Evaluating Heuristics in the Game Risk
An Aritifical Intelligence Perspective

Franz Hahn

August 23, 2010

Abstract

Risk[1] is a board game that provides a highly stochastic multi-agent environment. This article studies heuristics that could be the basis of an adaptive artificial player, using Artificial Intelligence (AI) to maximize its chances of winning the game. A round-robin evaluation of the heuristics using a specifically devised Risk simulator suggests the combination of three heuristics: supplying and reinforcing units to borders, and attacking with full force and only if the chance of winning the battle is above 50%.

1 Introduction

With the fast evolution of computers and other interactive technologies, and no indication of it slowing down, it has been the idea of many researchers to tackle real-world problems with computational engineering [7]. In order to do so, these real-world problems often have to be abstracted to a certain level. Solving the abstraction allows a prediction about the solution for the actual problem. Games often give a good substitute for real-world applications with similar features, such as partial observability and high-stochasticity. While more and more complex games hit the market, providing a rich context for problem solving [10], decades old games still have not been solved.

The video game market is a juggernaut of development. Launches of long awaited games can take in as much as summer blockbusters in the movie business. *Grand Theft Auto IV* racked up US$500 million in sales during its opening week in 2008, more than *Spider-Man 3* or *Pirates of the Caribbean 3* on their opening weekends, which gained US$380 and US$340 million respectively. While game development studios have pushed the perfectioning of game graphics, few have explored the depths of high-level AI opponents. Games that have done so, however, are usually remembered well for these strong AI opponents. Examples for this are *Pacman* and *Goldeneye 007* [3]. Some games even base their whole gameplay around the idea of teaching an AI, such as *The Sims* and *Black and White*.

Another big application field for an AI in the gaming industry are board games. Games like *Chess*, *Checkers* or *Backgammon* have been "solved" to an extent, where an AI can beat strong human players [2, 4, 11]. Cash prizes have been set out to programmers who can write an AI that beats expert players in deterministic games such as *Go* [1] and *Havannah* [5]. Furthermore, nondeterministic games with multiple players, multiple moves per turn, and a lot of stochastic elements as part of many moves have seen even less breakthroughs in terms of strong AIs yet. Examples for this game category are *Axis and Allies* [12] and *The Settlers of Catan* [9].

Challenges in developing strong AI include high stochasticity, requirement of deep planning despite vast action space due to combinatorial explosion of sequential actions, and opponent modelling. One game that embodies many of these challenges is Risk. Risk is a multi-player, non-cooperative, sequential, highly-stochastic, imperfect information game, yet to be played on a high-performing level by an AI. It is played on a board which resembles a map of the world, divided into several continents, which subsequently are divided into countries. The goal for the player is to conquer all countries, eliminating all opponents.

1.1 Problem Statement and Research Questions

This thesis focuses on the implementation and evaluation of heuristics to find a basis for a strong AI for the game Risk. The problem statement is therefore formulated as:

Is it possible to implement a strong AI for Risk, based solely on simple heuristics?

To develop an AI that plays fairly elaborate, it will have to be able to answer the problem "What is the best move in the given situation", or "What is the strength of my current position". Different heuristics have been investigated in order to help answer these questions. This leads to the following research questions:

[1] Risk is a Parker Brothers registered trademark for its world conquest game and game equipment.

RQ1 *Which individual heuristic has the strongest impact on the performance of the Risk AI?*

RQ2 *What combination of heuristics leads to the strongest play of the Risk AI?*

Since a player's turn in the game is divided into different phases explained later in this article, combinations of modular supply, attack and move heuristics are analysed. The presented research questions will explore the complexity of the problem statement, and their answers will lead to a moderately strong Risk AI.

1.2 Overview

This paper gives a description of Risk and background information in Section 2. The game's historical development is outlined in a short manner. Thereafter follows a recapitulation on the rules as well as variations that are investigated in this paper. Additionally, strategies that can be applied to improve efficiency of play are illustrated. Experimental setups intended to help evaluate the heuristics for a winning strategy are topic of Section 3, as well as a short illustrated description of the framework used to run the experiments. The results of these experiments are evaluated in Section 4, and interpreted subsequently in Section 5. Furthermore, Section 5 describes future research and concludes the article.

2 Background

This section gives a description of the background of Risk. The history of the game as well as a brief summary of it's rules is presented. Furthermore, two related scientific articles are outlined and their relation to this article is briefly described.

2.1 The Game Risk

Risk was invented by Albert Lamorisse in 1957 under the name "La Conquête du Monde" [6] meaning "The Conquest of the World". It has gone through a lot of variations and refinements ever since, including adaptions to many blockbuster movies such as Star Wars and Lord of the Rings. Other official Risk games were released with different map layouts and new additions to the functionality of the game, such as naval units or fortresses. In 2009, the game *Risk 1959* was released, which is a direct reproduction of the original game, including the original graphics and units made of wood. In 1988, the International Risk Tournament of Champions was brought into being, which celebrated its 20^{th} Anniversary in 2008[2].

The Risk world map is made up of 43 countries, all of which are connected to at least one other country. All countries are reachable from every other country, sometimes through a number of different connections and all countries belong to a unqiue set of interconnected countries, called continents. What continent a country belongs to is indicated through a specific color the country is drawn in. The interconnectivity in a continent is strong, with every country being connected to at least two other countries of the same continent. Connections between continents on the other hand are usually bottlenecks. Some continents are only reachable through as little as a single country. Holding ownership of all countries of a continent gives a bonus to a players army count at the beginning of his turn.

[2]The official website can be found at http://www.risktoc.org/, however it appears as though it is not being updated anymore.

Figure 1 depicts an undirected graph isomorphic to the game board, with vertices representing countries and edges representing adjacency between two countries [13]. A graph is a pair $G = (V, E)$, where V is a set of vertices (i.e. countries) and E is a set of edges between two vertices, for which holds $E \subseteq \{\{u, v\}|u, v \in V \wedge u \neq v\}$. The edges E in an undirected graph G describe a symmetric binary relation $\sim$ on V called the adjacency relation of G. For each edge $\{u, v\}$ the vertices u and v are said to be adjacent to each other, which is denoted by $u \sim v$.

The game itself is divided into two stages, the setup stage, where the players initialize their positions, and the play stage, in which the players take turns playing their moves. The game is won by whoever is left as the sole owner of countries[3]. Both aforementioned stages are subsequently discussed in detail.

[3]In the original rule set, finishing mission objectives provided by mission cards are an alternate way to win a game. For the sake of simplicity, mission cards are neglected in this thesis.

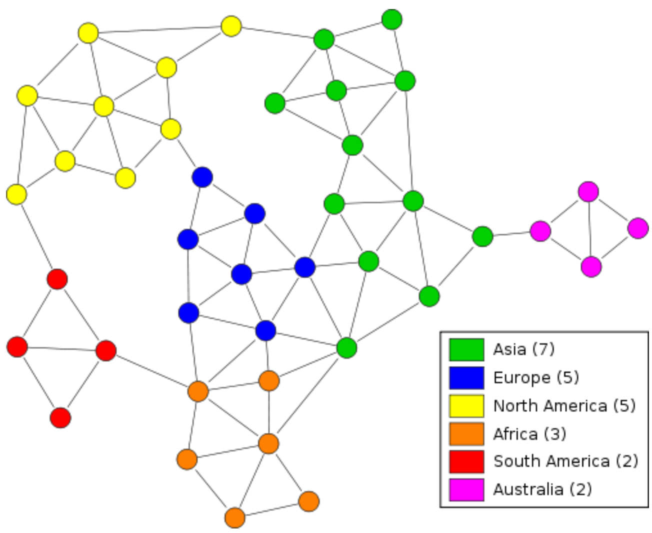

Figure 1: Planar graph representation of the connections between countries on the Risk game board. Each node represents one country. The colors indicate which continent each territory belongs to. Continent supply bonuses are indicated in the legend in parenthesis.

Stage 1: Setup
During the setup stage the game is initialized and prepared for the play stage. Each player throws a die, of which the highest value determines the starting player. The first player gets to claim a country of all unclaimed countries and place a unit on it. This is repeated by every player in turns until all countries are claimed. Once all countries are distributed, all players take turns distributing their remaining units on their countries. The number of supply units is calculated by:

$$\lceil \frac{2 \times CountryCount}{PlayerCount} \rceil$$

Stage 2: Play
While a player has to decide in the setup stage which countries to pick or where to supply his units, he is confronted with these decisions in every turn in the play stage until the game is over. Players take turns playing, going through three phases in their ply:

- Supply Phase
- Attack Phase
- Reinforce Phase

At the beginning of his turn, a player gets a certain number of supply units, dependent on the countries he holds, calculated based on the number of countries (CountryCount) and the sum of continent bonuses listed in Figure 1:

$$\max\{3, \lceil \frac{CountryCount}{3} \rceil\} + ContinentBonuses$$

He can distribute these on any number of countries. He then proceeds to the Attack Phase, in which he can move units from one of his countries onto an adjacent enemy country, rolling dice to determine the attack result as explained in the following subsection 'Battle in Risk', possibly capturing it. In the last phase, a player may move units between adjacent countries that belong to him, to reinforce borders for example.

Battle in Risk
Battle in Risk is handled through rolling dice. When two armies meet in battle, dice determine the number of losses on both sides. Depending on the number of units facing each other, different numbers of dice are used.

If the attacker attacks with one unit, he rolls one die, which is compared with the defenders dice. If the attacker attacks with two units, he rolls two dice, which are compared with the defenders dice. If the attacker attacks with three or more units, he rolls three dice, of which the lowest one is discarded, and the higher two are compared to the defenders dice.

If the defender chooses to defend with one unit, he rolls one die, which is compared to the highest of the attackers dice. If the defender chooses to defend with two or more units, he rolls two dice. ¡ When dice are compared, both sets of dice are sorted from highest to lowest, and the highest of both are compared. If both players have rolled two or more dice, the next two highest are compared. In case of an unequal number of dice in the sets, only the highest comparison counts. For each comparison that a player loses, he removes one unit in his army. In case of a tie, the defender wins the battle. If the defender has no units left the attackers remaining units are moved onto the former defenders country, and the attacker takes ownership of the country.

2.2 Related Work

Risk has been the topic of several research papers, two of which are particularly interesting for this article. At the beginning of a turn, with the attacker having an army of A pieces and the defender having an army of B pieces, choosing the number of battles to engage, or rather finding a threshold at which to cease attacking, depends on the objective chosen. This objective could be any of the following:

- Maximize the probability that the attacker defeats the defender.
- Maximize the expected number of pieces in the attacker's army at the end of the turn.
- Maximize the expected difference between the two armies at the end of the turn.
- Minimize the expected number of pieces in the defenders's army at the end of the turn.

The first article describes how operational research methodology has been applied to a simplified version of Risk, giving insights into useful strategy and objectives. The article outlines the difficulty to pick a favorable objective. Choosing one of these aforementioned objectives over another is completely unfounded when not considering the context. It appears that there is no intuitive advantage in selecting one of them in favor of any of the others.

Maliphant and Smith proceed by defining $P(A, D)$ as the probability of the attacker winning the battle, given the attacker's army strength A and the defender's army strength D, and decisions d_A and d_D of both players regarding the number of dice to be used in the battle. Depending on the number of dice used probabilities l_A and l_D of each player losing units dare calculated. The recurrence relationship is then defined as:

$$P(A,\ D) = \max_{d_A}\left(\min_{d_D}\left(\sum_{all\ l_A,l_D}(P(A-l_A, D-l_D)\times\right.\right.$$

$$\left.\left.\text{probability (A loses } l_A\text{, D loses } l_D \text{ given } d_A,\ d_D)\right)\right)$$

The second article to be discussed here compares search methods for the best plan on endgame movesets, randomly generated for this specific purpose. Comparing seven search algorithms resulted in a predominance of an evolutionary search approach in 85% of cases considered. Comparison was done based on number of opponents eliminated, plan completion probability and value of ending position when the moves do not complete the game. To evaluate the ending position an objective function was formulated, using one of the following heuristics:

- Maximize Overall Strength
- Maximize Expected Reinforcements
- Minimize No. of Defensive Fronts
- Maximize Average Front Strength
- Maximize Logistical Support
- Maximize Smallest Front Strength

The first two of these heuristics turned out to be the most influential to win games, while the last four combined had an impact lower than each of the first two.

3 Strategies and Heuristics

A person playing Risk the first time will figure out quickly that a couple of simple rules will benefit his chances to win the game. Controlling entire continents instead of countries scattered across the map increases unit supply and minimizes the number of countries adjacent to enemy territory. Adjusting unit supply on a country to repel neighboring armies increases chances of keeping this country over the course of the enemies turn and opens up possibilities to launch an attack that has its origins there. In order to find the most optimal plan to win a game, we will look closer at possible strategies.

Before explaining these tactics, a couple of abstractions that are used in our implementation of Risk have to be pointed out. There are two main differences to classic Risk. First of all, while all attacks are split up into smaller attacks of at most 3 attacking units and at most 2 defending units, we only give the AI the choice of the amount of units to attack with. It is not allowed to stop in between one of these small sub-attacks. When the AI attacks with 10 units for example, it will keep issuing small sub-attacks until either the target country is conquered, or all of those 10 units are dead. Secondly, an AI can issue as many reinforcement moves as it wants. The original rules of risk allow a single reinforcement move per turn.

Combat Dice Probabilities

Table 1 presents a matrix of victory probabilities for a battle between an attacker with A armies and a defender with D armies, for values of A not greater than 6, and D not greater than 10 [8]. If a player attacks with 3 units, and the target country holds 2 units, his chance of winning the battle is 65.6%. This is assuming, that the attacker follows through with his attack, regardless of whether he loses a lot of his units in the first dice throws.

The line drawn in Table 1 indicates the turning point, at which the attacker has a higher chance to win the battle than the defender. We derive a rule of thumb, that, as an attacker, having one unit more than the defender should on average win the battle.

Heuristics

The most basic strategy to follow is playing Random, which will be used as a baseline comparison throughout this article. This strategy can be applied to all three phases of the game, which will be called Random Supply, Random Attack and Random Reinforce respectively, while other strategies are unique for a game phase. Thus, we will investigate each game phase by itself in terms of strategies and tactics.

In order to have sufficient attack power, it is necessary to supply enough units at the right places in the supply phase. To achieve this, it might be beneficial to deploy units in countries that are adjacent to enemy territories. The heuristic Border Supply distributes its supply units randomly on border countries. An additional benefit can be drawn from good unit placement on the border in order to prepare for attack and defense. Taking the summation of all units in enemy countries y

Table 1: Probability that an attacker with A units wins the battle against a defender with D units, the indicated line showing the turning point at which the attacker has a higher chance to win the battle than the defender.

A \ D	1	2	3	4	5	6
1	0.417	0.106	0.027	0.007	0.002	0.000
2	0.754	0.363	0.206	0.091	0.049	0.021
3	0.916	0.656	0.470	0.315	0.206	0.134
4	0.972	0.785	0.642	0.477	0.359	0.253
5	0.990	0.890	0.769	0.638	0.506	0.397
6	0.997	0.934	0.857	0.745	0.638	0.521
7	0.999	0.967	0.910	0.834	0.736	0.640
8	1.000	0.980	0.947	0.888	0.818	0.730
9	1.000	0.990	0.967	0.930	0.873	0.808
10	1.000	0.994	0.981	0.954	0.916	0.860

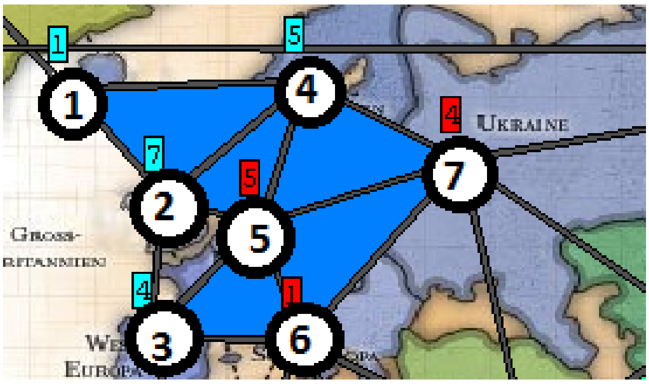

Figure 2: Example game state between two players.

adjacent to country x will give a measure which we call Border Security Threat (BST) in x.

$$BST_x = \sum_{y=1}^{n} AmountOfUnits_y$$

Dividing this BST by the units situated in x gives a Border Security Ratio (BSR) which can be compared among all border countries.

$$BSR_x = \frac{BST_x}{AmountOfUnits_x}$$

Countries with a high BSR are more likely to be conquered by an enemy player, since the number of enemy units in adjacent enemy countries are relatively higher than than the number of units on the country itself. Choosing countries with a high BSR to supply to will increase their defensive strength by lowering the BSR. Supplying units to countries with a lower BSR, meaning that they already have a better defensive stance, will increase their offensive strength, raising the chances of a successful attack from these countries. Using this measurement, we introduce a new supply heuristic called BSR Supply, that deploys its supply units to border countries according to the country's BSR.

Different tactics can be applied to use the BSR to find the right way to distribute a players army. Normalizing the BSR by dividing it by the sum of all BSRs of countries, a player owns, will give a direct measurement by which someone could arrange units. The Normalized Border Security Ratio (NBSR) is calculated by:

$$NBSR_x = \frac{BSR_x}{\sum_{z=1}^{n} BSR_z}$$

It gives a direct ratio of how the units could be distributed among countries. This, however, can be problematic with a low number of supply units. To illustrate this, the example depicted by Figure 2 is used. It shows an example game state between two players in Europe. The country names are numbers from one to seven, the number of units situated on them are illustrated by the rectangles above the country names, which also indicate ownership of the country by color. Given the game state, we can calculate BST, BSR and NBSR for country 5 as follows:

$$BST_5 = \sum_{y=1}^{n} AmountOfUnits_y = 7 + 4 + 5 = 16$$

$$BSR_5 = \frac{BST_5}{AmountOfUnits_5} = \frac{16}{5} = 3.2$$

$$NBSR_5 = \frac{BSR_5}{\sum_{z=1}^{n} BSR_z} = \frac{3.2}{3.2 + 4 + 1.25} = 0.37$$

All values for this game state are shown in Table 2. Assuming a supply of 3 units for the light gray player, the unit distribution would be 0.53 units on 2, 1.12 units on 3, and 1.35 units on 4. It is hard to decide how exactly to distribute the units based on these numbers. Is it reasonable to put one unit on each country? But that would be equivalent to a NBSR of $\frac{1}{3}$ for each, which is far from what the actual NBSR is. One way to tackle this problem is to use a threshold, below which all BSRs are set to 0. This would eliminate low-NBSR countries by putting them to 0 as well. In this article we have neglected the use of a threshold for simplicity's sake.

During the attack phase, it is most crucial not to enter battles that are unlikely to be won. When attacking randomly, this important information is left out. One option to maximize the chance of wining a battle is to always attack with full force, after deciding origin and destination of the attack, i.e., instead of using a random number of units, any attack launched uses the maximum number of units available. We will call this Full Force Attack.

While using all units to attack ensures, that a player doesn't waste units through randomly choosing to go with one unit against 50, there is still no anticipation of the likelihood of success of the attack. If the AI only has

Table 2: Values of BST, BSR and NBSR for the example game state given in Figure 2.

V	U	E	BST	BSR	NBSR
1	1	0	0	0	0
2	7	1	5	0.71	0.18
3	4	2	6	1.50	0.37
4	5	2	9	1.80	0.45
5	5	3	16	3.20	0.38
6	1	1	4	4.00	0.47
7	4	1	5	1.25	0.15

one unit on a country that is chosen as attack origin, it will follow through with it. We already determined, that it is a rule of thumb that attacking with one more unit than the defender has should result in a win for the attacker. This heuristic will be called High Chance Attack.

When relocating units it may prove efficient to move all units from inner territory countries onto border countries, in order to be able to fend off attacks in the subsequent turns. This is called Border Reinforce. Similar to the supply heuristics, favoring countries with a high BSR as destinations should prove efficient.

To illustrate the logic behind these strategies, Algorithms 1, 2 and 3 present an abstract attack heuristic in pseudo code. In initiation, the origins of attacks are gathered, which are given to the main method that selects the best suitable target for each attack, launches one after the other, checking via the update procedure, whether a new country is viable as attack origin after each attack was carried out. The overall methodology stays more or less the same throughout all supply, attack and reinforce heuristics. Merely the way origins and targets are selected is different.

Algorithm 1 initiation()

```
borderCountries ← getBorderCountries()
selectionChances ← ratio(borderCountries)
for all country ∈ borderCountries do
  if selectionChance(country) > threshhold then
    originCountries.add(country)
  end if
end for
main(originCountries)
```

Algorithm 2 main(originCountries)

```
while originCountries.isNotEmpty() do
  origin ← select(originCountries)
  targetCountries ← origin.getAdjEnemies()
  attackRatio ← selectionRatio(targetCountries)
  target ← select(targetCountries)
  unitCount ← originCountry.getRandUnitAmount()
  newCountry ← attack(origin, target, unitCount)
  update(newCountry)
end while
```

Algorithm 3 update(newCountry)

```
if newCountry then
  borderCountries ← getBorderCountries()
end if
selectionChances ← ratio(borderCountries)
for all country ∈ borderCountries do
  if selectionChance(country) > threshhold then
    originCountries.add(country)
  end if
end for
```

4 Experimental Setup

Before presenting the analysis, this section describes the experimental setup used to generate the datasets. Every action a player takes is saved in a log file. The log files are set up in a machine readable way, where every entry in the log file contains ten integers, characterizing the event:

1. Ply
2. Player
3. Game phase
4. Origin country (if applicable)
5. Destination country
6. Number of units moving
7. Number of enemy units encountered (if applicable)
8. Number of unit casualties (if applicable)
9. Number of enemy unit casualties (if applicable)
10. Time taken to complete the action

The outline of the log file entry differs dependent on the game phase. An example supply action log entry looks like this:

7	0	0	-1	42	3	-1	-1	-1	3

It means, in ply 17, player 0 supplied 3 units to country 42, which took 3 milliseconds to complete. The value -1 indicates "not applicable". An attack action log entry example looks like this:

22	1	1	10	12	1	2	1	0	5

In ply 22, player 1 attacked country 12 from country 10 with 1 unit. He faced 2 units and lost his one unit, while the enemy lost none. This action took 5 milliseconds. Lastly, a reinforce log entry looks like this:

7	0	2	2	1	1	-1	-1	-1	2

In ply 7, player 9 reinforced country 1 from country 2 with 1 unit, which took 2 milliseconds to complete.

This convenient log file setup allows a straightforward unconstrained analysis. It also theoretically allows a player to replay an entire game to any certain point. The choice of software to analyse the data is left arbitrary.

We let the AIs play 1000 games in a round-robin fashion, including self-play and evaluating each pair twice. So player 1 with configuration 2 plays against player 2 with configuration 4, as well as player 1 with configuration 4 versus player 2 with configuration 2. These repetitions of games can help identify possible advantages for the starting player.

5 Results and Discussion

Results are presented to be discussed in the next section. Configurations are made up of 3 digits, each indicating the heuristic used for the respective game phase. The first digit represents the supply heuristic used (1 is Random Supply, 2 is Border Supply and 3 is BSR Supply), the second digit shows the attack heuristic used (1 is Random Attack, 2 is Full Force Attack and 3 is High Chance Attack), and the third digit indicates the reinforce heuristic used (1 is Random Reinforce and 2 is Border Reinforce).

First we will look at the time complexity of the games based on the player configurations. Figure 3 shows the mean time a game with the player configurations on the x and y axis takes to play, in ms. Since the duration of games where both players use supply heuristic 1 is significantly higher, those configurations are plotted separately in Figure 4. We can see the combination of both player 1 and player 2 playing 112 taking an average of close to 1 second per game, while other configurations are taking less than 50 ms to complete a game. This is partly due to Random Supply and Random Attack being used, which are increasing the number of plies per game. This can be seen in Figures 5 and 6, which show the mean number of plies per game with the player configurations given on the x and y axis.

Combining the information of time taken per game and mean number of plies, leads us to the analysis of the heuristics by means of time complexity of the algorithms. Figures 7 and 8 show the mean time taken per ply based on the configurations indicated on the x (Player 1) and y (Player 2) axis. When the first player is playing any configuration using supply heuristic 2 or 3, we can see a drastic increase in time taken per ply between configurations of player 2 with supply heuristic 2 or 3, and

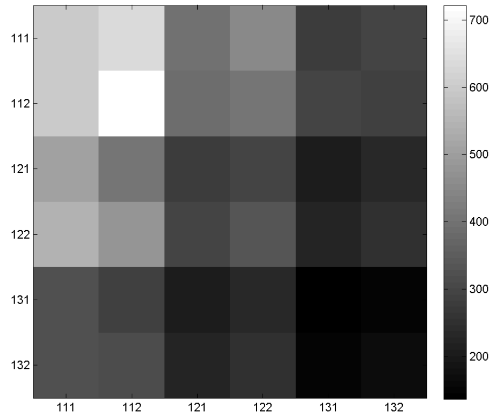

Figure 4: Mean time taken per game in ms plotted in grey scale. Player 1 playing with AI configuration on y axis against player 2 playing with AI configuration on x axis. The color map on the right shows the levels of brightness and the respective duration in ms.

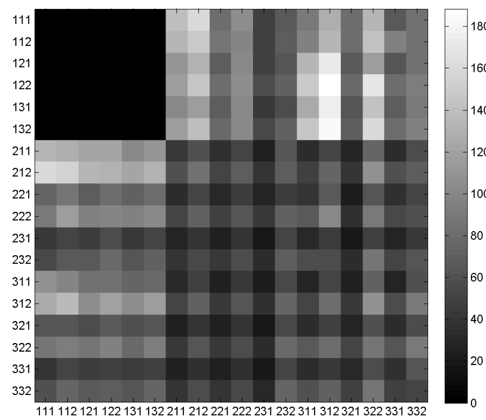

Figure 3: Mean time taken per game in ms plotted in grey scale. Player 1 playing with AI configuration on y axis against player 2 playing with AI configuration on x axis. The color map on the right shows the levels of brightness and the respective duration in ms. The 6x6 matrix in the upper left corner is plotted separately in Figure 4.

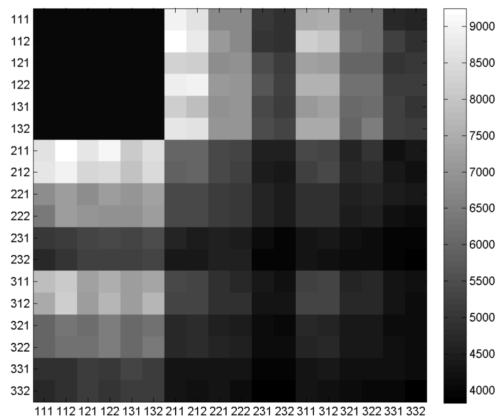

Figure 5: Mean number of plies per game of player 1 with AI configuration on y axis against player 2 playing with AI configuration on x axis plotted in grey scale. The color map on the right shows the levels of brightness and the respective number of plies. The 6x6 matrix in the upper left corner is plotted separately in Figure 6

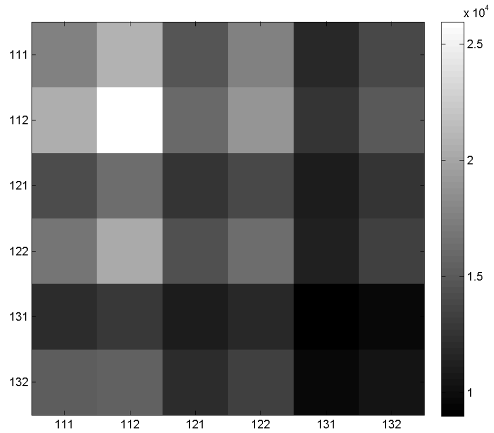

Figure 6: Mean number of plies per game of player 1 with AI configuration on y axis against player 2 playing with AI configuration on x axis plotted in grey scale. The color map on the right shows the levels of brightness and the respective number of plies.

reinforce heuristic 1, and configurations of player 2 with supply heuristic 2 or 3 and reinforce heuristic 2. We will analyse later in this section if this increase in time complexity does increase performance of the AI in a similar way. While configuration 121 versus 111 has the longest time per ply, it is not the worst configuration in terms of mean runtime.

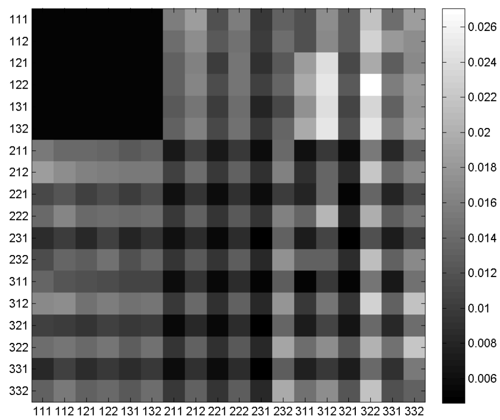

Figure 7: Mean time taken per ply in ms plotted in grey scale. Player 1 playing with AI configuration on y axis against player 2 playing with AI configuration on x axis. The color map on the right shows the levels of brightness and the respective time per ply. The 6x6 matrix in the upper left corner is plotted separately in Figure 8

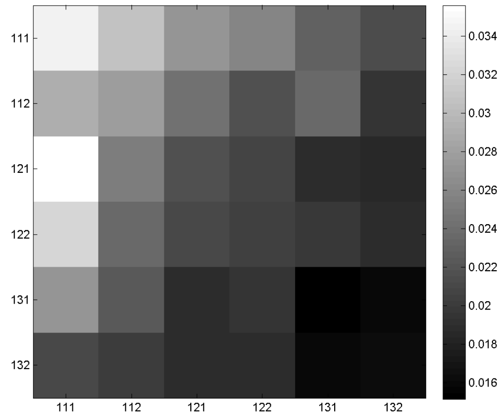

Figure 8: Mean time taken per ply in ms plotted in grey scale. Player 1 playing with AI configuration on y axis against player 2 playing with AI configuration on x axis. The color map on the right shows the levels of brightness and the respective time per ply.

Figure 9 shows the chance of player 1 to win a game against an AI with the same configuration. The complete data can be found in Tables 3 and 4. Individual configurations are enlisted on the x axis, the y axis indicates the win chance. The median is depicted by a line, the box shows the 25% to 75% confidence interval, while the whiskers show extreme data points that are not considered outliers. Outliers are plotted separately as crosses. The overall mean chance for player 1 to win a game with this setup is 66.59%, the 95% confidence interval lying between 58.42% and 74.77%. From this we can derive

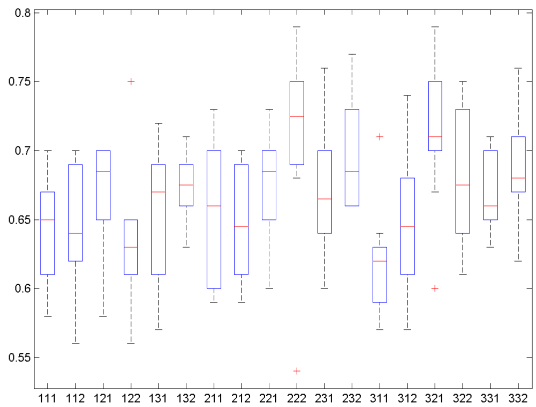

Figure 9: Win chance of player 1 winning, based on self-play game setups on x axis.

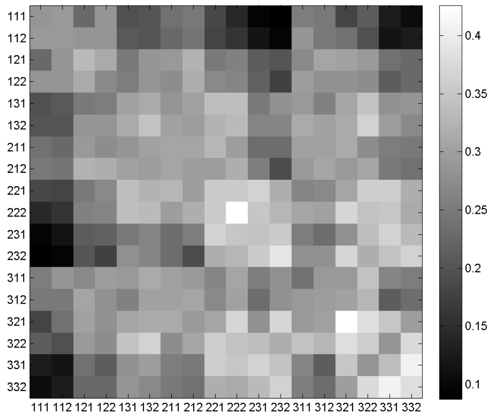

Figure 10: Advantage of player 1 with AI configuration on y axis against player 2 playing with AI configuration on x axis plotted in grey scale. The color map on the right shows the levels of brightness and the respective advantage in percent.

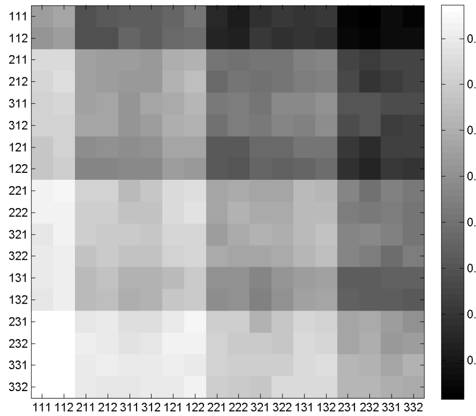

Figure 12: Win chance of player 1 with AI configuration on y axis against player 2 playing with AI configuraion on x axis, (see Tables 3 and 4) plotted in grey scale. The color map on the right shows the levels of brightness and the respective win chance.

an advantage for the first player which is illustrated in Figure 10. The closer the performance of the AI configurations get, the bigger the advantage of the first player. Running the strongest versus the weakest combination of heuristics, we note a marginal advantage for the first player. This can be explained with the fact, that one of the configurations is vastly better performing, and thus the sequence in which the players take their turns has little impact on the outcome of the game.

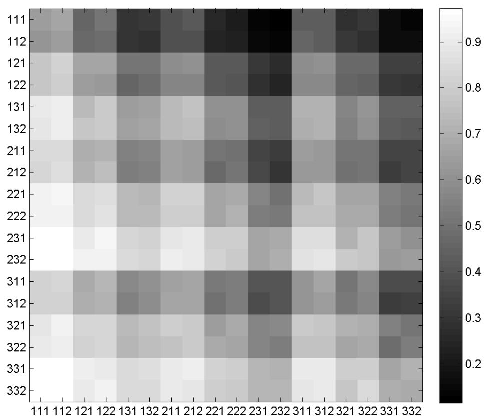

Figure 11: Win chance of player 1 with AI configuration on y axis against AI configuraion on x axis, plotted in grey scale. The color map on the right shows the levels of brightness and the respective win chance.

Figure 11 depicts a color map of the mean winning chance of player 1 playing with the configuration indicated by the y axis, against player 2 with the configuration indicated by the x axis. The color bar to the right shows the win chance by color. The data is sorted by supply heuristic, then by attack heuristic, and lastly by reinforcement heuristic used. Those entries that describe a configuration that uses the third attack heuristic (High Chance Attack) is in all columns brighter than the same configuration with the second attack heuristic (Full Force Attack). The same relation exists for second attack heuristic compared to first (Random Attack). Since brighter color means better performance, we can derive that High Chance Attack is the best performing attack heuristic, followed by Full Force Attack.

Taking the fact into account, that the transition from second to third attack heuristic in terms of strategy is an addition of a battle win probability check, based on the battle dice probability table shown earlier (see Table 1), this result was to be expected.

Furthermore, Figure 12 is a different representation of Figure 11. Rows and Columns have been swapped, to accentuate the influence of the supply strategy on performance of the Risk AI. Rows with configurations using the first supply heuristic are generally darker and thus performing worse than those with a higher order supply heuristic when the other strategies are kept the same. This shows the dominance of the second and third supply heuristic over the first.

The effect of advanced reinforcement strategies could not be determined. There is no clear indication for an enhancement of performance due to cleverer movement of units, mainly because with the attack heuristics implemented, a player uses his full army to attack, not leaving any units behind. And due to BSR supply, the borders do not get out of balance during a players turn.

Taking the strongest performing configuration, we now want to look at characteristics of games. Running 10000 self-play games of configuration 332, Figure 13 depicts the unit development of the winning and losing player respectively over game progress. Illustrated are the mean number of units at any ply, with 95% confidence. The x axis indicates the game progress in percent. The unit advantage of player 1 is given in Figure 14. We can derive with over 97.5% confidence, that at 60% into the game, the winner has a unit advantage, since the lower bound of the confidence interval exceeds 0 there.

Finally, after presenting the results received from the experiments, we want to go back to our initial problem statement and the corresponding research questions, which were to find the heuristic with the strongest impact on the performance of the AI, as well as the combination of heuristics that leads to the strongest play. The single strongest impact on the performance of the AI was achieved with improvements in the strategies used to attack. Since the overall goal of the game is to conquer all countries, an aggressive play style will end games quicker, as well as increase your chances to win, when playing any of the tested configurations. The best performing configuration overall are those, using BSR Supply and High Chance Attack, and utilizing either of the reinforce heuristics. The difference between using one or the other reinforce heuristic is marginal.

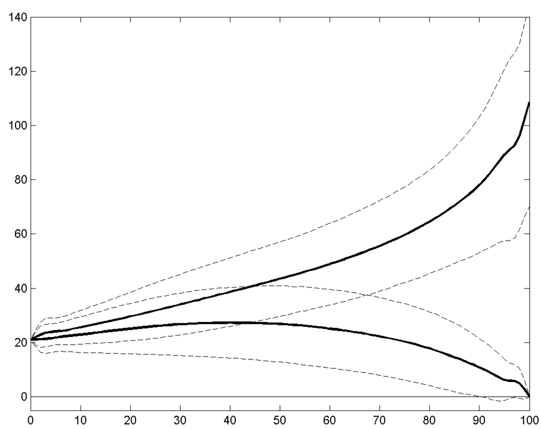

Figure 13: Units over time of winning and losing player in 10000 self-play games of configuration 332.

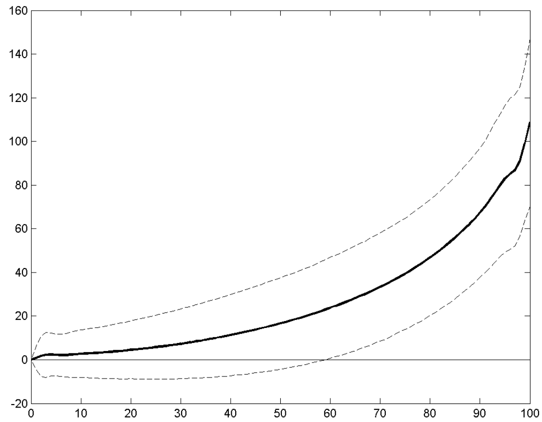

Figure 14: Unit difference over time of the winning and losing player running 332 in selfplay.

6 Conclusions

We have investigated the potential of a Risk AI, based on simple static heuristics. Having implemented three supply heuristics, three attack heuristics and two reinforce heuristics, we compared a total of 18 AI configurations in a round-robin evaluation. While the advantage of advanced supply and attack heuristics was confirmed, resulting in a vast performance gain, reinforcement strategies turned out to be not as resounding. However, the increase in time complexity per ply using the advanced Border Reinforce was immense. It can only be assumed that as the skill level of the AI increases, reinforcement will have a bigger impact on the performance.

The best performing heuristic incorporates the High Chance Attack and BSR Supply strategies. However we can make no clear distinction on whether the Random Reinforce strategy is superior to the simpler Border Reinforce. Due to the aggressive play style of the AI, always attacking with full force when it has more units in its army than the opponent, the nuances of a single unit reinforced elsewhere because of a high BSR are hardly noticeable.

6.1 Future Research

The more elaborate attack strategies increased the AIs performance significantly. However, they do not approach the level of human play, yet. In order to reach a higher level, a variety of heuristics as applied by humans (e.g. focussing on continent completion) can be added. Furthermore, an interconnection of the game phase algorithms be beneficial. Supplying units with foresight of planned attacks may increase performance further.

References

[1] Bouzy, B. and Cazenave, T. (2001). Computer go: An ai oriented survey. *Artificial Intelligence*, Vol. 132, No. 1, pp. 39 – 103.

[2] Campbell, M., Hoane, A., and Hsu, F. (2002). Deep blue. *Artificial Intelligence*, Vol. 134, No. 1-2, pp. 57 – 83.

[3] Gerstmann, J. (1997). Goldeneye 007 review. [Online; accessed 28-June-2010].

[4] Heinze, M., Ortiz-arroyo, D., Larsen, H., and Rodriguez-Henriquez, F. (2005). Fuzzeval: A fuzzy controller-based approach in adaptative learning for backgammon game. *In Mexican International Conference on Artificial Intelligence (MICAI), volume 3789 of LNAI*, pp. 224 – 233, Springer-Verlag.

[5] Joosten, B. (2009). Creating a havannah playing agent.

[6] Lamorisse, A. (1957). La conquête du monde. [Online; acscessed 28-June-2010].

[7] Oka, R. (1994). The real world computing program. *Artificial Intelligence Review*, Vol. 8, No. 2-3, pp. 105 – 111.

[8] Osborne, J. (2003). Markov chains for the risk board game revisited. *Mathematics Magazine*, Vol. 76, pp. 129–135.

[9] Pfeiffer, M. (2004). Reinforcement learning of strategies for settlers of catan. *International Conference on Computer Games: Artificial Intelligence, Design and Education.*

[10] Sanford, K. and Hopper, T. (2009). Videogames and complexity theory: Learning through game play.

[11] Schaeffer, J., Burch, N., Bjrnsson, Y., Kishimoto, A., Müller, M., Lake, R., Lu, P., and Sutphen, S. (2007). Checkers is solved. *Science*, Vol. 317, No. 5844, pp. 1518 – 1522.

[12] St-Pierre, D. L. (2009). Mind game: (axis and allies). M.Sc. thesis, Department of Computer Science, University of Glasgow.

[13] Wikipedia (2010). Risk (game) — wikipedia, the free encyclopedia. [Online; accessed 8-June-2010].

Table 3: Round-Robin evaluation of heuristics measured in winning probability for player 1 in n=1000 games (mean $\pm \frac{1}{2}$ of a 95% confidence interval)

1 \ 2	111	112	121	122	131	132	211	212	221
111	0.64±0.08	0.68±0.07	0.46±0.06	0.51±0.1	0.3±0.06	0.31±0.11	0.39±0.09	0.42±0.09	0.26±0.07
112	0.62±0.1	0.64±0.09	0.47±0.11	0.48±0.09	0.3±0.09	0.28±0.12	0.38±0.13	0.39±0.08	0.25±0.06
121	0.77±0.08	0.82±0.05	0.67±0.09	0.67±0.08	0.51±0.1	0.51±0.08	0.59±0.11	0.61±0.07	0.41±0.08
122	0.78±0.08	0.81±0.04	0.65±0.09	0.63±0.1	0.46±0.16	0.49±0.06	0.56±0.07	0.57±0.09	0.41±0.07
131	0.89±0.05	0.91±0.06	0.74±0.09	0.79±0.07	0.65±0.1	0.65±0.12	0.73±0.11	0.77±0.08	0.6±0.11
132	0.89±0.06	0.92±0.03	0.78±0.08	0.79±0.08	0.66±0.09	0.67±0.05	0.73±0.11	0.75±0.09	0.59±0.1
211	0.84±0.07	0.84±0.06	0.7±0.1	0.71±0.08	0.55±0.11	0.57±0.07	0.65±0.1	0.65±0.09	0.51±0.09
212	0.83±0.1	0.86±0.08	0.72±0.12	0.75±0.11	0.53±0.1	0.55±0.06	0.66±0.11	0.65±0.08	0.48±0.11
221	0.93±0.04	0.94±0.07	0.84±0.08	0.86±0.07	0.73±0.12	0.73±0.07	0.82±0.09	0.82±0.09	0.68±0.08
222	0.93±0.07	0.93±0.04	0.84±0.1	0.87±0.06	0.73±0.09	0.74±0.06	0.8±0.07	0.81±0.07	0.67±0.08
231	0.96±0.04	0.97±0.03	0.9±0.05	0.93±0.04	0.83±0.06	0.81±0.07	0.89±0.06	0.89±0.07	0.8±0.07
232	0.97±0.05	0.96±0.03	0.93±0.05	0.93±0.06	0.85±0.08	0.83±0.07	0.91±0.05	0.9±0.05	0.83±0.08
311	0.82±0.09	0.83±0.09	0.68±0.1	0.73±0.13	0.58±0.08	0.61±0.08	0.66±0.06	0.66±0.09	0.52±0.04
312	0.82±0.09	0.82±0.07	0.7±0.06	0.71±0.1	0.55±0.07	0.59±0.09	0.67±0.08	0.68±0.11	0.49±0.09
321	0.89±0.07	0.93±0.03	0.83±0.06	0.83±0.06	0.74±0.06	0.77±0.1	0.8±0.08	0.8±0.09	0.64±0.08
322	0.9±0.05	0.9±0.05	0.82±0.06	0.84±0.12	0.72±0.09	0.75±0.07	0.76±0.11	0.79±0.04	0.68±0.08
331	0.97±0.03	0.96±0.05	0.91±0.05	0.9±0.05	0.84±0.09	0.86±0.05	0.9±0.06	0.92±0.05	0.81±0.09
332	0.96±0.03	0.97±0.04	0.9±0.03	0.93±0.03	0.85±0.07	0.85±0.09	0.89±0.07	0.89±0.06	0.8±0.07

Table 4: Round-Robin evaluation of heuristics measured in winning probability for player 1 in n=1000 games (mean $\pm \frac{1}{2}$ of a 95% confidence interval)

1 \ 2	222	231	232	311	312	321	322	331	332
111	0.22±0.05	0.13±0.06	0.11±0.06	0.43±0.13	0.43±0.09	0.29±0.09	0.31±0.08	0.16±0.05	0.14±0.05
112	0.23±0.08	0.14±0.06	0.13±0.07	0.45±0.12	0.43±0.09	0.31±0.08	0.29±0.08	0.16±0.06	0.16±0.06
121	0.42±0.05	0.31±0.09	0.27±0.1	0.59±0.06	0.61±0.12	0.47±0.1	0.47±0.09	0.33±0.07	0.33±0.08
122	0.4±0.1	0.29±0.07	0.25±0.08	0.57±0.1	0.57±0.1	0.46±0.15	0.44±0.09	0.31±0.12	0.3±0.05
131	0.61±0.08	0.42±0.09	0.43±0.1	0.71±0.09	0.71±0.09	0.57±0.12	0.62±0.07	0.45±0.12	0.44±0.08
132	0.6±0.07	0.45±0.05	0.43±0.1	0.7±0.11	0.72±0.08	0.55±0.11	0.61±0.09	0.43±0.09	0.42±0.07
211	0.5±0.13	0.35±0.08	0.32±0.1	0.64±0.09	0.63±0.09	0.52±0.11	0.51±0.11	0.35±0.1	0.35±0.1
212	0.51±0.12	0.36±0.11	0.29±0.1	0.62±0.08	0.63±0.1	0.5±0.08	0.51±0.09	0.33±0.09	0.35±0.08
221	0.68±0.11	0.56±0.11	0.49±0.12	0.74±0.08	0.77±0.09	0.67±0.11	0.68±0.1	0.55±0.09	0.52±0.09
222	0.71±0.14	0.54±0.05	0.53±0.11	0.77±0.08	0.76±0.08	0.68±0.1	0.68±0.1	0.54±0.11	0.52±0.1
231	0.81±0.09	0.67±0.09	0.68±0.07	0.85±0.08	0.86±0.08	0.72±0.07	0.78±0.08	0.64±0.07	0.61±0.11
232	0.8±0.08	0.67±0.1	0.7±0.07	0.88±0.08	0.88±0.05	0.79±0.07	0.77±0.09	0.63±0.08	0.64±0.1
311	0.54±0.12	0.4±0.08	0.41±0.1	0.62±0.08	0.67±0.11	0.51±0.07	0.58±0.11	0.37±0.08	0.38±0.12
312	0.54±0.09	0.37±0.08	0.4±0.08	0.62±0.12	0.65±0.1	0.53±0.13	0.56±0.14	0.32±0.11	0.33±0.05
321	0.69±0.09	0.56±0.11	0.58±0.09	0.79±0.04	0.78±0.09	0.71±0.1	0.7±0.07	0.55±0.1	0.51±0.11
322	0.66±0.1	0.56±0.08	0.54±0.11	0.77±0.08	0.77±0.09	0.67±0.07	0.68±0.1	0.48±0.11	0.53±0.05
331	0.81±0.06	0.72±0.08	0.71±0.06	0.89±0.03	0.89±0.06	0.8±0.07	0.8±0.06	0.67±0.06	0.71±0.06
332	0.79±0.09	0.73±0.08	0.72±0.09	0.88±0.08	0.9±0.08	0.78±0.1	0.84±0.06	0.7±0.09	0.69±0.08