agents. Because of this, the data is biased toward those playstyles. We feel that the optimization requirements are general enough that the effect of this bias is not too severe, but we would ideally have a wider variety of algorithms, to confirm that these values would persist even after introducing additional algorithms.

VII. Future Work

We intend to investigate the possibility of collecting large amounts of data from a wider variety of algorithms. We also have reason to believe that this methodology could act as an effective evaluation function for an evolutionary algorithm designed to tune values in a game. This would work to automate a tedious but still essential step in the game design process. In addition to testing for fairness and rule integrity, designers could also use AI to promote certain playstyles, by tuning a game such that certain strategies play more effectively than others.

References

[1] E. Kraaijenbrink, F. van Gils, Q. Cheng, R. van Herk, and E. van den Hoven, "Balancing skills to optimize fun in interactive board games," in *Human-Computer Interaction–INTERACT 2009*. Springer, 2009, pp. 301–313.

[2] A. Jaffe, A. Miller, E. Andersen, Y.-E. Liu, A. Karlin, and Z. Popovic, "Evaluating competitive game balance with restricted play." in *AIIDE*, 2012.

[3] C. Salge and T. Mahlmann, "Relevant information as a formalised approach to evaluate game mechanics," in *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*. IEEE, 2010, pp. 281–288.

[4] V. Hom and J. Marks, "Automatic design of balanced board games," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, 2007, pp. 25–30.

[5] J. Krucher, "Algorithmically balancing a collectible card game," Bachelor's Thesis, ETH Zurich, 2015.

[6] T. Mahlmann, J. Togelius, and G. N. Yannakakis, "Evolving card sets towards balancing dominion," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*. IEEE, 2012, pp. 1–8.

[7] Ticket to ride. Accessed: 2016-05-15. [Online]. Available: http://en.wikipedia.org/wiki/Ticket_to_Ride_(board_game)