# Automated Playtesting in Collectible Card Games using Evolutionary Algorithms: a Case Study in HearthStone

**Article** *in* Knowledge-Based Systems · April 2018

DOI: 10.1016/j.knosys.2018.04.030



**5 authors**, including:

Pablo García-Sánchez
University of Granada
**155** PUBLICATIONS **1,271** CITATIONS


Alberto Paolo Tonda
French National Institute for Agriculture, Food, and Environment (INRAE)
**204** PUBLICATIONS **1,737** CITATIONS


Antonio Mora
University of Granada
**253** PUBLICATIONS **2,323** CITATIONS


Giovanni Squillero
Polytechnic University of Turin
**318** PUBLICATIONS **3,485** CITATIONS

# Automated Playtesting in Collectible Card Games using Evolutionary Algorithms: a Case Study in HearthStone

P. García-Sánchez[a], Alberto Tonda[b], Antonio M. Mora[c], Giovanni Squillero[d], Juan J. Merelo[e]

[a]*Dept. of Computer Engineering. University of Cádiz, Spain*
[b]*UMR 782 GMPA, INRA, Thiverval-Grignon, France*
[c]*Dept. Computer Sciences and Technology, Universidad Internacional de La Rioja, Spain*
[d]*Politecnico di Torino, Italy*
[e]*Dept. of Computer Architecture and Computer Technology, University of Granada, Spain*

---

**Abstract**

Collectible card games have been among the most popular and profitable products of the entertainment industry since the early days of *Magic: The Gathering*™ in the nineties. Digital versions have also appeared, with *HearthStone: Heroes of WarCraft*™ being one of the most popular. In Hearthstone, every player can play as a hero, from a set of nine, and build his/her deck before the game from a big pool of available cards, including both neutral and hero-specific cards. This kind of games offers several challenges for researchers in artificial intelligence since they involve hidden information, unpredictable behaviour, and a large and rugged search space. Besides, an important part of player engagement in such games is a periodical input of new cards in the system, which mainly opens the door to new strategies for the players. Playtesting is the method used to check the new card sets for possible design flaws, and it is usually performed manually or via exhaustive search; in the case of Hearthstone, such test plays must take into account the chosen hero, with its specific kind of cards. In this paper, we present a novel idea to improve and accelerate the playtesting process, systematically exploring the space of possible decks using an Evolutionary Algorithm (EA). This EA creates *HearthStone* decks which are then played by an AI versus established human-designed decks. Since the space of possible combinations that are play-tested is huge, search through the space of possible decks has been shortened via a new heuristic mutation operator, which is based on the behaviour of human players modifying their decks.

Results show the viability of our method for exploring the space of possible decks and automating the play-testing phase of game design. The resulting decks, that have been examined for balancedness by an expert player, outperform human-made ones when played by the AI; the introduction of the new heuristic operator helps to improve the obtained solutions, and basing the study on the whole set of heroes shows its validity through the whole range of decks.

*Keywords:* Genetic algorithm, HearthStone, Collectible Card Games, Artificial Intelligence

---

## 1. Introduction

Collectible card games (CCGs) are turn-based card games where players set up their decks in advance, carefully selecting cards in order to have the opportunity to exploit powerful combinations later, during an actual match. In this process, known as *deckbuilding*, players usually choose the cards from a large set (with hundreds or even thousands of possibilities), acquired in digital or physical sealed packs. Since every card has specific features, complex and rich gameplays usually emerge.

CCGs gained popularity in the 1990s, thanks to *Magic: The Gathering*™. In the following decades, the genre has been revitalized by the advent of digital versions (digital collectible card games, DCCGs), such as *Clash Royale*™ or *HearthStone: Heroes of WarCraft*™, both downloaded more than 50 million times [1].

Cards are clearly the main component of the game, and in order to increase players' engagement, available sets are updated regularly. On average, every two to six months a new set of cards is added to the game, and sometimes older ones are removed. Such *expansions* need to be thoroughly analyzed to avoid *breaking the game* [2]. However, testing these new cards is a difficult issue, because as the number of cards increases, also does the number of interactions or "combos", that can be potentially unbalanced and may affect players' enjoyment of the game.

Besides taking into account obvious constraints and checking for under- or over-powered cards, the end goal is the increase of satisfaction, engagement, and ultimately *fun* for all players. However, the huge dimension of most games makes it very hard to fully test the impact on all facets of the game of new contents before including them in one of the released expansion packs.

The situation calls for automatic testing tools to help designers to identify possible flaws or weaknesses, and to assess their impact in advance. This process is called *playtesting*, which is a way of searching the space of possible decks for superior ones.

Automatic deckbuilding methods may be useful to automatically test feasible and interesting decks, not only for developers, but also to players, as a decision-aid tool inside the game using their current collection as input. Moreover, they allow to add extra content, for example challenges where players have to defeat an Artificial Intelligence (AI) agent (or bot) with an original optimized deck.

In this paper, we propose a playtesting tool that can be used during the development of new content for a DCCG. More specifically, we focused on automatic deckbuilding in *HearthStone*, arguably the most successful DCCG nowadays. We propose to automatically build and test viable and competitive decks using an Evolutionary Algorithm (EA) [3].

Even though deckbuilding is a cornerstone of CCGs [4], there is not a lot of work in literature dealing with this specific issue. Indeed the possibility of optimizing deckbuilding in *HearthStone* was firstly explored by the authors in [5]. However that paper just presented a proof-of-concept study considering a limited experimental setup. In fact, we addressed as future work the possibility to evolve decks for different Heroes, and to perform a play-by-play analysis of the decks, to try and assess the generality of our approach.

Therefore, we have extended the search space of possible decks by including all available Heroes in the game. Since this extension boosts the size of the search space and thus the time of search, a new heuristic deck mutation operator has been added to the evolutionary algorithm. Also, as this method is intended to be a playtesting methodology, the best way to evaluate it is to use an expert to validate the decks obtained automatically by our algorithm, which is what we have done in this paper.

The rest of the paper is structured as follows. Section 2 introduces a few key concepts related to our subject. The state of the art and related works are commented in Section 3. The proposed method is described in Section 4, followed by the specific experimental setup in Section 5. Obtained results and findings about the evolutionary deckbuilding and playtesting process are exposed, followed by an analysis on the used cards and their impact on the matches in Sections 6 and 7 respectively. Finally, reached conclusions and an outline of future directions are presented.

## 2. Background

In this section, we summarize the terminology used, then the particular CCG we are working with, Hearthstone, and finally introduce the method we are applying, an Evolutionary Algorithm.

### *2.1. Collectible card games: Concepts and terminology*

CCGs became known to the wide public with the release of *Magic: The Gathering* in 1993. The culture that evolved around them developed specific terms, defined in the following.

#### *2.1.1. Deckbuilding*

The deck that a player deploys in a CCG must be carefully chosen, starting from a common pool of available cards. Building a deck is one of the most important parts of the experience: while playing the cards optimally is obviously important, players can only hope to draw a card that already is in their deck. Deckbuilding is a complex activity that requires understanding the current state of the game as well as evaluating other players; not surprisingly, it monopolizes the vast majority of articles and discussions among players on the Internet.

#### *2.1.2. Metagame*

All important activities associated with the experience, yet perceived by players as *peripheral* to the game itself, are cumulatively termed *metagame*. In the context of CCGs, metagame describes the *type* of decks that one is expected to find in a specific *ladder* (i.e., a competitive, ranked system), that is, "what everyone else is playing" [6].

#### *2.1.3. Mana curve*

In most CCGs each card has a *cost*, indicating the number of resources needed to play it, and usually called "casting cost". The card cost is used for balance, as resources increase over time, and cheaper cards tend to be weaker but can be played early in the game, while expensive cards are potentially game-changers. Such cost is generally called "mana" after the term used for resources in *Magic: The Gathering*™. The *mana curve*[1] is a histogram plot representing a deck by counting the number of cards per each casting cost.

[1] http://hearthstone.gamepedia.com/Mana_curve

From such mana curves, it's easy to understand the reference archetype. In any case, the mana curve should usually be somehow balanced, because completely lacking cards with a certain casting cost might leave the player unable to be effective in the early, mid or late game.

*2.1.4. Deck archetypes*

Since CCGs have a huge card pool, players can create decks with many different behaviours. A *deck archetype* is a category of deck formed by a specific subset of cards that allows a particular style of play. While each CCG features its own exclusive archetypes such as *Suicide Black* in Magic: The Gathering and *Malylock Warlock* in HearthStone [7], there are a few broader typologies that all decks can be roughly reduced to.

- *Aggro*, short for "aggression", is a deck driven by a relatively simple strategy: the player attempts to finish the game in its early stages, quickly consuming lots of resources to inflict the maximum possible damage to the opponent. Typically, if players with *Aggro decks* cannot end the game fast enough, they will eventually lose in the mid or late game. This kind of decks will typically have a considerable number of low-cost cards, with a mana curve shifted to the left.

- *Combo* is a deck where the main objective of the players is to survive until they manage to draw all the necessary pieces of a combination. Combos usually include two or more synergistic cards that allow the player to unleash a considerable amount of damage (ideally lethal) over the span of a single turn, securing the game. Players with these deck archetypes may lose if the opponent is able to produce a significant attack before all the pieces of the combination are gathered, or if the opponent is prepared to somehow counter it.

- *Control* is a deck chosen to keep the opponent in check, neutralizing early-game threats to prolong the match until the late game, where they can finish off using high-cost, high-value cards. Players with Control decks risk losing if they cannot find good answers for the cheap, effective threats of Aggro decks, or if they fail to counter the lethal combinations of Combo decks. This kind of decks will typically feature a mana curve shifted to the right.

*2.2. HearthStone*

*HearthStone: Heroes of WarCraft* is an online DCCG launched in 2013 by Blizzard Entertainment. Players build a deck of 30 cards from a huge card pool that can be expanded buying random packs, or converting owned cards to in-game currency to buy new ones. To win, players need to reduce the health of the other human opponent (or Hero) from 30 to 0, using the two types of cards available: *spells*, that affect the battleground and are then discarded, and *minions*, that stay in play and can attack the enemy Hero or other minions. Also, *weapons* can be seen as a sub-set of spells that allow the hero to attack other characters during several turns using special abilities. Each card has an associated cost (or number of *crystals*, equivalent to mana), that is reduced after a card is played, but replenished at the beginning of the turn. Each player starts with 1 crystal, and every turn 1 more is added, up to a maximum of 10 crystals.

Deckbuilding is limited to the neutral card pool and the cards that belong to the *class* of the Hero chosen for the game: Druid, Mage, Hunter, Paladin, Priest, Rogue, Shaman, Warlock, and Warrior. Besides, every Hero comes with a different Hero Power (costing 2 crystals to use), that in conjunction with their card set, matches every Hero to different deck archetypes. For example, Priest's healing abilities are a very powerful choice for Control decks, but not so convenient for Aggro ones.

Figure 1 shows a screenshot of a match confronting a Hunter versus a Mage.


Figure 1: Screenshot of a HearthStone match.

### *2.3. Evolutionary Algorithms*

Evolutionary Computation is a scientific field that involves a large number of bio-inspired methods, problems and tools. Evolutionary Algorithms (EAs) are a set of techniques from this field, that are usually applied to optimization problems [3]. These algorithms are inspired by the process of natural selection, giving the best, usually called *fittest*, solutions (or *individuals*) a higher probability to mate and generating new solutions that inherit their information, sometimes with flaws, in a process called *mutation*. Thus, iteratively, individuals will recombine to form new and, hopefully and due to the selection process, better solutions for the target problem.

At every iteration, or *generation*, different *genetic operators* are applied to the individuals to change them (*mutation*); or to recombine from existing ones (*parents*) to generate new individuals (*offspring*). At the end of each iteration, the least fit individuals are removed, and the process continues until a termination criterion is met, usually a fixed number of generations [8].

One of the advantages of the EAs is that they can obtain optimal, or near-optimal, solutions hard to find for a human expert. Moreover, they do not require human knowledge to solve the problem. Indeed, they have been extensively used in the field of videogames, mainly for the automatic generation and refinement of AI engines [9, 10, 11, 12]. EAs can also be used to find common patterns in generated decks, and therefore, test new sets of cards helping to identify outperforming cards, for instance. Thus, in the EA, every deck is evolved by simulating a variety of games against hand-made decks to calculate its fitness value (a measure of its quality).

## 3. Related Work

Card games have been a field of research since the nineties, when the automatic play of digital versions of 'classical games', such as Solitaire [13] or Poker [14], were an object of research for the first time. However, the rise of CCGs created an interesting testbed for AI research in which players deal with an environment with uncertainty and hidden information [15]. In addition, the huge number of different combinations of cards can produce many different - and sometimes unpredicted - effects.

However, CCGs have not been very prolific among the computational intelligence research community, and just a few works on the subject can be found. This can be explained due to the lack of simulation software to test new autonomous players or decks.

There are, however, some applications such as Magic Workstation[2] or Apprentice[3], which let the player manage card collections and play against other (human) players online. Recently, a HearthStone simulator named *MetaStone* [16] has arisen as a real option for testing AI approaches. This is the tool used for our work, as described in Section 5.1.

Most of the existing works in the CCGs scope are devoted to the presentation of AI approaches. Cowling et al. [17], for instance, applied Monte Carlo Tree Search (MCTS) to deal with the imperfect information of the game Magic: The Gathering. The authors considered determinization (i.e. assuming the hidden and random information is known by the players) so an advanced MCTS approach could be applied. Their approach was able to outperform a rule-based system modeling several human-expert heuristics in the game. This expert-based rule system was able to win almost half of the matches against strong human players.

Bursztein described in [18] an agent based in statistics which applies learning methods to predict, with a high accuracy level, the cards that its opponent will play in the following turns. Its forecasting ability decreases over time, but the preliminary results in the first five turns are excellent; however, this is to be expected, as the number of options available to the player increases with time.

Wanderley et al. [19] presented a system able to generate original and unexpected combinations of cards in HearthStone, following a creativity-focused method. However, that work only tries to generate 'uncommon' card combinations, calculating efficiency and rarity metrics from a previously built database of combos (extracted from human plays), and not automatically playing against an existing AI.

Mahlmann et al. [4] conducted a complete study on the usage of EAs for game balancing in the card game Dominion. In this game, 10 cards, each one with a specific rule, are placed at the beginning of the game, meaning that every game will have a different gameplay dynamics, and implying the players have to adapt their strategies according to this rule set. In this paper, each individual of the EA is a vector of 10 cards (from a pool of

[2] `http://www.magicworkstation.com/`

[3] `http://apprentice.nu/`

25 available). Authors propose different fitness to measure the balance of this set, calculated after simulating a large number of games, based on the difference in points. However, our approach is more related to the detection of possible card interactions, and not with the rules of the game. Deeper analysis and experiments need to be carried out because of the existence of 9 different heroes, implying a higher number of restrictions and combinations during the game. AI assets can be automatically created, and automatically tailored to the play style of the target AI. For example, it can be used to generate decks for the one-player campaign mode of the game. This can be especially useful when new sets of cards are added.

The present work is the follow-up of a previous contribution [5] that introduced an EA able to create and optimize decks of cards. Results for two out of the nine characters in HearthStone, Mage and Hunter, were reported. The approach, while preliminary, still yielded promising outcomes.

Here, as stated in the Introduction, we extend this research line, with 7 more heroes and a new game-based (human-like) mutation operator to enhance the performance of evolutionary algorithm.

## 4. Proposed approach

We propose a novel, generic approach to assist the playtesting process when a new CCG is created or new cards are introduced in existing ones, targeting HearthStone as a specific case study. The methodology uses an EA to automatically generate competitive decks, using a target metagame and an AI to assess their effectiveness. Eventually, cards that appear too frequently in the evolved decks, or decks that have an extremely high win ratio, can be analyzed by an expert and possibly identified as unbalanced.

The presented methodology is extensively used for all possible classes of Heroes in HearthStone. With respect to our previous work [5], we included a new evolutionary operator, named `Smart Mutation`, able to modify decks following a human-like heuristic, i.e. replacing one card with other of a similar cost.

For the fitness function, each deck is evaluated simulating several matches against a wide and diverse set of human-made decks, taken from the most competitive in the target metagame.

### 4.1. Evolutionary framework used: $\mu GP$

The framework used in the experiments is $\mu GP$ [20], a general-purpose EA designed to easily tackle different optimization problems out-of-the-box, thanks to its flexible definition of individual structure and external evaluator. Moreover, $\mu GP$ is able to self-adapt the activation probabilities of the evolutionary operators, freeing the user from choosing these parameters. The project is available on SourceForge[4].

Decks in HearthStone are formed by 30 cards, and therefore, an individual in our problem is defined as a deck (or decklist) of 30 cards, taken from the available pool for a specific class. Initially, such individuals are randomly generated. The evolutionary operators available in $\mu GP$ collectively allow the EA to replace a card with any other card, and cross over two decks, including the one explained in next subsection.

### 4.2. `Smart Mutation`: a customizable heuristic operator

During deckbuilding, human players are unlikely to replace an expensive card (e.g. mana cost 7-8) with a very cheap one (e.g. mana cost 1-2), but they will instead try to preserve the shape of the mana curve (see 2.1.3) in their deck, swapping cards with similar casting costs. Following this idea, the heuristic mutation operator, implemented as `Smart Mutation` in the $\mu GP$ framework, replaces a random card in a parent deck with another selected with uniform probability among all available cards within $+1/-1$ of the casting cost of the original.

### 4.3. Fitness evaluation

The fitness of a specific deck is computed resorting to the same strategy outlined in our previous work [5]. A good metric to measure the quality of a deck is the number of victories against other decks. Due to the stochastic nature of the game, a single execution of a game against a deck would not be statistically significant [21], so for each different opponent at least 16 games are recommended. Also, while the total number of victories obtained is important, at the same time we desire a deck with a fair chance to win against all decks in the metagame, and not one that mercilessly slaughters specific opponents and loses badly against other ones, so the standard deviation of the victories by deck is a good versatility measure. Moreover, as the EA can

[4] `http://ugp3.sourceforge.net/`

freely manipulate decks, swapping any card for any other, crossing two decks and so on, it is possible that it will obtain decks that violate the rules of the game: for example, by having more than 2 copies of the same card, or more than 1 copy of a Legendary one. For all these reasons, the fitness function is divided into three parts, evaluated following a lexicographical order:

1. Number of errors (minimize): this metric takes into account the number of errors in the decklist (repeated cards). Decks that have this fitness value bigger than 0 are not evaluated further, and all their remaining fitness values are set to the lowest possible amount. This fitness value is to be minimized.
2. Number of victories (maximize): straightforwardly, this is the total number of victories obtained by the decklist played 16 times against each of the decks in the target metagame. This fitness value is to be maximized.
3. Standard deviation of victories (minimize): this value is computed by evaluating the number of victories obtained against each opponent, and computing the standard deviation with regards to the number of victories against other opponents. If the deck obtains the same number of victories against all opponents, its standard deviation will be optimal. This fitness value is to be minimized.

Lexicographical order implies comparing a vector of the three metrics following the aforementioned order. This means that when comparing two individuals, first, the number of errors of each one is considered, choosing the one with less errors as the winner. In case both of them have the same number of errors, the number of victories is used for the comparison, choosing the individual with more victories. Finally, standard deviation is only used if there is again a draw between both individuals.

We have applied this technique in order to deal with restrictions, but there are several other approaches [22]. One method consists in setting the fitness to its worst value (e.g.: MIN_DOUBLE when maximising) if the individual do not met a specific constraint. Other approaches aim to 'fix' the individuals during the application of crossover or mutation operators, including valid elements. However, in this paper we have chosen the lexicographical fitness comparison because it derogates this issue within the evaluation process. This also allows us to differentiate the versatility of the decks without adding extra parameters to weigh each part.

Pseudo-code of the EA is described in Algorithm 1. The computational complexity of EAs is a function of several parameters, such as population size, number of generations, probability of mutations and crossovers [23]. It must be noted, however, that in the proposed approach the computational cost of executing a single fitness evaluation vastly dominates the rest of the steps of the algorithm, as it is common for many real-world problems.

**Algorithm 1** Pseudo-code of the EA adopted in the proposed approach.

```
population ← initializePopulation()
evaluate(population)
while stopping criterion not met do
  for child in offspring do
    parents ← tournamentSelection(population)
    choose evolutionary operator to apply
    if operator = crossover then
      offspring ← offspring + crossover(parentA,parentB)
    else if operator = mutation then
      offspring ← offspring + mutate(parentA)
    else if  operator = smartMutation then
      offspring ← offspring + smartMutation(parentA)
    end if
  end for
  evaluate(offspring)
  update internal parameters
  population ← population + offspring
  sort(population)
  reduce population to initial size by removing worst individuals
end while
```

## 5. Experimental evaluation

This section describes the experimental setup, the parameters used for the proposed approach, as well the fitness evaluation for the case of HearthStone. First, the engine used for this specific case study, called MetaStone, is described; then, a preliminary analysis of this engine execution with several available AI behaviors is performed, to establish an appropriate fitness function, and to assess the behavior of the engine. Parameter set for the EA and hardware setup are described at the end of this section.

### *5.1. MetaStone*

MetaStone [16] is an open-source HearthStone simulator that allows the manual creation of decks using the cards available in HearthStone and the simulation of matches between decks, obtaining several statistical measures such as the number of turns taken until victory or damage inflicted. Different heuristics can be selected for the AI engine, based on a score given to the actions that are evaluated in each turn, taking into account a combination of weights for the minions/spells used.

- Play Random Behavior (PRB): each turn the actions (moves) to play are selected randomly.
- Greedy Optimize Move (GOM): in each turn the AI selects each move ordered by score.
- Greedy Optimize Turn (GOT): in each turn the AI selects, among the combination of all possible moves, the one with the highest score, computed given the current game situation.
- Flat MonteCarlo Tree (FMC): during a given number of iterations, the AI simulates random moves until the possible ends of the match, to calculate their score taking into account possible future game states; then, it picks the combination of moves with the highest score.

### *5.2. Opponent decks analysis*

A set of different human-designed decks, used during a specific game season, have been considered as opponents. More specifically, the decks include MidRange Hunter, Mage Tempo, Aggro Paladin, Mech Shaman, Shadow Madness Priest, MalyLock Warlock, Control Warrior, Oil Rogue, and MidRange Druid. The detailed description of each one, and the justification of the used metagame season, can be seen in Appendix A.

In order to get an estimation on how well MetaStone can play the human-designed decks, we run a first tournament, in which each deck was paired against every other deck for 32 games, using all combinations of the 4 possible AIs (360 combinations). Thus, 11,520 games were played, showing that aggressive decks win more times without taking into account the AI used

Table 1: Number of games won by the human-made decks in our preliminary analysis (of a total of 11,520) aggregated by deck, after confronting all deck/AI combinations (every deck was played using all the available AIs against the others).

| Deck name | Games Won | Percentage of victories |
|---|---|---|
| Midrange Hunter | 1,669 | 14.48% |
| Aggro Paladin | 1,502 | 13.03% |
| Mage Tempo | 1,494 | 12.96% |
| Control Warrior | 1,295 | 11.24% |
| Midrange Druid | 1,286 | 11.16% |
| Mech Shaman | 1,233 | 10.70% |
| Shadow Madness Priest | 1,122 | 9.73% |
| Warlock MalyLock | 1,101 | 9.55% |
| Oil Rogue | 809 | 7.02% |

(Table 1). Dividing by deck/AI combination (Table 2) this also applies for most of the aggro decks, although the other types of decks also benefit from using the Greedy methods by obtaining a larger number of victories. From those results we discovered that the AI `GOT` obtained the best percentage of victories, winning 4,320 games out of the 11,520 (37.5%). Therefore, we set this AI as the one to beat during the rest of the experiments. Focusing on the deck behavior using this AI, Table 3 shows the percentage of victories of each one.

Table 2: Number of games won in our preliminary analysis (of a total of 11,520) of all decks/AI combinations.

| Deck | FMC | GOM | GOT | PRB |
|---|---|---|---|---|
| Aggro Paladin | 302 | 420 | 579 | 201 |
| Control Warrior | 187 | 395 | 359 | 354 |
| Mage Tempo | 281 | 442 | 648 | 123 |
| Mech Shaman | 183 | 399 | 417 | 234 |
| Midrange Druid | 304 | 341 | 470 | 171 |
| Midrange Hunter | 417 | 437 | 561 | 254 |
| Oil Rogue | 52 | 289 | 419 | 49 |
| Shadow Madness Priest | 90 | 445 | 510 | 77 |
| Warlock MalyLock | 95 | 462 | 357 | 187 |
| Total by AI | 1911 | 3630 | 4320 | 1650 |

Table 3: Number of games won using the GOT AI with each deck. Each deck shown in this table played against other decks using the same AI (256 games per deck).

| Deck name | Games Won | Games Lost | Percentage of victories |
|---|---|---|---|
| Aggro Paladin | 182 | 74 | 71.09% |
| Mage Tempo | 177 | 79 | 69.14% |
| Shadow Madness Priest | 152 | 104 | 59.37% |
| Midrange Hunter | 143 | 113 | 55.85% |
| Mech Shaman | 119 | 137 | 46.48% |
| Oil Rogue | 106 | 150 | 41.40% |
| Control Warrior | 104 | 152 | 40.62% |
| Midrange Druid | 85 | 171 | 33.20% |
| Warlock MalyLock | 83 | 173 | 32.42% |

### *5.3. Parameters*

After deciding which opponent decks and AI will be used in the fitness function, the rest of the parameters of the experimental setup is explained. Each candidate decklist (individual) is played $t = 16$ times against previous human-designed decks, with the exception of the deck featuring the same Hero ($e = 8$). This number of games has been set in order to achieve enough statistical confidence [21], but also to take advantage of the number of cores available in our computer, in order to parallelize the evaluation process. Therefore, each individual is tested 128 times in every evaluation.

As our evolutionary process is trying to create the best possible deck for that hero, we decided to avoid mirror matches against the corresponding human-designed deck, with the additional benefit to save computational time.

$\mu GP$ has been configured with the parameters reported in Table 4 for all the experiments. $\mu$ is the population size, the number of individuals stored at each iteration; $\lambda$ is the number of new individuals produced at each iteration; $\mathcal{G}$ is the total number of iterations, called generations. Several parameters are self-adapted during the run, using an inertia $\alpha$ that regulates their rate of change: among those, the mutation strength $\sigma$ that controls the amount of mutations performed per individual, and the probability of applying each operator. The considered operators can swap a single card in a deck for another randomly selected one (`singleParameterAlterationMutation`), combine two individuals using a one (`onePointCrossover`) or two (`twoPointCrossover`) cut point, or use a mutation that is not completely random, but follows a more human-like heuristic decision (`Smart Mutation`).

Table 4: Parameters used by the EA ($\mu GP$). The activation probabilities of the operators are self-adapted. For more information on the parameters, see [20] or visit `https://sourceforge.net/p/ugp3/wiki/Home/`.

| Parameter | Meaning | Value |
|---|---|---|
| $\mu$ | Population size | 10 |
| $\lambda$ | Operators applied | 10 |
| $\alpha$ | Self-adapting inertia | 0.9 |
| $\sigma$ | Initial mutation strength | 0.9 |
| $\tau$ | Size of the tournament selection | [2-4] |
| $G$ | Number of generations | 200 |
| $S$ | Strategy | $(\mu + \lambda)$ |
| $R$ | Replacement mechanism | Generational |
| $e$ | Number of opponent decks | 8 |
| $t$ | Number of games per opponent deck | 16 |
| Operators used | `singleParameterAlterationMutation` | |
| | `onePointCrossover` | |
| | `twoPointCrossover` | |
| | `Smart Mutation` | |

*5.4. Hardware Setup*

All the runs for each class have been executed on different nodes of a cluster. Each node featured 16 Intel(R) Xeon(R) CPU E5520 @2.27GHz processors, 16 GB RAM, CentOS 6.8 and Java Version 1.8.0_80. One of the most interesting features of EAs is their inherent possibility of evaluating solutions simultaneously, so the parallelization process was straightforward. Nevertheless, it is worth to mention that every run required approximately 30 days of computational time, as every individual played 128 simulated games per evaluation; however, with a higher parallelization capacity and ad-hoc optimization, computational times can be reduced. We will address this as one of the issues in future work.

In the following section, we first conduct an analysis of the obtained results just focusing on the AI (deckbuilding) performance. Then, the obtained decks and their cards are analyzed, aiming to discover clues on those that are over- or underpowered, i.e. those which could unbalance the game.

## 6. Results

Table 5 shows the percentage of wins for each generated deck by our method (with and without using Smart Mutation) and the original human-

Table 5: Percentage of wins obtained by the human-designed decks in the preliminary analysis (Subsection 5.2), versus the percentage of wins obtained by the best individuals evolved using our method with and without Smart Mutation at the end the run for each class. Best results are highlighted.

| Class | Human | No Smart Mutation | Smart Mutation |
|---|---|---|---|
| Paladin | 71.094 % | **82.813** % | 81.250 % |
| Mage | 69.141 % | 76.563 % | **81.250** % |
| Priest | 59.375 % | 78.906 % | **83.594** % |
| Hunter | 55.859 % | 76.563 % | **78.906** % |
| Shaman | 46.484 % | **72.656** % | 68.750 % |
| Rogue | 41.406 % | 67.188 % | **77.344** % |
| Warrior | 40.625 % | 67.188 % | **73.438** % |
| Druid | 33.203 % | 78.125 % | **80.469** % |
| Warlock | 32.422 % | 60.938 % | **67.188** % |

designed decks. As it can be seen, all generated decks by our method obtain better percentage of victories than the original ones. Using Smart Mutation has implied an improvement in all the classes with the exception of Shaman and Paladin. Paladin and Shaman are notable exceptions to the overall improvements introduced by Smart Mutation. It must be noted, however, that in these cases the version of the EA involving Smart Mutation lags only a few percentage points behind the other, while in almost all other cases the new version vastly outperforms the old. This behavior can be explained by the stochastic elements involved in a single run of an EA, and several repetitions of a run in a specific conditions would be needed to assess the goodness of this addition with reasonable statistical confidence, a task that is outside the scope of this work. Globally, the usage of Smart Mutation still seems beneficial.

It is worth mentioning that standard deviation has also been reduced with respect to the human decks from 14.57 to 7.44 without Smart Mutation, and to 5.83 with it. The behavior of this evolved decks is therefore, more balanced in terms of AI playing, than the human ones, even if the fitness function was not aimed at that, unlike the proposed approach by Mahlmann et al. [4].

Histogram of card costs (also called *Mana curves*, already explained in Section 2) of each deck is depicted in Figure 2. With some exceptions, almost all generated decks have curves shifted to the left, a clear marker of aggressive decks.

Focusing on the mere AI-based performance, the average number of turns-to-victory for all decks is around 8.24, with Warlock being the fastest (7.63) and Shaman the slowest (12.29). A whole statistics set for each deck/enemy combination can be seen in Table B.7 in the Appendix section. This can be checked in the generated deck lists depicted in Table B.6, that will be analyzed in more depth in next section, also pointing to find some clues or indicators regarding card unbalancing.

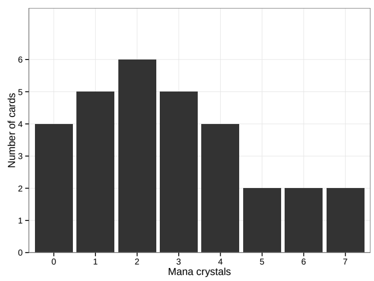
It is interesting to notice that the *Shadow Madness Priest* deck usually is the hardest to defeat for almost all the evolved decks. This may be due to the already mentioned strategy of changing the hero's healing power to inflict damage instead, easy to master and exploit by a greedy AI player.

## 7. Discussion

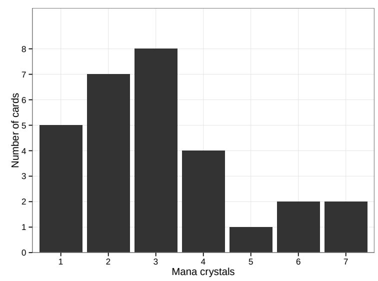
In this paper we set out to create a method that could be used for automated playtesting and general design of optimized decks in the game of Hearthstone. The proposed approach, which uses EAs and a new mutation operator, is able to discover decks with a satisfying win ratio against competitive human-designed decks in the target metagame. For all Heroes considered, the final win ratio outperformed the corresponding decks from season 18, see Table 5.

As mentioned in our previous work [5], while the proposed approach seems effective, there's an important weakness to consider: MetaStone is a viable solution for fitness evaluation, but it does not reach human players' level of play, a problem common to many other games [24]. Moreover, the best available AI exploits a greedy approach, so it is difficult to claim that our method generates decks suitable to play against humans, as human players can deploy an extremely large variety of strategies. Constraining the evolution to evaluating fights against a specific AI has the risk of overfitting to the particular AI's play style, leaving the generalization and real-world applicability as open questions [21].

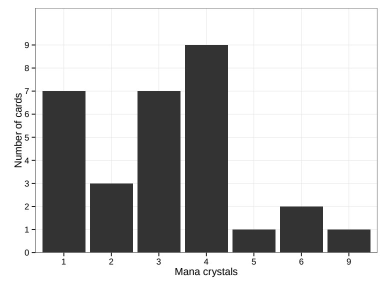
Despite this problem, interesting results have been obtained. We will first discuss notable features of the evolved decks, and then present results from a manual inspection of selected games. All the HearthStone expertise comes from one of the authors, an average competitive player, able to reach rank 10 in the season ladder (ranks 1-10 contain more or less the top 10% of the registered users [25]), who played over 13,000 matches since the Open Beta of the game.



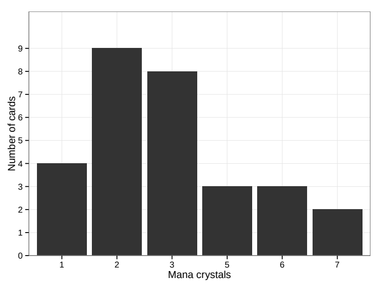

(a) Druid



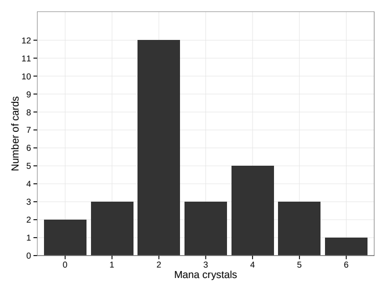

(b) Hunter



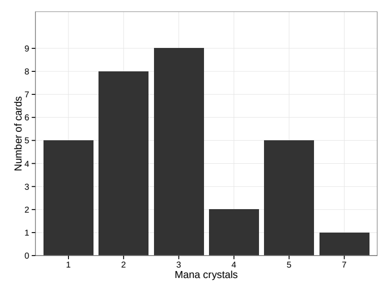

(c) Paladin



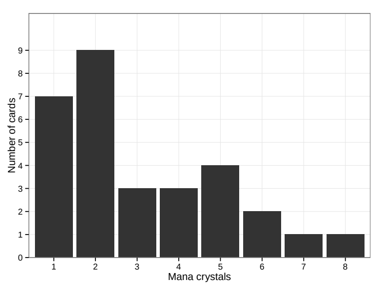

(d) Priest



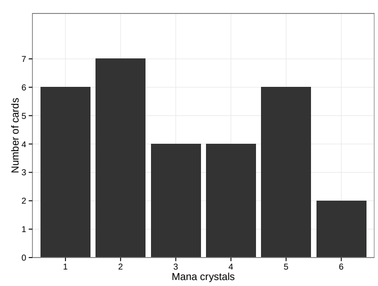

(e) Rogue



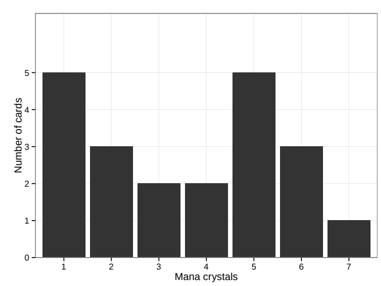

(f) Shaman




(g) Warlock




(h) Warrior




(i) Mage

Figure 2: Number of cards by cost, also called *Mana Curves*, of each generated deck. As it can be seen, they are balanced to the left, meaning the generated decks tend to be aggressive and quick to play, as they have a larger number of low-resources (crystals) cards.

*7.0.1. Deck overview*

The first thing we can appreciate is that, even lacking any specific instructions to do so, the EA is able to create decks with multiple copies of the same card. This makes sense, as increasing the probability of drawing a card reduces the variance in the deck's performance. Secondly, the evolved decks feature a large number of cards that are considered effective by human players. In Table B.6, cards highlighted in **bold** are often found in human-designed decks. Nevertheless, a few cards that are widely considered suboptimal, highlighted in *italics*, also found their way into the best evolved decks. Interestingly, some of these *suboptimal* choices might indeed be effective against the greedy AI: cards such as *Master Swordsmith*, *Recruiter* or *Eydis Darkbane*, if left unchecked on the board, can quickly spiral out of control, offering a cumulative advantage to the player that controls as turns go by. Usually human players are able to quickly dispose of them, as they are not particularly hard to remove. However, as described below in more detail, the greedy AI does not always sacrifice its minions to remove potential threats.

*7.0.2. Specific cards and combinations*

*Dr. Boom.* This card is a 7-crystal 7/7 minion that spawns two other 1/1 Boombot minions when it enters play. When Boombots die, they inflict 1-4 damage to a random enemy target. Dr. Boom appears in 3 out of the 9 evolved decks, and this is not surprising, as it has long been considered unbalanced [26] (players gave it the nickname *Dr. Balanced* to make fun of the situation).

*Ragnaros the Firelord and Sylvanas Windrunner.* While not as clearly unbalanced as Dr. Boom, these two Legendary cards have seen so much play that Blizzard recently decided to ban them from HearthStone's Standard format, in order to increase the variety of decks [27]. Unsurprisingly, they were found in 2 of the evolved decks.

*Weapons.* The weapons found in the evolved decks are almost always the most effective in the game. Cards like *Arcanite Reaper* for Warriors and *Truesilver Champion* for Paladins are clearly better than all other alternatives, and are included in almost all decks of those classes. Thus, this is a precious indication for a designer: new weapons to be introduced in the game should always be compared with the ones found by the proposed approach.

*Flood the board + Savage Roar.* Featured in the evolved Druid deck, this combination uses several cards that flood the board with a large number of minions (e.g. *Echoing Ooze*, *Living Roots*, *Imp Master*) and a card that increases the attack of all minions by 2. This strategy is reminiscent of the combo in the human-made Druid deck, that used *Savage Roar* and *Force of Nature*. However, Force of Nature might be too hard to use for the greedy AI, as this card creates three 2/2 minions that can immediately attack, but die at the end of the turn. Interestingly, combos based on *Savage Roar* were at the center of several controversies[5] and have been recently limited by Blizzard, with a radical change made to *Force of Nature*[6].

*Kill Command + Beast-type minions.* The evolved Hunter decklist has several minions of type Beast, or card that generate such minions (*Stonetusk Boar*, *Animal Companion*), and *Kill Command*, a powerful spell that increases its effect if a Beast is in play under your control.

*Spell-enhancing minions + Spells.* In the evolved Mage deck, it is easy to identify a few popular (*Mana Wyrm*, *Sorcerer's Apprentice*) and less popular (*Archmage*, *Frigid Snobold*) that all enhance spells, by reducing their cost, increasing the damage output, or activating other effects. Unsurprisingly, the spells included in the deck all deal damage, with the exception of *Mirror Image*, a defensive spell.

Surprisingly, a similar strategy emerges in the evolved Priest deck, that is unusually aggressive, as the class is considered to be more suited to control decks. *Velen's Chosen* and *Azure Drake* both increase spells' effectiveness, and the deck features 7 damage-dealing spells.

*Buff spells + Divine shield minions.* The evolved Paladin deck has 4 minions with *divine shield*, an ability that prevents all damage inflicted to the creature, the first time it is damaged in the game. This ability is particularly effective in combination with *buffs*, cards that increase the statistics of a minion on the board: and the deck plays 5 buffs. This preference might also be a side-effect of how the greedy AI deals (not very effectively) with divine-shield minions.

[5] `http://www.pcgamer.com/does-hearthstones-druid-combo-need-to-be-nerfed/`
[6] `http://www.hearthpwn.com/cards/237-force-of-nature`

*Evolved Oil Rogue.* Interestingly, the evolved Rogue deck features cards that are also included in the human-made version used as an opponent during the evaluation. *Gadgetzan Auctioneer* and *Questing Adventurer* capitalize on the Rogue playing many cards on the same turn. Moreover, the spells selected by the EA not only have low costs (thus, they can be potentially played in the same turn), but they are usually all included in human-designed decks.

*Murlocs.* Murlocs in HearthStone are a minion type with considerable card synergy. In the evolved Warlock deck we can see a small set of Murlocs (*Murloc Raider*, *Murloc Tidehunter*), along with the *Murloc Warleader*, that increases the stats of all friendly Murlocs in play.

*Demons.* The evolved Warlock deck also exploits several minions of type Demon (*Mistress of Pain*, *Illidan Stormrage*), along with cards that greatly increase their power (*Demonfuse*, *Demonheart*). This is particularly interesting for Mistress of Pain, as every time it deals damage, this minion also restores that much health to the Hero.

*Wounded minions + Rampage.* The evolved Warrior deck exploits several minions that are automatically wounded to activate their ability (*Imp Master*) or obtain benefits when wounded (*Amani Berserker*), plus a card, *Rampage*, that greatly increases a wounded creature's stats. Further synergy comes from *Whirlwind*, a spell that deals 1 damage to all minions on the board, thus wounding them.

*7.0.3. Gameplay analysis*

In order to assess the capabilities of MetaStone and investigate its response to specific cards, we manually inspect over 100 games played by the evolved decks against decks in the selected metagame. This inspection allowed us to highlight a few key points, and conclude that indeed MetaStone is sometimes unable to exploit the weak points of some otherwise powerful minion, or react correctly when faced with unusual threats.

*Fel Reaver.* Fel Reaver is a 5-crystal 8/8 minion, with extremely good stats for its cost, and a huge drawback: every time the opponent plays a card, the controller of Fel Reaver has to destroy the top 3 cards of their deck. This minion caught our attention, as it appears in 4 out of the 9 evolved decks. A human player would respond to the appearance of Fel Reaver by playing as many cards as possible, in order to annihilate the opponent's deck; but an AI might not be able to correctly operate in such a peculiar situation. As we expected, MetaStone cannot deal with Fel Reaver. During all inspected games, it kept playing as usual, without exploiting this big minion's weakness.

*Baron Geddon.* This card is a 7-crystal 7/5 minion that deals 2 damage to all creatures and players at the end of every turn it is in play. While Baron Geddon is not widely exploited by human players, it had some moments of popularity in metagames featuring aggressive decks with a large number of weak minions. From analyzed games where Baron Geddon is played, it seems that MetaStone's AI is unable to properly assess it, often underestimating the consequences of its presence on the board.

*Trading.* Sacrificing a player's own minions to kill strategic minions in the opponent's field is commonly known as *trading*. Overall, MetaStone's AI with the settings we selected plays pretty aggressively, rarely trading, but very often attacking the opposing player. It is thus not surprising that basically all evolved decks can be classified as Aggro or Mid-Range, with mana curves unbalanced towards cheap, cost-effective cards (see Figure 2). The only exception might be the Mage deck, that features a considerable number of costly cards, as the Mage class can count on expensive but efficient removals such as *Flamestrike*.

## 8. Conclusions and future work

This paper presents a tool intended to support the design of new levels of card games based on the application of an Evolutionary Algorithm (EA) to the optimization of decks in the Collectible Card Game (CCG) *HearthStone*. So far, these decks had to be analyzed by a human expert in order to find unbalanced cards or combinations of cards, that is, over- or under-powered ones in the comparison with the rest of the set. In this paper we propose a method that finds these unbalanced cards or combinations of same automatically. Every deck has been evaluated against the best human-designed decks of a specific season using the Framework *MetaStone AI* for the simulation of the matches.

The fact that cards of combinations of them that tip the balance of the game can be found automatically makes the proposed approach useful in the design process of the game, helping a designer test new cards before launching extensions that include them to the market. This can be done in order to avoid the inclusion of an undesired unbalance in the game, with the consequent need of banning or limiting cards after their introduction.

It is worth mentioning that the obtained results show that all the generated decks for each game class, which in this case corresponds to different hero types, have outperformed the human-designed ones. This proves the viability of our method for finding optimal decks for a given AI, expanding the conclusions reached in our previous work to this new search space that includes all types of cards. This points to a new and not totally unintended result of this method: to create optimized decks for players, which can then use them online.

Although the generated decks have, in general, an aggressive behavior, due to the greedy approach of the AI used for evaluation, the evolved decks feature several strong cards frequently chosen by human players (when the build their own deck). Some of the cards appearing multiple times across evolved decks, such as *Dr. Boom*, are later identified by an expert as unbalanced; while others, such as *Fel Reaver*, probably owe their recurrent appearance to the play style of the AI.

In this paper we have also tested a new human-like heuristic, which has been used by the evolutionary algorithm to modify the decks during the evolution. It has been applied as a context-aware or game-based mutation operator in the EA. This operator swaps cards in a deck for cards of similar cost, trying to preserve the so-called "mana curve", that is, the distribution of mana costs in a deck; this follows a common practice that human players consider important for a deck's effectiveness. According to the obtained results, the operator is proven to deliver better decks in the majority of reported experiments.

Finally, we think that this specific evolutionary approach can be generalized to any game for which: 1) users have the possibility of creating decks, or user-specific pools taken from a large amount of elements, such as a 6-Pokémon team in the *Pokémon*™games, or a vehicle that can be configured with a limited set of unique features; 2) there is an evaluation of the goodness of a player's choice available, for example, playing the decks against decks in the current metagame, or by pairing evolved Pokémon teams against the best ones in championships; and 3) new content is routinely added to the game, so a long and time-consuming playtesting phase is usually needed before the content is released.

There are several open lines of future work. First, we will conduct a and more advanced and in-depth evaluation of the statistics of the matches that MetaStone offers, going beyond the examination of just the number of victories. We will also apply game mining techniques over the game logs. We could also adapt the fitness function in the EA to consider balancing-related terms in the evaluation of decks, following previous proposals such as Mahlmann's [4].

In addition, given the limitations of the used MetaStone's AI, we plan to improve it by optimizing the score function it uses to establish the board position and to decide which action should be taken in every turn. Moreover, other different game AIs can be compared, and even the possibility of testing the decks against human players will be considered, with the objective to obtain more human-like results. Cards and human-made decks from the newest expansions, that completely altered the metagame, will be also included in the experiments.

**Acknowledgements**

This work has been partially funded by projects SPIP2017-02116, EphemeCH TIN2014-56494-C4-3-P, DeepBio TIN2017-85727-C4-2-P and TEC2015-68752.

The authors would like to thank Antonio Fernndez-Ares for proof-reading the paper. Special thanks to Github user `@demilich1` and all contributors for creating MetaStone, and for answering our questions on design of its AI.

[1] Hearthstone now has 50 million players, http://www.polygon.com/2016/4/26/11511890/hearthstone-50-million-players-2016, accessed 15 January 2017.

[2] E. Ham, Rarity and power: Balance in collectible object games, Game Studies 10 (1).

[3] A. E. Eiben, J. E. Smith, Introduction to Evolutionary Computing, Natural Computing Series, Springer, 2015.

[4] T. Mahlmann, J. Togelius, G. N. Yannakakis, Evolving card sets towards balancing Dominion, in: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2012, Brisbane, Australia, June 10-15, 2012, IEEE, 2012, pp. 1–8.

[5] P. García-Sánchez, A. P. Tonda, G. Squillero, A. M. García, J. J. M. Guervós, Evolutionary deckbuilding in Hearthstone, in: IEEE Conference on Computational Intelligence and Games, CIG 2016, Santorini, Greece, September 20-23, 2016, IEEE, 2016, pp. 1–8.

[6] M. Carter, M. R. Gibbs, M. Harrop, Metagames, paragames and orthogames: a new vocabulary, in: M. S. El-Nasr, M. Consalvo, S. K. Feiner (Eds.), International Conference on the Foundations of Digital Games, FDG '12, Raleigh, NC, USA, May 29 - June 01, 2012, ACM, 2012, pp. 11–17.

[7] Identifying deck archetypes, `https://tempostorm.com/articles/identifying-deck-archetypes`, accessed: 21 March 2017 (2016).

[8] A. Fernández-Ares, P. García-Sánchez, A. M. Mora, P. Á. C. Valdivieso, J. J. M. Guervós, M. I. G. Arenas, G. Romero, It's time to stop: A comparison of termination conditions in the evolution of game bots, in: A. M. Mora, G. Squillero (Eds.), Applications of Evolutionary Computation - 18th European Conference, EvoApplications 2015, Copenhagen, Denmark, April 8-10, 2015, Proceedings, Vol. 9028 of Lecture Notes in Computer Science, Springer, 2015, pp. 355–368.

[9] P. Spronck, I. Sprinkhuizen-Kuyper, E. Postma, Improving opponent intelligence through offline evolutionary learning, International Journal of Intelligent Games and Simulation 2 (1) (2003) 20–27.

[10] N. Cole, S. J. Louis, C. Miles, Using a genetic algorithm to tune first-person shooter bots, in: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2004, 19-23 June 2004, Portland, OR, USA, IEEE, 2004.

[11] R. K. Small, C. B. Congdon, Agent Smith: Towards an evolutionary rule-based agent for interactive dynamic games, in: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2009, Trondheim, Norway, 18-21 May, 2009, IEEE, 2009, pp. 660–666.

[12] A. M. Mora, A. Fernández-Ares, J. J. Merelo-Guervós, P. García-Sánchez, C. M. Fernandes, Effect of noisy fitness in real-time strategy games player behaviour optimisation using evolutionary algorithms, J. Comput. Sci. Technol. 27 (5) (2012) 1007–1023.

[13] R. Bjarnason, A. Fern, P. Tadepalli, Lower bounding Klondike solitaire with Monte-Carlo planning, in: A. Gerevini, A. E. Howe, A. Cesta, I. Refanidis (Eds.), Proceedings of the 19th International Conference on Automated Planning and Scheduling, ICAPS 2009, Thessaloniki, Greece, September 19-23, 2009, AAAI, 2009.

[14] G. V. den Broeck, K. Driessens, J. Ramon, Monte-carlo tree search in Poker using expected reward distributions, in: Z. Zhou, T. Washio (Eds.), Advances in Machine Learning, First Asian Conference on Machine Learning, ACML 2009, Nanjing, China, November 2-4, 2009. Proceedings, Vol. 5828 of Lecture Notes in Computer Science, Springer, 2009, pp. 367–381.

[15] I. Frank, D. A. Basin, Search in games with incomplete information: A case study using Bridge card play, Artif. Intell. 100 (1-2) (1998) 87–123.

[16] MetaStone - A Hearthstone Simulator, `http://www.demilich.net/metastone/`, accessed: 30 March 2017.

[17] P. I. Cowling, C. D. Ward, E. J. Powley, Ensemble determinization in monte carlo tree search for the imperfect information card game Magic: The Gathering, IEEE Trans. Comput. Intellig. and AI in Games 4 (4) (2012) 241–257.

[18] E. Bursztein, I am a legend: Hacking HearthStone using statistical learning methods, in: IEEE Conference on Computational Intelligence and Games, CIG 2016, Santorini, Greece, September 20-23, 2016, IEEE, 2016, pp. 1–8.

[19] L. F. W. Góes, A. R. D. Silva, J. Saffran, A. Amorim, C. França, T. Zaidan, B. M. P. Olimpio, L. R. O. Alves, H. Morais, S. Luana, C. Martins, Honingstone: Building creative combos with honing theory for a digital card game, IEEE Trans. Comput. Intellig. and AI in Games 9 (2) (2017) 204–209.

[20] E. Sanchez, M. Schillaci, G. Squillero, Evolutionary Optimization: the $\mu$GP toolkit, Springer Science & Business Media, 2011.

[21] J. J. Merelo, F. Liberatore, A. Fernández-Ares, R. H. García-Ortega, Z. Chelly, C. Cotta, N. Rico, A. M. Mora, P. García-Sánchez, There is noisy lunch: A study of noise in evolutionary optimization problems, in: A. C. Rosa, J. J. Merelo-Guervós, et al. (Eds.), Proceedings of the 7th International Joint Conference on Computational Intelligence (IJCCI 2015) - Volume 1: ECTA, Lisbon, Portugal, November 12-14, 2015., SciTePress, 2015, pp. 261–268.

[22] T. P. Runarsson, X. Yao, Search biases in constrained evolutionary optimization, IEEE Trans. Systems, Man, and Cybernetics, Part C 35 (2) (2005) 233–243.

[23] M. Srinivas, L. M. Patnaik, Adaptive probabilities of crossover and mutation in genetic algorithms, IEEE Trans. Systems, Man, and Cybernetics 24 (4) (1994) 656–667.

[24] P. García-Sánchez, A. P. Tonda, A. M. Mora, G. Squillero, J. J. M. Guervós, Towards automatic StarCraft strategy generation using genetic programming, in: 2015 IEEE Conference on Computational Intelligence and Games, CIG 2015, Tainan, Taiwan, August 31 - September 2, 2015, IEEE, 2015, pp. 284–291.

[25] Battle.net: Hearthside Chat - You're better than you think!, `http://us.battle.net/hearthstone/en/blog/15955974/hearthside-chat-youre-better-than-you-think-9-18-2014`, accessed: 24 April 2016 (2014).

[26] Face off: Should Dr. Boom be nerfed?, `http://www.pcgamer.com/face-off-should-dr-boom-be-nerfed/`, accessed: 13 March 2017.

[27] Ragnaros and Sylvanas will only be playable in Wild once Hearthstones next set launches, `http://bit.ly/2nwRF6H`, accessed: 30 March 2017.

## Appendix A. Opponent decks description

For the experimental evaluation, we consider the metagame of Season 18 of *HearthStone* competitive play, featuring the base set, the adventures *Curse of Naxxramas* and *Blackrock Mountain*, and the expansions sets *Goblin vs Gnomes* and *The Grand Tournament*, which overall include 694 cards. We choose this set of cards because it features a good representation of different deckbuilding strategies, from which we select 4 human-designed Aggro decks (Hunter, Mage, Paladin, Shaman), 3 Control (Priest, Warrior, Warlock) and 2 Combo (Druid, Rogue). Another reason is that the expansion *League of Explorers*, released in Season 19, considerably altered deckbuilding, due to the presence of *Reno Jackson*[7] and other powerful cards that activate only

[7] `http://hearthstone.gamepedia.com/Reno_Jackson`

if the content of one's deck respects specific constraints (e.g. no more than one copy of each card).

The considered decks, summarized in the following, have been taken from the website of Tempo Storm[8], an American e-sports professional video game team, and selected among the ones able to reach the highest rank in the competitive ladder during season 18.

*Appendix A.1. MidRange Hunter (Aggro)*

This Hunter deck is slower than similar Aggro decks, trading cheap cards for cost-effective minions that are harder to remove, and thus more difficult to deal with for Control decks.

*Appendix A.2. Mage Tempo (Aggro)*

In CCGs, *Tempo* is basically a measurement of the speed of a player's progression through the game. This Aggro Mage deck uses cards that are able to improve one's progression, while at the same time slowing down the opponent, making enemy minions unusable for one or more turns.

*Appendix A.3. Aggro Paladin (Aggro)*

A fast, effective Aggro deck, that attempts to swarm the battlefield with a lot of weak but cheap minions. It includes a few ways to neutralize problematic answers from the opponent.

*Appendix A.4. Mech Shaman (Aggro)*

This Aggro deck exploits the synergy of some Shaman cards with a specific category of minions, the *Mechs*. The Mechs are not as cheap as the minions used in other Aggro decks, but they are harder for the opponent to deal with, and interact nicely with each other, as some Mechs provide bonuses to all other Mechs in play.

*Appendix A.5. Shadow Madness Priest (Control)*

A classical Priest Control deck, that makes use of a few twists. The Priest's Hero power normally would cure minions or the player; but there are a few Priest cards (with the keyword *Shadow*) that change this ability into inflicting an equal amount of damage. This deck tries to switch between

[8]`https://tempostorm.com/articles/meta-snapshot-18-from-warrior-to-warrior`

curing and dealing damage depending on board conditions, to keep the match under control until it can finish off the opponent using relatively powerful creatures.

*Appendix A.6. MalyLock Warlock (Control)*

The Warlock's default power allows the player to draw extra cards, in exchange for life points. Exploiting this feature, this deck tries to go through the deck, finally obtaining a single, expensive, powerful creature: the dragon Malygos. Malygos increases the amount of damage dealt by all of the player's spells by a large quantity, allowing the Warlock to quickly close the game the turn after Malygos enters the field.

*Appendix A.7. Control Warrior (Control)*

The Warrior's Hero power allows it to cumulate *Armor*, a sort of shield that protects the life points: before damaging the player's hit points, the opponent has to destroy all the Armor. Interestingly, while there is a cap for the hit points, there is no maximum limit for Armor. The Warrior tries to use the Armor to survive the early game, removing the most pernicious threats, while waiting for powerful, expensive minions that will be extremely effective in the late game.

*Appendix A.8. Oil Rogue (Combo)*

Oil rogue is a combo deck that exploits the Rogue's ability to play multiple cards in the same turn, reducing their costs thanks to the aid of other cards. In the very first turns this Rogue deck will try to remove the opponent's threats, all the while slowly building a large hand of cards, to finally unleash lethal damage in one single turn.

*Appendix A.9. MidRange Druid (Combo)*

This Combo deck aims at using a combination of 2 cards, *Force of Nature* and *Savage Roar*, that can inflict from 14 to 30 damage to the opponent, depending on board conditions. However, the combined cost of the two cards is 9 (6+3), thus the deck has to stall for time in the early game, and slowly build a *ramp* by using specific Druid cards that increase its resources faster than the opponent's.

## Appendix B. Generated decks

Table B.6 shows the generated decks by our method. These decks have been explained in section 7.0.1. Statistics of each deck (average turns or damage done, among others) are also presented in Table B.7.

Table B.6: Decks generated by the EA. Each card shows its mana cost, the rarity (F=Free, C=Common, R=Rare, E=Epic, L=Legendary) and an * indicates it is specific to the class.

| DRUID | | HUNTER | | MAGE | |
|---|---|---|---|---|---|
| MINIONS | SPELLS | MINIONS | SPELLS | MINIONS | SPELLS |
| **Ancient of War** 7 (E) * | **Claw** 1 (F) * | Amani Berserker 2 (C) | **Animal Companion** 3 (F) * | Amani Berserker 2 (C) | **Arcane Missiles** 1 (F) * |
| **Baron Geddon** 7 (L) | **Claw** 1 (F) * | **Baron Geddon** 7 (L) | **Animal Companion** 3 (F) * | Archmage 6 (F) | **Dragon's Breath** 5 (C) * |
| **Dire Wolf Alpha** 2 (C) | **Innervate** 0 (F) * | **Dr. Boom** 7 (L) | Arcane Shot 1 (F) * | **Baron Geddon** 7 (L) | **Fireball** 4 (F) * |
| Drakonid Crusher 6 (C) | **Living Roots** 1 (C) * | Drakonid Crusher 6 (C) | Arcane Shot 1 (F) * | Bomb Lobber 5 (R) | **Fireball** 4 (F) * |
| Drakonid Crusher 6 (C) | **Living Roots** 1 (C) * | **Flame Juggler** 2 (C) | Deadly Shot 3 (C) * | Bomb Lobber 5 (R) | **Flamecannon** 2 (C) * |
| **Echoing Ooze** 2 (E) | *Mark of the Wild* 2 (F) * | **Flame Juggler** 2 (C) | Explosive Shot 5 (R) * | Darkscale Healer 5 (F) | **Flamecannon** 2 (C) * |
| *Eydis Darkbane* 3 (L) | Power of the Wild 2 (C) * | Gelbin Mekkatorque 6 (L) | **Kill Command** 3 (F) * | *Eydis Darkbane* 3 (L) | **Flamestrike** 7 (F) * |
| **Fel Reaver** 5 (E) | **Savage Roar** 3 (F) * | **Knife Juggler** 2 (R) | **Kill Command** 3 (F) * | **Fel Reaver** 5 (E) | **Frostbolt** 2 (F) * |
| Gilblin Stalker 2 (C) | Starfall 0 (F) * | **Knife Juggler** 2 (R) | Multi-Shot 4 (F) * | Frigid Snobold 4 (C) | Mirror Image 1 (F) * |
| Gilblin Stalker 2 (C) | **Swipe** 4 (F) * | Light's Champion 3 (R) | Powershot 3 (R) * | *Gadgetzan Jouster* 1 (C) | |
| **Grove Tender** 3 (R) * | **Swipe** 4 (F) * | Light's Champion 3 (R) | **Tracking** 1 (F) * | *Gadgetzan Jouster* 1 (C) | |
| Imp Master 3 (R) | **Wrath** 2 (F) * | *Mad Bomber* 2 (C) | | Hogger 6 (L) | |
| Imp Master 3 (R) | | Mechanical Yeti 4 (C) | | Illidan Stormrage 6 (L) | |
| *Pit Fighter* 5 (C) | | Mechanical Yeti 4 (C) | | **Knife Juggler** 2 (R) | |
| Savage Combatant 4 (R) * | | **Mind Control Tech** 3 (R) | | **Leper Gnome** 1 (C) | |
| Secretkeeper 1 (R) | | *Recombobulator* 2 (E) | | **Mana Wyrm** 1 (C) * | |
| Stormwind Knight 4 (F) | | *Stonetusk Boar* 1 (F) | | *Skycap'n Kragg* 7 (L) | |
| *Target Dummy* 0 (R) | | *Stonetusk Boar* 1 (F) | | Sorcerer's Apprentice 2 (C) * | |
| | | **Twilight Drake** 4 (R) | | Spellbreaker 4 (C) | |
| | | | | Spider Tank 3 (C) | |
| | | | | Stampeding Kodo 5 (R) | |

| PALADIN | | PRIEST | | ROGUE | |
|---|---|---|---|---|---|
| MINIONS | SPELLS | MINIONS | SPELLS | MINIONS | SPELLS |
| **Alexstrasza** 9 (L) | **Blessing of Kings** 4 (F) * | Argent Commander 6 (R) | Holy Fire 6 (R) * | **Acidic Swamp Ooze** 2 (F) | **Backstab** 0 (F) * |
| Argent Squire 1 (C) | Blessing of Kings 4 (F) * | **Argent Horserider** 3 (C) | Holy Fire 6 (R) * | Chillwind Yeti 4 (F) | **Backstab** 0 (F) * |
| Clockwork Gnome 1 (C) | Blessing of Might 1 (F) * | **Argent Horserider** 3 (C) | **Holy Nova** 5 (F) * | **Cutpurse** 2 (R) * | *Betrayal* 2 (C) * |
| *Dragonhawk Rider* 3 (C) | **Consecration** 4 (F) * | *Argent Watchman* 2 (R) | **Holy Nova** 5 (F) * | **Cutpurse** 2 (R) * | **Cold Blood** 1 (C) * |
| Drakonid Crusher 6 (C) | **Consecration** 4 (F) * | **Azure Drake** 5 (R) | Holy Smite 1 (F) * | *Dancing Swords* 3 (C) | **Eviscerate** 2 (C) * |
| **Echoing Ooze** 2 (E) | **Muster for Battle** 3 (R) * | **Baron Geddon** 7 (L) | **Light of the Naaru** 1 (R) * | Dragonkin Sorcerer 4 (R) | **Eviscerate** 2 (C) * |
| **Enhance-o Mechano** 4 (E) | **Muster for Battle** 3 (R) * | Blood Knight 3 (E) | **Light of the Naaru** 1 (R) * | **Gadgetzan Auctioneer** 6 (R) | Sabotage 4 (E) * |
| **Fel Reaver** 5 (E) | **Seal of Champions** 3 (C) * | *Bloodfen Raptor* 2 (F) | Mind Blast 2 (F) * | **Harrison Jones** 5 (L) | **Sap** 2 (F) * |
| *Goblin Sapper* 3 (R) | WEAPONS | **Dr. Boom** 7 (L) | Mind Blast 2 (F) * | *Hungry Crab* 1 (E) | **Sap** 2 (F) * |
| *Goldshire Footman* 1 (F) | **Coghammer** 3 (E) * | Dreadscale 3 (L) | **Power Word: Shield** 1 (C) * | *Master Swordsmith* 2 (R) | WEAPONS |
| *Goldshire Footman* 1 (F) | **Truesilver Champion** 4 (F) * | **Echoing Ooze** 2 (E) | **Velen's Chosen** 3 (C) * | *Master Swordsmith* 2 (R) | Assassin's Blade 5 (F) * |
| **Hogger** 6 (L) | **Truesilver Champion** 4 (F) * | **Echoing Ooze** 2 (E) | **Velen's Chosen** 3 (C) * | Mechanical Yeti 4 (C) | |
| Lowly Squire 1 (C) | | *Mad Bomber* 2 (C) | | Mechanical Yeti 4 (C) | |
| Lowly Squire 1 (C) | | *Mad Bomber* 2 (C) | | **Mechwarper** 2 (C) | |
| Scarlet Purifier 3 (R) * | | **Mind Control Tech** 3 (R) | | *Micro Machine* 2 (C) | |
| **Shielded Minibot** 2 (C) * | | **Mind Control Tech** 3 (R) | | Nightblade 5 (F) | |
| **Shielded Minibot** 2 (C) * | | *Razorfen Hunter* 3 (F) | | Nightblade 5 (F) | |
| Silvermoon Guardian 4 (C) | | Recombobulator 2 (E) | | **Questing Adventurer** 3 (R) | |
| **Violet Teacher** 4 (R) | | | | Scarlet Crusader 3 (C) | |
| | | | | Secretkeeper 1 (R) | |

| SHAMAN | | WARLOCK | | WARRIOR | |
|---|---|---|---|---|---|
| MINIONS | SPELLS | MINIONS | SPELLS | MINIONS | SPELLS |
| **Annoy-o-Tron** 2 (C) | **Crackle** 2 (C) * | *Bloodfen Raptor* 2 (F) | **Darkbomb** 2 (C) * | **Abusive Sergeant** 1 (C) | **Heroic Strike** 2 (F) * |
| **Antique Healbot** 5 (C) | **Feral Spirit** 3 (R) * | *Bloodfen Raptor* 2 (F) | Demonfuse 2 (C) * | **Abusive Sergeant** 1 (C) | **Mortal Strike** 4 (R) * |
| **Dr. Boom** 7 (L) | **Feral Spirit** 3 (R) * | Bomb Lobber 5 (R) | Demonheart 5 (E) * | Amani Berserker 2 (C) | Rampage 2 (C) * |
| Draenei Totemcarver 4 (R) * | *Forked Lightning* 1 (C) * | Bomb Lobber 5 (R) | Demonheart 5 (E) * | Amani Berserker 2 (C) | Rampage 2 (C) * |
| **Echoing Ooze** 2 (E) | *Forked Lightning* 1 (C) * | Dragonkin Sorcerer 4 (R) | **Imp-losion** 4 (R) * | Dread Corsair 4 (C) | **Whirlwind** 1 (F) * |
| **Echoing Ooze** 2 (E) | **Lava Burst** 3 (R) * | *Evil Heckler* 4 (C) | **Mortal Coil** 1 (F) * | **Fel Reaver** 5 (E) | WEAPONS |
| **Flame Juggler** 2 (C) | **Lightning Bolt** 1 (C) * | **Flame Juggler** 2 (C) | **Mortal Coil** 1 (F) * | **Fel Reaver** 5 (E) | **Arcanite Reaper** 5 (F) * |
| **Flame Juggler** 2 (C) | **Lightning Storm** 3 (R) * | **Flame Juggler** 2 (C) | **Siphon Soul** 6 (R) * | Frostwolf Warlord 5 (F) | **Arcanite Reaper** 5 (F) * |
| Flesheating Ghoul 3 (C) | **Lightning Storm** 3 (R) * | Illidan Stormrage 6 (L) | | Gilblin Stalker 2 (C) | **Battle Axe** 1 (F) * |
| Goldshire Footman 1 (F) | | Mistress of Pain 2 (R) * | | *Hungry Crab* 1 (E) | |
| **Grim Patron** 5 (R) | | Mistress of Pain 2 (R) * | | Imp Master 3 (R) | |
| **Knife Juggler** 2 (R) | | Murloc Raider 1 (F) | | **Kor'kron Elite** 4 (F) * | |
| **Knife Juggler** 2 (R) | | **Murloc Tidehunter** 2 (F) | | **Kor'kron Elite** 4 (F) * | |
| **Leeroy Jenkins** 5 (L) | | **Murloc Warleader** 3 (E) | | **Leper Gnome** 1 (C) | |
| Scarlet Crusader 3 (C) | | Ogre Brute 3 (C) | | **Piloted Shredder** 4 (C) | |
| Scarlet Crusader 3 (C) | | **Questing Adventurer** 3 (R) | | *Pint-Sized Summoner* 2 (R) | |
| Secretkeeper 1 (R) | | **Ragnaros the Firelord** 8 (L) | | **Questing Adventurer** 3 (R) | |
| *Shattered Sun Cleric* 3 (F) | | *War Golem* 7 (F) | | **Questing Adventurer** 3 (R) | |
| Silvermoon Guardian 4 (C) | | Worgen Infiltrator 1 (C) | | *Recruiter* 5 (E) | |
| **Thunder Bluff Valiant** 5 (R) * | | Worgen Infiltrator 1 (C) | | Saboteur 3 (R) | |
| **Thunder Bluff Valiant** 5 (R) * | | Young Priestess 1 (R) | | **Shieldmaiden** 6 (R) * | |
| | | Young Priestess 1 (R) | | **Sylvanas Windrunner** 6 (L) | |

Table B.7: Statistics of each evolved deck, averaged per opponent.

| Evolved Druid VS | Games Won | Games Lost | Damage Dealt | Healing Done | Mana Spent | Cards Played | Turns Taken | Cards Drawn | Minions Played | Spells Cast | Hero Power | Weapons Equipped |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Midrange Hunter S18 | 14 | 2 | 32 | 0 | 21.4375 | 11.8125 | 7.0625 | 7.625 | 5.3125 | 4.375 | 2.125 | 0 |
| Mage Tempo S18 | 13 | 3 | 26.75 | 0 | 24.875 | 12.0625 | 7.375 | 8.1875 | 5.375 | 5 | 1.6875 | 0 |
| Aggro Paladin S18 | 12 | 4 | 32.625 | 0 | 21.4375 | 11.0625 | 6.9375 | 7.4375 | 5.1875 | 3.75 | 2.125 | 0 |
| Shadow Madness Priest S18 | 12 | 4 | 43.8125 | 0 | 27.9375 | 13 | 8.125 | 8.5625 | 6.4375 | 4.0625 | 2.5 | 0 |
| Oil Rogue S18 | 12 | 4 | 33.5 | 0 | 33.0625 | 14.125 | 8.5 | 9 | 7.3125 | 5.0625 | 1.75 | 0 |
| Mech Shaman S18 | 14 | 2 | 33.875 | 0 | 24.8125 | 12.125 | 7.6875 | 8.1875 | 5.8125 | 4.5 | 1.8125 | 0 |
| Warlock MalyLock S18 | 14 | 2 | 33.75 | 0 | 28.625 | 13.625 | 8.4375 | 8.875 | 6.25 | 5 | 2.375 | 0 |
| Control Warrior S18 | 12 | 4 | 33.1875 | 0 | 31.125 | 14.625 | 8.625 | 9.375 | 6.8125 | 5.375 | 2.4375 | 0 |
| *Average* | - | - | 33.69 | 0.00 | 26.66 | 12.80 | 7.84 | 8.41 | 6.06 | 4.64 | 2.10 | 0.00 |
| Evolved Hunter VS | Games Won | Games Lost | Damage Dealt | Healing Done | Mana Spent | Cards Played | Turns Taken | Cards Drawn | Minions Played | Spells Cast | Hero Power | Weapons Equipped |
| Midrange Druid S18 | 14 | 2 | 44.3125 | 3.5 | 27.5 | 11.6875 | 8.25 | 8.375 | 6.25 | 3.5 | 1.9375 | 0 |
| Mage Tempo S18 | 11 | 5 | 42.5 | 1.5 | 26.75 | 12.75 | 8.25 | 8.4375 | 5.9375 | 4 | 2.8125 | 0 |
| Aggro Paladin S18 | 11 | 5 | 43.0625 | 1 | 19.75 | 9.75 | 7 | 7.1875 | 4.6875 | 3.1875 | 1.875 | 0 |
| Shadow Madness Priest S18 | 12 | 4 | 60.1875 | 2.5 | 31.5625 | 14.25 | 9.25 | 9.625 | 6.5 | 4.875 | 2.875 | 0 |
| Oil Rogue S18 | 11 | 5 | 40.5 | 1 | 26.375 | 11.875 | 8.0625 | 8.125 | 6.4375 | 2.875 | 2.5625 | 0 |
| Mech Shaman S18 | 14 | 2 | 38.6875 | 1 | 22.0625 | 10.1875 | 7.375 | 7.5 | 4.5625 | 3.9375 | 1.6875 | 0 |
| Warlock MalyLock S18 | 15 | 1 | 40.5625 | 2 | 25.4375 | 11.5 | 8.125 | 8.375 | 5.5 | 3.625 | 2.375 | 0 |
| Control Warrior S18 | 13 | 3 | 41.0625 | 1 | 26.6875 | 12.875 | 8.625 | 8.625 | 6.0625 | 3.625 | 3.1875 | 0 |
| *Average* | - | - | 43.86 | 1.69 | 25.77 | 11.86 | 8.12 | 8.28 | 5.74 | 3.70 | 2.41 | 0.00 |
| Evolved Mage VS | Games Won | Games Lost | Damage Dealt | Healing Done | Mana Spent | Cards Played | Turns Taken | Cards Drawn | Minions Played | Spells Cast | Hero Power | Weapons Equipped |
| Midrange Druid S18 | 14 | 2 | 40.875 | 0.375 | 25.3125 | 11.375 | 7.625 | 7.625 | 6.125 | 3.0625 | 2.1875 | 0 |
| Midrange Hunter S18 | 12 | 4 | 48.6875 | 0.125 | 30.5 | 14.125 | 8.75 | 8.75 | 6.8125 | 4.25 | 3.0625 | 0 |
| Aggro Paladin S18 | 13 | 3 | 44.3125 | 0.5 | 23.75 | 12.0625 | 7.75 | 7.75 | 5.8125 | 3.0625 | 3.1875 | 0 |
| Shadow Madness Priest S18 | 8 | 8 | 60.0625 | 1.25 | 28.4375 | 13.0625 | 8.125 | 8 | 6.5625 | 3.625 | 2.875 | 0 |
| Oil Rogue S18 | 13 | 3 | 43.375 | 0.125 | 35.8125 | 14.375 | 9.0625 | 9 | 7.375 | 4.25 | 2.75 | 0 |
| Mech Shaman S18 | 13 | 3 | 37.8125 | 0.5 | 23.9375 | 10.75 | 7.8125 | 7.8125 | 5.6875 | 2.8125 | 2.25 | 0 |
| Warlock MalyLock S18 | 16 | 0 | 47.1875 | 1.125 | 33.8125 | 14.25 | 9.1875 | 9.1875 | 7.4375 | 3.5 | 3.3125 | 0 |
| Control Warrior S18 | 15 | 1 | 39.5625 | 1.75 | 31.875 | 12.8125 | 8.75 | 8.75 | 6.625 | 3.75 | 2.4375 | 0 |
| *Average* | - | - | 45.23 | 0.72 | 29.18 | 12.85 | 8.38 | 8.36 | 6.55 | 3.54 | 2.76 | 0.00 |
| Evolved Paladin VS | Games Won | Games Lost | Damage Dealt | Healing Done | Mana Spent | Cards Played | Turns Taken | Cards Drawn | Minions Played | Spells Cast | Hero Power | Weapons Equipped |
| Midrange Druid S18 | 16 | 0 | 25.625 | 0.25 | 20.875 | 10.9375 | 7.25 | 7.25 | 5.3125 | 3.0625 | 2.4375 | 0.9375 |
| Midrange Hunter S18 | 13 | 3 | 27.625 | 0 | 17.5 | 9.4375 | 6.5625 | 6.5625 | 5.25 | 2.4375 | 1.75 | 0.625 |
| Mage Tempo S18 | 13 | 3 | 39.125 | 0.25 | 26.375 | 12.9375 | 8.1875 | 8.1875 | 6.75 | 3.1875 | 2.875 | 0.625 |
| Shadow Madness Priest S18 | 10 | 6 | 46.625 | 0.5 | 25.6875 | 12.6875 | 7.875 | 7.875 | 6.8125 | 3 | 2.6875 | 1.0625 |
| Oil Rogue S18 | 14 | 2 | 27.125 | 0.125 | 20.625 | 10.875 | 7.1875 | 7.1875 | 6.25 | 2.375 | 2.1875 | 0.75 |
| Mech Shaman S18 | 11 | 5 | 31.0625 | 0 | 17.3125 | 9.75 | 6.6875 | 6.6875 | 5.625 | 2.25 | 1.875 | 0.625 |
| Warlock MalyLock S18 | 14 | 2 | 31 | 0.5 | 23.25 | 12 | 7.625 | 7.625 | 6 | 3.0625 | 2.75 | 1.0625 |
| Control Warrior S18 | 13 | 3 | 25.125 | 0.375 | 23.5 | 12 | 7.75 | 7.75 | 6.375 | 2.875 | 2.625 | 0.875 |
| *Average* | - | - | 31.66 | 0.25 | 21.89 | 11.33 | 7.39 | 7.39 | 6.05 | 2.78 | 2.40 | 0.82 |
| Evolved Priest VS | Games Won | Games Lost | Damage Dealt | Healing Done | Mana Spent | Cards Played | Turns Taken | Cards Drawn | Minions Played | Spells Cast | Hero Power | Weapons Equipped |
| Midrange Druid S18 | 12 | 4 | 43.8125 | 10.8125 | 24.4375 | 14.3125 | 8 | 9.625 | 6.0625 | 5.3125 | 2.9375 | 0 |
| Midrange Hunter S18 | 14 | 2 | 38.4375 | 9.6875 | 24.125 | 13.3125 | 7.75 | 8.9375 | 6.125 | 4.9375 | 2.25 | 0 |
| Mage Tempo S18 | 13 | 3 | 40.9375 | 8 | 25.6875 | 13.625 | 8 | 9.5 | 5.875 | 5.75 | 2 | 0 |
| Aggro Paladin S18 | 11 | 5 | 49.5 | 11.0625 | 21.5 | 13.5625 | 7.625 | 9.25 | 5.5 | 5.4375 | 2.625 | 0 |
| Oil Rogue S18 | 15 | 1 | 33.1875 | 6 | 23.625 | 11.5 | 7.25 | 7.625 | 6 | 4.125 | 1.375 | 0 |
| Mech Shaman S18 | 14 | 2 | 32.25 | 7.125 | 23.125 | 12.5 | 7.5625 | 8.5 | 5.8125 | 4.8125 | 1.875 | 0 |
| Warlock MalyLock S18 | 14 | 2 | 44.375 | 9.375 | 27.1875 | 15.125 | 8.375 | 9.625 | 6.75 | 5.875 | 2.5 | 0 |
| Control Warrior S18 | 14 | 2 | 35.3125 | 8.4375 | 27.4375 | 15.4375 | 9 | 10.3125 | 7.625 | 4.875 | 2.9375 | 0 |
| *Average* | - | - | 39.73 | 8.81 | 24.64 | 13.67 | 7.95 | 9.17 | 6.22 | 5.14 | 2.31 | 0.00 |
| Evolved Rogue VS | Games Won | Games Lost | Damage Dealt | Healing Done | Mana Spent | Cards Played | Turns Taken | Cards Drawn | Minions Played | Spells Cast | Hero Power | Weapons Equipped |
| Midrange Druid S18 | 13 | 3 | 30.3125 | 2.25 | 23.4375 | 10 | 7.5 | 7.5 | 6.625 | 1.9375 | 1.4375 | 1.4375 |
| Midrange Hunter S18 | 13 | 3 | 32.4375 | 1.125 | 24 | 10.8125 | 7.5625 | 7.375 | 7.1875 | 1.9375 | 1.6875 | 1.6875 |
| Mage Tempo S18 | 8 | 8 | 32.375 | 0 | 25.1875 | 11.8125 | 7.6875 | 7.8125 | 7.5625 | 2.375 | 1.875 | 1.9375 |
| Aggro Paladin S18 | 11 | 5 | 35.3125 | 0 | 19.0625 | 9.9375 | 7 | 7 | 6.0625 | 2 | 1.875 | 1.875 |
| Shadow Madness Priest S18 | 9 | 7 | 48 | 1.125 | 30 | 13.125 | 8.5 | 8.4375 | 7.875 | 3 | 2.25 | 2.25 |
| Mech Shaman S18 | 15 | 1 | 27.0625 | 0 | 23.25 | 11.125 | 7.875 | 8.4375 | 6.5625 | 2.5625 | 2 | 2 |
| Warlock MalyLock S18 | 15 | 1 | 28.25 | 0.375 | 25.125 | 10.8125 | 7.875 | 7.8125 | 7.0625 | 2.1875 | 1.5625 | 1.5625 |
| Control Warrior S18 | 15 | 1 | 20.3125 | 1.125 | 24.875 | 10.8125 | 8 | 8.1875 | 7.0625 | 1.875 | 1.875 | 1.875 |
| *Average* | - | - | 31.76 | 0.75 | 24.37 | 11.05 | 7.75 | 7.82 | 7.00 | 2.23 | 1.82 | 1.83 |
| Evolved Shaman VS | Games Won | Games Lost | Damage Dealt | Healing Done | Mana Spent | Cards Played | Turns Taken | Cards Drawn | Minions Played | Spells Cast | Hero Power | Weapons Equipped |
| Midrange Druid S18 | 12 | 4 | 65.25 | 8.9375 | 39.25 | 18.375 | 11.25 | 11.4375 | 6.5625 | 5.4375 | 6.3125 | 0.0625 |
| Midrange Hunter S18 | 14 | 2 | 66.1875 | 15.75 | 46.875 | 21.5625 | 12.625 | 14.125 | 8.125 | 6.4375 | 6.8125 | 0.1875 |
| Mage Tempo S18 | 5 | 11 | 62.375 | 6.9375 | 32.625 | 16.3125 | 10.125 | 10.625 | 5.75 | 5.3125 | 5.1875 | 0.0625 |
| Aggro Paladin S18 | 9 | 7 | 98.6875 | 11.125 | 45.4375 | 20.75 | 12.25 | 12.5 | 6.875 | 7.0625 | 6.5 | 0.3125 |
| Shadow Madness Priest S18 | 13 | 3 | 83.625 | 11.5625 | 49.9375 | 24.3125 | 14 | 14.875 | 8.3125 | 7.3125 | 8.25 | 0.4375 |
| Oil Rogue S18 | 11 | 5 | 60.75 | 8.5625 | 42.4375 | 19.6875 | 11.875 | 12.375 | 7.625 | 6.25 | 5.75 | 0.0625 |
| Warlock MalyLock S18 | 15 | 1 | 72 | 9.4375 | 48.375 | 21.8125 | 13.25 | 13.625 | 8.125 | 6.4375 | 7 | 0.25 |
| Control Warrior S18 | 9 | 7 | 72.5 | 9.625 | 49.8125 | 21.875 | 12.9375 | 13.9375 | 8.375 | 6.5625 | 6.75 | 0.1875 |
| *Average* | - | - | 72.67 | 10.24 | 44.34 | 20.59 | 12.29 | 12.94 | 7.47 | 6.35 | 6.57 | 0.20 |
| Evolved Warrior VS | Games Won | Games Lost | Damage Dealt | Healing Done | Mana Spent | Cards Played | Turns Taken | Cards Drawn | Minions Played | Spells Cast | Hero Power | Weapons Equipped |
| Midrange Druid S18 | 10 | 6 | 43.8125 | 0 | 32.9375 | 13.25 | 8.875 | 9.4375 | 7.8125 | 1.5625 | 2.875 | 1 |
| Midrange Hunter S18 | 10 | 6 | 36.5 | 0 | 27.0625 | 11.6875 | 8.0625 | 8.4375 | 6.8125 | 1.75 | 2.3125 | 0.8125 |
| Mage Tempo S18 | 9 | 7 | 30.5 | 0 | 27.6875 | 12.5625 | 8.0625 | 8.4375 | 7.0625 | 2.3125 | 2.375 | 0.8125 |
| Aggro Paladin S18 | 6 | 10 | 42.375 | 0 | 24.75 | 10.0625 | 7.25 | 7.4375 | 6 | 1.75 | 1.625 | 0.6875 |
| Shadow Madness Priest S18 | 14 | 2 | 31.5 | 0 | 33.9375 | 14.9375 | 9.5625 | 10.1875 | 8.1875 | 1.6875 | 3.8125 | 1.25 |
| Oil Rogue S18 | 11 | 5 | 38.3125 | 0 | 33.5 | 14 | 8.875 | 9.0625 | 8.1875 | 2 | 2.875 | 1.125 |
| Mech Shaman S18 | 12 | 4 | 28.3125 | 0 | 24.875 | 10.625 | 7.625 | 7.9375 | 6.3125 | 2 | 1.75 | 0.5625 |
| Midrange Hunter S18 | 12 | 4 | 39.375 | 0 | 32 | 13.875 | 8.8125 | 9.5 | 7 | 1.875 | 3.625 | 1.375 |
| *Average* | - | - | 36.34 | 0.00 | 29.59 | 12.63 | 8.39 | 8.80 | 7.17 | 1.87 | 2.66 | 0.95 |
| Evolved Warlock | Games Won | Games Lost | Damage Dealt | Healing Done | Mana Spent | Cards Played | Turns Taken | Cards Drawn | Minions Played | Spells Cast | Hero Power | Weapons Equipped |
| Midrange Druid S18 | 13 | 3 | 47 | 0.125 | 29.75 | 12.9375 | 8.0625 | 10 | 7.4375 | 3.5 | 2 | 0 |
| Midrange Hunter S18 | 9 | 7 | 43.125 | 0 | 22.8125 | 10.125 | 6.875 | 8.125 | 6.6875 | 2.1875 | 1.25 | 0 |
| Mage Tempo S18 | 9 | 7 | 41.0625 | 0 | 29 | 11.5625 | 7.875 | 9.1875 | 7.25 | 2.9375 | 1.375 | 0 |
| Aggro Paladin S18 | 10 | 6 | 45.875 | 0.1875 | 22.8125 | 9.4375 | 6.9375 | 7.9375 | 6.3125 | 2.125 | 1 | 0 |
| Shadow Madness Priest S18 | 11 | 5 | 53.8125 | 0.125 | 31 | 12.75 | 8.3125 | 10.0625 | 7.75 | 3.25 | 1.75 | 0 |
| Oil Rogue S18 | 11 | 5 | 35.9375 | 0 | 26.75 | 10.8125 | 7.5 | 8.6875 | 7 | 2.625 | 1.1875 | 0 |
| Mech Shaman S18 | 12 | 4 | 43.3125 | 0 | 24.8125 | 10.3125 | 7.5 | 8.75 | 6.5 | 2.5625 | 1.25 | 0 |
| Control Warrior S18 | 11 | 5 | 33.875 | 0 | 28.6875 | 11.6875 | 8 | 9.375 | 7.4375 | 2.75 | 1.5 | 0 |
| *Average* | - | - | 43.00 | 0.05 | 26.95 | 11.20 | 7.63 | 9.02 | 7.05 | 2.74 | 1.41 | 0.00 |