

Toward a Competitive Agent Framework for Magic: The Gathering

Michael Bowling and Shaelyn Rivers-Green and Deric Siglin and Lori Alvin and Chris Alvin*

Furman University
chris.alvin@furman.edu

Abstract

Magic: The Gathering is a collectible card game that is traditionally a human-based game: two or more players compete with their own custom deck to eliminate their opponent(s). *Magic* has again surged in popularity with the release of *Magic: Arena*, a digital substitute for the physical card game. What *Magic: Arena* lacks is meaningful gameplay with a competitive, intelligent agent. We propose a strategy-based framework for basic decision making in *Magic: The Gathering*. We describe these procedures and provide experimental evidence for the effectiveness of our approach.

1 Introduction

Magic: The Gathering (*Magic*, for short) is viewed as the pinnacle of competitive collectible card games due to its long-lasting success. Wizards of the Coast, designers and publishers of *Magic*, recognized the early success of the game with both gamers and collectors. Other gaming companies recognized this success in the 1990s with different competing efforts; many were unsuccessful, while some gained traction. More than two decades later, *Magic* is still considered to be the leader in competitive card games even in the digital realm with their proprietary software *Magic: Arena*. Games such as *Hearthstone* and *Legends of Runeterra* have made inroads against *Magic*, but the longevity, richness, and classic nature of *Magic* always seems to win even with the digital client *Magic: Arena*. As boardgames and card games continue to be digitized, there is often a desire to make such games playable for a single player. That is, digital game clients need to offer a competitive agent player for realistic competitive play. This is a particularly difficult problem since *Magic: The Gathering* has been shown to be Turing complete (Churchill, Biderman, and Herrick 2019).

In this paper we summarize *Magic* decks and gameplay (Section 2) and motivate development of a decision-making framework in Section 3. In Section 4, we describe our framework and the algorithms for the decisions it can make. We conclude with brief experimental analyses in Section 5 and a discussion of limitations and future considerations (Section 7).

2 Magic Decks and Gameplay

Magic is a rich and complex game: the comprehensive rule set (Wizards of the Coast 2020c) is more than 200 pages and is consistently updated and expanded. In this section we give a terse overview of the game and how it is played.

The Goal. The overarching goal in a game of *Magic* is to reduce an opponent’s life total from 20 down to 0. Life is deducted from a player for many reasons including, but not limited to, damage from spells and creatures.

Decks. In this paper we consider *constructed* best-of-one (BO1) games. In this format, a deck consists of at least 60 cards. Except for basic lands, a constructed deck may not have more than 4 copies of any card. Since *Magic* is a consistently evolving game, our discussions will consider cards that were in the *Standard* format on *Magic: Arena* (Wizards of the Coast 2020b) when the research was completed, but are now classified as *Historic*. A deck of 60 cards commonly contains between 23 and 27 lands with the remaining cards being creatures, enchantments, artifacts, and other spells. Land cards supply the player with ‘mana’: the resource used to ‘cast’ spells. All cards other than land are considered spells and thus have a corresponding *casting cost*: resources a player must pay to play a card.

The game of *Magic* uses five colors: Black (B), Blue (U), Green (G), Red (R), and White (W); each of the five colors maintains a trademarked symbol which we do not use here. Mana is created by ‘tapping’ (turning a card sideways) lands that produce the corresponding color mana (e.g., swamps produce black mana, plains produce white mana, etc.). There are many spells that require ‘generic’ mana to be cast: mana of any color can be used to pay for such mana costs. For example, the enchantment *Cavalcade of Calamity* (Wizards of the Coast 2020a) can be cast with 1 generic and 1 red mana. We abbreviate the casting cost of *Cavalcade of Calamity* 1R and say it has a *converted mana cost* (CMC) of 2.

Gameplay. In a BO1 game there are two players. Each player is unaware of the contents of the other player’s deck, although traditionally decks follow a ‘meta’ or common deck-type. Both players individually shuffle the contents of their decks and draw the top 7 cards. A player is then tasked with a decision to keep their current hand or *mulligan* by reshuffling their hand and library together and drawing 7 new cards. We assume the use of the *London mulligan* in

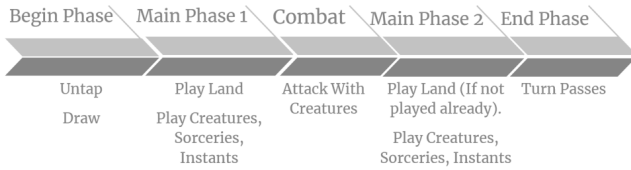


Figure 1: A summary of the phases in a *Magic* turn.

which a player places m cards on the bottom of their deck, where m is the number of times they have performed a mulligan (e.g., $m = 0$ if a player does not mulligan, but if a player mulligans once, $m = 1$ and the player will begin with $7 - m = 6$ cards in their opening hand).

We do not describe all the intricacies of a turn in *Magic* and instead focus on a high-level overview of the phases depicted in Figure 1. Each turn begins with players ‘untapping’ cards, which allows the player to use these resources again, and drawing one card from their deck (the first player does not draw a card on their first turn). In the first main phase, the player may play a land as well as any spells in which they have sufficient mana. The combat phase allows the current player to initiate combat against the opponent by choosing attacking creatures. The opponent then decides how to use their own creatures to block the opponent’s attack. The player then has a second main phase in which to play a land (if they did not play a land in main phase 1) and play any spells in which they have sufficient mana. The final phase officially concludes a player’s turn and passes the turn to the opponent.

The Goal of this Work. As this is a work in progress we do not describe all decision-making required in a game of *Magic*. The focus of this paper is a framework for making two fundamental and difficult decisions: (1) whether a player should mulligan and (2) which card to play next. Section 4 describes our representation scheme and our algorithms for making these decisions.

3 The Bot in Magic: Arena

While *Magic: Arena* boasts a “Bot Match” mode in which a user can play against an autonomous opponent named *Sparky*, it is not meaningfully competitive: *Sparky* sometimes makes choices that a typical player would not. For example, it is often the case that a player needs to evaluate the largest threat on the board and then attempt to either remove it or play around it. In a game against *Sparky*, we noticed that the bot pacified (prohibits a creature from attacking and blocking) the largest numeric threat on the board ignoring the real threat: the only creature with flying. A win using the flying creature followed shortly after.

Even for human players *Magic* is a difficult game to master and mistakes are made. However, when an intelligent agent makes a clearly flawed play, players perceive the agent as non-competitive. By most accounts, *Sparky* seems to be a reasonable AI opponent, but there are times in which its autonomous nature is revealed. *Sparky* is our motivation for developing our Strategies Framework as a means of decision-making for our agent play.

Turn	1	2	3
Land	Swamp (B)	Plains (P)	Swamp (B)
Play	Vampire of the Dire Moon (B) <i>Lifelink</i> (Damage dealt by this creature also causes you to gain that much life)	Ajani's Pridemate (1W) Whenever you gain life, put a +1 / +1 counter on Ajani's Pridemate.	Bloodthirsty Aerialist (1BB) <i>Flying</i> Whenever you gain life, put a +1 / +1 counter on Bloodthirsty Aerialist.

Figure 2: An example synergistic 3-turn sequence.

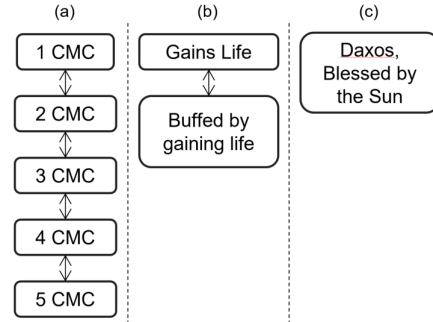


Figure 3: Three strategies for the lifegain deck (Djinnmaster 2020a).

4 The Strategies Framework

One way to build a deck in *Magic* is to identify synergies among cards. For example, in a life-gain deck (Djinnmaster 2020a), we attempt to gain life either through spells or creatures. The ability to gain life then synergizes with creatures that grow in power and toughness each time you gain life. *Power* refers to the amount of damage a creature deals in combat to an opposing creature or player when it attacks and *toughness* refers to the amount of damage needed to destroy the creature. As an example, *Bloodthirsty Aerialist* (Wizards of the Coast 2020a) is a creature with power 2 and toughness 3 (a 2/3 creature) with the text: “Flying. Whenever you gain life, put a +1/+1 counter on Bloodthirsty Aerialist”. With life-gain, it does not take long for *Bloodthirsty Aerialist* to become a deep threat by growing in power and toughness.

Any deck allows a player to construct sequences of desired cards to play based on particular synergies in a deck. For example, it is advantageous to play one or two creatures that gain life followed by a *Bloodthirsty Aerialist*. As an example, we observe in Figure 2 a Turn 1 swamp and *Vampire of the Dire Moon* (Wizards of the Coast 2020a); Turn 2 plains and *Ajani's Pridemate* (Wizards of the Coast 2020a); Turn 3 swamp and a *Bloodthirsty Aerialist*. With this sequence of plays it is conceivable that at the beginning of Turn 4 the *Bloodthirsty Aerialist* might be a 3/4 (3 power and 4 toughness) and *Ajani's Pridemate* might be up to a 4/4 based on *Vampire of the Dire Moon's* lifelink.

Our strategy framework attempts to formalize these synergies and thus make decisions exploiting such synergies among cards in a deck. We say a *strategy node* is a set of constraints describing a subset of cards in a deck D . Figure 3(a) depicts a constraint as being any card that costs 1

mana (1 CMC) such as *Vampire of the Dire Moon*. Another example constraint is if a creature must be capable of gaining the player life such as *Daxos, Blessed by the Sun* (Wizards of the Coast 2020a). We note that since *Daxos, Blessed by the Sun* gains a player life when a creature enters the battlefield or when a creature is removed, it is its own strategy (Figure 3(c)). Generally, we say a *strategy* is a bidirectional sequence of m strategy nodes $S = (n_1, n_2, \dots, n_m)$.

As our goal is to consistently play toward our strategies, we define the idea of completing steps in a strategy. If a *Magic* card c in play satisfies the constraints of a strategy node n in a strategy S ($n \in S$), then n is a *completed node* by c . If all strategy nodes in a strategy S are completed, we say S is a *completed strategy*. If a *Magic* card c in a player’s hand satisfies the constraints of a strategy node n in a strategy S ($n \in S$) and the mana resources are available to pay for the spell, then n is a *completable node* by c .

We say that a deck D defines a corresponding set of strategies \mathcal{S}_D . A subset of strategies for our life-gain deck (Djinnmaster 2020a) are shown in Figure 3.

4.1 The Decision to Mulligan

For a deck D and its corresponding set of strategies \mathcal{S}_D , our goal is to decide if a collection of cards $H \subset D$ (‘hand’ in card game parlance) is satisfactory to begin the game. Intuitively, a satisfactory opening hand is one that will play toward completing a strategy or set of strategies. Thus, we define two measures of a hand H working toward completing strategies \mathcal{S}_D : *completable width* and *completable depth*.

Given a hand H , we compute the number of strategies in which at least one of the strategy nodes can be completed by cards in H ; we refer to this value as w . Since the number of strategies for a deck ($|\mathcal{S}_D|$) may differ, we compute the *completable width* as a ratio: $w/|\mathcal{S}_D|$. For completable depth, we count the number of nodes that can be completed by H in all strategies \mathcal{S}_D ; call this value d . We also define $N_{\mathcal{S}_D}$ to be the overall number of nodes defined by all strategies of a deck (\mathcal{S}_D). We compute *completable depth* for H as $d/N_{\mathcal{S}_D}$.

As an example, assume the strategies in Figure 3 for the life-gain deck and a hand containing 3 swamps, 1 plains, *Vampire of the Dire Moon*, *Ajani’s Pridemate*, and *Bloodthirsty Aerialist*. We compute the completable width as $2/3$ since these cards can satisfy constraints and can complete nodes in strategy (a) and (b), but not (c). There are 8 total nodes in the three strategies in Figure 3 and we identify that the cards in hand can complete 3 nodes in (a), 2 nodes in (b), and 0 in (c). The completable depth is thus $5/8$.

It is important to recall that with each turn, a player draws a fresh card from their deck. Thus, the mulligan decision must be made with imperfect information. We wish for an opening hand to result in a competitive game. Hence, the intent of these measures is to assess the versatility of the current hand with respect to the overall strategies: we wish for a hand to target a particular strategy (completable depth), but also be open to pursuing other strategies (completable width) based on the uncertainty of card draw.

The decision to mulligan is thus based on comparing *both* the completable width and completable depth to a mulligan

threshold, $T \in [0, 1]$. We found $T = 0.25$ to be an experimentally reasonable value. Considering the example in the previous paragraph, we find the hand to be satisfactory and decline to mulligan since both $2/3 > T$ and $5/8 > T$. As an extreme example, we would mulligan a hand containing 7 lands since no nodes would be completable (completable width and completable depth are 0); similarly for a hand containing no lands since we would not be able to play any cards in hand.

4.2 The Decision of Which Card to Play

Our strategy-based approach proposes a simple algorithm for making the decision of which *Magic* card to cast. The input to our algorithm consists of the current game board state \mathcal{B} (i.e., all current cards in play including lands, creatures, etc.), the player’s hand H , and the set of strategies for the deck. The algorithm computes (a) what land to play from H , if applicable, and (b) which card to play from H . In general, we score each strategy and use those scores to ‘vote’ for an individual card in H to play; we discuss details below.

Our algorithm proceeds as follows. We first identify our land resources from \mathcal{B} and the possible land resources in H . Next, for all cards in H , we identify the subset of cards that can be played this turn with available resources, $P \subseteq H$.

Let S_m be a strategy with m nodes. Our goal is to compute *score* (S_m), a score for each strategy. Foremost, *score* (S_m) begins with the number of constituent nodes: $5 \cdot m$. Our rationale behind weighting toward longer strategies is that there are usually larger payoffs at the end of more intricate strategies. We next consider completed and completable nodes in S_m . We then determine how many nodes in S_m are completed considering \mathcal{B} as well as the number of nodes that are completable considering P . We take the sum of these completed and completable nodes, n_c , and compute a ratio: $100n_c/m$. Hence, *score* (S_m) = $5m + 100n_c/m$.

We choose an individual card using each strategy’s weight as a ‘vote’. We initialize the score for each playable card: $\forall c \in P, \text{score}(c) := 0$. For each strategy, $S_m \in \mathcal{S}_D$, we choose the node with greatest index that can be completed by one or more playable cards (call this playable set $V \subseteq P$). We then add the score of S_m to the score of each card: for all $c \in V, \text{score}(c) += \text{score}(S_m)$. Our choice of card to play is then $c_{\text{play}} := \text{argmax}_{c \in P} \text{score}(c)$; c_{play} also dictates a land card $c_{\text{land}} \in H$ that must be played to cast c_{play} .

As an example, consider the sequence of plays in Figure 2 completing the 3 top-most nodes of the strategy in Figure 3(a). We compute the score of this strategy as $5 \cdot 5 + \frac{100 \cdot 3}{5} = 85$. Now assume our hand contains one swamp and two *Twinblade Paladins* (Wizards of the Coast 2020a), a creature with mana cost 3W and grows by its controller gaining life. In this case, strategy (a) in Figure 3 would associate a score of 85 toward each Paladin in hand while strategy (c) would contribute 0 since the card in hand is not *Daxos, Blessed by the Sun*. Strategy (b) would contribute $5 \cdot 2 + \frac{100 \cdot 2}{2} = 110$ toward each Paladin.

Table 1: Descriptors of interior table data in Table 2 and Table 3.

Win % On the Play No Mulligan	Win % On the Play Mulligan	Total Games in Match
Win % On the Draw No Mulligan	Win % On the Draw Mulligan	Overall Win %

Table 2: Results of first-to-100 matches for agents in the left column using Life-Gain (LG) and Mono-Red (MR) decks; inner tables described by the template in Table 1.

	Forge LG			Forge MR			Strategy LG		
Forge MR	35.00	26.31	139						
	21.43	0.00	28.06						
Strategy LG	58.62	38.10	198	100.0	90.0	122			
	57.58	31.25	49.49	84.84	67.65	81.97			
Strategy MR	25.58	14.29	126	61.54	50.00	179	14.29	0.00	112
	9.52	0.00	20.63	64.38	22.22	55.87	0.00	0.25	10.71

5 Experimental Analyses

We implemented our framework leveraging the open source *Forge* codebase (Collectible Card Game Headquarters 2020). *Forge* is a tool supported by *Magic* enthusiasts that allows a player to build decks across *Magic*’s history consisting of more than 25000 cards. *Forge* is unique in that it is a tool that implements some bot functionality and thus provides a reasonable proof-of-concept testing ground for our strategy-based approach compared to a tool like *cockatrice* (Cockatrice Consortium 2020) which is simply a cross-platform virtual tabletop for online play among human competitors.

Using Mono-Red (MR) (Djinnmaster 2020b) and Life-Gain (LG) (Djinnmaster 2020a) decks, we set our strategy-based decision processes against *Forge*’s AI. Our implementation can supersede *Forge*’s implementation about when to mulligan and which cards to play; all other decisions (e.g. attacking, blocking, etc.) used *Forge*’s procedures.

Each match is a “first to 100” set of BO1 games in which play stops when one competitor reaches 100 wins; hence, the number of BO1 games in a first-to-100 match may not be consistent among matches. In this format, the loser of the previous game goes first (“on the play”) in the next game and the winner of the previous game will go second (“on the draw”) in the next game. It is well-known in *Magic* that a player being “on the draw” gives a higher probability of loss compared to “on the play”. We observe in Table 2 that this intuitive rule holds. *Magic* is a game of variance due, in part, to the land system and size of a standard deck (60 cards). However, over many iterations we expect an agent to win approximately 50% of games against itself (a self-mirror match). We report reasonable win-rates (42.53%, 47.37%, 49.75%, and 49.75%) for the two agents playing self-mirror matches with our two decks.

We now consider all other match results reported in Table 2. Foremost, we observe the relative power of the LG deck compared to the MR deck: win-rates for MR against LG are low at 10.71%, 18.03%, 20.63%, and 28.06%. We attribute these win-rates to MR hitting hard and fast while

Table 3: Results of first-to-100 matches with mulligans disallowed for agents in the left column using Life-Gain (LG) and Mono-Red (MR) decks; inner tables described by the template in Table 1 minus the middle column.

	Forge LG		Forge MR		Strategy LG	
Strategy LG	47.47	190	83.33	129		
	47.25	47.37	75.76	77.52		
Strategy MR	27.00	134	69.12	167	26.26	134
	20.59	25.37	53.53	59.88	22.86	25.37

LG gains life attempting to withstand the MR assault. The life-gain sustains the deck into the mid-game while growing larger creatures to eventually take over the game.

We evaluate the effectiveness of the strategy-based agent compared to *Forge*’s agent by considering matches with a common opponent. We use a 1-tailed, 2-sample proportion test with $H_0 : p1 - p2 = 0$, where $p1$ is the proportion of wins by the strategy deck (S) and $p2$ corresponds to wins with the *Forge* (F) deck. In LG matches against the FMR deck, we had a p -value of 0.028 indicating that our SLG deck performed better than the FLG deck; when comparing against the SMR deck, the p -value was 0.0185 again indicating the SLG deck performed better. We note no significant difference between SLG and FLG in direct play, as the SLG deck won 49.49% of the games. We note that there was no statistical significance between the SMR and FMR decks when compared against either the FLG or SLG decks, however in direct play the SMR deck won 55.87%. As a work in progress, these win-rates indicate our strategy-based approach is competitive with *Forge* when using the same deck, but is not exemplary. Hence, we have room to evolve and improve.

To consider the effectiveness of our mulligan procedures, we executed best-to-100 matches between our strategy-based agent and *Forge* (Table 3) except that our agent was not allowed to mulligan. We compare corresponding cells in Table 2 and Table 3. In some cases we observe a moderate increased win-rate with mulligans disallowed for MR and slight decrease for LG; these differences were not significant. We attribute this variation to MR being a single-color deck that can function well on few lands and thus a mulligan to start the game with one fewer card may be detrimental. In comparison, LG is a two-color deck and thus prone to land issues in the early game if a better hand is not allowed. Regardless, LG is a robust deck compared to MR.

6 Related Works

For an established game like *Magic: The Gathering*, the literature is surprisingly sparse when it comes to coordinated, automated gameplay as we have implemented and tested. Surprisingly, much of the literature arises as student projects and not peer-reviewed publications. We begin with peer-reviewed works from Peter I. Cowling and his co-authors.

In (Ward and Cowling 2009; Cowling, Ward, and Powley 2012), Cowling, et al. apply a Monte Carlo Tree search to *Magic*. *Magic* is a game with imperfect information (e.g., opponent hand, next card a player will draw, etc.) and thus

one of the main criticisms is the use of determinization to expand the tree during the Monte Carlo Tree search. However, it is common practice for players of *Magic* to “reason over clairvoyance” (Russell and Norvig 2009). In particular, players reason over past play experiences with their deck, speculate on cards in an opponent’s hand, speculate on cards in an opponent’s deck list, or on drawing a particular card. Hence, determinization is common practice by human players of *Magic*. No doubt the work of Cowling, et al. has moved automated decision making in *Magic* forward. However, their contributions focus on searching a space, an expensive operation considering the number of decisions a typical player makes in a game. It is unclear if the authors’ algorithms are efficient enough for a competitive best 2-of-3 match which is allotted no more than an hour in practice. In contrast, our contribution provides a heuristic for strategic play with a simpler, more efficient data structure and a method of reasoning over that structure. While our contribution lacks algorithms for all decision-making procedures required for a typical game of *Magic*, it describes an underlying framework to implement those decisions: attacking, blocking, and “playing around the opponent.”

It is also the case that (Ward and Cowling 2009; Cowling, Ward, and Powley 2012) focus entirely on gameplay decisions during play. While the algorithms they provide extend to most of the decisions in a typical *Magic* game, these works do not mention or analyze one of the most important decisions: the mulligan. Instead, their experiments defined and used a set of acceptable fixed permutations of cards in their decks. While their experiments are statistically relevant, we believe that our contribution helps fill this important decision-making gap.

Chodoriwsky (Chodoriwsky 2006) describes representing a deck using a synergy graph: an undirected graph in which each card is a vertex and each edge represents a synergistic relationship between the cards. As a result, the author is able to compute a synergy metric for a given deck as well as other measures related to synergistic locality. While this is an interesting approach to evaluating the strength of a deck, the work provides a static analysis of constituent cards in a deck. In total, it provides a data structure that may serve as the foundation for automated deck construction as well as strategy generation; however, it does not address the issues of turn-by-turn decision-making as we have done.

Deckbuilding is another area of interest in which there is existing research. In (Bjørke and Fludal 2017), the authors describe with few details their genetic approach to constructing the best possible *sealed pool* deck: ‘sealed’ is a draft format in which a player receives six 15-card booster packs and attempts to make the best deck possible for play against other sealed decks. The evaluation function for this work ended up being actual gameplay of each deck in the Forge (Collectible Card Game Headquarters 2020) framework (the same framework we used for our experiments in Section 5). In their genetic approach, successful decks propagate to future generations. While the work in (Bjørke and Fludal 2017) is admirable, our goal is to establish a framework for gameplay, not deckbuilding.

Existing literature including (Fink, Pastel, and Sapro

2015; Pawlicki, Polin, and Zhang 2014) attempt to use machine learning techniques to predict the monetary value of a card based on its features: card type, keywords, mana cost, rarity, etc. A work by Zilio, et al. (Zilio, Prates, and Lamb 2018) attempts to evaluate if a card properly predicts where it falls on the *Magic* “color pie” (Rosewater 2017). These ideas are in stark contrast with our evaluation of cards. As described in the Section 3, one of the most impact aspects of *Magic: The Gathering* is evaluating the perceived threat of a card. While our approach performs card evaluation based on similar features, our goal in evaluating the cards is for purposes of gameplay, not investment purposes or correctness of card design.

7 Conclusions and Future Considerations

Magic: The Gathering requires constant analysis and decision-making. This work demonstrates that a strategy-based framework can be used to make many important decisions: when to mulligan and which cards to play. As a work in progress we have focused on essential decisions required from a player in a typical game; for example, we implemented attacking and blocking algorithms, but did not include them in our tests. While our strategy system is effective for these decisions, we have yet to consider how the strategy can be used, for example, to identify an opposition’s deck archetype. Intuiting the composition of an opponent’s deck is important to “play around” anticipated spells. For example, if an opponent plays few creatures but uses colors that have spells allowing the opponent to destroy all creatures, we may wait to play any creature spells until such a spell is cast. Our approach also casts spells at “sorcery speed”; we do not consider casting spells at “instant-speed”. Future iterations will synergize our strategy-based framework with these other decision procedures.

8 Acknowledgements

This work was supported by the Furman University Department of Computer Science and The Furman Advantage.

References

- Bjørke, S. J., and Fludal, K. A. 2017. Deckbuilding in magic: The gathering using a genetic algorithm. Master’s thesis, “Norwegian University of Science and Technology”.
- Chodoriwsky, J. 2006. Synergy in magic: The gathering an application of graph theory.
- Churchill, A.; Biderman, S.; and Herrick, A. 2019. Magic: The gathering is turing complete. *CoRR*. <http://arxiv.org/abs/1904.09828>.
- Cockatrice Consortium. 2020. Cockatrice. <https://cockatrice.github.io/>.
- Collectible Card Game Headquarters. 2020. Forge - slightly magic. <https://www.slightlymagic.net/wiki/Forge>.
- Cowling, P. I.; Ward, C. D.; and Powley, E. J. 2012. Ensemble determinization in monte carlo tree search for the imperfect information card game magic: The gathering. *IEEE Transactions on Computational Intelligence and AI in Games* 4(4):241–257.

- Djinnmaster. 2020a. Lifegain deck. <https://www.mtgoldfish.com/deck/visual/3057086>.
- Djinnmaster. 2020b. Mono red deck. <https://www.mtgoldfish.com/deck/visual/3084492>.
- Fink, D.; Pastel, B.; and Sapra, N. 2015. Predicting the strength of magic: The gathering cards from card mechanics. *Cs 229: Machine Learning Final Project, December 2015*.
- Pawlicki, M.; Polin, J.; and Zhang, J. 2014. Prediction of price increase for magic: The gathering cards.
- Rosewater, M. 2017. Mechanical color pie. <https://magic.wizards.com/en/articles/archive/making-magic/mechanical-color-pie-2017-2017-06-05>.
- Russell, S., and Norvig, P. 2009. *Artificial Intelligence: A Modern Approach*. USA: Prentice Hall Press, 3rd edition.
- Ward, C. D., and Cowling, P. I. 2009. Monte carlo search applied to card selection in magic: The gathering. In *2009 IEEE Symposium on Computational Intelligence and Games*, 9–16.
- Wizards of the Coast. 2020a. Gatherer—the magic card database. <https://gatherer.wizards.com/>.
- Wizards of the Coast. 2020b. Magic: Arena. <https://magic.wizards.com/en/mtgarena>.
- Wizards of the Coast. 2020c. Magic:the gathering: Rules. <https://magic.wizards.com/en/game-info/gameplay/rules-and-formats/rules>.
- Zilio, F.; Prates, M.; and Lamb, L. 2018. Neural networks models for analyzing magic: The gathering cards. In Cheng, L.; Leung, A. C. S.; and Ozawa, S., eds., *Neural Information Processing*, 227–239. Cham: Springer International Publishing.