

Deck-Building Methods for the *Magic: The Gathering* Booster Draft Format

Nathaniel Gordon¹

¹Northeastern University
360 Huntington Ave
Boston, MA 02115
gordon.n@northeastern.edu

Abstract

The booster draft format is a widely popular format in both the physical and digital variations of popular trading card game *Magic: The Gathering*. In this paper, the deck-construction component of the booster draft play format is formatted as a combinatorial optimization problem. Both an evaluation function and several heuristic-based optimization schemes for generating solutions the problem are proposed. Experimentally-generated decks produced by applying annealing and genetic evolution methods to human-generated draft pools were shown outperform a random control in a variety of critical metrics, including optimal mana cost curve and expert-informed card rankings.

Introduction

The rise in popularity of the trading card game *Magic: The Gathering* (MTG) has only been accelerated by the 2018 release of *Magic: The Gathering Arena*, a digital version of the classic product. While publisher Wizards of The Coast (WotC) has not released official player count data, popular streaming website Twitch reported an average monthly viewer count of 10 million just a year after the game's release, already nearing half the viewership of popular competitor *Hearthstone* (ESPN 2020).

Arena's popularity has spurred interest in the development of autonomous agents, or 'bots' to autonomously play MTG. From the perspective of algorithm design, the implementation of such a bot requires the development of agents to solve several particular problems, each presenting different challenges.

One of *Arena's* most popular game modes is *booster draft*, consists of three main phases:

1. *Drafting*: Players take turns drafting from randomized packs of cards. Each player's selections contribute to their *draft pool*, the selection of cards they will be limited to in subsequent phases.
2. *Deck-building*: Players select a subset of cards from their draft pool to construct a deck to be used in the final phase.
3. *Dueling*: Players are paired up to compete in a battle using the decks they have constructed. Typically, winners are

determined using the player's record after competing in four or eight duels against other players.

In this paper, several methods for constructing high-performance decks during the deck-building phase of MTG booster draft is proposed. To achieve this, a evaluation function is developed in order to generate an estimate of deck quality. Then, this evaluation function is applied to several combinatorial optimization methods to produce theoretically highly-performant decks. Finally, an experiment is run to compare these methods with regard to how they satisfy various metrics of the evaluation function.

Background

The rules of MTG vary widely across the various official and unofficial play formats for the game. The key rules of booster draft will define the approach to optimizing deck-building for this particular format.

A player or autonomous agent attempting to construct a deck for a booster draft is given a draft pool. This pool consists of 45 cards that were selected during the drafting phase of gameplay. Each draft in the pool is likely selected with some overarching philosophy, which could be influenced by the other cards in the pack, knowledge of competitor's philosophies, or personal preference. Furthermore, all 45 of the pool's cards belong to a particular MTG *set*. A set typically consists of around 300 card types, and is designed by WotC to be a balanced card ecosystem for MTG play to occur.

To compete in the dueling phase of a booster draft, a deck of no less than 40 cards must be constructed. These cards may be selected from either 1), the player's draft pool or 2), an unlimited supply of the 5 basic land cards. While there is no limit to maximum deck size (in physical formats, the ability for the deck to be shuffled governs the theoretical maximum), the player is encouraged to opt for the minimum size of 40 to reduce variance in their deck's performance when cards are drawn randomly during the dueling phase. Additionally, no more than 4 copies of any card (besides basic lands) may appear in the deck. This is extremely unlikely to be a relevant rule in booster draft, however, as a player is in control of what cards they add to the draft pool.

A final noteworthy constraint to the booster draft format is the limited time allowed for deck construction to occur.

While the time allowance for this phase has varied, an objective for this paper will be to generate methods that reliably terminate in under one minute.

Related Work

Deck-building in trading card games is often framed as a combinatorial optimization problem. Common approaches to solving such a problem include heuristic methods, metaheuristic search methods, and sequential decision making approaches. Some heuristic searches are governed by the rigorous rules and synergy rankings present in the cards (Stiegler et al. 2016). Others choose to use human-generated rankings to evaluate the popularity of each card, a potential indicator of how likely it should be included in the deck (Karsten 2017b). Alternatively, metaheuristic search methods rely on simulation to generate win-rate statistics for proposed decks. An implementation of this method with a genetic algorithm is presented, but the simulation time required to perform deck analysis makes them untenable for the time-constraints presented in booster draft (Garcia-Sanchez et al. 2016). Another approach is to formulate the deck construction process as a sequential decision-making problem such as an Markov Decision Problem (MDP). While this method has been shown to run in real-time scenarios, it relies on lengthy training times in advance (Chen et al. 2018). The comparatively smaller optimization space and lesser reliance on synergy present in the booster draft format makes many of the more complex and computationally intensive methods ill-suited to the task.

Deck Evaluation

Card Features

The many cards of MTG are possess notorious levels of depth in terms of the rules implications present on any given card. Cards that appear identical in many categories may differ in critical, yet subtle ways that may cause them to be drawn to completely different deck-building archetypes. However, the limited number of card types available in the booster draft format (and the additional limits of the draft pool) make powerful synergies between cards much less likely to occur. Consequently, the deck-evaluation scheme proposed will not examine multi-card synergies beyond a few critical metrics, but will instead place greater emphasis on the power of each card in the deck independently.

Color MTG cards belong to one (or none) of the five colors of magic: black, blue, green, red, and white. In order to play cards of a given color, mana sources of the corresponding color, most often land cards, must be present.

In the deck construction process, the player must balance including the most powerful cards from the draft pool in their deck while not diluting the deck with an excess of colors to supply mana for. A general rule of booster draft deck construction is to only select cards belonging to exactly two of the colors, with limited exceptions in specific sets where drawing from three colors is acceptable, but not ideal (Karsten 2018).

Land-to-Spell Ratio In a similar vein to color-count issues, the balance between land cards (which produce mana) and spells (which require mana to be played) is a critical concern. The long-time MTG wisdom of playing 17 lands and 23 spells is supported by a statistical analysis including both basic probabilistic reasoning and a regression applied to top-performing decks (Karsten 2017a).

Mana Cost While closely tied to color, total mana cost is also a pertinent consideration when constructing a deck. While high mana cost cards are typically more likely to influence the outcome of a game, MTG limits the number of lands cards a player may play per turn to one, making it challenging to play these powerful cards early into a game.

The deck-building consideration is the following: a deck with a lack of high mana cost cards will be at a power disadvantage as the game goes on, while an overabundance may increase vulnerability early in the game. To address this, a *curve* metric that describes the ideal number of cards with each mana cost to use in a deck has been crafted (Karsten 2014).

Power Ranking With the much lower impact of synergistic interactions in the booster draft format, the power assigned to a given card for evaluation purposes will not depend on other cards in the deck. Instead, card power will be determined as an aggregation of human-assigned rankings. In particular, rankings were sourced from MTG veteran pro player Luis Scott-Vargas and crowd-sourced ranking system Draftaholics Anonymous (Scott-Vargas 2018) (Draftaholics Anonymous 2018). These rankings are scaled comparatively such that the best card in the set is worth approximately 2.5 times that of the lowest-ranked card.

Evaluation Function

The metrics used to evaluate the strength of a deck in regards to each card feature can be combined to form a generalized deck-evaluation function. However, one notable metric is color, as non-adherence to the optimal two-color policy is massively detrimental to the deck's performance. Rather than scoring the deck by a color-based metric, the draft pool will be pre-processed to guarantee that cards from only two colors remain. This is done in two steps: first, all cards that belong to a color that is represented by 6 or fewer cards are immediately removed. Then, the color with the least sum of power over all cards that represent it is eliminated until only 2 colors remain. This procedure is outlined in Algorithm 1. Because most drafters are aware of the two-color policy and choose their draft picks accordingly, this trimming of the draft pool will rarely have any effect on the optimal outcome.

Similarly, the 17-land policy will also treated as a strict constraint in the framing of the combinatorial optimization problem. In particular, the feature vector will be defined as having 23 dimensions, corresponding to the 23 spells to be chosen for the deck.

With fixed constraints handled in the pre-processing phase, the evaluation function only needs to be a function of two metrics: fit to the mana cost curve and total card power.

Algorithm 1: Draft Pool Color Pre-Processing

Result: Trimmed draft pool
colors = B, U, G, R, W;
pool P;
for color in(colors) **do**
 if count(P, color) ≤ 6 **then**
 remove(P, color);
 remove(colors, color);
 end
end
while len(colors) > 2 **do**
 weakest_color = argmin(count(P, color));
 remove(P, weakest_color);
 remove(colors, weakest_color);
end

To evaluate a deck’s fit to the optimal curve, the number of cards that exceed the curve limit for a particular mana cost will be totaled, and the curve score will be determined as the percent of non-violating cards in the deck. For the purposes of this calculation, all cards with mana cost six or greater will be considered to have a cost of 6. If $D(x)$ is the number of cards in the deck and $C(x)$ the desired curve for mana cost x , the the curve score can be determined as follows:

$$CurveScore(D) = \frac{\sum_{i=1}^6 D(i) - C(i) * H(D(i) - C(i))}{\sum_{i=1}^6 D(i)} \quad (1)$$

Please note that $H(n)$ is the unit step function in this application. Furthermore, this scoring scheme has been left intentionally lenient, as decks do not need to perfectly comply with the optimal curve policy to excel.

The calculation of the overall utility of a proposed deck is straightforward once the curve score and sum of independent card scores, denoted *CardScore* have been calculated:

$$Utility(D) = CurveScore(D) * CardScore(D) \quad (2)$$

In other words, the total card utility of a deck receives a penalty equal to the percentage of cards that do not comply with the optimal curve policy.

Optimization Methods

Two heuristic optimization methods will be examined for use with the defined evaluation function.

Simulated Annealing

Simulated annealing is an optimization method in the ‘dual-phase’ family of algorithms. That is, it exploits the state space to approach a balance between global and local optimization methods. The annealing algorithm relies on a running temperature variable to dictate its willingness to accept a sub-optimal solution in hopes that it is ultimately moving towards a global maximum.

Different annealing schedules have been developed to be well-suited to particular problems. For the purposes of

this project, a schedule that utilizes a distorted Cauchy-Lorentz schedule was selected (SciPy community 2020b). An overview of the simulated annealing procedure can be found in Algorithm 2.

Algorithm 2: Simulated Annealing

Result: Cooled Solution
temperature T;
temp_threshold;
cooling_factor α ;
random solution S;
while $T > temp_threshold$ **do**
 for inner_reps_count **do**
 solution S’ near(S);
 $\delta = cost(S') - cost(S)$;
 if $\delta < 0$ **then**
 S = S’;
 else
 if prob($exp(\delta/T)$) **then**
 S = S’;
 end
 end
 end
 $T = \alpha T$
end

Genetic Algorithm

Genetic evolution algorithms, or more simply genetic algorithms (GAs) have been used as both primary heuristic and metaheuristic optimizers to construct decks for MTG (Sverre and Knut 2017). However, these efforts utilize simulated gameplay to evaluate the fitness of a particular deck, a method which is untenable in real-time solutions without shifting the computational effort to a lengthy training period (Chen et al. 2018). In this implementation, deck simulation is forgone in favor of the previously defined evaluation method, which combines optimal deck policies with expert analysis on each member card.

The GA functions by randomly initializing a number of initial solutions, known as the population. The fitness of each population member is then evaluated. The highest-fitness solutions are then combined in a process called *crossover*, where the underlying traits, or genes of each pair of solutions are spliced together. To introduce exploration of the search space into the process, the crossover phase is followed by a mutation phase, where random traits are introduced into the population. This process iterates until the population converges to a highly-fit solution. Like simulated annealing, this method does not guarantee finding a globally optimal solution (SciPy community 2020a). Pseudocode for this high-level procedure can be found in Algorithm 3.

Experiments

Data

To supply our experiment with a database for card statistic lookup, data on every card from the MTG set *Core Set 2019* was downloaded from the regularly-updated website Scryfall.com using their API (Scryfall 2020). This database was

Algorithm 3: High-Level Genetic Evolution

Result: Converged solution
random population P ;
while *not converged*(P) **do**
 fitness_scores = fitness(P);
 best_solutions = select(P , fitness_scores);
 crossover(best_solutions);
 mutate(best_solutions);
 P = best_solutions;
end

then augmented with the addition of aggregate card rankings for each card in the set.

The deck-building phase also requires a draft pool as an input. Over two million sample draft pools collected from MTG arena players were downloaded from the publicly available database at draftsim.com (Draftsim 2020).

Procedure

An experiment was run to compare the performance of three potential deck-building methods: simulated annealing, genetic evolution, and random selection. The simulated annealing and genetic evolution methods were implemented using the open-source library SciPy. An experimental trial consisted of running the three methods on a randomly selected sample draft pool and comparing the curve fit, total card power, and utility score of the solutions each method produced. Additionally, the deck lists produced from each method were stored for further analysis.

Results and Analysis

A experiment was run on 1500 human-constructed draft pools arbitrarily selected from the Draftsim database. In addition to tracking the utility of their selection, the curve fit and net card value metrics were also recorded for each of the three methods. The averaged results over the 1500 trials can be found in Table 1. Each trial was run in roughly 7 seconds of CPU time on consumer-grade hardware.

Unsurprisingly, the random method’s average card value was 1572 points, only slightly above the set’s average card value of 1570. Less predictably, the random method built decks that were on average more than 80% compliant with the curve policy. This is due to the tendency of human drafters to construct their pools with a curve distribution in mind, and will hesitate to select an overabundance of either costly or cheap cards. Moreover, this curve is roughly reflected in the base distribution of card costs in the set.

The decks constructed by the genetic algorithm had average card values only 17 points higher than those generated randomly, but surpassed the random method significantly in regard to their curve fit. This is indicative of the method not fully exploring the solution space to optimize card selection (as the average card value is not meaningfully greater than that of random selection), yet effectively prioritizing the curve metric.

The annealing method had the strongest performance across all metrics, with an average card pick 90 points higher

| Method | Curve Fit | Avg. Card Value | Utility |
|-----------|-----------|-----------------|---------|
| Annealing | 0.964 | 1659 | 36785 |
| Genetic | 0.936 | 1589 | 34214 |
| Random | 0.838 | 1572 | 30328 |

Table 1: Performance metrics for the three combinatorial optimization methods averaged over 1500 trials

than the genetic method. The ability to maintain a high adherence to the curve policy while prioritizing above-average cards displays the method’s ability to effectively explore the solution space.

With additional tuning, the genetic algorithm may be able to approach the annealing method in terms of deck quality. Increasing population size and mutation prevalence may both help the method explore the search space more fully, an aspect that it appears to be lacking in due to its lower card pick value. However, the genetic method is much slower than annealing and grows linearly with both population size and generations before convergence, indicating that solution quality may come at a significant loss in speed.

Conclusion

Several heuristic methods for MTG booster draft deck construction have been presented and shown to generate performant results in seconds of run time. These approaches offer solutions to close the gap between drafting and dueling in simulated booster draft tournament play. Further trials involving simulated play would be necessary to fully explore how effective the decks crafted from these heuristic methods are, which may be the subject of future work. Moreover, statistical analysis of any differing trends in human and bot drafting may reveal oversights stemming from this project’s reliance on human-generated draft pools.

Code Availability

All code for this project (including data) has been made available at the following public GitHub repository: <https://github.com/gordonng123/DraftToDeck>.

References

- Chen, Z.; Amato, C.; Nguyen, T.-H.; Cooper, S.; Sun, Y.; and El-Nasr, M. S. 2018. Q-DeckRec: A Fast Deck Recommendation System for Collectible Card Games. *IEEE Conference on Computational Intelligence and Games (CIG)*.
- Draftaholics Anonymous. 2018. P1P1 – Core Set 2019. URL <https://apps.draftaholicsanonymous.com/p1p1/M19/>.
- Draftsim. 2020. Draftsim.com. URL <https://draftsim.com/draft-data/>.
- ESPN. 2020. Can Magic: The Gathering Arena challenge Hearthstone’s digital card game supremacy? URL https://www.espn.com/esports/story/_/id/28752185/can-

magic-gathering-arena-challenge-hearthstone-digital-card-game-supremacy.

García-Sánchez, P.; Tonda, A.; Squillero, G.; Mora, A.; and Merelo, J. J. 2016. Evolutionary deckbuilding in hearthstone. *IEEE Conference on Computational Intelligence and Games (CIG)* 1–8.

Karsten, F. 2014. Finding the Optimal Mana Curve via Computer Simulation. URL <https://strategy.channelfireball.com/all-strategy/mtg/channelmagic-articles/how-many-lands-do-you-need-to-consistently-hit-your-land-drops/>.

Karsten, F. 2017a. How Many Lands Do You Need to Consistently Hit Your Land Drops? URL <https://strategy.channelfireball.com/all-strategy/mtg/channelmagic-articles/how-many-lands-do-you-need-to-consistently-hit-your-land-drops/>.

Karsten, F. 2017b. Magic Math - A New Way to Determine an Aggregate Deck List. URL <https://www.channelfireball.com/articles/magic-math-a-new-way-to-determine-an-aggregate-deck-list-rg-dragons/>.

Karsten, F. 2018. To Splash or Not to Splash in Limited? URL <https://strategy.channelfireball.com/all-strategy/mtg/channelmagic-articles/to-splash-or-not-to-splash-in-limited/>.

SciPy community. 2020a. `scipy.optimize.differential_evolution`. URL.

SciPy community. 2020b. `scipy.optimize.dual_annealing`.

Scott-Vargas, L. 2018. Core Set 2019 Constructed Set Review: Gold, Artifacts, Lands, and Top 5 Cards. <https://strategy.channelfireball.com/all-strategy/mtg/channelmagic-articles/core-set-2019-constructed-set-review-gold-artifacts-lands-and-top-5-cards/>.

Scryfall. 2020. Scryfall Magic: The Gathering Search. <https://scryfall.com/>.

Stiegler, A.; Messerschmidt, C.; Maucher, J.; and Dahal, K. 2016. Hearthstonedeck-construction with a utility system. *10th International IEEE Conference on Software, Knowledge, Information Management Applications (SKIMA)* 21–28.

Sverre, J. B.; and Knut, A. F. 2017. *Deckbuilding in Magic: The Gathering Using a Genetic Algorithm*. Master’s thesis, Norwegian University of Science and Technology.