

## Regular Paper

# Mad Science is Provably Hard: Puzzles in Hearthstone's Boomsday Lab are NP-hard

MICHAEL HOFFMANN<sup>1,a)</sup> JAYSON LYNCH<sup>2,b)</sup> ANDREW WINSLOW<sup>3,c)</sup>

Received: December 31, 2019, Accepted: September 10, 2020

**Abstract:** We consider the computational complexity of winning this turn (mate-in-1 or “finding lethal”) in Hearthstone as well as several other single turn puzzle types introduced in the Boomsday Lab expansion. We consider three natural generalizations of Hearthstone (in which hand size, board size, and deck size scale) and prove the various puzzle types in each generalization NP-hard.

**Keywords:** computational complexity, NP, video games

## 1. Introduction

Collectible card games (CCGs) involve buying some cards, picking a subset or *deck* of those cards, and playing against someone else who separately picked their own deck. Collectible card games are fun and popular! In 2017 Hearthstone reported over 70 million registered players [1].

With the release of the Hearthstone set, The Boomsday Project, Blizzard introduced a number of puzzles related to the game. All of the puzzles require changing the board to match some desired state in a single turn. The puzzle types are: “Lethal”, “Mirror”, “Board Clear”, and “Survival”.

In Lethal, the player must reduce their opponents health to zero. In Mirror, the player must make both sides of the board exactly the same. This means both boards must have minions of the same type in the same order, and any damage and status effects on those minions must be the same. In Board Clear, the player must ensure there are no more minions on the board, usually by destroying ones already in play. In Survival, the player must return their character to full health.

We will focus on the problem of finding lethal (analogous to “mate-in-1” from Chess) since it is the most common puzzle type; however, our proofs can be adapted to show NP-hardness for the other three puzzle types. Even before Blizzard released The Boomsday Project, there were numerous collections of challenging “lethal puzzles” available online, e.g., on websites such as [hearthstonepuzzles.net](http://hearthstonepuzzles.net), [hsdeck.com](http://hsdeck.com), and [reddit.com/r/HearthPuzzle/](https://www.reddit.com/r/HearthPuzzle/) (also see Ref. [2]). The frequent difficulty of such lethal puzzles leads to the consideration of the formal computational complexity of such puzzles - can these prob-

lems be shown to be computationally intractable under standard complexity-theoretic assumptions?

The Boomsday Project Labs allow for carefully designed situations with non-random decks, cards with reduced costs, complex board states, and even cards that are not normally available in the game. These are all useful tools for designing both puzzles and hardness proofs, however, we are also interested in the problem of finding lethal in which it would occur within a game itself. For this reason, we restrict our proofs to using cards normally available to players and we give a description of how the board state used in the reduction could have been created in a game of Hearthstone. However, for one proof, we do extensively use the puzzle property of the player knowing the contents and order of their deck.

**Related work.** The body of work on the computational complexity of games and puzzles has become quite expansive, however, only a small amount of it considers the mate-in-one question. Although deciding who will win in a two player game is frequently PSPACE- or EXP-complete<sup>\*1</sup>, the problem of ‘mate-in-1’ or ‘finding lethal’ is often far less computationally complex because many games have only a polynomial number of moves on any given turn and evaluating whether the new state of the board results in a win is computationally easy. Two examples where this problem is interesting are Conway’s Phutball for which mate-in-1 is NP-complete [7] and Checkers for which mate-in-1 is in P [8].

Although a fair amount of academic study has gone into collectible card games [10], [11], less is known about their computational complexity. Magic: The Gathering, perhaps the most well-know CCG, is one example that has been analyzed. A recent paper shows that deciding who wins in a game of Magic is undecidable [5], [6]. It has been shown that simply deciding if a move is legal is coNP-complete [4]. Mate-in-1 and Mate-in-2 for another CCG, Android Netrunner, was shown to be weakly NP-hard [3].

<sup>1</sup> ETH Zurich, Rämistrasse 101, 8092 Zürich, Switzerland

<sup>2</sup> University of Waterloo, 200 University Ave W, Waterloo, ON N2L 3G1, Canada

<sup>3</sup> UT Rio Grande Valley, 1201 W University Dr, Edinburg, TX 78539

<sup>a)</sup> [hoffmann@inf.ethz.ch](mailto:hoffmann@inf.ethz.ch)

<sup>b)</sup> [jayson.lynn@uwaterloo.ca](mailto:jayson.lynn@uwaterloo.ca)

<sup>c)</sup> [andrew.winslow@utrgv.edu](mailto:andrew.winslow@utrgv.edu)

<sup>\*1</sup> See Ref. [9] for several examples.

## 2. Hearthstone

Hearthstone is a popular online CCG made by Blizzard and themed after World of Warcraft. Players are able to purchase virtual cards with which to construct their decks. The game consists of players taking actions on their own turns, including casting spells, summoning and attacking with minions, and controlling heroes with the objective of reducing the enemy hero's health to zero.

New cards come out regularly in sets. Only the base set and most recent sets are allowed in the Standard format. However, all cards are allowed in the Wild format. Blizzard also occasionally changes cards to adjust game-play. Configurations of Hearthstone game considered here take place in the Wild format at the release of The Boomsday Project.<sup>\*2</sup> There are also Solo Adventures which can have unique rules and cards. The Boomsday Project Lab is an example of a Solo Adventure which includes custom rules and cards specifically to facilitate mate-in-1 like puzzles for Hearthstone.

### 2.1 Rules Overview

Here we present a very brief summary of some of the basic rules in Hearthstone which are relevant to the proofs. The cards themselves often have text which specifies additional abilities and rules that may make game-play differ from the typical behavior described in this section.

In Hearthstone, players use *mana* to pay for cards and abilities. Each turn they gain a *mana crystal*, up to 10, and then gain mana equal to their mana crystals. Players also have a *hero* which has *health* and a *hero power*. If a player's hero is ever reduced to zero or less health, that player loses. Hero powers are abilities that cost 2 mana and can be used once a turn.

There are four main types of cards which can be played: *minions*, *spells*, *weapons*, and *traps*.

**Spells** typically have a one time effect, such as healing, drawing cards, or doing damage, and are discarded after they are played. Minions are played onto the *Battlefield*. Each player cannot have more than 7 minion on the battlefield at a time.

**Minions** have *attack* and *health* which are both non-negative integers. If a minion's health is reduced to zero or lower, it is removed from the battlefield. Minions and Heroes can attack once a turn if they have a positive attack value. When a minion (or hero) attacks, the player chooses an opponent's minion or hero as the target. If the opponent has a minion with *taunt* then a minion with taunt must be selected as the target for any attacks. The attacking and attacked cards simultaneously deal damage to each other equal to their attack value. When a card takes damage its health is reduced by that amount. Minions can also have *abilities* which effect game-play while they are on the battlefield. They can also have *battlecry* or *deathrattle* which triggers an effect (similar to a spell) when the minion is played or dies respectively. Minions without card text (abilities, battlecry, deathrattle) are called *vanilla* minions. For example, Bloodfen Raptor and Pit Fighter are vanilla minions.

**Weapons** give a hero an attack value. They also have a *durability* which is reduced by 1 every time that hero attacks. If a weapon reaches zero durability it is destroyed. If a player plays a weapon while they already have one in play, the original weapon is destroyed and replaced by a new one.

**Traps** are not used in any of these proofs as their effects trigger on the opponents turn in response to some action taken by the opponent.

Not all cards can be included in a deck, most notably each hero has a class and decks can only contain *neutral* cards or cards from their class. Decks must normally contain exactly 30 cards, no more than 1 copy of any legendary card and no more than 2 copies of any other card. However, during the game cards may add or remove cards from players' decks and the above constraints only apply while building decks, not in the middle of a game. Games generally begin with cards in a deck being randomly shuffled. However, in The Boomsday Lab puzzles, players might have pre-determined decks with a specific card ordering.

### 2.2 Generalizing the Game

In the game of Hearthstone as available to players, the time and complexity of games are limited in several ways. For instance, each turn has a time limit of 75 seconds (plus animations), games are limited to 89 turns, decks are limited to 60 cards, hands to 10 cards, and boards to 14 minions. The Boomsday Lab puzzles do not have a time limit. In comparison, the computational complexity of problems is considered as the problem size grows to infinity.

In order to formalize finding lethal into a problem that can be analyzed, we consider generalizing the game in one of several ways, enabling puzzles of arbitrarily large size. For Hearthstone, we see three natural generalizations: arbitrarily large boards, hands, and decks. The game configurations obey all rules of Hearthstone, except that turns may take arbitrarily long, they may require arbitrarily many turns (and thus played cards and card copies) to reach, and will be either:

- *Board-scaled*: the board may have arbitrarily many minions (beyond the 7 permitted in the game).
- *Hand-scaled*: players' hands may have arbitrarily many cards (beyond the 10 permitted in the game).
- *Deck-scaled*: players' decks may have arbitrarily many cards (beyond the 60 permitted in the game).

Unless otherwise stated, the configurations all occur at turns with 10 mana. The *lethal problem* of a configuration of Hearthstone is as follows: can the current player reduce their opponents health to zero this turn?

For the Deck-scaled versions of the game we make a further alteration: each player knows the entire content of her deck, including card ordering (i.e., each player has *perfect information* about her deck). Normally cards are drawn uniformly at random from those in the deck. It would be very interesting to know whether the Deck-scaled version of the game remains hard (or is harder) with random draw.

### 2.3 A Preliminary Combo

The reductions below involve large numbers of specific cards.

<sup>\*2</sup> For further details of the rules of Hearthstone, see <http://www.hearthstone.com>.

Although the Boomsday Lab puzzles allow us the freedom to design the precise board state, we show a way of generating cards in a game. Here we describe a sequence of players yielding arbitrarily many of desired sets of cards (with the caveat of working with very low probability). In particular, many of the cards generate random cards of a given type. In these cases, we assume they happen to generate exactly the cards we desire.

**The setup.** On the prior turn, the opponent plays a Millhouse Manastorm, causing all spells cast this turn to cost 0 mana. Our board contains a Brann Bronzebeard, which causes our battlecries to trigger twice, and another vanilla minion. Playing Cabalist's Tome adds three random Mage spells to your hand. In this case, we assume it generates three copies of Unstable Portal which adds a random minion to your hand and reduces its mana cost by 3. Playing two of the three Unstable Portals generates a Spellslinger and Void Terror.

**A cyclic play sequence.** Since Spellslinger and Void Terror cards have mana cost 3, when obtained from Unstable Portal they have cost 0. Spellslinger has a battlecry which adds a random spell to each player's hand. Playing Spellslinger yields a Cabalist's Tome and a second spell, due to Brann Bronzebeard causing its battlecry to trigger twice. Void Terror has a battlecry which destroys adjacent minions. Playing Void Terror between the played Spellslinger and vanilla minion destroys them, recovering space on the board.

In our hand is now an arbitrary spell (generated from Spellslinger), an arbitrary minion (generated from Unstable Portal), and a new copy of Cabalist's Tome. No mana has been spent, so this process can be repeated, obtaining a new arbitrary spell and a minion at each iteration.

**Playing cards.** If we need to actually play these cards, we can generate Innervates to gain mana and alternate between generating Void Terrors and some other minion that costs 3 mana or less. This combo requires 4 free minion slots (out of a maximum of 7) and 4 free hand slots, leaving 3 board slots and 6 hand slots for other aspects of later constructions.

**Obtaining the setup.** If neither hero is a mage, we could have generated the initial Cabalist's Tome by playing Yogg-Saron, Hope's End, after having cast at least one spell during the game, and having it cast an Unstable Portal that generated a Spellslinger.

### 3. NP-hardness for Hearthstone

Here we give three different proofs for the three different generalized versions of Hearthstone. A large battlefield size is considered in Section 3.1, a large hand size is considered in Section 3.2, and a large deck size is considered in Section 3.3. The reductions are from 3-PARTITION and 2-PARTITION and use similar ideas. In all cases the opponent will have minions with taunt which we must destroy to be able to attack the opponent's hero. These will encode target sums and the attack values of our minions will encode our set of numbers which we want to partition. At this high level the reductions are very simple; the majority of the complication comes from properly constructing the needed attack and health values with cards in the game and a limited hand/deck/board space. It is also important to note that because some cards apply multiplicative factors to attack and health our

2-PARTITION reductions actually also yield strong NP-hardness for finding lethal in Hearthstone.

#### 3.1 Hardness of Board-Scaled Lethal

**Theorem 3.1.** *The lethal problem for board-scaled instances of Hearthstone is NP-hard.*

*Proof.* The reduction is from 3-PARTITION. Let  $A = \{a_1, a_2, \dots, a_{3n}\}$  be the input multiset of positive integers that sum to  $S$ . The goal is to partition  $A$  into  $n$  parts that each sum to  $S/n$ . The game state is as follows, and is described from the first-person perspective of the current player's turn.

**Your hero, hand, deck, and board.** Your hero is Anduin (Priest) with 1 health and Light's Justice equipped (from Blingtron 3000). Light's Justice is a weapon allowing the hero to attack for 1 damage. Your hand and deck are empty. Your board consists of  $3n$  vanilla minions with attack values  $4a_1, 4a_2, \dots, 4a_{3n}$  and 3 health each.

**The opponent's hero, hand, deck, and board.** The opponent's hero is Valeera Sanguinar (Rogue) with 1 health. The opponent's board consists of  $n$  minions, each with taunt, 5 attack,  $4S/n$  health, and no other special text.

**Lethal strategies.** To win, you must kill all  $n$  of the opponent's minions, then do 1 damage to the opponent. Attacking any minion with your weapon causes you to die. So the opponent's minions must be killed with your minions. The total attack of your minions is exactly  $4S$ , so to kill all enemy minions, each minion must not overdamage, i.e. each minion must do exactly its attack damage to some opponent minion.

Thus all of your minions must attack the enemy minions in a way that corresponds exactly to partitioning  $a_1, a_2, \dots, a_{3n}$  (your minions' attack values) into  $n$  groups of  $S/n$  each (your opponent's minions' health values). Such partitions are exactly the solutions to the 3-PARTITION instance.

**Achieving your board state.** Your board is constructed by using the preliminary combo to generate  $3n$  copies of Duskboar and  $S - n$  copies of Blessing of Kings. Duskboars have 4 attack and 1 health and the Blessing of Kings each add 4 attack and 4 health. These are cast on the Duskboars such that their attacks correspond to the values  $4a_1, 4a_2, \dots, 4a_{3n}$ . The extra minions needed for the combo are removed later via combo-generated Assassins.

**Achieving the opponent's board state.** The opponent's board can be constructed using the preliminary combo from Section 2.3, where the combo generates and the opponent plays  $n$  Evil Hecklers and  $S - n$  Blessing of Kings. Evil Hecklers have 5 attack, 4 health, and taunt. Blessing of Kings gives a minion +4 attack and +4 health. These can be distributed to construct the opponents board which contains  $n$  minions with taunt, each with  $4S/n + 1$  attack, and  $4S/n$  health.  $\square$

#### 3.2 Hardness of Hand-Scaled Lethal

**Theorem 3.2.** *The lethal problem for hand-scaled instances of Hearthstone is weakly NP-hard.*

*Proof.* We reduce from 2-PARTITION. Let  $A = \{a_1, a_2, \dots, a_n\}$  be the input set of (exponentially large in  $n$ ) integers that sum to  $S$ . The goal is to partition the integers into two sets that each sum to  $S/2$ .

**Your hero, hand, deck, and board.** Your hero is Jaina (Mage) with 1 health and a Light's Justice (created by an earlier Blingtron 3000). Your hand consists of  $n$  copies of Bolvar Fordragon with attack equal to  $b_i = 4a_i - 2$ ,  $n$  copies of Charge,  $6n$  copies of Innervate (to pay for the Bolvar Fordragon and Charge). Your board is empty. Charge allows minions to attack the turn they come into play and Innervate generates additional mana.

**Your opponent's hero and board.** Your opponent's hero is Uther (Paladin) with 2 health. Your opponents board consists of 2 vanilla minions with taunt, at least 7 attack,  $4S$  health (buffed via Blessing of Kings and Blessed Champion).

**Lethal strategies.** In order to win this turn (or at all), you must kill both large minions of the opponent, then do 2 total damage by attacking and using your hero power. To win, we must play the Bolvar Fordragons, give them Charge, and attack the minions with taunt such that they both die. Since the total attack of all of the Bolvar Fordragons and Charges equals the health of the two minions, we cannot succeed unless we cast Charge on every Bolvar Fordragon exactly once. Thus we must allocate the Bolvar Fordragons between the two enemy minions such that their attack adds up exactly to that of the health of the minions.

This partition is exactly the solution to the given 2-PARTITION instance. The scaling of the attack and health is to deal with the bonus to attack given by Charge and the possibility of  $a_i = 1$ .

**Achieving your board state.** In general we will use the infinite combo given earlier in this section to obtain the necessary cards. First, we generate all of the cards needed except for the Bolvar Fordragons, as well as some additional cards specified shortly. We need to create the Bolvar Fordragons and between them cause many minions to die until we reach the correct values for our set being partitioned.

Assume the  $b_i$  are ordered from largest to smallest. We will fill our hand with  $n$  copies of Unstable Portal,  $b_1$  copies of Stonetusk Boar,  $b_1/2$  copies of Innervate, and 1 Bestial Wrath. The Bestial Wrath is cast on an opponent's beast, say Bloodfen Raptor, to ensure we have a way of killing all of the Stonetusk Boars. We play an Unstable Portal which summons a Bolvar Fordragon. We then play a copy of Stonetusk Boar and attack the opponent's Bloodfen Raptor  $b_1 - b_2$  times. This causes the Bolvar Fordragons to gain  $b_1 - b_2$  attack.

We then cast another Unstable Portal to obtain and play another Bolvar Fordragon, then cast  $b_2 - b_3$  copies of Stonetusk Boar, and attack with them. Both Bolvar Fordragons gain  $b_2 - b_3$  attack from the minions that die. We repeat this pattern until we have the  $n$ th copy of Bolvar Fordragon and we attack with  $b_n - 1$  copies of Stonetusk Boar. Since Bolvar Fordragon starts with 1 attack, we've now caused them to gain the exact amount of attack to have one equal to each of our 3-PARTITION values.  $\square$

### 3.3 Hardness of Deck-Scaled Lethal

**Theorem 3.3.** *The lethal problem for deck-scaled instances of Hearthstone is NP-hard.*

*Proof.* We reduce from 2-PARTITION. Let  $A = \{a_1, a_2, \dots, a_n\}$  be the input set of (exponentially large in  $n$ ) integers that sum to  $S$ . The goal is to partition the integers into two sets that each sum to  $S/2$ .

**Your hero, hand, and board.** Your champion is Uther (Paladin). Your hand consists of 8 Pit Fighters and 1 Solemn Vigil. Your board consists of 4 frozen vanilla minions.

**Your opponent's hero, hand, deck, and board.** Your opponent's hero is Uther (Paladin). Your opponent's board consists of 2 Target Dummies buffed to  $S/2$  health (via Blessing of Kings and Blessed Champion to get a large attack and then swapping attack and health with Crazy Alchemist) and Millhouse Manastorm. Your opponent played Millhouse Manastorm last turn, so all spells you cast this turn are free. Your opponent's deck consists of 1 Bluegill Warrior.

**Your deck.** Your deck contains the following cards: Solemn Vigil (SV), Blessing of Kings (BoK), Blessed Champion (BC), Pit Fighter (PF), and Anyfin Can Happen (ACH). A Bluegill Warrior (BW) had been played earlier and died. No other murlocs have died in this game, thus ACH will always summon BW. The sequence of the first 4 cards, called the *setup sequence*, is SV, PF, SV, PF.

For an integer  $n (= b_1b_2 \dots b_k$  in binary) with  $b_{k-1}b_k = 00$ , define the *encoding sequence* of  $n$  as a polynomial length sequence of left *bit shifts* (equivalent to multiplying by 2) and *increments* by 8 (equivalent to incrementing by 100 in binary) to obtain  $n$ . We will be interested in using this to encode the input sequence to the 2-PARTITION instance.

Define the *integer card sequence* of  $n$  to be the sequence of cards obtained by replacing each bit shift and increment in an encoding sequence by BC and BoK, respectively, appending ACH and BC to the beginning of the sequence, and then replacing each card with a SV followed by the card. For example, the following sequence of cards encodes 1110100100: [SV, ACH, SV, BC, SV, BC, SV, BoK, SV, BC, SV, BoK, SV, BC, SV, BC, SV, BoK, SV, BC, SV, BC, SV, BC, SV, BoK]. The complete deck consists of the setup sequence, followed by the integer card sequence for each  $a_i$ , followed by an ACH.

**Lethal strategies.** In order to win this turn, both Target Dummies must be killed and 1 damage dealt to the opponent. To have any possibility of lethal, damage must be done by obtaining ACH via draw. Thus both Pit Fighters must be drawn and played, leaving only one open slot to play minions.

Since you've spent all 10 mana, nothing else with positive mana cost (namely the other Pit Fighters in your hand) can be played. Thus your hand has between 8 and 10 cards for the remainder of the turn. Moreover, the interleaving of SV with other spells implies that any attempt to play SV with a hand of 9 cards causes one of the two drawn cards, namely another SV, to be burnt, preventing further card draw.

At the end of each integer card sequence, to continue to draw into the deck without burning an ACH, we need a minion to die and trigger Cult Master to draw an additional card. Since Bluegill Warriors are the only minions that can attack, each must be killed at the end of the integer card sequence in which it was drawn and thus only the buffs from one integer card sequence can be applied to a given Bluegill Warrior.

Since these buffs on the BW yield a total attack value of  $S$ , if any buff is not played on a BW then the BWs will have a total attack less than  $S$  and thus the two Target Dummies cannot be



killed. Thus any lethal turn involves each BW being buffed with exactly the buffs in the integer card sequence they belong to, and attacking one of the two large minions.

So any lethal play sequence consists of buffing BW to attack values corresponding to  $a_1, a_2, \dots, a_n$  and attacking each into one of two opponent Target Dummies, followed by a final ACH being drawn and attacking the opponent with the final BW. Since buffed Bluegill Warriors have  $S$  total attack damage, killing both Target Dummies requires partitioning the attack values of the buffed BW into two subsets, each of  $S/2$  total attack value.

**Strong hardness.** Note that unlike the prior reduction from 2-PARTITION, this reduction establishes strong NP-hardness. This is due to the encoding of exponentially large numbers as minion attack and health values in a polynomial number of cards, one of which (Blessed Champion) doubles minion attack. because the input to the problem is given by a polynomial number of cards and prior plays in the game. The exponentially large minion sizes are efficiently encoded by a series of plays of BC and BoK.  $\square$

### 3.4 Adapting to Other Puzzle Types

In addition to finding “Lethal”, our proofs can be adapted to show the other three puzzle types introduced in the Boomsday Lab are also NP-hard. In general these proofs require us to carefully construct appropriate minions to remove powerful minions with taunt and allow us to attack the enemy hero. We will show that there are minions which could be chosen that we can attack instead of the enemy hero to accomplish the other goals.

#### 3.4.1 Survival

In this puzzle, the player’s objective is to restore their hero to full, normally 30, health. To adapt to this case, in each reduction we give our opponent a Mistress of Mixtures which has two attack, two health, and upon dying restores 4 health to each hero. We make sure the Mistress of Mixtures only has one health remaining, perhaps by previously damaging it with a Stonetusk Boar. We have your character’s health set to 28, so even if you must attack the Mistress of Mixtures to kill it you will take 2 damage but then regain 4 health returning you to full.

#### 3.4.2 Board Clear

In this puzzle, the player’s objective is to kill all minions on the board. This is achieved in our board scaled reduction. In the other two reductions either your opponent has a leftover Bloodfen Raptor or you have leftover frozen minions (which we will assume are also Bloodfen Raptors for the sake of simplicity). To fix this issue, we give your opponent an Explosive Sheep which is a 1 attack, 1 health minion that does 2 damage to all other minions when it dies. Instead of attacking your opponent once you’ve gotten rid of the taunt minions in the way, attack the Explosive Sheep whose damage will kill off the remaining unwanted Bloodfen Raptors. In this case we will set your hero’s health to 2, so it is greater than the damage done by attacking the Explosive Sheep.

#### 3.4.3 Mirror

In this puzzle the player must make both sides of the board identical. We note that if the player clears the board, then both sides will be identically empty, fulfilling the technical requirement if not the spirit of the puzzle. We use the same augmenta-

tion as we did with the board clear goal and note that the player has no way of playing the same minions as their opponent and thus cannot fulfill the mirror requirement if any of their opponent’s minions are on the board. Thus the only solution is one that involves clearing the board.

## 4. Open Problems

Given the ability to generate an arbitrary number of Hearthstone cards on a single turn, it is not clear Hearthstone puzzles are in NP, or even PSPACE. Obtaining upper bounds on the complexity is a clear open question. Also, for these puzzles, we assume perfect information. If we exploit imperfect information and randomness, even with a bounded number of plays we might suspect the problem is PSPACE-hard.

Hearthstone puzzles also occur on a single turn, thus eliminating the 2-player aspect of the game. What is the complexity of deciding if a player in a game of Hearthstone has a forced win?

Although Magic and Hearthstone are likely the two most famous CCG’s at the moment, there have been a number of other such games in the past. It would be interesting to see other examples studied, as well as a general framework for understanding when such games are computationally intractable.

Finally, we have yet to see a formalization of the problem of deck construction and the meta-game involved in most competitive CCGs. Since deck building is such an important and integral part of many of these games, it would be interesting to have a more formal understanding of the questions and process involved.

**Acknowledgments** We wish to thank Jeffrey Bosboom for significant feedback and discussion about this paper, as well as LaTeX expertise. We would also like to thank the other participants and especially the organizers (Erik Demaine and Godfried Toussaint) of the Bellairs Research Institute Winter Workshop on Computational Geometry 2015.

## References

- [1] Activision Blizzard Announces Better-Than-Expected and Record First-Quarter 2017 Financial Results, available from (<https://www.businesswire.com/news/home/20170504006637/en/Activision-Blizzard-Announces-Better-Than-Expected-Record-First-Quarter-2017>).
- [2] Some Hearthstone Players Challenge Themselves To Spot Lethal Combos, available from (<https://compete.kotaku.com/hearthstone-brain-teaser-puzzles-help-keep-players-shar-1792530321>).
- [3] Bosboom, J. and Hoffmann, M.: Netrunner Mate-in-1 or-2 is Weakly NP-Hard, arXiv preprint arXiv:1710.05121 (2017).
- [4] Chatterjee, K. and Ibsen-Jensen, R.: The Complexity of Deciding Legality of a Single Step of Magic: The Gathering, *ECAI*, pp.1432–1439 (2016).
- [5] Churchill, A.: Magic: The Gathering is Turing Complete, available from (<http://www.toothyat.net/hologram/Turing/>).
- [6] Churchill, A., Biderman, S. and Herrick, A.: Magic: The Gathering is Turing Complete (2019).
- [7] Demaine, E.D., Demaine, M.L. and Eppstein, D.: Phutball Endgames are Hard, *More Games of No Chance*, Nowakowski, R.J. (Ed.), pp.351–360, Cambridge University Press (2002). Collection of papers from the MSRI Combinatorial Game Theory Research Workshop, Berkeley, California (2000).
- [8] Fraenkel, A.S., Garey, M.R., Johnson, D.S., Schaefer, T. and Yesha, Y.: The complexity of checkers on an  $N \times N$  board, *19th Annual Symposium on Foundations of Computer Science (SFCS 1978)*, pp.55–64 (online), DOI: 10.1109/SFCS.1978.36 (1978).
- [9] Hearn, R.A. and Demaine, E.D.: *Games, Puzzles, and Computation*, CRC Press (2009).
- [10] Ward, C.D. and Cowling, P.I.: Monte Carlo search applied to card selection in Magic: The Gathering, *2009 IEEE Symposium on Com-*

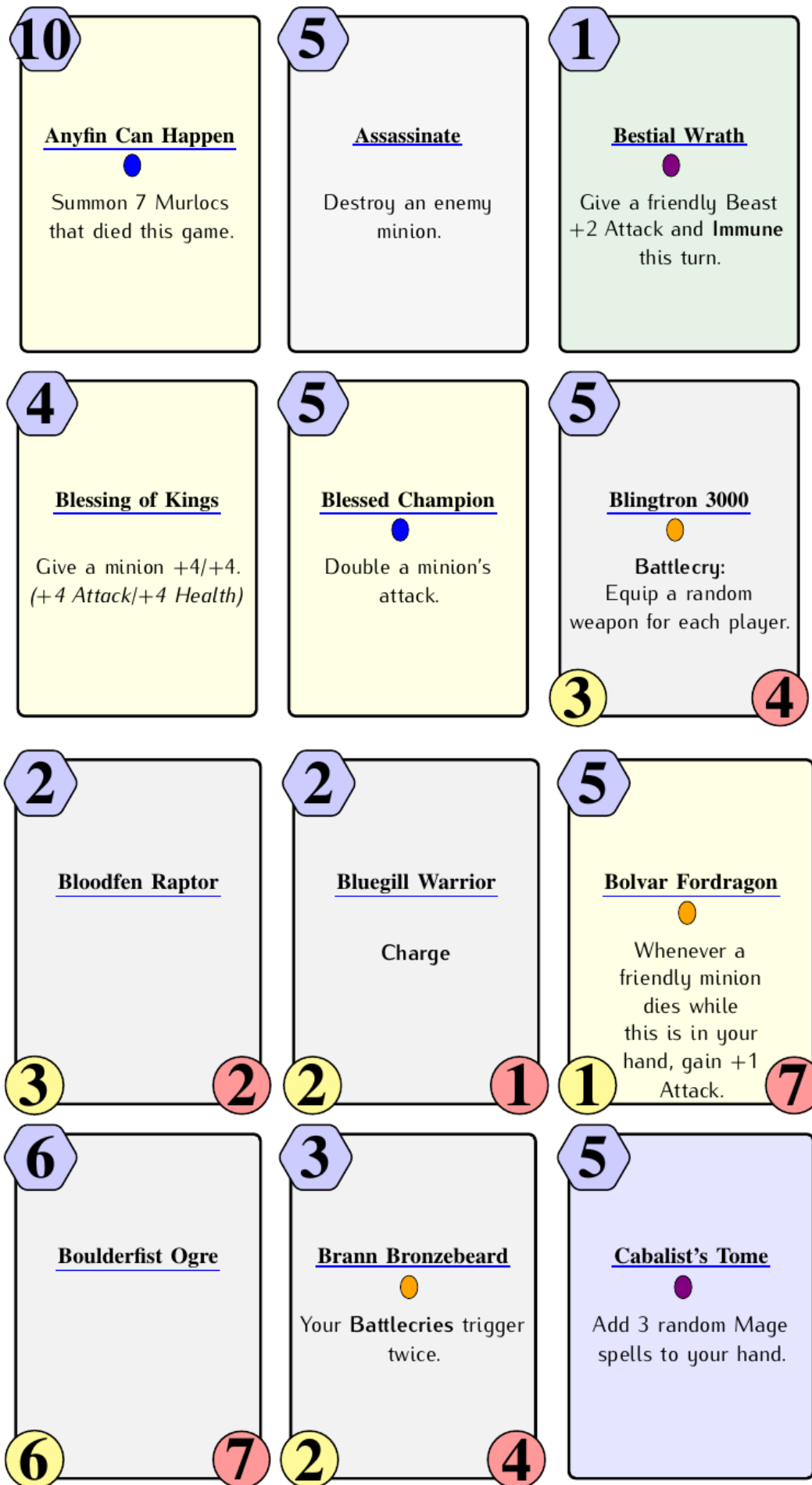
*putational Intelligence and Games*, pp.9–16 (2009).

- [11] Zhang, S. and Buro, M.: Improving Hearthstone AI by learning high-level rollout policies and bucketing chance node events, *2017 IEEE Conference on Computational Intelligence and Games (CIG)*, pp.309–316, IEEE (2017).

## Appendix

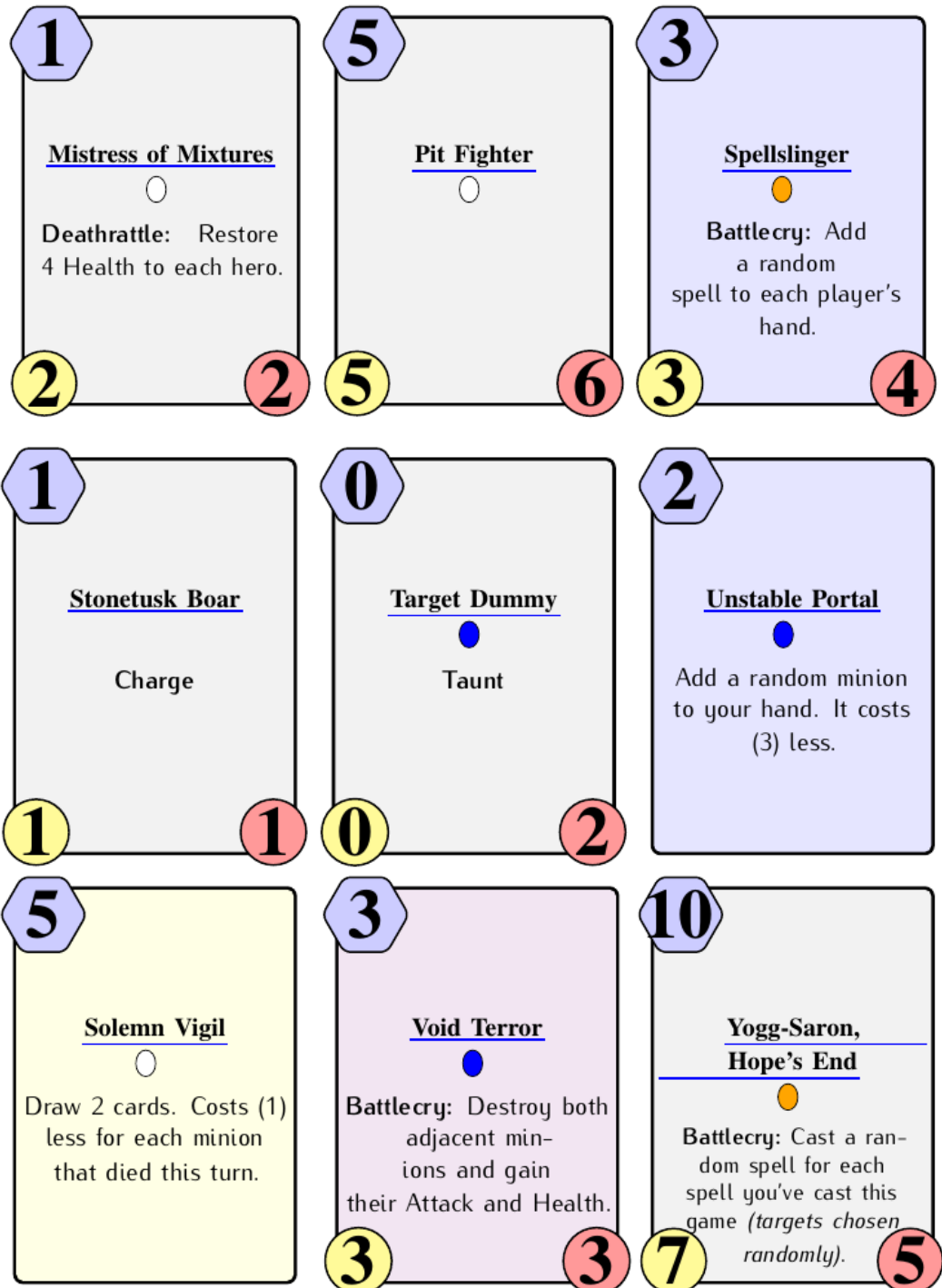
### A.1 Hearthstone Card Details

Below is a list of cards used in the constructions and their details. The name of the card is the top line of text and is underlined. Below that is a colored oval denoting the rarity of the card followed by any ability and rules text of the card. The mana cost is given in the upper left hand corner. The attack of the card (for minions and weapons) is given in the lower left. The number in the lower right is the durability for weapons or the health of a minion.











**Michael Hoffmann** is a professor in the department of computer science at ETH Zürich and is on the CGAL editorial board. Dr. Hoffmann's research interests include computational geometry and graph algorithms.



**Jayson Lynch** received a Ph.D. from MIT 2020 for work on the computational complexity of motion planning problems under Erik Demaine. Jayson is now a Postdoctoral Researcher at the University of Waterloo continuing to do work on computational geometry and origami, graph algorithms, resource efficient computing, and the computational complexity of games and puzzles.



**Andrew Winslow** is a software engineer at Google and former faculty at the University of Texas Rio Grande Valley. His research focuses on computational geometry and algorithmic self-assembly.