Optimizing Buying Strategies in Dominion

Nikolas A. Koutroulakis
Georgia Southern University, nikolaskoutroulakis@gmail.com

Follow this and additional works at: https://scholar.rose-hulman.edu/rhumj
Part of the Discrete Mathematics and Combinatorics Commons, Number Theory Commons, and the Probability Commons

Recommended Citation
Available at: https://scholar.rose-hulman.edu/rhumj/vol25/iss1/2
Optimizing Buying Strategies in Dominion

Cover Page Footnote
The author acknowledges Georgia Southern University's Talon high-performance computing cluster for providing resources that contributed to the research results reported within this paper. URL: https://its.georgiasouthern.edu/crts/research/talon-cluster/

This article is available in Rose-Hulman Undergraduate Mathematics Journal: https://scholar.rose-hulman.edu/rhumj/vol25/iss1/2
Optimizing Buying Strategies in Dominion

By Nikolas Koutroulakis

Abstract.
Dominion is a deck-building card game that simulates competing lords growing their kingdoms. Here we wish to optimize a strategy called Big Money by modeling the game as a Markov chain and utilizing the associated transition matrices to simulate the game. We provide additional analysis of a variation on this strategy known as Big Money Terminal Draw. Our results show that players should prioritize buying provinces over improving their deck. Furthermore, we derive heuristics to guide a player’s decision making for a Big Money Terminal Draw Deck. For optimal play, we show that after turn 6, it is better to buy a second Smithy than a Silver, and after turn 8, a second Smithy is better than a Gold.

1 Introduction

Deck-building games are card games centered around strengthening a deck relative to competing players. Dominion, a game designed by Donald X. Vaccarino, and released in 2008, is the first of a now popular genre called deck-building games. Players in a game of Dominion represent competing lords growing their kingdoms and using their wealth to purchase land. This “land” is known in the game as victory points. The player with the most victory points at the end of the game is the winner. Every player starts the game with a 10-card deck, and play rotates by turns among the players. Each turn, a player draws a 5-card hand from their individual deck. They are permitted to take one action and can additionally buy a single card from the reserve containing treasure cards, action cards, and victory points. The goal of Dominion is to accumulate more victory points (Provinces, Estates, and Duchies) than any other player. However, these cards tend to weaken a player’s deck so many players consider it more prudent to abstain from buying them earlier in the game.

There are two basic phases of play in a game of Dominion. The first can be thought of as the deck building phase. Here, the goal is simply to improve one’s deck to perform well in the next stage of play called greening.

Mathematics Subject Classification. 00A08, 60J20
Keywords. recreational mathematics, board games, markov chains
**Definition 1.1** (Greening). A player begins *greening* on turn $t$ if and only if they have bought their first victory card. A player is considered to be greening from turn $t$ until the end of the game.

While in the greening stage, the player focuses all of his attention on buying victory points, often times at the expense of their deck's performance. The greening stage can be further broken down into two phases. In the first phase, it is common wisdom among many players that Estates and Duchies are not bought until there are few Provinces left in the game. Estates and Duchies tend to clutter up a player's hand with very little "return on investment." Thus, in the first phase of greening a player buys Provinces.

The second phase of greening begins when there are very few Provinces left to purchase. Players then scramble to buy the smaller valued Estates and Duchies to pad their victory points as much as possible. This second phase of greening lasts only a few turns, and the strategy, known as the "Penultimate Province Rule" is well understood [2]. We do not consider this phase at all, accepting the conventional algorithms.

There are three types of cards in a player's deck: victory points, treasure cards, and action cards. Victory points are the ultimate measure of who wins the game and are comprised of Province, Estates, and Duchies. Treasure cards and action cards allow a player to improve the value of his hand to eventually allow for the purchase of the most victory points. Treasure cards are straightforward. There are three types: copper, which cost 0 and are worth 1, silver, which cost 3 and are worth 2, and gold, which cost 6 and are worth 3. These cards can be used in later turns to purchase more cards.

The third type of cards are action cards; these require more explanation since there are numerous ways action cards might improve a player's hand. Action cards can allow a player to draw more cards, discard cards (putting cards in a discard pile which will be recycled later), trash cards (permanently remove cards from their deck), or add to the number of cards the player is allowed to buy. Attack cards will improve a player's hand at the expense of another player's, while defense cards will shield a player from their effects. Action cards can also be used to increase the number of actions a player is permitted to take in a turn, and action cards can be used to increase the value of a hand, that is, increase the buying power a player has on a given turn. The value of a hand is different from the *value of the deck*. The value of a deck is the sum of all the treasure cards in the deck. Usually, there is an issue of how to factor the presence of value that is added to the deck by action cards (this sort of money is called virtual money). However, we will not need to consider this in the present analysis. Here we will assume that the value of a hand and a deck are only the sum of the values of treasure cards.

The game begins with the deck of cards and an empty discard pile. On each turn, the player first plays actions cards and then purchases more cards. When a card is purchased, it immediately goes into the player's discard pile. When the player draws all of his cards, they re-shuffle their discard pile, and it becomes their deck once again. Note, at the end of a player's turn, all of the cards played or purchased and any cards remaining in their
hand are discarded.

Finally, the cards that are not yet bought by a player are said to be in a supply pile. Because there are a finite number of each card type for a player to purchase, the game ends when either every Province is purchased or the supply pile for any three distinct card types is depleted.

There are several general strategies for building a deck in Dominion, referred to as deck archetypes \cite{3}. Every strategy in dominion is some variant of these archetypes. Combo decks rely on 2 or 3 action cards that synergize well with one another. Engines are decks that focus on chaining several different action cards together in order to achieve a large payload, capable of purchasing multiple cards in a single turn. Rush decks aim to finish the game as quickly as possible. The idea is to rapidly buy victory points while simultaneously depleting 3 supply piles before other players even have the chance to start buying victory points themselves. This strategy is most commonly used against engines, which typically take a bit of time before they are able to green. The flip-side of Rush decks are Slog decks, which aim to extend the game for as long as possible. This is done by playing cards that are intended to weaken other players’ decks, either by making them trash cards, or conversely by forcing them to add undesirable cards to their deck.

We examine a deck archetype known as Big Money. A Big Money deck is a deck primarily concerned with buying treasure cards. While these are usually augmented with the purchase of action cards, this is not the focus of the strategy. In fact for certain card types, a Big Money deck that only buys 1 or 2 of that card type can be very difficult to beat.

We model the game of Dominion as a Markov chain where each turn is considered as a state. The transition matrix of this Markov chain is a matrix whose rows and columns are indexed with every possible deck in the game, and whose entries are the probability of getting from the deck indexed by the row to the deck indexed by the column. This value is determined by a multi-variable piece-wise function that models the player’s decision at a given turn based on the properties of their deck. When deciding what to do, our player will consider three factors: the most expensive treasure card he can afford at a given turn, the money density of his deck, and the number of turns that have already passed.

In Section 2, we begin with some preliminaries and describe our methods in Section 3. The results in Section 4 demonstrate that for a Big Money strategy, a player is always better off buying a Province the first chance they can get. It is never a good idea for a player to pass up buying a Province to improve his deck, even early in the game. Thus, Provinces should always be the first priority for Big Money decks. Then, Section 5 provides some nice heuristics for buying a kind of action card that allows a player to draw additional cards into their hand, called terminal draw cards. These heuristics come in the form of inequalities that tell a player when to buy a terminal draw card over each kind of treasure cards. Additionally, some bounds are provided for the player that suggest
when to buy a Smithy over a Silver or Gold, and when it is guaranteed that the Smithy is always the better buy. We conclude with a discussion on the limitations of this analysis.

2 Big Money Strategy

Beginner players often fall into the trap of spending too much time building their deck, leaving too little time to use the deck to gather victory points. While there are sophisticated deck-builds that allow the purchase of multiple victory points in a turn, a beginner is unlikely to get the timing right to use these strategies efficiently. These challenges lead the improving player to discover a simple but effective kind of deck-building strategy called Big Money.

Whereas Combo Decks are generally themed around two or three synergistic action cards, and Engine Decks are decks designed to draw as many cards as possible in a single turn, Big Money has a simpler approach. For a Big Money strategy, a player will focus on buying high value treasure cards to increase the value of the deck, so it is more likely the player can afford to buy Provinces. This fairly simple strategy is surprisingly effective, and its more intent focus will often beat out inexperienced players who unwittingly employ more elaborate strategies. While it is possible to beat out Big Money with other strategy schema, the simple and surprisingly effective nature of Big Money style decks make it a natural starting point for studying Dominion strategies. The basic Big Money algorithm is as follows:

Algorithm 1 Basic Big Money Buying Algorithm

if \( \text{valueOfHand} \leq 2 \) then
    \( \text{Copper} \leftarrow \text{Copper} + 1 \)
end if

else if \( 3 \leq \text{valueOfHand} \leq 5 \) then
    \( \text{Silver} \leftarrow \text{Silver} + 1 \)
else if \( 6 \leq \text{valueOfHand} \leq 7 \) then
    \( \text{Gold} \leftarrow \text{Gold} + 1 \)
else if \( \text{valueOfHand} \geq 8 \) then
    \( \text{Province} \leftarrow \text{Province} + 1 \)
end if

Algorithm 1 provides simple rules for what to buy and when, but it does not clearly recommend when to start greening, that is, when to start buying victory points over treasure cards. To help refine the algorithm, we define the notion of money density.

Definition 2.1. Given a deck \( D = (c, s, g, p, e) \) of size \( n \) cards where the values in the 5-tuple represent the number of Copper, Silver, Gold, Provinces, and Estates, respectively in \( D \), define

1. the value function \( \mathcal{F} : D \rightarrow \mathbb{N} \) that maps a card in the deck to its treasure value,
2. the deck value \( v_D = \sum_{x \in D} \mathcal{T}(x) \), and

3. the money density of the deck \( m_D = \frac{v_D}{n} \).

If the input for \( \mathcal{T} \) is a set of cards \( H \), then \( \mathcal{T}(H) \) will be implicitly understood as the value of the set; in other words, \( \mathcal{T} \) is a function mapping a set of cards to the sum of its treasure values, i.e. \( \mathcal{T}(H) = \sum_{x \in H} \mathcal{T}(x) \).

The notations \( D \) and \( (c, s, g, p, e) \) will be used interchangeably, and we sometimes write \( v = v_D \) when the deck referenced is clear. It is straightforward to find the expected value of a card.

**Proposition 2.1.** Let \( X \) be a random variable denoting the value of a card drawn uniformly at random from a deck \( D \) of \( n \) cards. Define a hand \( H \) to be a set of five cards \( \{C_1, C_2, C_3, C_4, C_5\} \) drawn without replacement from our deck, and let \( X_i \) be the value of \( C_i \) for \( i = 1, 2, 3, 4, 5 \). Then, the expected value of a card is \( E(X) = m_D \) and the expected value of a 5-card hand \( H \) is \( E(H) = 5m_D \).

**Proof.** By definition,

\[
E(X) = \sum_{x \in D} \mathcal{T}(x) p(x) \tag{1}
\]

where \( p(x) \) is the probability of \( x \), and \( x \) is a particular card in the deck. For a set of cards \( D \), the probability of drawing any one card is \( \frac{1}{n} \). So,

\[
E(X) = \sum_{x \in D} \mathcal{T}(x) p(x) = \sum_{x \in D} \mathcal{T}(x) \frac{1}{n} = \frac{1}{n} \sum_{x \in D} \mathcal{T}(x) = \frac{v_D}{n}. \tag{2}
\]

Since a hand \( H \) contains 5 cards,

\[
E(\mathcal{T}(H)) = E(X_1 + ... + X_5) = E(X_1) + ... + E(X_5) = 5E(X_1) = 5 \frac{v_D}{n} = 5m_D. \tag{3}
\]

Note that higher money density results in higher expected value, and that higher expected value increases a player’s ability to buy Provinces or improve his deck. Our paper focuses on the following question regarding the Big Money strategy: Is there an optimal value of \( m_D \) at which to begin buying victory points over treasure cards? We also analyze an extension of the Big Money strategy called Terminal Draw Big Money.
3 Markov Modeling of Big Money Strategy

Looking at the basic Big Money decision algorithm, there is not much we can do to optimize performance from turn to turn. It is fairly obvious, for example, that a player should never buy a lower valued treasure card if they can buy a higher valued treasure card. Moreover, ideally a player should always abstain from buying treasure cards whenever their deck's money density is greater than the value added by the most expensive card they can afford that turn. For instance, if a player's money density is greater than one, but they can only afford to buy a Copper, adding that card into their deck would decrease the expected value of their draw. This leaves us with the macroscopic consideration of when a player is ready to start greening.

A player is greening when he begins the process of acquiring victory points. Greening as soon as possible allows a player to gain an early lead. However, it also means the player will buy Provinces less frequently later in the game. On the other hand, a player who delays greening will have a more effective deck late in the game, but has missed earlier opportunities to buy Provinces. This raises the question: When is the right time to start greening? For instance, should a player on turn 4 buy a Province or a Gold? Which leads to a better outcome? Can a player glean insight from knowing the money density of their deck? Deck archetypes such as Engines favor strategies that postpone greening in order to improve their deck earlier in the game. This is so that the Engine can rapidly out-buy other players’ decks in the last few turns of the game. Are there any such benefits in doing so for a Big Money style deck?

Our approach to this question models the game as a Markov chain, making use of the associated transition matrix. The transition matrix will have rows and columns indexed by every deck reachable in an $n$-turn game. The rows are indexed by a player's deck at the start of a turn, and the columns are indexed by the player's deck by the end of that turn. The entry corresponding to each row and column denotes the probability of getting from the state indexing the row to the state indexing the column. The transition matrix itself represents every possible - and perhaps impossible - way a player's deck could change in a game of length $n$. Since the starting deck in a game of Dominion is 7 Copper, 0 Silver, 0 Gold, 3 Estates, and 0 Provinces, the first row and column are both identified with the ordered tuple (7, 0, 0, 0, 3). It is well know that the $n^{th}$ power of a transition matrix gives the probability distribution of the $n^{th}$ state. So, we can find the probability of a deck being in a particular state after $n$ turns by multiplying the matrix by itself $n$ times. We will compute this matrix by running a program in Java; first it will be necessary to describe some intermediary functions that are used to compute the cell values.

It is important to draw attention to two assumptions that we make in order to compute the Markov chain.

1. Taking each turn as a state in a Markov chain is equivalent to assuming that a player shuffles every turn. This is not so in a real game of Dominion.
2. We assume that if the player will always buy the most expensive treasure they can afford, even if its value is less than the money density of their deck.

While these assumptions are somewhat concerning, the impact that they would have on the final results are likely to be negligible. We justify this assertion in Section 4.

Similar to how a deck was represented by the number of cards of each type, a 5-card hand \( H \) will be represented with the 3-tuple \((c, s, g)\) where \( c \) is the number of Copper, \( s \) is the number of Silver, and \( g \) is the number of Gold in the hand.

**Proposition 3.1.** The probability of getting a hand \( H = (c, s, g) \) from a deck \( D \) containing subsets \( C \), \( S \), and \( G \), of all Copper, Silver, and Gold cards, respectively, is given by the equation:

\[
p_D(H) = \binom{|C|}{c} \binom{|S|}{s} \binom{|G|}{g} \binom{|D|-|C|-|S|-|G|}{5-c-s-g} \binom{|D|}{5}^{-1}
\]

**Proof:** When a player draws 5 cards, there are \( \binom{|D|}{5} \) possible hands they could get. In the numerator, we have all the possible ways we could draw \( c \) Copper, \( s \) Silver, and \( g \) Gold from \( D \). Hence, their ratio is the probability of drawing a 5-card hand of \( c \) Copper, \( s \) Silver, and \( g \) Gold. \(\blacksquare\)

**Definition 3.1.** Given a deck \( D \) and a hand \( H \subset D \), let the value of \( H \) be

\[
V = \mathcal{T}(H) = \sum_{x \in H} \mathcal{T}(x)
\]

and let the solution set of \( V \) be the set of hands \((c, s, g) \subset D\) where

\[
\text{sol}_D(V) = \{(c, s, g) \in \mathbb{N}^3 : c + 2s + 3g = V \text{ and } c + s + g \leq 5\}.
\]

**Corollary 3.1.** The probability of a drawing a hand of value \( k \) from a deck \( D \) is

\[
Pr(V = k) = \sum_{(c,s,g) \in \text{sol}_D(k)} p_D(H)
\]

To compute this probability, we need a method for finding the solution set of \( V \). Our program does this by first iterating over all the possible solutions within a given range, and then passing all of the solutions as an input to another method that sums their probabilities together. To do this, we solve the Diophantine equation, \( c + 2s + 3g = V \), to express \( c, s, \) and \( g \) in terms of any arbitrary integers, \( k, m \in \mathbb{Z} \). We used nested for loops on \( k \) and \( m \) to iterate over a sufficiently small range of values needed for \( c, s, g \in \mathbb{Z} \).
where \( 0 \leq c + s + g \leq 5 \). Though it is difficult to define a nested for loop that only iterates over solutions which meet this restriction, we can certainly define an algorithm that cuts down on the search time required for finding them by ensuring that every solution can be found within a small range of values.

**Theorem 3.1.** The hand \((c, s, g)\) is in \(sol_D(V)\) only if there exists integers \(-5 \leq m \leq 0\) and \(-5 \leq k \leq 2\) such that \(c = k - m\), \(s = -2k - m - V\), and \(g = k + m + V\).

**Proof.** Let \(c + 2s + 3g = V\). Then for some \(k, m \in \mathbb{Z}\),

\[
\begin{align*}
c + g &\equiv V \pmod{2} \quad \implies \quad c + g = 2k + V \quad \implies \quad g = 2k + V - c, \quad \text{and} \\
c + 2s &\equiv V \pmod{3} \quad \implies \quad -2c - s \equiv V \pmod{3} \quad \implies \quad s = -2c - 3m - V
\end{align*}
\]

We now substitute \(g\) and \(s\) into our equation.

\[
V = c + 2s + 3g = c + 2(-2c - 3m - V) + 3(2k + V - c) = c - 4c - 6m - 2V + 6k + 3V - 3c = 6k - 6m - 6c - 2V + 3V
\]

Then rearrange this equation to solve for \(c\).

\[
\begin{align*}
V &= 6k - 6m - 6c - 2V + 3V \\
0 &= 6k - 6m - 6c \\
c &= k - m
\end{align*}
\]

Thus

\[\quad (c, s, g) = (k - m, -2k - m - V, k + m + V)\]

is an integer representation that satisfies \(c + 2s + 3g = V\) for some \(k, m \in \mathbb{Z}\).

There are between 0 and 5 treasure cards in a hand, so \(0 \leq c + s + g \leq 5\), where \(c\), \(s\) and \(g\), are non-negative. Replacing \(c\), \(s\), and \(g\) with their values from Theorem 3.1 gives

\[0 \leq c - 2c - 3m - V + 2k + V - c \leq 5\]

which reduces to

\[-5 \leq m \leq 0\]
Similarly, $0 \leq c \leq 5$ implies that $0 \leq k - m \leq 5$, so

$$m \leq k \leq 5 + m$$

and $0 \leq s \leq 5$ implies that $0 \leq -2k - m - V \leq 5$, so

$$\frac{-5 - m - V}{2} \leq k \leq \frac{-m - V}{2}$$

Since $0 \leq c, s, g \leq 5$ holds only if each of these inequalities are true, we can use the most restrictive pair of inequalities as parameters for $k$. Because the lowest value $m$ can take on is $-5$, we know that $k$’s absolute lower bound is also $k = -5$. Likewise, when $m = 0$, its highest possible value is $5 + m = 5 \geq k$, so we know that $k$ must at least be between $-5$ and $5$ in order for for $c$ to be between $-5$ and $0$.

We can proceed similarly with the other 2 inequalities. The expression $\frac{(-5 - m - V)}{2}$ is at its minimum when $V = 15$, and $m = 0$. On the other hand, the formula $\frac{(-m - V)}{2}$ has its maximum when $m = -5$, and $V = 0$ so,

$$\frac{1}{2}(-5 - m - V) \leq k \leq \frac{1}{2}(-m - V) \implies -10 \leq k \leq \frac{5}{2}$$

Since $k$ is an integer, and since we have already shown that $k \geq -5$, we can adjust this to $-5 \leq k \leq 2$, and we have proven the claim. ■

**Algorithm 2** Find Solutions for a Hand of Value $v$

1: function FINDSOLUTIONS($v$)
2:     solutions ← empty list
3:     for $m$ in $-5$ to $0$ do
4:         for $k$ in $5$ to $2$ do
5:             $x \leftarrow k - m$
6:             $y \leftarrow -2x - 3m - v$
7:             $z \leftarrow 2k + v - x$
8:             if ($x \geq 0$) and ($y \geq 0$) and ($z \geq 0$) then
9:                 solutions ← solutions + [$x, y, z$]
10:          end if
11:     end for
12: end for
13: return solutions
14: end function

Theorem 3.1 ensures that Algorithm 2 correctly computes $\text{sol}_D(V)$ for $V$. The converse of Theorem 3.1 holds true when $c, s, g \geq 0$, since $(c, s, g) \in \text{sol}_D(V)$ requires $c, s, g$ to be natural numbers. Thus, a solution will be appended to $solutions$ when and only when it belongs in $\text{sol}_D(V)$. 
Algorithm 1 shows us that there are two inputs determining a player’s decision: what he should buy, and what is possible for him to buy. We can represent this with the following definitions.

**Definition 3.2.** For a range of values, \( \{v_k, v_k + 1, \ldots, v_n - 1, v_n\} \subseteq \mathbb{Z}^+ \), the probability that a hand is drawn with a value within this range is given by the equation

\[
Pr(v_k \leq X \leq v_n) = \sum_{k=v_k}^{v_n} Pr(V = k)
\]

Since we wish to examine what the effect of growing money density prior to greening will be, we define a multi-variable piece-wise function to model this. Let \( g \) represent the function in variables \( t, m, D_1 \) and \( D_2 \) where \( t \) is the maximum turns a player waits until he greens, \( m \) is the minimum money density a player requires before greening, \( D_1 = (c_1, s_1, g_1, p_1, e_1) \) is the deck in the prior state before the turn is played and \( D_2 = (c_2, s_2, g_2, p_2, e_2) \) is the deck after the turn. Note that on each turn exactly one card is purchased, so the coordinate differences will be zero in four cases and one in the last case.

We define \( f_1 \) and \( f_2 \) as piece-wise functions as follows:

**Definition 3.3 (The Decision Function).**

\[
f_1(D_1, D_2) = \begin{cases} 
Pr(0 \leq X \leq 2) & c_2 - c_1 = 1 \\
Pr(3 \leq X \leq 5) & s_2 - s_1 = 1 \\
Pr(6 \leq X \leq 7) & g_2 - g_1 = 1 \\
Pr(8 \leq X) & \text{otherwise}
\end{cases}
\]

and

\[
f_2(D_1, D_2) = \begin{cases} 
Pr(0 \leq X \leq 2) & c_2 - c_1 = 1 \\
Pr(3 \leq X \leq 5) & s_2 - s_1 = 1 \\
Pr(6 \leq X) & g_2 - g_1 = 1 \\
0 & \text{otherwise}
\end{cases}
\]

Then \( g \) is given as:

\[
g_{D_1 D_2}(m, t) = \begin{cases} 
0 & |D_1| \neq |D_2| + 1 \\
f_1(D_1, D_2) & t_{D_1} > t, m_{D_1} > m \\
f_2(D_1, D_2) & \text{otherwise}
\end{cases}
\]
where $t_{D_1}$ is the number of turns that have actually passed, and $m_{D_1}$ is money density for $D_1$. Note $t_{D_1}$ and $m_{D_1}$ are dependent on the other variables as $t_{D_1} = |D_1| - 10$ and $m_{D_1} = (c_1 + 2s_1 + 3g_1)/|D_1|$.

**Remark 3.2.** Fifteen is the highest value a hand can have (this would be a hand containing 5 Gold), so the sum for $x \geq 8$ can be computed by the sum $Pr(8 \leq X \leq 15)$ in this case.

A quick example will illustrate how this function is used.

**Example 3.1.** A deck $D_1$ with 7 Copper, 0 Silver, 0 Gold, 3 Estates and 0 Provinces is represented with the ordered tuple $D_1 = (7, 0, 0, 0, 3)$. Let $m = t = 0$, meaning we begin greening immediately. Before the first turn, $m_{D_1} = 0.7$. If on the first turn a player gains 1 Copper, then $D_2 = (8, 0, 0, 0, 3)$. The probability of this happening is $f_1(D_1, D_2) = f_2(D_1, D_2)$ so

$$f_1(D_1, D_2) = Pr(0 \leq X \leq 2) = \sum_{k=0}^{2} Pr(V = k) = Pr(V = 0) + Pr(V = 1) + Pr(V = 2) = \sum_{H \in sol_{D_1}(0)} p_{D_1}(H) + \sum_{H \in sol_{D_1}(1)} p_{D_1}(H) + \sum_{H \in sol_{D_1}(2)} p_{D_1}(H).$$

If $C$ is the random variable denoting the number of Copper that can be drawn in a 5-card hand,

$$Pr(C \geq 2) = \sum_{k=2}^{5} \binom{7}{k} \binom{3}{5-k} \binom{10}{5} = 1.$$ 

Thus $Pr(C < 2) = 0$, which in turn implies that $sol_{D_1}(0) = sol_{D_1}(1) = \emptyset$. This makes sense; after 3 Estates are drawn, there is nothing left to draw but 2 more Copper. So, we have

$$g_{D_1D_2}(0, 0) = f_1(D_1, D_2) = \sum_{x \in sol_{D_1}(2)} p(x) = \binom{7}{3} \frac{1}{\binom{10}{5}} = \frac{1}{12} \quad (6)$$

Now that we have our decision algorithm, we can define our transition matrix. Our model assumes that the player shuffles his deck each turn. This assumption allows us to define the probability space of an $n$-turn game as a Markov chain where each turn is a state.

Here is the symmetric transition matrix for a one turn game employing this simple Big Money strategy, with indices to indicate the the properties of each deck:
Remark 3.3. Each cell of this particular matrix is determined by the function $g$ with values at $g_{D_1D_2}(0,0)$, where $D_1$ is the deck denoted by the row indices, $D_2$ is a deck denoted by one of the column indices. Assuming $|D_2| = |D_1| + 1$ the values for the $f$ functions $g$ is composed of are determined by which card was gained.

Each of the entries lining the farthest left row shows respectively the amount of Copper, Silver, Gold, and Provinces that are in a player’s deck; likewise for the column labels. As previously mentioned, the rows are the possible states of the deck prior to the states indicated by the columns. The probability of getting from one state to the next is found at the intersection of each row and column.

There are only two possibilities for a player on the first turn: they can either gain a Silver or gain a Copper. It should be apparent that if we multiply this matrix by itself, then every entry in the matrix will be 0. This is expected, since for each hand at the beginning of the turn, the deck will always gain a Copper at the least. So, the probability of having a deck like $(7,1,0,0,3)$ by the end of the second turn is always going to be 0, because in the worst case, a player will at least have the deck $(8,1,1,0,3)$.

For a matrix to be useful in capturing every possible state of an $n$-turn game, it must be large enough to index every deck that is reachable in $n$ turns. We draw inspiration from Van der Heijden [1] in defining a reachable deck.

Definition 3.4. $D_2$ is reachable from $D_1$ in $t \geq 0$ turns if and only if

- $D_1 = D_2$, or
- There is a number of turns, $0 \leq k \leq t$ and a non-zero probability of deck $D_2$ $k$ turns from deck $D_1$

Remark 3.4. For a game of $t \geq 0$ turns, $D_2$ is unreachable from $D_1$ if $|D_2| > |D_1| + t$. It is only possible to gain one card per turn, so the maximum cardinality for any deck reachable by $D_1$ in $t$ turns must be $|D_1| + t$. 
Though fairly obvious, this fact is nevertheless important for defining the bounds of our matrix. While we don’t need to worry about excluding unreachable states in our matrix (save for purposes of efficient computation), we need to at least ensure that our matrix contains an index for every reachable hand.

Recalling that a starting deck always has ten cards, \((7,0,0,0,3)\), our transition matrix can therefore be defined as follows:

**Definition 3.5** (Big Money Markov Matrix). Given \(n \geq 10\), define the set

\[ I = \{(c,s,g,p,e) \mid c + s + g + p + e \leq n, \ c \geq 7, \ \text{and} \ e = 3\}. \]

Then let \(G(m, t) = [g_{ij}]\) be the \(|I| \times |I|\) matrix whose rows and columns are both indexed by \(I\) such that \(g_{i,j} = g_{D_i,D_j}(m, t)\).

Now, we show that this in fact defines a transition matrix for an \(n\)-turn game of Dominion using the Big Money strategy.

**Lemma 3.5.** \(G\) is matrix whose rows and columns are indexed by every hand reachable in an game of Dominion with a deck of \(n \geq 10\) cards.

**Proof.** Each entry \(g_{i,j}\) in \(G\) is uniquely determined by \(D_i, D_j \in I\), and every game begins with a starting deck of \((7,0,0,0,3)\). Let \(D = (c,s,g,p,e)\). If \(D \notin I\), then either \(|D| > n\), \(c < 7\), or \(e \neq 3\). Since every player starts with 7 Copper and there is no way to trash Copper, any decks with less than 7 Copper are unreachable. Likewise, since our strategy at no point buys Estates, any deck where \(e \neq 3\) is not reachable. Finally, by Remark 3, if \(|D| > n\), then \(|D|\) is unreachable in \(n - 10\) turns from the starting deck 10. Therefore if \(D \notin I\), then \(D\) is unreachable, and by contra-position, \(D\) is reachable only if \(D \in I\). \(\Box\)

After considering the indices of the matrix, we look at the values of the cells.

**Theorem 3.2.** Set \(g_{i,j} = g_{D_i,D_j}\), that is, the entry in the row corresponding to deck \(D_i\) and the column of deck \(D_j\). Then \(g_{i,j}\) is the probability of getting to \(D_i\) from \(D_j\).

**Proof.** Each turn a player can either buy Copper, Silver, Gold, or Province. First, any deck \(D_j\) with a number of cards not equal to \(|D_i| + 1\) cannot be reached using this strategy. By definition of \(g_{D_1,D_2}(m, t)\) the value of \(g_{i,j}\) is 0 when \(|D_i| \neq |D_j| + 1\).

Now, assume the conditions for greening have been met, so the output for \(g_{D_1,D_2}(m, t)\) will depend on \(f_1(D_i,D_j)\). Suppose the number of Provinces in \(D_j\) is one greater than that of \(D_i\), then the differences of Copper, Gold, and Silver between the two decks, must be 0 or else contradict our assumption that \(|D_i| = |D_j| + 1\). Thus the value at \(g_{i,j}\) is given by \(Pr(8 \leq X)\). This of course is the probability of a having enough money to buy a Province, which means that it is the probability the player will buy another Province assuming the conditions specified are met.
If the additional card is instead either a Copper, Silver, or Gold, then the probability distribution function handled by each of these conditions would instead be the value at \( g_{i,j} \).

On the other hand if the conditions for greening are not met, the value of \( g_{D_i D_2}(m, t) \) is instead determined by the function \( f_2(D_i, D_j) \). This time, if the extra card in \( D_j \) is a Province, then the probability at \( g_{i,j} \) would be 0, since the greening conditions have not been met. The probabilities of all of the other cases are nearly identical, with the exception that this time the probability of getting a Gold is given by \( Pr(6 \leq X) \). Because the conditions for buying a Province have not been met, it makes sense that any additional value in a hand would not change the fact that Gold will be bought instead, since it is the second highest card in terms of cost.

Therefore, the values at each \( g_{i,j} \) is the probability of going from \( D_i \) to \( D_j \) in one turn using the Big Money strategy.

We have demonstrated that, in fact, \( G \) is a matrix that does contain the probabilities of all reachable hands from \((7, 0, 0, 0, 3)\) in \( n - 10 \) turns. This is the state transition matrix of our model.

**Remark 3.6.** An alternative definition for \( G \) uses an infinite transition matrix that is not restricted to all hands being reachable in \( n \) turns. Our matrices can then be viewed as sub-matrices of this infinite transition matrix.

### 4 Results

The following tables display probabilities and expected values generated from the transition matrix defined in Section 3. We used a Java program on Georgia Southern University’s Talon cluster to complete the computations. The code computing the Markov chain is available upon request.
The first table gives the probability distributions for the number of Provinces for a 15 turn game of Dominion. Each column shows a separate distribution based on money density constraints on greening. The column on the far left in Table 1 denotes the number of Provinces gained by the end of the game. Each column thereafter is a distinct probability distribution, separated based on the amount of money density required for greening, ranging from 0.7 to 1.8. Table 1 has no other condition for greening other than the minimum money density.

The expected values corresponding to these distributions are given below in Table 2.

<table>
<thead>
<tr>
<th>pts</th>
<th>0.7</th>
<th>0.8</th>
<th>0.9</th>
<th>1.0</th>
<th>1.1</th>
<th>1.2</th>
<th>1.3</th>
<th>1.4</th>
<th>1.5</th>
<th>1.6</th>
<th>1.7</th>
<th>1.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.19</td>
<td>0.19</td>
<td>0.19</td>
<td>0.19</td>
<td>0.20</td>
<td>0.22</td>
<td>0.29</td>
<td>0.46</td>
<td>1.00</td>
<td>3.97</td>
<td>16.46</td>
<td>57.94</td>
</tr>
<tr>
<td>1</td>
<td>3.66</td>
<td>3.66</td>
<td>3.66</td>
<td>3.66</td>
<td>3.69</td>
<td>3.88</td>
<td>4.56</td>
<td>6.37</td>
<td>11.09</td>
<td>27.64</td>
<td>49.91</td>
<td>40.01</td>
</tr>
<tr>
<td>2</td>
<td>18.90</td>
<td>18.90</td>
<td>18.90</td>
<td>18.91</td>
<td>18.87</td>
<td>19.33</td>
<td>21.34</td>
<td>27.48</td>
<td>39.03</td>
<td>53.29</td>
<td>32.12</td>
<td>2.05</td>
</tr>
<tr>
<td>3</td>
<td>36.11</td>
<td>36.11</td>
<td>36.11</td>
<td>36.04</td>
<td>36.51</td>
<td>38.64</td>
<td>43.89</td>
<td>41.03</td>
<td>14.99</td>
<td>1.51</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>29.00</td>
<td>29.00</td>
<td>29.00</td>
<td>29.00</td>
<td>29.09</td>
<td>29.06</td>
<td>28.96</td>
<td>20.43</td>
<td>7.80</td>
<td>0.11</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>10.35</td>
<td>10.35</td>
<td>10.35</td>
<td>10.35</td>
<td>10.42</td>
<td>9.86</td>
<td>6.11</td>
<td>1.37</td>
<td>0.05</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1.66</td>
<td>1.66</td>
<td>1.66</td>
<td>1.66</td>
<td>1.61</td>
<td>1.13</td>
<td>0.11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0.12</td>
<td>0.12</td>
<td>0.12</td>
<td>0.12</td>
<td>0.08</td>
<td>0.01</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0.7</th>
<th>0.8</th>
<th>0.9</th>
<th>1.0</th>
<th>1.1</th>
<th>1.2</th>
<th>1.3</th>
<th>1.4</th>
<th>1.5</th>
<th>1.6</th>
<th>1.7</th>
<th>1.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.28</td>
<td>3.28</td>
<td>3.28</td>
<td>3.28</td>
<td>3.28</td>
<td>3.24</td>
<td>3.1</td>
<td>2.81</td>
<td>2.43</td>
<td>1.8</td>
<td>1.19</td>
<td>0.44</td>
</tr>
</tbody>
</table>

Table 2: The expected values for the probability distributions in Table 1

Notice that as money density requirements increase, the expected number of Provinces for a given deck decreases. This implies that the opportunity cost, that is, the loss associated with choosing a sub-optimal strategy, is always higher when purchasing a treasure card over a Province.
Optimizing Buying Strategies in Dominion

Table 3: The probability distributions for a 15 turn game where the additional condition of waiting for 8 turns have been applied to each of these distributions.

<table>
<thead>
<tr>
<th>pts</th>
<th>0.7</th>
<th>0.8</th>
<th>0.9</th>
<th>1.0</th>
<th>1.1</th>
<th>1.2</th>
<th>1.3</th>
<th>1.4</th>
<th>1.5</th>
<th>1.6</th>
<th>1.7</th>
<th>1.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.39</td>
<td>0.39</td>
<td>0.39</td>
<td>0.39</td>
<td>0.39</td>
<td>0.39</td>
<td>0.39</td>
<td>0.40</td>
<td>0.50</td>
<td>1.02</td>
<td>3.97</td>
<td>16.46</td>
</tr>
<tr>
<td>1</td>
<td>6.15</td>
<td>6.15</td>
<td>6.15</td>
<td>6.15</td>
<td>6.15</td>
<td>6.15</td>
<td>6.15</td>
<td>6.22</td>
<td>7.13</td>
<td>11.31</td>
<td>27.65</td>
<td>49.91</td>
</tr>
<tr>
<td>2</td>
<td>26.00</td>
<td>26.00</td>
<td>26.00</td>
<td>26.00</td>
<td>26.00</td>
<td>26.00</td>
<td>26.01</td>
<td>26.30</td>
<td>29.62</td>
<td>39.36</td>
<td>53.27</td>
<td>32.12</td>
</tr>
<tr>
<td>3</td>
<td>39.27</td>
<td>39.27</td>
<td>39.27</td>
<td>39.27</td>
<td>39.27</td>
<td>39.29</td>
<td>39.95</td>
<td>43.11</td>
<td>40.45</td>
<td>14.98</td>
<td>1.51</td>
<td>0.00</td>
</tr>
<tr>
<td>4</td>
<td>22.92</td>
<td>22.92</td>
<td>22.92</td>
<td>22.92</td>
<td>22.92</td>
<td>22.93</td>
<td>23.18</td>
<td>18.42</td>
<td>7.81</td>
<td>0.12</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>5</td>
<td>4.96</td>
<td>4.96</td>
<td>4.96</td>
<td>4.96</td>
<td>4.96</td>
<td>4.94</td>
<td>3.89</td>
<td>1.22</td>
<td>0.06</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>6</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
<td>0.29</td>
<td>0.06</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>7</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Table 4 shows expected values, this time for the distributions in Table 3.

<table>
<thead>
<tr>
<th>0.7</th>
<th>0.8</th>
<th>0.9</th>
<th>1.0</th>
<th>1.1</th>
<th>1.2</th>
<th>1.3</th>
<th>1.4</th>
<th>1.5</th>
<th>1.6</th>
<th>1.7</th>
<th>1.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.94</td>
<td>2.94</td>
<td>2.94</td>
<td>2.94</td>
<td>2.94</td>
<td>2.94</td>
<td>2.94</td>
<td>2.91</td>
<td>2.75</td>
<td>2.43</td>
<td>1.8</td>
<td>1.19</td>
</tr>
</tbody>
</table>

Table 4: The expected values for the probability distributions in Table 3

Tables 3 and 4 offer further support for this explanation. Each table is laid out exactly as before, but show decks which greened on two conditions: the minimum required money density, and the condition that certain number of turns must pass before the player can green. Eight was chosen as the typical number of turns to begin greening as it is halfway through the 15 turn game. Table 3 allows us to see the result of spending too much time focused on improving a deck.

Notice that the outcomes for the smaller densities are slightly worse, but as they get higher the distributions in Table 3 converge with that of Table 1. This suggests the decrease in the expected value for larger money density requirements are explained by the lower number of turns available for greening.

Overall, the results of our analysis show that with a basic Big Money strategy, a player can only hurt their chances of winning by opting for treasure cards over Provinces. As higher money densities are prioritized, the expected value for a 15 turn game decreases towards 0. Note that we chose to look at the probabilities for a 15 turn game, since it balances our interests regarding computational resources, and applicability to a real game. In a game with 2-4 players, we would expect a majority of the Provinces to be gone around this point. So, any strategy worth playing will have at least greened prior to turn 15.

We made two assumptions during our analysis to make the computation easier. First, we assumed that players shuffle every turn. Second, we assumed that the player always
buys the most expensive treasure they can afford, even if its value is less than the money density of their deck. For the first assumption, we would actually expect to see the money density grow faster in our model than in a real game. This is because the expected value updates each turn, whereas for a regular game of Dominion, the expected value updates only after the next shuffle. This is particularly relevant, as we will see in the Section 6 the limitation of such an assumption. However, this fact gives even stronger support the conclusion that waiting for money density to pass a certain threshold is not an optimal strategy. A deck with a more rapidly increasing $m_D$ would of course reach its target money density faster. Since a deck's $m_D$ increases less quickly in a real game, we can infer that it will waste more time reaching its target money density. So, if the deck in our model spends too much time trying to build up to its target $m_D$, we would naturally expect a deck in a real game to take even longer, and therefore perform worse.

For the second assumption, we argue that buying a card $x$ with $\mathcal{T}(x) > m_D$ happens infrequently, and it when it does occur, the impact on the performance of the deck is negligible. First consider buying a silver. Our tables show that reaching a money density of 1.8 by turn 15 is incredibly rare. Beyond this point, the expected value for buying a single Province reaches 0, so a player should have no concerns the cost of the silver will be greater than the money density. That leaves the case of buying a copper despite having a money density greater than one.

Proposition 3.1 and Definition 3.3 implies that the probability of getting a hand with a value less than two decreases asymptotically, assuming the player is only buying treasure cards. This is because each turn, a player at least buys a copper, making it less and less likely for a player to get a hand with a value less than two. To see this played out in an extreme case, consider $D = (997,0,0,0,3)$, with $P_D(V \leq 2) \approx 6.018 \times 10^{-8}$. This implies it is most probable to get a hand with a value less than or equal to two on turns one and two. On turn three, $P(V \leq 2) = 0.016317$ for $(7,2,1,0,3), (7,1,2,0,3)$, i.e., every possible $D$ such that $m_D > 1$ on turn three. However, if a player is able to buy a province they will see this probability increase. For example, the deck with the highest probability of buying a copper on turn five is $D = (7,3,1,1,3)$ with $P(V \leq 2) = 0.031302$, and $m_D = 1.066$. If on the next turn, we transition from $D$ to $D'$ with an extra copper, our money density decreases only to $m_{D'} = 1.0625$.\(^1\)

Hence, while it is possible that sub-optimal hands may be played in our strategy, such occurrences are uncommon and are unlikely to have a significant impact on the player's performance for a particular game. Even when they do occur, the impact on the deck is negligible. Therefore, the effect is almost certainly “averaged out”, and should not affect our aggregate results. On these grounds, we feel it is justified to neglect this in our analysis.

\(^1\)These values are computed using functions within the "Deck" class, designed for computing the final state of the Markov Chain.
5 Terminal Draw Decks

There are other strategies for Dominion that utilize action cards to improve one’s deck. In this section, we discuss a variation of Big Money decks, called Terminal Draw decks. We give guidelines as to when a player should purchase a particular terminal draw card known as a Smithy.

A Terminal Draw deck strategy utilizes terminal draw cards which allows a player to draw cards from their deck to their hand. Terminal draw cards are so named because no more actions are permitted if a terminal draw card is the first action played. In contrast to the basic Big Money algorithm, one or two terminal draw cards are bought prior to greening when a player wants to use the Terminal Draw strategy.

The most basic of this kind of strategy is known as Big Smithy. A Smithy allows a player to draw 3 additional cards. There are other kinds of terminal draw strategies utilizing cards like Council Room, which allows you to draw 4 additional cards, and has the additional effect of allowing other players to draw cards as well. We derive a formula that helps a player choose between buying the most expensive treasure card he/she can afford, or a terminal draw card.

**Theorem 5.1.** The expected value of a deck \( S \) with one terminal draw card is given by

\[
E(S) = \frac{5v_1 n + 5v_1 (c - 1)}{n(n-1)}
\]

where \( n \geq 10 \) is the number of cards in the deck, \( v_1 \) is the value of the deck, and \( c \geq 1 \) is the number of extra cards drawn by playing the terminal draw card.

**Proof.** Consider the probability distribution for a deck with one terminal draw card. This distribution can be partitioned by the set containing all the subsets of \( S \) with cardinality 5 that contain the terminal draw card and the set containing those that do not. This is the power-set of \( S \) restricted to subsets of \( S \) with a cardinality greater than 5. That is to say, \( \mathcal{P}_{|X|=5}(S) = T \cup T^c \), and \( T \cap T^c = \emptyset \), where: \( T = \{X \in \mathcal{P}(S) \mid |X| = 5 \text{ and } \tau \in X\} \) for the terminal draw card \( \tau \). Since selecting cards from a deck can be modelled with a hyper-geometric distribution, by (3.1) we know that

\[
Pr(T) = \frac{\binom{1}{1}\binom{n-1}{4}}{\binom{n}{5}} = \frac{(n-1)! 5!(n-5)!}{4!(n-5)! n!} = \frac{5}{n}
\]

This implies that
Pr(T\textsuperscript{c}) = 1 - \frac{5}{n} = \frac{n-5}{n}

and that the expected value is E(T\textsuperscript{c}) = \frac{5v_1}{n-1}, since we can remove the Smithy from consideration, and there are no action cards to change how the hand is played. Furthermore,

E(T) = \frac{(5 + c - 1)v_1}{n-1}

because when \( c + 5 \) cards are drawn, at least one of them is the terminal draw card which has a value of 0. From this, we can now express the expected value of the terminal draw deck \( S_1 \) as

\[
E(S) = Pr(T\textsuperscript{c}) \frac{v_1}{n-1} + Pr(T) \frac{(5 + c - 1)v_1}{n-1} = \left( \frac{n-5}{n} \right) \frac{5v_1}{n-1} + \left( \frac{5}{n} \right) \frac{(5 + c - 1)v_1}{n-1}.
\]

We can further simplify this equation.

\[
E(S) = \left( \frac{n-5}{n} \right) \frac{5v_1}{n-1} + \left( \frac{5}{n} \right) \frac{(5 + c - 1)v_1}{n-1} = \frac{5v_1 n - 25v_1 + 25v_1 + 5v_1 (c-1)}{n(n-1)} = \frac{5v_1 n + 5v_1 (c-1)}{n(n-1)} \tag{9}
\]

**Theorem 5.2.** Let \( n \geq 10 \) be the number of cards in a deck, \( v_1 \) be the value of the deck, and \( c \geq 1 \) be the number of cards the terminal draw card adds to a players hand. If the most expensive treasure card a hand can afford has value \( v \), then it is preferable to buy a terminal draw card over the treasure card if and only if \( v_1 \geq \frac{vn}{c} \)

**Proof.** To know whether or not to buy a terminal draw card over a treasure card, we want to compare the expected value of our deck with the terminal draw card on the next turn, against the expected value of our deck with the most expensive treasure card we can afford on our next turn. We know that we can only buy card card at a time, either the terminal draw or the treasure, so both decks will have \( n + 1 \) cards on the next turn. Moreover, if \( v_1 \) denotes our current deck value, after drawing a terminal draw card the value remains \( v_1 \) because the value of a terminal draw card is zero. However, after drawing the treasure card the value of the deck increases to \( v_1 + v \).
Using Proposition 2.1 and Theorem 5.1, we show \( E(S) \geq E(D) \iff v_1 \geq \frac{vn}{c} \).

\[
\begin{align*}
\frac{E(S)}{E(D)} & \geq \frac{5v_1(n + 1) + 5v_1(c - 1)}{(n + 1)n} \geq \frac{5(v_1 + v)}{n + 1} \\
v_1(n + 1) + v_1(c - 1) & \geq (v_1 + v)n \\
v_1n + v_1c & \geq v_1n + vn \\
v_1c & \geq \frac{vn}{c} \\
v_1 & \geq \frac{vn}{c}
\end{align*}
\]

This elegant formula has a simple and natural interpretation: Increasing the number of cards in a deck reduces the utility of a terminal draw card, all else equal. So, \( v_1 \) must grow at a faster rate than \( n \) in order for a terminal draw card to eventually become the better option. For treasure cards with a greater value \( v \), we must have a larger deck value \( v_1 \) to justify the purchase of a terminal draw card. However, the required deck value for buying a terminal draw card can be lowered if a player buys a terminal draw card that allows him/her to draw more cards.

The formula given in Theorem 5.2 is intuitive. Terminal Draw cards are less attractive when:

1. There are high valued treasure cards available
2. The number of cards in one’s deck is high, diminishing the added value of a terminal draw card, and
3. The number of cards that the terminal draw card adds to a hand is small

**Corollary 5.1.** Any terminal draw card drawing at least three cards should be acquired on the first turn if possible.

**Proof.** Applying Theorem 5.2, on the first turn, \( v_1 = 7, n = 10, \) and since the highest value card possible to draw is a Silver, \( v = 2 \). We have \( 7 \geq \frac{2^{10}}{c} \iff c \geq \frac{2^{10}}{7} \approx 3 \)

**Remark 5.2.** Draw cards that add one or two cards to a player’s deck are usually not thought of as terminal draw since these cards are typically accompanied with additional perks, such as extra actions, attacks, reactions to attacks, and so on. So, our analysis of their capacity only as terminal draw cards would not be able to account for the full benefit they provide a player’s deck.
A natural follow up question to ask is: When is it in the player’s best interest to buy a second terminal draw card? We derive a necessarily more complex formula for the expected value of a deck with two terminal draw cards. We consider the opportunity cost as before, alongside the added risk of drawing both terminal cards at the same time.

Per the Big Smithy strategy, there are not additional actions in this hand. So, the second terminal draw is a dead card, no better than an Estate. Therefore, we will need to find the expected value of a terminal draw deck with two terminal draw cards in order to determine when a player should buy his second. Keeping the same definitions for $c$, $v$, and $n$, we prove the following.

**Theorem 5.3.** The expected value of a deck $S_2$ with two terminal draw cards is given by

$$Pr(S_2) = \frac{5v_2n^2 - 20v_2 - 5nv_2 - 30cv_2 + 10ncv_2}{n(n-1)(n-2)}$$

where $n \geq 10$ is the number of cards in the deck, $v_2$ is the value of the deck, and $c \geq 1$ is the number of extra cards drawn by playing the terminal draw card.

**Proof.** Let $\tau_i$, denote a 5-card hand with $i$ terminal draw cards in the initial draw for $i = 0, 1, 2$. Then,

$$P(\tau_1) = \binom{2}{1} \binom{n-2}{4} \binom{n}{5} = \frac{2(n-2)!}{4!(n-6)!} \left( \frac{5!(n-5)!}{n!} \right) = \frac{10(n-5)}{n(n-1)}$$

$$P(\tau_2) = \frac{2}{2} \binom{n-2}{3} \binom{n}{5} = \frac{(n-2)!}{3!(n-5)!} \left( \frac{5!(n-5)!}{n!} \right) = \frac{20}{n(n-1)}$$

$$P(\tau_0) = 1 - (P(\tau_1) + P(\tau_2)) = 1 - \left( \frac{10(n-5) + 20}{n(n-1)} \right) = \frac{n^2 - 11n + 30}{n(n-1)}$$

Now, to find the expected value, let us consider each case in turn. In the denominator we have $n - 2$ since we assume knowledge about whether the terminal draw cards are in the hand. Accordingly, we have:

$$E(\tau_0) = \frac{5v_2}{n-2}$$

$$E(\tau_1) = \frac{(5 + c - 1)v_2}{n-2}$$

$$E(\tau_2) = \frac{(5 + c - 2)v_2}{n-2}$$

The reason we decrease the numerators in the latter two cases is because we know the terminal draw cards already in the hand are guaranteed to contribute a value of zero.
Let $v_2$ be a deck of size $n \geq 12$ with value $v_2$ containing exactly two terminal draw cards of the same type that add $c$ cards to a player's hand when played. Let $S_1$ be the deck formed from $S_2$ by removing one terminal draw card and replacing it with a treasure card of value $v$. Then $E(S_2) \geq E(S_1)$ if and only if

$$v_2 \geq \frac{v(n-2)(n+c-1)}{c(n-4)} .$$

**Proof.** Because of the additional treasure card, the value of $S_1$ is $v_2 + v$. We have the following:

$$\frac{5v_2(n+1)^2 + 10v_2 - 15(n+1)v_2 - 30v_2c + 10(n+1)v_2c}{n(n+1)(n-1)} \geq \frac{5(v_2 + v)(n+1) + 5(v_2 + v)(c-1)}{n(n+1)} \quad \text{if and only if}$$

$$v_2(n^2 + 2 - 3nv - 6c + 2nc) - v_2(n^2 - 3n + nc - 2c + 2) \geq v_2(nc - 4c) \geq v(n^2 - 3n + nc - 2c + 2) \geq v(n-2)(n+c-1) \geq \frac{v(n-2)(n+c-1)}{c(n-4)} .$$

$$\blacksquare$$
Remark 5.3. Note instead, if $n$ is the number of cards in the deck before drawing the second terminal draw or treasure card, the formula becomes

$$v_2 \geq \frac{v(n - 1)(n + c)}{c(n - 3)}.$$  \hspace{1cm} (11)

This inequality is not as elegant or intuitive as the inequality in Theorem 5.2, and the early game turns move quickly enough to make it impractical to apply in an actual game (save of course if one were playing with some incredibly accommodating friends). However, we are still able to provide decision-making guidelines by parametrizing the inequality in terms of the number of turns $t$, and providing appropriate bounds.

We narrow our focus to a specific terminal draw card called Smithy that allows a player to draw three additional cards into their hand, that is, we set $c = 3$. We will find the upper and lower for the turn numbers where a Smithy should be acquired given the best and worst cases for how quickly a player’s money density increases.

First, we make the assumption that after turn two, it is impossible to gain any Copper. We justify this assumption with the incredibly low probability of acquiring two Coppers in the first two turns. On turn one, the probability of getting a Copper is the probability of pulling all three Estates, or $\frac{3}{10} \cdot \frac{2}{9} \cdot \frac{1}{8} = 0.08333334$. For either of the turns that buys a Copper, the other turn will be guaranteed five Copper, and so a Silver will be bought (since Smithies are not yet preferable to a Silver). The probability of buying a second Copper on the third turn given the first was bought is $0.0353535$, so the probability of both happening is $0.002945$. After that the probability of getting a Copper on any given turn is less than or equal to $0.02797$. So while this may not be the lower bound for three out of 100 games playable games, it is sufficient for the vast majority of games.

Recall that for terminal draw cards that draw greater than three, the first card ought to be acquired as soon as possible. Since it is always the case that at least one hand in the first two turns will have enough to buy a Smithy, we know that the first Smithy will be acquired before the third turn. Moreover, the second Smithy cannot be acquired prior to the third turn, since at one of the two hands will not have enough to buy the Smithy. With this in mind, we state the following theorem.

Theorem 5.5. Assume the player has already bought their first Smithy on either turn one or two.

1. Let $t \geq 3$ be the index of a turn where a player can buy a Silver, for turns $t < 5$ the player should buy a Silver over a Smithy, and for turns $t > 6$ the player should buy a Smithy over the Silver.

2. Instead, if $t$ is the index of a turn where a player can only buy a Gold card. For turns $t < 8$, the player should always buy a Gold over a Smithy. For turns $t > 13$, the players should always buy a Smithy over the Gold.
Proof. After two turns, we are assuming a player has 12 cards with value 9 (seven Copper and one Silver) including one Smithy. Because we want to calculate a conservative lower bound, we assume the player buys a Gold on every turn thereafter. So, the value of an $n$-card deck is $3(n - 12) + 9$. Let $t' = n - 12$ represent the number of turns which have passed after the first two. Suppose the player has the option to buy a Silver or a second Smithy. For $c = 3$, we apply the inequality from Remark 5.3.

$$v_1 \geq \frac{v(n-1)(n+c)}{c(n-3)}$$

$$3t' + 9 \geq \frac{2(11 + t')(15 + t')}{3(9 + t')}$$

$$3(3t' + 9)(9 + t') \geq 2(11 + t')(15 + t')$$

$$9(t')^2 + 108t' + 243 \geq 2(t')^2 + 52t' + 330$$

$$7(t')^2 + 56t' \geq 87$$

$$(t')^2 + 8t' \geq \frac{87}{7}$$

$$(t')^2 + 8t' + 16 \geq \frac{199}{7}$$

$$(t' + 4)^2 \geq \frac{199}{7}$$

$$-\sqrt{\frac{199}{7}} \leq t' + 4 \leq \sqrt{\frac{199}{7}}$$

$$-4 - \sqrt{\frac{199}{7}} \leq t' \leq -4 + \sqrt{\frac{199}{7}}$$

$$-1 - \sqrt{\frac{199}{7}} \geq t \geq -1 + \sqrt{\frac{199}{7}}$$

The last step is justified because $t = 3 + t' \implies t' = t - 3$. Simplifying the expression, we get that $t \leq -6.3318$, or $t \geq 4.3318$. Since it is nonsense to talk about a negative turn, we dispense with it, and so $t \geq 4.3318 \implies t \geq 5$.

Now again for the upper bound. Because we want a conservative bound, we assume after the second turn the player will only be able to buy Silver. Using the same notation as above, we have:

$$v_1 \leq \frac{v(n-1)(n+c)}{c(n-3)}$$

$$2t' + 8 \leq \frac{2(11 + t')(15 + t')}{3(9 + t')}$$
\[3(t' + 4)(9 + t') - (11 + t')(15 + t') \leq 0\]
\[4(t')^2 + 26t' + 114 \leq 0\]
\[2(t')^2 + 13t' - 57 \leq 0\]
\[(t' - 3)(2t' + 19) \leq 0\]
\[-\frac{19}{2} \leq t' \leq 3\]
\[-\frac{25}{2} \leq t \leq 6\]

This is the range for which it is possible to \(E(S_2) \leq E(S_1)\). Taken together we know that the optimal time to purchase a Smithy over a Silver is \(5 \leq t \leq 6\).

We can proceed similarly, for Gold. The lower bound can be obtained with the formula
\[3t' + 9 \geq \frac{3(11 + t')(15 + t')}{3(9 + t')}\]
\[(3t' + 9)(9 + t') \geq (11 + t')(15 + t')\]
\[3(t')^2 + 36t' + 81 \geq (t')^2 + 26t' + 165\]
\[2(t')^2 + 10t' \geq 84\]
\[(t')^2 + 5t' + \frac{25}{4} \geq 84 + \frac{25}{4}\]
\[\left(t' + \frac{5}{2}\right)^2 \geq \frac{193}{4}\]
\[t' \geq \frac{-5 + \sqrt{193}}{2}\]
\[-6.4462 \leq t \leq 7.4462 \approx 8\]

For the upper bound, consider
\[2t' + 8 \leq \frac{3(11 + t')(15 + t')}{3(9 + t')}\]
\[(9 + t')(2t' + 8) \leq (11 + t')(15 + t')\]
\[2(t')^2 + 26t' + 72 \leq (t')^2 + 26t' + 165\]
\[(t')^2 \leq 93\]
\[-\sqrt{93} \leq t' \leq \sqrt{93}\]
\[3 - \sqrt{93} \leq t \leq 3 + \sqrt{93}\]

\(3 - \sqrt{93}\) is negative, leaving us with \(t \leq 3 + \sqrt{93} \approx 12.6437 \leq 13\)

So, the bound for when it is possibly to purchase Gold is \(8 \leq t \leq 13\)
Remark 5.4. Theorem 5.5 suggests that Silver beats Smithy when $t < 5$, and Smithy beats Silver when $t > 6$. Similarly, Gold beats Smithy for $t < 8$, and Smithy beats Gold for $t > 13$. For $t = 5, 6$, it is not always the case that a Silver is better than a Smithy, and vice versa. Likewise it is possible that a Smithy is preferable to a Gold when $8 \leq t \leq 13$ but this is not always the case.

6 Discussion

6.1 Results Summary

By modeling Dominion as a Markov chain, we showed that the opportunity cost is always higher for buying treasure cards as opposed to Provinces for Big Money decks. We also showed that it pays to buy a terminal draw card of +3 or more on the first two turns for Terminal Draw decks. For Big Smithy, we showed that the second Smithy should be considered over a Silver only after the fifth turn. It is not until after the sixth turn that the second Smithy is always guaranteed to be a better buy than a Silver, so turns five and six deserve more analysis. One solution is to apply the formula for turns five and six. Perhaps a more practical one is to notice that a player’s decision on a given turn will not effect his deck until the next shuffle. For a Big Money Terminal Draw Deck, turn five is the start of one shuffle, and the player shuffles again at the end of turn six. So, a player might reason on turn five or six that they could buy the Smithy, provided he does not pass up a gold. A similar rationale can be employed when considering the bounds for Gold.

6.2 Limitations of the Model

There are a few limitations to this analysis. We have already mentioned that our model assumes that the deck is shuffled every turn. While in theory this analysis could extend to other strategy schema, in practice the problem becomes unwieldy. For example, while the nature of Engines might make the assumption of shuffling every turn more appropriate, this very nature would open up a much more expansive list of possible states and algorithms for reaching these states, making a purely analytic and fully comprehensive analysis untenable. This would be true in general for schema that rely on the use of several action cards: especially multiple kinds of action cards.

Computing a Markov chain for a Basic Big Money strategy alone proved to be computationally expensive - another limitation of this paper. Initially, the goal was to calculate the distribution for a 20 turn game. After weeks of computing the distribution for a single game, we decided it best to drop the computations down to 15 turns. The computations for a game with 15 turns still took 12 hours per batch of simulations with the program running in exponential time complexity. Markov Chain simulation is intrinsically an exponential time computation, and no speedup techniques overcome this limitation.
6.3 Ideas for Further Research

Big Money, one of the first effective strategies discovered by beginner Dominion players, is a natural starting place for analyzing Dominion strategies, but that leaves other more sophisticated strategies for future research: Combos, Rush, Sludge, and Engines. As mentioned in Section 1, Combo decks use cards whose strengths play off of each other for a more powerful hand. Rush decks attempt to end the game as quickly as possible by depleting supply piles. Sludge decks attempt to draw a game out by junking up a player’s hand with dead cards. Perhaps the most interesting of these archetypes to analyze would be Engines. Engines are decks that are built to chain multiple actions together to draw a large portion of the deck to achieve turns with large payouts.

Future research could investigate the impact of early game decisions on overall performance. For instance, a card known as Chapel allows a player to trash up to four cards from their hand. These are often used in Engines to remove cards that drag the performance of the deck down. When $m_D \geq 1$, it is useful to trash Copper in order to make drawing a higher valued treasure card more likely. However, it might be surprising to learn that many experienced players will even use Chapels to trash Copper when $m_D < 1$. In the short term, the deck’s expected value takes a hit. Nevertheless, experienced players recognize that this decision is an improvement on the deck in the long term. This is because as cards are added to the deck, the Chapel is seen less frequently, and it is seen even less frequently with multiple Coppers in the same hand. Moreover, it is often less productive later in the game to spend an action Chapel-ing away Copper as opposed to playing a different action card. With this in mind, a player might trash as much Copper as they can, even forgoing the chance to buy another treasure card with the Copper they trashed. The analysis presented in Section 5 is limited to describing the performance of a deck immediately after a turn is taken. Therefore, it is not sufficient to account for benefits particular actions can have in the long term. Focusing only on the short-term benefits of an action fails to appreciate scenarios such as the one outlined above, where it is a better play to sacrifice the performance of a deck in the short term for better long-term performance.

Another direction of future research is to model the game more realistically by treating the set of turns between shuffles as a single state. With this model, each state depends only on the prior state, which leads to a computationally tractable calculation. Notably, the first two turns in our model have

$$\binom{10}{5} \times \binom{11}{5} = 116424$$

different possibilities to consider. However, if we are considering shuffles as states, we only have
\[
\binom{10}{5} \binom{5}{5} = 252
\]

different ways of drawing two hands in the first five turns. Moreover in the first two turns there are really only two ways to receive a hand with respect to the amount of Copper: as a 3-4 split, or a 2-5 split. However in our model, it is possible for a to get a hand value of 3, 4, or 5 on the first turn, then 3, 4, 5, or 6 on the second turn, for a total of 12 possible combinations. Though the math would be a bit more complicated, modelling Dominion as a Markov chain with shuffles as states could potentially see a dramatic rise in computational efficiency.

References


Nikolas Koutroulakis