Simplified 1-Round Hearthstone Game

Sanyang Liu, CID: 02040950
Yaxuan Wang, CID: 02051576
Ran Zhang, CID: 02031320

November 2023

1 Introduction

In the realm of digital collectible card games, Hearthstone stands as a paragon of strategic complexity and competitive gameplay. Developed by Blizzard Entertainment, this intricately designed game melds elements of chance, skill, and decision-making, creating a dynamic environment that captivates millions of players worldwide. This report endeavors to illuminate the strategic intricacies of Hearthstone through the lens of game theory, in which we shall model strategic interactions among rational decision-makers, and provide a powerful framework for understanding the equilibrium points and optimal strategies within complex gaming environments. By simplifying the structure of Hearthstone into a strategic model, we aim to not only unveil the underlying equilibrium, shedding light on the rational decision-making processes that guide players through the game, but also provide insights that may have broader implications for the application of game theory in diverse strategic contexts.

2 Game Settings

The game being analysed in this report is a simplification of the process of Hearthstone, which only considers actions within one round. It takes two players, Player A (attacker) and Player B (defender).
At the start of the game, both Player A and B are dealt with their minions respectively. Minions are characterised by their Attacks and Healths (e.g. minion m can be represented as $m[m_{atk}, m_{hp}]$), both are integers greater than or equal to 1. The Attacks and Healths of both players' minions are transparent, that is, mutually acknowledged throughout the game.
Player B sets up defense first, where B needs to assign a secret called 'reborn' on one of B's minions which can only be triggered once. Note that A does **NOT** know which minion is chosen by B.
Then, A attacks B, which means A sequentially gives a command to **EACH** of A's minions **ONCE**. There are 2 types of commands: either battle with one of the B's minions, or do nothing. In Hearthstone, if players decide not to attack

opponent's minions, they usually attack opponent's hero instead of doing nothing.
Suppose A commands A's minion a to battle B's minion b, b_{atk} amount of health will be taken away from a_{hp}, and a_{atk} to b_{hp} analogously, that is:

$$a_{hp} \leftarrow a_{hp} - b_{atk},$$

$$b_{hp} \leftarrow b_{hp} - a_{atk}.$$

For instance, if $a[2,3]$ battles with $b[1,4]$, they will end up with $a[2,2]$ and $b[1,2]$. If the Health of a minion is reduced to 0 or below, it is considered to be dead and removed from the game, unless it has 'reborn' cast on it, which leaves its Attack as it was but changes its Health to 1.
In Hearthstone, both Players A and B would aim to maximise their fighting competence after each round. Therefore, we set the payoff of Player A to be the sum of the Attacks of all alive minions of A minus that of Player B at the end of this game after all A's minions have executed commands accordingly. The payoff of Player B shall be the negation of this value.

3 Game Formulation

3.1 Classification of the Game

This simplification of Hearthstone is a 2-players game where players operate sequentially. It is also a zero-sum game since the payoff of Player B is the negation of Player A. Therefore, we could apply the Min-max Theorem to further analyse this game in detail later in Section 4.

3.2 Strategies For Players

Suppose there are m minions for A, and n minions for B. We claim that there are **at most** $\sum_{k=0}^{m} P_k^m n^k$ pure strategies for Player A, where P_k^m is k-permutation of m, and exactly n pure strategies for Player B. The pure strategies for B are trivial since there are n choices for B to cast the 'reborn' card. As for Player A, A can choose k ($k \in \{0, 1, ..., m\}$) minions to attack sequentially, and each minion has n potential battle options. Hence there are $\sum_{k=0}^{m} P_k^m n^k$ strategies in total. However, some of these could be invalid, when the targeted B's minion is already dead before the battle, in which case we assign an extremely negative value to its corresponding payoff, to avoid affecting finding the equilibrium.

Example 1. *Suppose A is dealt with minions Sam[2,3] and Andrew[1,2], B has minions Alice[1,3] and Bob[3,2]. Consider the case where Player B casts the 'reborn' secret on Alice, and Player A commands Sam to battle Bob then Andrew to battle Bob again. What happens is that Bob is already dead before Andrew even gets a chance, so Andrew has 2 choices (battle Alice or not battling at all) instead of 3.*

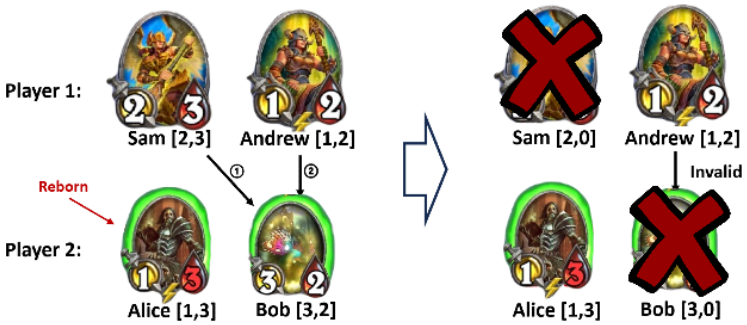

Figure 1: Simple but cute diagram of pure strategy being invalid in Example 1.

Example 1 shows that some of the pure strategies of A might be invalid in practice with $\sum_{k=0}^{m} P_k^m n^k$ being the upper bound of the number of pure strategies of A. We use the convention of a list of tuples to denote each of Player A's pure strategy. Each tuple (a,b) represents one single battle between A's minion with index 'a' and B's minion with index 'b'. Battles are scheduled sequentially according to this list. For instance, in the previous Example 1, Player A's strategy is denoted as [(0,1),(1,1)], with Sam battling Bob first and then Andrew battling Bob again.

3.3 Payoff Equivalent and Dominated Strategies

In a real Hearthstone game, both Player A and B can have at most 7 minions. In this case, there are approximately 4.8×10^9 pure strategies for Player A. All possible outcomes of the game should be simulated to compute the full payoff table. In Example 1 with both players having 2 minions, A has at most 13 pure strategies and B has 2 strategies. We compute the full payoff table by simulating all outcomes:

To determine the equilibria of a game with a large number of strategies, we need to simplify the full payoff table into a **reduced table**. The game with the reduced payoff table is called the **reduced game**. To obtain this result, we notice that there are strictly dominated strategies (e.g. $[(1,1)]$ is strictly dominated by $[(0,0)]$) which we can delete without eliminating any equilibrium by iterative deletion theorem. For payoff equivalent strategies (e.g. $\{[], [(0,0)], [(1,0)]\}$ is a set of strategies with the same payoff), we delete all but one of them so that the equilibrium value remains the same.

By these approaches, we get the reduced payoff table for Example 1 as shown below:

	0	1
[]	-1	-1
[(0,0)]	-1	-1
[(0,1)]	0	-3
[(1,0)]	-1	-1
[(1,1)]	-2	-2

	0	1
[(0,0),(1,0)]	-1	0
[(0,0),(1,1)]	-2	-2
[(0,1),(1,0)]	0	-3
[(0,1),(1,1)]	-1000	-1

	0	1
[(1,0),(0,0)]	-1	0
[(1,0),(0,1)]	0	-3
[(1,1),(0,0)]	-2	-2
[(1,1),(0,1)]	-1	-4

Table 1: The full payoff table.

		Player B	
		b_1[1]	b_2[2]
Player A	a_1[3]	-1	-1
	a_2[4]	-1	0
	a_3[5]	0	-3

Table 2: The reduced payoff table.

From this reduced table, we discover the game is degenerate: if B chooses b_1, A has 3 pure best responses $[(0,1)]$, $[(0,1),(1,0)]$, $[(1,0),(0,1)]$ in a_3 to b_1.

4 Finding Equilibria

This section will now take a deeper look at Example 1 and find the equilibrium by using the upper envelope method and linear programming.

4.1 Upper Envelope Method

Since the reduced game is a zero-sum game, B would choose a min-max strategy to minimise A's maximum gain, equivalent to B's loss. By setting B's mixed strategy $\boldsymbol{\beta} = (q, 1-q)$, A's expected payoffs for each pure strategies are:

$$g(a_1, \boldsymbol{\beta}) = -1,$$
$$g(a_2, \boldsymbol{\beta}) = -q,$$
$$g(a_3, \boldsymbol{\beta}) = -3 - 3q.$$

By using Python code [6], we can plot all $g(a_i, \boldsymbol{\beta})$ against q and identify the upper envelope with its minimum marked out as in Fig. 2:

[1] B chooses to reborn minion 0.
[2] B chooses to reborn minion 1.
[3] $a_1 = \{[], [(0,0)], [(1,0)]\}$.
[4] $a_2 = \{[(0,0),(1,0)], [(1,0),(0,0)]\}$.
[5] $a_3 = \{[(0,1)], [(0,1),(1,0)], [(1,0),(0,1)]\}$.
[6] see find_equi.py in `https://github.com/iivvyy-w/GameTheory-CW1/tree/main/Equi`

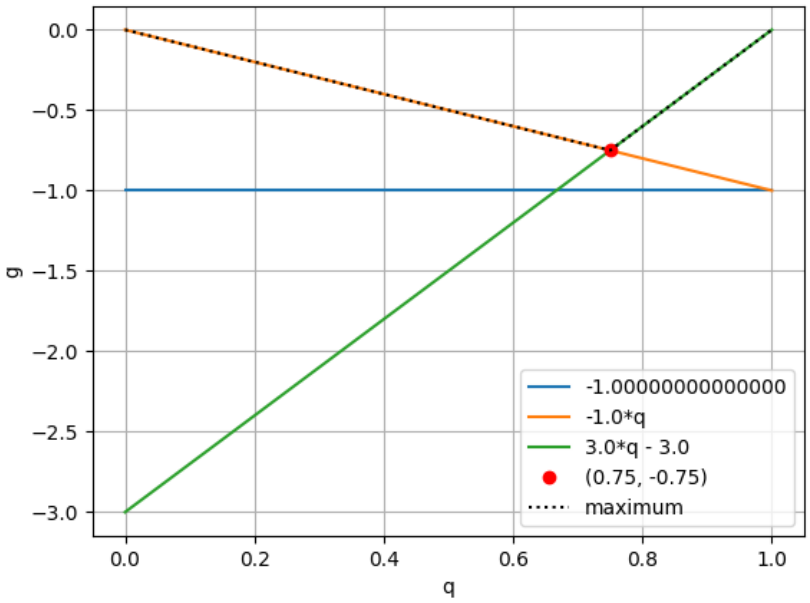

Figure 2: The upper envelope diagram for the reduced game in Example 1.

From this graph, it is clear that $g(a_2, \boldsymbol{\beta})$ and $g(a_3, \boldsymbol{\beta})$'s intersection at $q = \frac{3}{4}$, $\boldsymbol{\beta} = (\frac{3}{4}, \frac{1}{4})$ is the min-max strategy for B. In this case, the expected payoff for A is -0.75. Meanwhile, for A, the preferred strategy would only mix between a_2 and a_3 in the form of $\boldsymbol{\alpha} = (0, p, 1-p)$ because the intersection involves these two pure strategies. The strategy a_1 is discarded given $\boldsymbol{\beta}$ since it provides a payoff that is below -0.75. Now the expected payoffs for B with pure strategies b_1 and b_2 are:

$$g(\boldsymbol{\alpha}, b_1) = -p,$$
$$g(\boldsymbol{\alpha}, b_2) = -3 + 3p.$$

Solving the equation $-p = -3 + 3p$ we obtain $p = \frac{3}{4}$ and $\boldsymbol{\alpha} = (0, \frac{3}{4}, \frac{1}{4})$. The unique equilibrium of this reduced example is

$$(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \left(\left(0, \frac{3}{4}, \frac{1}{4} \right), \left(\frac{3}{4}, \frac{1}{4} \right) \right).$$

4.2 Linear Programming

Suppose now A has m minions and B has n minions. Let $\mathbf{P} \in \mathbb{R}^{s \times n}$ be the reduced payoff matrix, $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_s)$ be A's strategy and $\boldsymbol{\beta} = (\beta_1, \beta_2, ..., \beta_n)$

be B's strategy. By the minimax theorem of Von Neumann (1928) and the relationship between equilibria and max-min/min-max strategies, the problem of finding equilibria can be reformulated to a set of linear programming problems as the following:

$$\begin{aligned} \max_{\boldsymbol{\alpha},v} \quad & v \\ \text{s.t.} \quad & \mathbf{P}^T\boldsymbol{\alpha} \geq v \\ & \sum_{i=1}^{s} \alpha_i = 1 \\ & \alpha_i \geq 0,\ \forall i. \end{aligned} \tag{1}$$

$$\begin{aligned} \min_{\boldsymbol{\beta},v} \quad & v \\ \text{s.t.} \quad & \mathbf{P}\boldsymbol{\beta} \leq v \\ & \sum_{j=1}^{n} \beta_j = 1 \\ & \beta_j \geq 0,\ \forall j. \end{aligned} \tag{2}$$

Then, the optimal solution $(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ is the equilibrium of the reduced game. We use the simplex method from *scipy.optimize.linprog*[7] to solve the reduced mixed equilibrium for Example 1. The result is

$$(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) = \left(\left(0, \frac{3}{4}, \frac{1}{4}\right), \left(\frac{3}{4}, \frac{1}{4}\right) \right).$$

and the value of the game is $v = -0.75$.

4.3 Equilibria for the Original Game

After solving the equilibrium of the reduced game, we need to recover the equilibria of the original game with the full payoff table. In Example 1, we have $\boldsymbol{\alpha}^* = (0, \frac{3}{4}, \frac{1}{4})$ mixing a_2 and a_3, which are sets of payoff equivalent strategies. There are infinitely many ways to split the probability $\frac{3}{4}$ between pure strategies $[(0,0),(1,0)]$ and $[(1,0),(0,0)]$ from set a_2 and arrive at the same equilibrium value. This applies to a_3 as well and therefore the original game has an infinite number of equilibria in the form:

$$\left(p_1,\ \frac{3}{4} - p_1,\ p_2,\ p_3,\ \frac{1}{4} - p_2 - p_3\right)$$

over strategies

$$\left[(0,0),(1,0)\right],\ \left[(1,0),(0,0)\right],\ \left[(0,1)\right],\ \left[(0,1),(1,0)\right],\ \left[(1,0),(0,1)\right]$$

and $p = 0$ on other pure strategies. On the other hand, the equilibrium for Player B remains at $(\frac{3}{4}, \frac{1}{4})$ in the original game since B's strategies are not reduced.

[7] see game.py in `https://github.com/iivvyy-w/GameTheory-CW1/blob/main/Game`

5 Conclusion and Future Endeavors

In this report, we establish the settings for the simplified 1-round game of Hearthstone between 2 players. To quantitatively solve the game, we compute the payoff table and simplify it into a reduced form. The upper envelope method and linear programming are introduced to solve the equilibrium of the reduced game before recovering into an infinite family of equilibria in the original full game.

Now, we connect the strategies within the equilibria found in Section 4.3 with the real situations in Hearthstone. If A plays a_2, both A's minions $a[2,3], a[1,2]$ will survive from focusing on $b[1,3]$ which has lower Attack. If A plays a_3, $a[2,3]$ will die and $a[1,2]$ will survive from focusing on the more threatening $b[3,2]$. Based on these outcomes, we conclude that a_2 represents a relatively conservative strategy while a_3 is more aggressive. Regarding other strategies not in the equilibria of A, such as $[(0,0),(1,1)]$ being strictly dominated by $[(0,0),(1,0)]$, it reflects that in Hearthstone, focusing attacks on one minion is always better than splitting when aiming to control the board (i.e. have more minions alive). In the equilibria of B, b_1 is assigned with a higher probability. This happens in Hearthstone where the defensive player always chooses the conservative strategies that give secrets to the minion with lower Attack and higher Health to guarantee some board control.

In Example 1, the initial difference between the sums of Attacks of A's minions and B's is -1. After playing the mixed strategies in the equilibria, the value of the game is -0.75, indicating that A could benefit from them. In alternative settings of minions for both A and B, the solution might be completely different. Suppose the initial difference in the sums of Attacks of A and B is greater than or equal to 0, signifying that A has an advantage at the beginning. Then, A tends to mix more aggressive strategies (e.g. to kill more B's minions even though B might choose some of them to reborn).

To further choose a single strategy in Player A's payoff-equivalent strategies set, we could design additional criteria to evaluate the strategies. For example, we can choose the strategy that minimises the sum of Healths for B's surviving minions. In Example 1, $[(0,1)]$ and $[(0,1),(1,0)]$ have the same payoff. Because the later one cause more damage to B's minions, we should choose $[(0,1),(1,0)]$.

In a real Hearthstone game with multiple rounds, it is difficult to simulate all outcomes since the number of strategies grows exponentially as the number of rounds increases. Although optimised strategies in each round do not guarantee an optimised winning rate, our analysis on the simplified 1-round game is still extremely important and useful in certain decks and at endgames.