How to Play Fantasy Sports Strategically (and Win)

Martin B. Haugh
Imperial College Business School
Imperial College
m.haugh@imperial.ac.uk

Raghav Singal
Department of IE&OR
Columbia University
rs3566@columbia.edu

This version: May 23, 2019
First version: 17 April, 2018

Abstract

Daily Fantasy Sports (DFS) is a multi-billion dollar industry with millions of annual users and widespread appeal among sports fans across a broad range of popular sports. Building on the recent work of Hunter, Vielma and Zaman (2016), we provide a coherent framework for constructing DFS portfolios where we explicitly model the behavior of other DFS players. We formulate an optimization problem that accurately describes the DFS problem for a risk-neutral decision-maker in both double-up and top-heavy payoff settings. Our formulation maximizes the expected reward subject to feasibility constraints and we relate this formulation to mean-variance optimization and the out-performance of stochastic benchmarks. Using this connection, we show how the problem can be reduced to the problem of solving a series of binary quadratic programs. We also propose an algorithm for solving the problem where the decision-maker can submit multiple entries to the DFS contest. This algorithm is motivated in part by some new results on parimutuel betting which can be viewed as a special case of a DFS contest. One of the contributions of our work is the introduction of a Dirichlet-multinomial data generating process for modeling opponents' team selections and we estimate the parameters of this model via Dirichlet regressions. A further benefit to modeling opponents' team selections is that it enables us to estimate the value in a DFS setting of both insider trading and and collusion. We demonstrate the value of our framework by applying it to DFS contests during the 2017 NFL season.

Keywords: Fantasy sports, portfolio optimization, Dirichlet regression, parimutuel betting, order statistics, binary quadratic programming,.

1. Introduction

Daily Fantasy Sports (DFS) has become a multi-billion dollar industry [1, 25, 32, 40, 42] with millions of annual users [17, 40]. The pervasiveness of fantasy sports in modern popular culture is reflected by the regular appearance of articles discussing fantasy sports issues in the mainstream media. Moreover, major industry developments and scandals are now capable of making headline news as evidenced [13, 23] in Figures 1(a) and 1(b) below. The two major DFS websites are FanDuel and DraftKings and together they control approximately 95% of the U.S. market [25, 32]. Approximately 80% of DFS players have been classified as *minnows* [34] as they are not believed to use sophisticated techniques for decision-making and portfolio construction. Accordingly, these users provide financial opportunities to the so-called *sharks* who do use sophisticated techniques [21, 28, 34, 41] when constructing their fantasy sports portfolios. The goal of this paper is to provide a coherent framework for constructing fantasy sports portfolios where we explicitly model the behavior of other DFS players. Our approach is therefore strategic and to the best of our knowledge, we are the first academic work to develop such an approach in the context of fantasy sports.

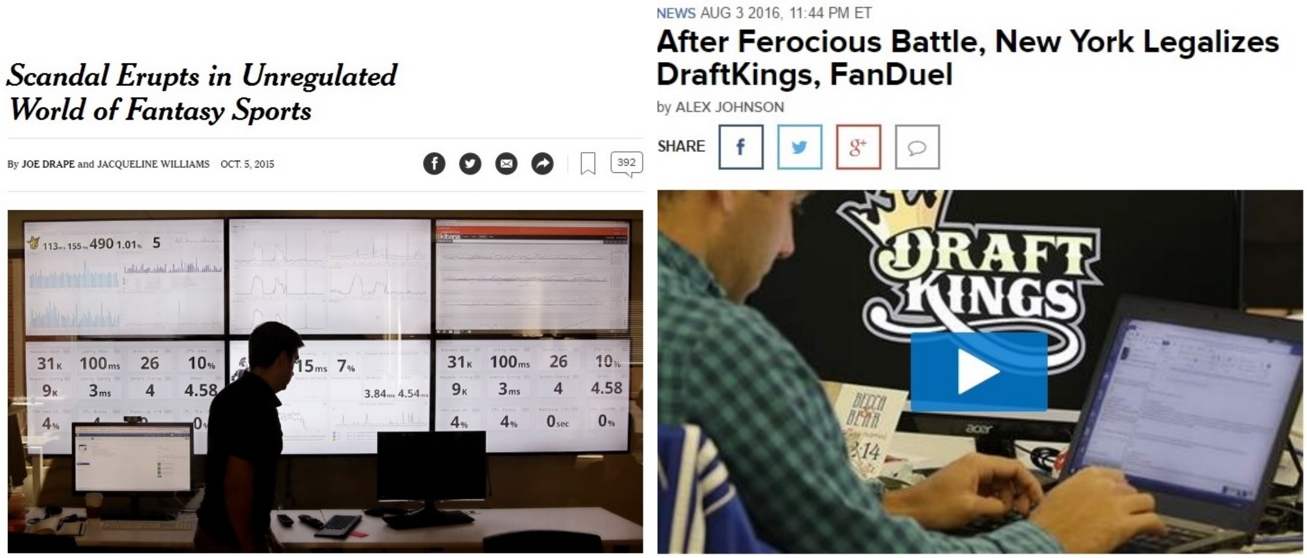
The number of competitors in a typical DFS contest might range from two to hundreds of thousands with each competitor constructing a fantasy team of real-world athletes, e.g. National Football League (NFL) players in a fantasy football contest, with each portfolio being subject to budget and possibly other constraints. The performance of each portfolio is determined by the performances of the real-world athletes in a series of actual games, e.g. the series of NFL games in a given week. The competitors with the best performing entries then earn a monetary reward, which depends on the specific payoff structure, e.g. *double-up* or *top-heavy*, of[1] the DFS contest.

Several papers have already been written on the topic of fantasy sports. For example, Fry, Lundberg, and Ohlmann [16] and Becker and Sun [5] develop models for season-long fantasy contests while Bergman and Imbrogno [6] propose strategies for the survivor pool contest, which is also a season long event. Multiple papers have been written of course on so-called office pools (which pre-date fantasy sports contests) where the goal is to predict the maximum number of game winners in an upcoming elimination tournament such as the *March Madness* college basketball tournament. Examples of this work include Kaplan and Garstka [24] and Clair and Letscher [9]. There has been relatively little work, however, on the problem of constructing portfolios for daily fantasy sports. One notable exception is the recent work of Hunter et al. [22], which is closest to the work we present in this paper. They consider a winner-takes-all payoff structure and aim to maximize the probability that one of their portfolios (out of a total of N) wins. Their approach is a greedy heuristic that maximizes their portfolio means, that is, expected number of fantasy points, subject to constraints that lower bound their portfolio variances and upper bound their inter-portfolio correlations. Technically, their framework requires the solution of linear integer programs and they apply their methodology to fantasy sports contests which are *top-heavy* in their payoff structure as opposed to winner-takes-all. Their work has received considerable attention, e.g. [11], and the authors report earning[2] significant sums in real fantasy sports contests based on the National Hockey League (NHL) and Major League Baseball (MLB).

There are several directions for potential improvement, however, and they are the focus of the work in this paper. First, Hunter et al. [22] do not consider their opponents' behavior. In particular, they do not account for the fact that the payoff thresholds are stochastic and depend on both the performances of the real-world athletes as well as the unknown team selections of their fellow fantasy sports competitors. Second, their framework is only suitable for contests with the top-heavy payoff structure and is in general not suitable for the double-up payoff structure. Third, their approach is based on (approximately) optimizing for the winner-takes-all payoff, which is only a rough approximation to the top-heavy contests they ultimately

[1]Loosely speaking, in a double-up contest a player doubles her money if her entry is among the top 50% of submitted entries. In a top-heavy contest, the rewards are skewed towards the very best performing entries and often decrease rapidly in the rank of the entry. See Section 2 for further details.

[2]They donated their earnings to charity and we have done likewise with our earnings from playing DFS competitions during the 2017 NFL season. The results of these real-world numerical experiments are described in Section 6.



(a) New York Times headline, 5 October 2015

(b) NBC News headline, 3 August 2016

Figure 1: Fantasy sports in the news.

target. In contrast, we directly model the true payoff structure (top-heavy or double-up) and seek to optimize our portfolios with this objective in mind.

Our work makes several contributions to the DFS literature. First, we formulate an optimization problem that accurately describes the DFS problem for a risk-neutral decision-maker in both double-up and top-heavy settings. Our formulation seeks to maximize the expected reward subject to portfolio feasibility constraints and we explicitly account for our opponents' unknown portfolio choices in our formulation. Second, we connect our problem formulation to the finance literature on mean-variance optimization and in particular, the mean-variance literature on outperforming stochastic benchmarks. Using this connection, we show how our problems can be reduced (via some simple assumptions and results from the theory of order statistics) to the problem of solving a series of binary quadratic programs. The third contribution of our work is the introduction of a Dirichlet-multinomial data generating process for modeling opponents' team selections. We estimate the parameters of this model via Dirichlet regressions and we demonstrate its value in predicting opponents' portfolio choices.

We also propose a greedy algorithm for solving the top-heavy problem where the decision-maker can submit multiple entries to the DFS contest. This algorithm is motivated by some new results for the optimization of wagers in a parimutuel contest which can be viewed as a special case of a DFS contest albeit with some important differences. Parimutuel betting in the horse-racing industry has long been a topic of independent interest in its own right, particularly in economics [4, 33, 39, 43], where it has been used to test theories related to market efficiency and information aggregation. We develop a greedy algorithm that is optimal for the parimutuel contest and use this to motivate a similar algorithm for the more complex DFS contests. We also show the DFS objective function for the top-heavy problem with multiple entries is monotone submodular, which enables us to invoke a classic result [30] on submodular maximization to provide further support for our algorithm.

We demonstrate the value of our framework by applying it to both double-up and top-heavy DFS contests in the 2017 NFL season. Despite the fact that DFS contests have a negative net present value (NPV) on average (due to the substantial cut taken by the major DFS websites), we succeeded in earning a net profit over the course of the season. That said, model performance in DFS contests based on a single NFL season has an inherently high variance and so it is difficult to draw meaningful empirical conclusions from just one NFL season. Indeed other sports (baseball, ice hockey, basketball etc.) should have a much lower variance and we believe our approach is particularly suited to these sports.

We also use our model to estimate the value of "insider trading", where an insider, e.g. an employee of the DFS contest organizers, gets to see information on opponents' portfolio choices before making his own team selections. This has been a topic of considerable recent media interest [12, 13]; see also Figure 1(a), which refers to the case of a DraftKings employee using data from DraftKings contests to enter a FanDuel DFS contest in the same week and win $350,000. This problem of insider trading is of course also related to the well known value-of-information concept from decision analysis. While insider trading does result in an increase in expected profits, the benefits of insider trading are mitigated by superior modeling of opponents' team selections. This is not surprising: if we can accurately predict the distribution of opponents' team selection, then insider information will become less and less valuable.

It is also straightforward in our framework to study the benefits of a stylized form of collusion in DFS contests. Specifically, we consider the case where a number N_{collude} of DFS players combine to construct a single portfolio of $N_{\text{collude}} \times E_{\max}$ entries for a given contest, where $E_{\max}$ is the maximum number of permitted entries per DFS player. In contrast, we assume that non-colluders choose identical portfolios of $E_{\max}$ entries. We show the benefits of this type of collusion can be surprisingly large in top-heavy contests. This benefit is actually twofold in that colluding can simultaneously result in a significant increase in the total expected payoff and a significant reduction in the downside risk of the payoff. In practice, however, it's highly unlikely that non-colluding players will choose identical portfolios and so we argue that the benefits of collusion to a risk-neutral player are likely to be quite small.

Beyond proposing a modeling framework for identifying how to construct DFS portfolios, our work also has other implications. To begin with, it should be clear from our general problem formulation and solution approach that high levels of "skill" are required to play fantasy sports successfully. But this is not necessarily in the interest of the fantasy sports industry. In order to maintain popular interest (and resulting profit margins), the industry does not want the role of skill to be too great. Indeed a recent report from McKinsey & Company [26] on fantasy sports makes precisely this point arguing, for example, that chess is a high-skill and deterministic game, which is why it is rarely played for money. In contrast, while clearly a game of high skill, poker also has a high degree of randomness to the point that amateur players often beat professionals in poker tournaments. It is not surprising then that poker is very popular and typically played for money. The framework we have developed in this paper can be used by the fantasy sports industry to determine whether the current DFS game structures achieve a suitable balance between luck and skill. One simple "lever" to adjust this balance, for example, would be to control the amount of data they release regarding the teams selected by the DFS players. By choosing to release no information whatsoever, it will become more difficult for skillful players to estimate their models and take advantage of their superior modeling skills. The industry can also use (as we do) our framework to estimate the value of insider trading and collusion and propose new rules / regulations or payoff structures to counter these concerns.

A recent relevant development occurred in May 2018 when the U.S. Supreme Court struck down a 1992 federal law – the Professional and Amateur Sports Protection Act – that prohibited states from authorizing sports gambling. As a result, some states are taking advantage of this ruling by passing their own sports betting laws and encouraging gambling with the goal of raising additional tax revenue. This remains a controversial development but would certainly appear to be a positive development for the fantasy sports industry. To the extent that individual states seek to regulate online gambling and DFS, the "skill-versus-luck" debate (referenced in the preceding paragraph) may continue to play a role as it has done historically in the federal regulation of gambling in the U.S.

The remainder of this paper is organized as follows. In Section 2, we formulate both the double-up and top-heavy versions of the problem while we outline our Dirichlet regression approach to modeling our opponents' team selections in Section 3. In Section 4, we use results from mean-variance optimization (that relate to maximizing the probability of outperforming a stochastic benchmark) to solve the double-up problem. We then extend this approach to solve the top-heavy problem in Section 5, where we develop and justify our greedy algorithm in the context of parimutuel betting. We present numerical results based on the 2017 NFL season for both problem formulations in Section 6. In Section 7 we discuss the value of information and in particular, how much an insider can profit from having advance knowledge of his opponents' team selections. We also consider the benefits of collusion there. We conclude in Section 8, where we also discuss some directions for ongoing and future research. Various technical details and additional results are deferred to the appendices.

2. Problem Formulation

We assume there are a total of P athletes / real-world players whose performance, $\boldsymbol{\delta} \in \mathbb{R}^P$, in a given round of games is random. We assume that $\boldsymbol{\delta}$ has mean vector $\boldsymbol{\mu}_{\boldsymbol{\delta}}$ and variance-covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\delta}}$. Our decision in the fantasy sports competition is to choose a portfolio $\mathbf{w} \in \{0,1\}^P$ of athletes. Typically, there are many constraints on $\mathbf{w}$. For example, in a typical NFL DFS contest, we will only be allowed to select $C = 9$ athletes out of a total of $P \approx 100$ to 300 NFL players. Each athlete also has a certain "cost" and our portfolio cannot exceed a given budget B. These constraints on $\mathbf{w}$ can then be formulated as

$$\begin{aligned} \sum_{p=1}^{P} w_p &= C \\ \sum_{p=1}^{P} c_p w_p &\leq B \\ w_p &\in \{0,1\}, \quad p = 1, \ldots, P \end{aligned}$$

where c_p denotes the cost of the p^{th} athlete. Other constraints are also typically imposed by the contest organizers. These constraints include positional constraints, e.g. exactly one quarterback can be chosen, diversity constraints, e.g. you can not select more than 4 athletes from any single NFL team, etc. These constraints can generally be modeled as linear constraints and we use $\mathbb{W}$ to denote the set of binary vectors $\mathbf{w} \in \{0,1\}^P$ that satisfy these constraints.

A key aspect of our approach to constructing fantasy sports portfolios is in modeling our opponents, that is, other DFS players who also enter the same fantasy sports contest. We assume there are O such opponents and we use $\mathbf{W}_{\text{op}} := \{\mathbf{w}_o\}_{o=1}^{O}$ to denote their portfolios with each $\mathbf{w}_o \in \mathbb{W}$.

Once the round of NFL games has taken place, we get to observe the realized performances $\boldsymbol{\delta}$ of the P NFL athletes. Our portfolio then realizes a points total of $F := \mathbf{w}^\top \boldsymbol{\delta}$ whereas our opponents' realized points totals are $G_o := \mathbf{w}_o^\top \boldsymbol{\delta}$ for $o = 1, \ldots, O$. All portfolios are then ranked according to their points total and the cash payoffs are determined. These payoffs take different forms depending on the structure of the contest. There are two contest structures that dominate in practice and we consider both of them. They are the so-called double-up and top-heavy payoff structures.

2.1. The Double-Up Problem Formulation

Under the double-up payoff structure, the top r portfolios (according to the ranking based on realized points total) each earn a payoff of R dollars. Suppose now that we enter $N << O$ portfolios[3] to the contest. Then, typical values of r are $r = (O+N)/2$ and $r = (O+N)/5$ with corresponding payoffs of $R = 2$ and $R = 5$ assuming an entry fee of 1 per portfolio. The $(r = (O+N)/2, R = 2)$ case is called a double-up competition whereas the $(r = (O+N)/5, R = 5)$ is called a *quintuple-up* contest. We will refer to all such contests

[3] There is usually a cap on N, denoted by $E_{\max}$, imposed by the contest organizer, however. Typical cap sizes we have observed can range from $E_{\max} = 1$ to $E_{\max} = 150$.

as "double-up" contests except when we wish to draw a distinction between different types of double-up contests, e.g. (true) double-up versus quintuple-up. In practice of course, the contest organizers take a cut and keep approximately 15% of the entry fees for themselves. This is reflected by reducing r appropriately and we note that this is easily accounted for in our problem formulations below. We also note that this means the average DFS player loses approximately 15% of her initial entry. In contrast to financial investments then, DFS investments are on average NPV-negative and so some skill is required in portfolio construction to overcome this handicap.

While it is possible and quite common for a fantasy sports player to submit multiple entries, that is, multiple portfolios, to a given contest, we will consider initially the case where we submit just $N = 1$ entry. Given the double-up payoff structure, our fantasy-sports portfolio optimization problem is to solve

$$\max_{\mathbf{w}\in\mathbb{W}} \mathbb{P}\left\{\mathbf{w}^\top\boldsymbol{\delta} > G^{(r')}(\mathbf{W}_{\text{op}},\boldsymbol{\delta})\right\}, \tag{1}$$

where we use $G^{(r)}$ to denote the r^{th} order statistic of $\{G_o\}_{o=1}^O$ and we define $r' := O + 1 - r$. Note that we explicitly recognize the dependence of $G^{(r)}$ on the portfolio selections $\mathbf{W}_{\text{op}}$ of our O opponents and the performance vector $\boldsymbol{\delta}$ of the NFL athletes.

2.2. The Top-Heavy Problem Formulation

The top-heavy payoff structure is more complicated than the double-up structure as the size of the cash payoff generally increases with the portfolio ranking. In particular, we first define payoffs

$$R_1 > \cdots > R_D > R_{D+1} := 0$$

and corresponding ranks

$$0 := r_0 < r_1 < \cdots < r_D.$$

Then, a portfolio whose rank lies in $(r_{d-1}, r_d]$ wins R_d for $d = 1, \ldots D$. In contrast to the double-up structure, we now account for the possibility of submitting $N > 1$ entries to the contest. We use $\mathbf{W} := \{\mathbf{w}_i\}_{i=1}^N$ to denote these entries and $F_i := \mathbf{w}_i^\top\boldsymbol{\delta}$ to denote the realized fantasy points total of our i^{th} entry. It is then easy to see that our portfolio optimization problem is to solve[4]

$$\max_{\mathbf{W}\in\mathbb{W}^N} \sum_{i=1}^{N}\sum_{d=1}^{D}(R_d - R_{d+1})\,\mathbb{P}\left\{\mathbf{w}_i^\top\boldsymbol{\delta} > G_{-i}^{(r'_d)}(\mathbf{W}_{-i},\mathbf{W}_{\text{op}},\boldsymbol{\delta})\right\} \tag{2}$$

where $r'_d := O + N - r_d$, $G_{-i}^{(r)}$ is the r^{th} order statistic of $\{G_o\}_{o=1}^O \cup \{F_j\}_{j=1}^N \setminus F_i$ and $\mathbf{W}_{-i} := \mathbf{W} \setminus \mathbf{w}_i$.

Later in Section 5, we will discuss our approach to solving (2) and we will argue (based on our parimutuel betting formulation in Section 5.1) that diversification, i.e., choosing N different entries, is a near-optimal strategy. For top-heavy payoffs where the reward R_d decreases rapidly in d, it should be clear why diversification might be a good thing to do. Consider the extreme case of a winner-takes-all structure, for example. Then, absent pathological instances[5], replication of entries means you are only giving yourself one chance to win. This is accounted for in (2) by the fact that your $\mathbf{w}_i^{th}$ entry is "competing" with your other $N-1$ entries as they together comprise $\mathbf{W}_{-i}$. In contrast, when you fully diversify, you are giving yourself N separate chances to win the prize in total. (We are ignoring here the possibility of sharing the prize.)

[4]The probability term in (2) involves a strict inequality ">" but we note that the objective should also include an additional term for $\mathbb{P}(\mathbf{w}_i^\top\boldsymbol{\delta} = G_{-i}^{(r'_d)}(\mathbf{W}_{-i},\mathbf{W}_{\text{op}},\boldsymbol{\delta}))$ in which case a *share* of the reward $(R_d - R_{d+1})$ would be earned. To keep our expressions simple, we don't include this term in (2) as it is generally negligible (except when replication is used) but we do account correctly for such ties in all of our numerical results.

[5]For example, if the best team could be predicted in advance with perfect accuracy, then choosing and replicating this team would be optimal since by replicating this entry you will be (a) guaranteed to win and (b) gain a greater share of the reward if some of your competitors also chose it. If none of your competitors chose the team, you will earn the entire reward for yourself.

We note that the top-heavy payoff structure is our main concern in this paper. That said, it should be clear that the double-up formulation of (1) is a special case of the top-heavy formulation in (2). We will therefore address the double-up problem before taking on the top-heavy problem. Before doing this, however, we must discuss the modeling of our opponents' portfolios $\mathbf{W}_{\text{op}}$.

3. Modeling Opponents' Team Selections

A key aspect of our modeling approach is that there is value to modeling our opponents' portfolio choices, $\mathbf{W}_{\text{op}}$. This is in direct contrast to the work of Hunter et al. [22] who ignore this aspect of the problem and focus instead on constructing portfolios that maximize the expected number of fantasy points, subject to possible constraints[6] that encourage high-variance portfolios. Based on numerical simulations of DFS contests during the 2017 NFL season, we noted it is possible to obtain significant gains in expected dollar payoffs by explicitly modeling $\mathbf{W}_{\text{op}}$. This is partly due to the well-known fact that some athletes are (often considerably) more / less popular than other athletes and because there is some predictability in the team selections of DFS players who may be responding to weekly developments that contain more noise than genuine information. To the best of our knowledge, we are the first to explicitly model $\mathbf{W}_{\text{op}}$ and embed it in our portfolio construction process. That said, we certainly acknowledge that some members of the fantasy sports community also attempt to be strategic in their attempted selection of less popular athletes and avoidance of more popular athletes, other things being equal; see for example Gibbs [19].

If we are to exploit our opponents' team selections, then we must be able to estimate $\mathbf{W}_{\text{op}}$ reasonably accurately. Indeed it is worth emphasizing that $\mathbf{W}_{\text{op}}$ is not observed before the contest and so we must make do with predicting / simulating it, which amounts to being able to predict / simulate the $\mathbf{w}_o$'s. To make things clear, we will focus on the specific case of DFS in the NFL setting. Specifically, consider for example the following NFL contest organized by FanDuel [14]. Each fantasy team has $C = 9$ positions which must consist of 1 quarterback (QB), 2 running backs (RB), 3 wide receivers (WR), 1 tight end (TE), 1 kicker (K) and 1 "defense" (D). We now write $\mathbf{w}_o = (\mathbf{w}_o^{\text{QB}}, \mathbf{w}_o^{\text{RB}}, \ldots, \mathbf{w}_o^{\text{D}})$ where $\mathbf{w}_o^{\text{QB}}$ denotes the quarterback component of $\mathbf{w}_o$, $\mathbf{w}_o^{\text{RB}}$ denotes the running back component of $\mathbf{w}_o$ etc. If there are P_{QB} QBs available for selection then $\mathbf{w}_o^{\text{QB}} \in \{0,1\}^{P_{\text{QB}}}$ and exactly one component of $\mathbf{w}_o^{\text{QB}}$ will be 1 for any feasible $\mathbf{w}_o$. In contrast, $\mathbf{w}_o^{\text{RB}}$ and $\mathbf{w}_o^{\text{WR}}$ will have exactly two and three components, respectively, equal to 1 for any feasible $\mathbf{w}_o$. We refer to $\mathbf{w}_o^{\text{QB}}$, $\mathbf{w}_o^{\text{RB}}$ etc. as the *positional marginals* of $\mathbf{w}_o$. Moreover, it follows that $P_{\text{QB}} + P_{\text{RB}} + \cdots + P_{\text{D}} = P$ since there are P athletes in total available for selection.

In order to model the distribution of $\mathbf{w}_o$, we will use a classic result from copula theory [29], namely Sklar's theorem [37]. This theorem states that we can write

$$F_{\mathbf{w}_o}\left(\mathbf{w}_o^{\text{QB}}, \ldots, \mathbf{w}_o^{\text{D}}\right) = C\left(F_{\text{QB}}\left(\mathbf{w}_o^{\text{QB}}\right), \ldots, F_{\text{D}}\left(\mathbf{w}_o^{\text{D}}\right)\right) \tag{3}$$

where $F_{\mathbf{w}_o}$ denotes the CDF of $\mathbf{w}_o$, F_{QB} denotes the marginal CDF of $\mathbf{w}_o^{\text{QB}}$ etc., and C is the copula of $\mathbf{w}_o^{\text{QB}}, \ldots, \mathbf{w}_o^{\text{D}}$. We note that C, which is only defined uniquely on $\text{Ran}(F_{\text{QB}}) \times \cdots \times \text{Ran}(F_{\text{D}})$, models the dependence structure of the positional marginals. The representation in (3) is convenient as it allows us to break our problem down into two separate sub-problems:

1. Modeling and estimating the positional marginals $F_{\text{QB}}, \ldots, F_{\text{D}}$.
2. Modeling and estimating the copula C.

Moreover, it turns out that the representation in (3) is particularly convenient from an estimation viewpoint as we will have sufficient data to estimate the positional marginals reasonably well whereas obtaining sufficient data to estimate the copula C is challenging. We note that this is often the case in copula modeling

[6] They included constraints that encouraged high-variance portfolios because they too were focused on top-heavy contests where very few contestants earn substantial payoffs. It is intuitively clear that high-variance portfolios are desirable for such contests. We discuss this property in further detail in Sections 4 and 5 in light of the results from mean-variance optimization that we bring to bear on the problem.

applications. For example, in the equity and credit derivatives world in finance, there is often plentiful data on the so-called marginal risk-neutral distributions but relatively little data on the copula C. We begin with the positional marginals.

3.1. The Positional Marginals

To simplify matters, we will focus here on the selection of the QB from the total of P_{QB} that are available. We assume a Dirichlet-multinomial data generating process for a random opponent's selection. Specifically, we assume:

- $\mathbf{p}_{\text{QB}} \sim \text{Dir}(\boldsymbol{\alpha}_{\text{QB}})$ where $\text{Dir}(\boldsymbol{\alpha}_{\text{QB}})$ denotes the Dirichlet distribution with parameter vector $\boldsymbol{\alpha}_{\text{QB}}$.
- A random opponent then selects QB k with probability p^k_{QB} for $k = 1, \ldots, P_{\text{QB}}$, i.e., the chosen QB follows a $\text{Multinomial}(1, \mathbf{p}_{\text{QB}})$ distribution.

Note $\mathbf{p}_{\text{QB}} := \{p^k_{\text{QB}}\}_{k=1}^{P_{\text{QB}}}$ lies on the unit simplex in $\mathbb{R}^{P_{\text{QB}}}$ and therefore defines a probability distribution over the available quarterbacks. It is important to note that $\mathbf{p}_{\text{QB}}$ is not known in advance of the DFS contest. Moreover, they do not appear to be perfectly predictable and so we have to explicitly model[7] their randomness. Accordingly, it is very natural to model $\mathbf{p}_{\text{QB}}$ as following a Dirichlet distribution.

3.1.1. Available Data

In most fantasy sports contests, it is possible to obtain some information regarding $\mathbf{W}_{\text{op}}$ once the contest is over and the winners have been announced. In particular, it is often possible to observe the realized ownership proportions which (because O is assumed large) amounts to observing $\mathbf{p}_{\text{QB}}, \mathbf{p}_{\text{RB}}, \ldots, \mathbf{p}_{\text{D}}$ after each contest. We therefore assume we have such data from a series of historical contests. In practice, we will also have access to other observable features, e.g. expected NFL player performance $\boldsymbol{\mu}_{\boldsymbol{\delta}}$, home or away indicators, quality of opposing teams etc. from these previous contests.

3.1.2. Dirichlet Regression

We can then use this data to build a Dirichlet regression model for estimating the marginal distributions of $\mathbf{w}_o$. We do this by assuming that the parameter vector $\boldsymbol{\alpha}_{\text{QB}} \in \mathbb{R}^{P_{\text{QB}}}$ is predictable. In particular, we assume

$$\boldsymbol{\alpha}_{\text{QB}} = \exp(\mathbf{X}_{\text{QB}} \boldsymbol{\beta}_{\text{QB}}) \tag{4}$$

where $\boldsymbol{\beta}_{\text{QB}}$ is a vector of parameters that we must estimate and $\mathbf{X}_{\text{QB}}$ is a matrix (containing P_{QB} rows) of observable independent variables that are related to the specific features of the NFL games and QBs underlying the DFS contest. To be clear, the exponential function in the r.h.s. of (4) is actually an $P_{\text{QB}} \times 1$ vector of exponentials.

For example, in a DFS contest for week t, we might assume

$$\boldsymbol{\alpha}_{\text{QB},t} = \exp(\beta^0_{\text{QB}} \mathbf{1} + \beta^1_{\text{QB}} \mathbf{f}_{\text{QB},t} + \beta^2_{\text{QB}} \mathbf{c}_{\text{QB},t} + \beta^3_{\text{QB}} \boldsymbol{\mu}_{\text{QB},t}) \tag{5}$$

where $\mathbf{f}_{\text{QB},t} \in \mathbb{R}^{P_{\text{QB}}}$ is an estimate of $\mathbf{p}_{\text{QB}}$ for week t that we can obtain from the FantasyPros website [15], $\mathbf{c}_{\text{QB},t} \in \mathbb{R}^{P_{\text{QB}}}$ are the (appropriately scaled) week t costs of the QBs in the contest, and $\boldsymbol{\mu}_{\text{QB},t}$ is an (appropriately scaled) sub-vector of $\boldsymbol{\mu}_{\boldsymbol{\delta}}$ for week t whose components correspond to the QB positions in $\boldsymbol{\mu}_{\boldsymbol{\delta}}$. Other features are of course also possible. For example, we might also want to include expected returns $\boldsymbol{\mu}_{\text{QB},t} / \mathbf{c}_{\text{QB},t}$ (where division is understood to be component-wise), home-away indicators, quality of opponents etc. as features.

[7] In initial unreported experiments, we assumed $\mathbf{p}_{\text{QB}}$ was fixed and known but this led to over-certainty and poor performance of the resulting portfolios.

We can estimate the $\boldsymbol{\beta}_{\text{QB}}$ vector by fitting a Bayesian Dirichlet regression. Assuming we have data from weeks $t = 1$ to $t = T - 1$ and a flat prior on $\boldsymbol{\beta}_{\text{QB}}$, then the posterior satisfies

$$\begin{aligned} p(\boldsymbol{\beta}_{\text{QB}} \mid \{\mathbf{p}_{\text{QB},t}\}_{t=1}^{T-1}) &\propto \underbrace{p(\boldsymbol{\beta}_{\text{QB}})}_{\propto 1} p(\{\mathbf{p}_{\text{QB},t}\}_{t=1}^{T-1} \mid \boldsymbol{\beta}_{\text{QB}}) \\ &\propto \prod_{t=1}^{T-1} \text{Dir}(\mathbf{p}_{\text{QB},t} \mid \underbrace{e^{\mathbf{X}_{\text{QB},t}\boldsymbol{\beta}_{\text{QB}}}}_{=\boldsymbol{\alpha}_{\text{QB},t}}) \\ &\propto \prod_{t=1}^{T-1} \frac{1}{\text{B}(\boldsymbol{\alpha}_{\text{QB},t})} \prod_{k=1}^{P_{\text{QB}}} \left(p_{\text{QB},t}^{k}\right)^{\alpha_{\text{QB},t}^{k}-1} \end{aligned} \tag{6}$$

where $\text{B}(\boldsymbol{\alpha}_{\text{QB},t})$ is the normalization factor for the Dirichlet distribution. We fit this model using the Bayesian software package `STAN` [38].

3.1.3. The Other Positions

It should be clear that we can handle the other positions in a similar fashion. In the case of the three selected WRs for example, we assume $\mathbf{p}_{\text{WR}} \sim \text{Dir}(\boldsymbol{\alpha}_{\text{WR}})$ and that a random opponent then selects her three WRs according[8] to a Multinomial$(3, \mathbf{p}_{\text{WR}})$ distribution. We can again use Dirichlet regression to estimate the parameter vector $\boldsymbol{\beta}_{\text{WR}}$ where $\boldsymbol{\alpha}_{\text{WR}} = \exp(\mathbf{X}_{\text{WR}}\boldsymbol{\beta}_{\text{WR}})$.

3.2. The Copula

Returning to (3), the question arises as to what copula C should we use? This is a difficult question to answer as we will generally have little or no data available to estimate C. To be clear, the data we do have available is typically data on the positional marginals. In order to obtain data that is useful for estimating C, we would need to have access to the teams selected by contestants in historical DFS contests. Such data is hard to come by although it is often possible to obtain a small sample of selected teams via manual[9] inspection on the contest web-sites. As a result, we restrict ourselves to three possible choices of C:

1. The *independence* copula C_{ind} satisfies $C_{\text{ind}}(u_1, \ldots, u_m) = \prod_{i=1}^m u_i$ where m is the number of positional marginals. As the name suggests, the independence copula models independence among the selected positions so that when a contestant is choosing her TE position, for example, it is done so independently of her selections for the other positions. The independence copula can therefore be interpreted as the copula of a non-strategic contestant.
2. The *stacking* copula C_{stack} is intended to model the well known [3] stacking behavior of some strategic contestants. In the NFL setting, for example, it is well known that the points scored by a given team's QB and main WR[10] are often strongly positively correlated. Selecting both players then becomes attractive to contestants who understand that positive correlation will serve to increase the overall variance of their entry, which is generally a desirable feature in top-heavy contests as we will argue in Section 5. Rather than explicitly defining the stacking copula (which would require further notation), we simply note that it is straightforward to simulate a value of $\mathbf{w}_o$ when C is the stacking copula. We first generate the QB position, i.e., $\mathbf{w}_o^{\text{QB}}$. We then select the first WR to be the main WR from the generated QB's team. The remaining 2 WRs are generated from the Multinomial$(2, \mathbf{p}_{\text{WR}})$ distribution. It is easy to ensure that all 3 WRs are different; see footnote 8.

[8] In fact, the rules of a DFS contest are likely to state that the same player can not be chosen more than once. In that case, we could simply repeatedly draw from the Multinomial$(3, \mathbf{p}_{\text{WR}})$ distribution until 3 different WRs are selected. Alternatively (but equivalently), we could draw each WR sequentially adjusting the multinomial distribution each time so that once selected, a WR cannot be selected again.

[9] These web-sites are generally not easy to "scrape" nor do the owners look favorably on web-scrapers.

[10] By "main" WR of a team, we refer to the WR with the highest expected points among all the WRs in the same team.

3. The *mixture* copula sets $C(\cdot) := (1-q)C_{\text{ind}}(\cdot) + qC_{\text{stack}}(\cdot)$ for some $q \in (0,1)$. Note that a finite mixture of copulas remains a copula. While very little data on complete team selections is available, as mentioned above, it is possible to observe a small sample of teams and such a sample could be used to estimate q. We can then interpret q as being the probability that a random contestant will be a "stacker" and therefore be represented by the stacking copula.

We note that different contest structures tend to result in more or less strategic behavior. Top-heavy contests, for example, encourage high variance teams, which suggests that stacking might be more common in those contests. Indeed that is what we observe and so the estimated q is typically higher for top-heavy contests. Finally, we note that the main fantasy sports companies will themselves have access to the team selections of all players and these companies could easily fit more sophisticated copulas to the data. This might be of general interest to these companies but it might also be useful to help them understand the skill-luck tradeoff in playing fantasy sports.

3.3. Generating Random Opponents' Portfolios

Suppose now that the Dirichlet regression model has been fit for each of the positional marginals and that we have also estimated the q parameter for the mixture copula. It is then easy to simulate a candidate $\mathbf{w}_o$. We first generate Stack $\sim$ Bernoulli(q) and if Stack $= 0$, we use the independence copula, which amounts to generating the player in each position independently of the selection for all of the other positions. For example, to generate the QB selection, we must:

(i) First draw a sample $\mathbf{p}_{\text{QB}}$ from the $\text{Dir}(\boldsymbol{\alpha}_{\text{QB}})$ distribution.

(ii) Then draw a sample from the $\text{Mult}(1, \mathbf{p}_{\text{QB}})$ distribution.

(iii) This draw then defines our chosen QB, i.e., it sets one component of $\mathbf{w}_o^{\text{QB}}$ to 1 with the others being set to 0.

We repeat this for all positions. If Stack $= 1$, however, then we use the stacking copula and therefore follow the same steps except we must set the first WR to be the main WR from the selected QB's team. At this point, we only have a *candidate* $\mathbf{w}_o$ as there is no guarantee that the resulting $\mathbf{w}_o$ is feasible, i.e., that $\mathbf{w}_o \in \mathbb{W}$. We therefore use an accept-reject approach whereby candidate $\mathbf{w}_o$'s are generated according to the steps outlined above and are only accepted if they are feasible, that is, if $\mathbf{w}_o \in \mathbb{W}$. In fact, we impose one further condition: we insist that an accepted $\mathbf{w}_o$ uses up most of the available budget. We impose this condition because it is very unlikely in practice that a fantasy player in a DFS contest would leave much of her budget unspent. This is purely a behavioral requirement and so we insist the cost of an accepted $\mathbf{w}_o$ satisfy $\mathbf{c}^\top \mathbf{w}_o \geq B_{\text{lb}}$ for some lower bound $B_{\text{lb}} \leq B$ that we get to choose. Algorithm 1 below describes how to generate O random opponents' portfolios $\mathbf{W}_{\text{op}}$ and it therefore (implicitly) defines the distribution of $\mathbf{W}_{\text{op}}$.

4. Solving the Double-Up Problem

As mentioned earlier, we first tackle the double-up problem since our solution to this problem will help inform how we approach the top-heavy problem. We begin first by recalling a result from mean-variance optimization and in particular, the problem of maximizing the probability of exceeding a stochastic benchmark. Our approach to solving both double-up and top-heavy problems will be a mean-variance optimization based on this result.

4.1. Mean Variance Optimization and Outperforming Stochastic Benchmarks

We consider[11] a one-period problem where at time $t = 0$ there are P financial securities available to invest in. At time $t = 1$ the corresponding random return vector $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_P)$ is realized. Let $\boldsymbol{\mu}_{\boldsymbol{\xi}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\xi}}$ denote

[11] The material and results in this subsection follow Morton et al. [27] and they should be consulted for further details and related results. In this subsection, we will sometimes use the same notation from earlier sections to make the connections between the financial problem of this subsection and the DFS problem more apparent.

Algorithm 1 Sampling O Opponent Portfolios

Require: $(\boldsymbol{\beta}_{\text{QB}}, \ldots, \boldsymbol{\beta}_{\text{D}})$, $(\mathbf{X}_{\text{QB}}, \ldots, \mathbf{X}_{\text{D}})$, $\mathbf{c}$, B_{lb}, q

$(\boldsymbol{\alpha}_{\text{QB}}, \ldots, \boldsymbol{\alpha}_{\text{D}}) = (\exp(\mathbf{X}_{\text{QB}}\boldsymbol{\beta}_{\text{QB}}), \ldots, \exp(\mathbf{X}_{\text{D}}\boldsymbol{\beta}_{\text{D}}))$
$(\mathbf{p}_{\text{QB}}, \ldots, \mathbf{p}_{\text{D}}) \sim (\text{Dir}(\boldsymbol{\alpha}_{\text{QB}}), \ldots, \text{Dir}(\boldsymbol{\alpha}_{\text{D}}))$
for $o = 1 : O$ **do**
  Stack $\sim$ Bernoulli(q)
  Reject = True
  while Reject **do**
    $(k_{\text{QB}}, k_{\text{RB}}, \ldots, k_{\text{D}}) \sim (\text{Mult}(1, \mathbf{p}_{\text{QB}}), \text{Mult}(2, \mathbf{p}_{\text{RB}}), \ldots, \text{Mult}(1, \mathbf{p}_{\text{D}}))$
    % Mult$(2, \mathbf{p}_{\text{RB}})$ etc. should be understood as being without replacement; see footnote 8
    if Stack = 1 **then**
      Replace $k_{\text{WR}}(1)$ with main WR from team of k_{QB}
    end if
    Let $\mathbf{w}_o$ denote the portfolio corresponding to $(k_{\text{QB}}, \ldots, k_{\text{D}})$
    if $\mathbf{w}_o \in \mathbb{W}$ and $\mathbf{c}^\top \mathbf{w}_o \geq B_{\text{lb}}$ **then**
      Reject = False **and**
      Accept $\mathbf{w}_o$
    end if
  end while
end for
return $\mathbf{W}_{\text{op}} = \{\mathbf{w}_o\}_{o=1}^{O}$

the mean return vector and variance-covariance matrix, respectively, of $\boldsymbol{\xi}$. The goal is then to construct a portfolio $\mathbf{w} = (w_1, \ldots, w_P)$ with random return $R_{\mathbf{w}} = \mathbf{w}^\top \boldsymbol{\xi}$ that maximizes the probability of exceeding a random benchmark R_b. Mathematically, we wish to solve

$$\max_{\mathbf{w} \in \mathbb{W}} \mathbb{P}(R_{\mathbf{w}} - R_b \geq 0) \tag{7}$$

where $\mathbb{W}$ includes the budget constraint $\mathbf{w}^\top \mathbf{1} = 1$ as well as any other linear constraints we wish to impose. If we assume $R_{\mathbf{w}} - R_b$ has a normal distribution so that $R_{\mathbf{w}} - R_b \sim \text{N}(\mu_{\mathbf{w}}, \sigma^2_{\mathbf{w}})$ for some[12] $\mu_{\mathbf{w}}$ and $\sigma^2_{\mathbf{w}}$ that depend on $\mathbf{w}$, then (7) amounts to solving

$$\max_{\mathbf{w} \in \mathbb{W}} 1 - \Phi\left(-\frac{\mu_{\mathbf{w}}}{\sigma_{\mathbf{w}}}\right) \tag{8}$$

where $\Phi(\cdot)$ denotes the standard normal CDF. Let $\mathbf{w}^*$ be the optimal solution to (8). The following result is adapted from Morton et al. [27] and follows from the representation in (8).

Proposition 4.1. *Suppose $R_{\mathbf{w}} - R_b \sim N(\mu_{\mathbf{w}}, \sigma^2_{\mathbf{w}})$ for all $\mathbf{w} \in \mathbb{W}$.*

(i) *Suppose $\mu_{\mathbf{w}} < 0$ for all $\mathbf{w} \in \mathbb{W}$. Then*

$$\mathbf{w}^* \in \left\{\mathbf{w}(\lambda) : \mathbf{w}(\lambda) \in \arg\max_{\mathbf{w} \in \mathbb{W}} (\mu_{\mathbf{w}} + \lambda \sigma^2_{\mathbf{w}}),\ \lambda \geq 0\right\}. \tag{9}$$

(ii) *Suppose $\mu_{\mathbf{w}} \geq 0$ for some $\mathbf{w} \in \mathbb{W}$. Then*

$$\mathbf{w}^* \in \left\{\mathbf{w}(\lambda) : \mathbf{w}(\lambda) \in \arg\max_{\mathbf{w} \in \mathbb{W},\, \mu_{\mathbf{w}} \geq 0} (\mu_{\mathbf{w}} - \lambda \sigma^2_{\mathbf{w}}),\ \lambda \geq 0\right\} \tag{10}$$

[12] If the benchmark R_b is deterministic, then $\mu_{\mathbf{w}} := \mathbf{w}^\top \boldsymbol{\mu}_{\boldsymbol{\xi}} - R_b$ and $\sigma^2_{\mathbf{w}} := \mathbf{w}^\top \boldsymbol{\Sigma}_{\boldsymbol{\xi}} \mathbf{w}$.

so that $\mathbf{w}^*$ is mean-variance efficient.

Proposition 4.1 is useful because it allows us to solve the problem in (8) efficiently. In particular, we determine which of the two cases from the proposition applies. This can be done when $\mathbb{W}$ is polyhedral by simply solving a linear program that maximizes $\mu_{\mathbf{w}}$ (which is affine in $\mathbf{w}$) over $\mathbf{w} \in \mathbb{W}$. If the optimal mean is negative, then we are in case (i); otherwise we are in case (ii). We then form a grid Λ of possible λ values and for each $\lambda \in \Lambda$, we solve the appropriate quadratic optimization problem (defining $\mathbf{w}(\lambda)$) from (9) or (10) and then choose the value of λ that yields the largest objective in (7) or (8). See Algorithm 2 in Section 4.2 below for when we apply these results to our double-up problem.

4.2. The Double-Up Problem

Recall now the double-up problem as formulated in (1). We define $Y_{\mathbf{w}} := \mathbf{w}^\top \boldsymbol{\delta} - G^{(r')}$ and note that

$$\begin{aligned} \mu_{Y_{\mathbf{w}}} &:= \mathbf{w}^\top \boldsymbol{\mu}_{\boldsymbol{\delta}} - \mu_{G^{(r')}} \\ \sigma^2_{Y_{\mathbf{w}}} &:= \mathbf{w}^\top \boldsymbol{\Sigma}_{\boldsymbol{\delta}} \mathbf{w} + \sigma^2_{G^{(r')}} - 2\mathbf{w}^\top \boldsymbol{\sigma}_{\boldsymbol{\delta}, G^{(r')}} \end{aligned} \tag{11}$$

where $\mu_{G^{(r')}} := \mathrm{E}\left[G^{(r')}\right]$, $\sigma^2_{G^{(r')}} := \mathrm{Var}\left(G^{(r')}\right)$ and $\boldsymbol{\sigma}_{\boldsymbol{\delta}, G^{(r')}}$ is a $P \times 1$ vector with p^{th} component equal to $\mathrm{Cov}(\delta_p, G^{(r')})$. Our approach to solving (1) is based on Proposition 4.1 and is presented in Algorithm 2 below. While this algorithm will deliver the optimal solution in the event that each $Y_{\mathbf{w}} \sim \mathrm{N}(\mu_{Y_{\mathbf{w}}}, \sigma^2_{Y_{\mathbf{w}}})$, it should yield a good approximate solution even when the $Y_{\mathbf{w}}$'s are not normally distributed. Indeed the key insights yielded by Proposition 4.1 don't rely on the normality of the $Y_{\mathbf{w}}$'s. Specifically, if $\mu_{Y_{\mathbf{w}}} < 0$ for all $\mathbf{w} \in \mathbb{W}$, then it seems intuitively clear that we need to select a team $\mathbf{w}$ that simultaneously has a high mean and a high variance. The appropriate balance between mean and variance in the objective function will then be determined by λ. Similarly, if there is at least one $\mathbf{w} \in \mathbb{W}$ such that $\mu_{Y_{\mathbf{w}}} > 0$, then intuition suggests we can search for a team $\mathbf{w}$ with a large (and positive) mean and a small variance. Again, the appropriate balance between the two will be determined by λ. Not insisting on the normality of $Y_{\mathbf{w}}$ also gives us the freedom to consider using non-normal distributions for $\boldsymbol{\delta}$. Indeed this parallels the situation in the asset allocation literature in finance where the mean-variance paradigm remains[13] very popular despite the well-known fact that asset returns have heavy tails and therefore are not normally distributed.

Algorithm 2 Optimization for the Double-Up Problem with a Single Entry

Require: $\mathbb{W}$, Λ, $\boldsymbol{\mu}_{\boldsymbol{\delta}}$, $\boldsymbol{\Sigma}_{\boldsymbol{\delta}}$, $\mu_{G^{(r')}}$, $\sigma^2_{G^{(r')}}$, $\boldsymbol{\sigma}_{\boldsymbol{\delta}, G^{(r')}}$ and Monte Carlo samples of $(\boldsymbol{\delta}, G^{(r')})$

1: **if** $\exists \mathbf{w} \in \mathbb{W}$ with $\mu_{Y_{\mathbf{w}}} \geq 0$ **then**
2: **for all** $\lambda \in \Lambda$ **do**
3: $\mathbf{w}_\lambda = \underset{\mathbf{w} \in \mathbb{W},\, \mu_{Y_{\mathbf{w}}} \geq 0}{\mathrm{argmax}} \left\{ \mu_{Y_{\mathbf{w}}} - \lambda \sigma^2_{Y_{\mathbf{w}}} \right\}$
4: **end for**
5: **else**
6: **for all** $\lambda \in \Lambda$ **do**
7: $\mathbf{w}_\lambda = \underset{\mathbf{w} \in \mathbb{W}}{\mathrm{argmax}} \left\{ \mu_{Y_{\mathbf{w}}} + \lambda \sigma^2_{Y_{\mathbf{w}}} \right\}$
8: **end for**
9: **end if**
10: $\lambda^* = \underset{\lambda \in \Lambda}{\mathrm{argmax}}\ \mathbb{P}\{Y_{\mathbf{w}_\lambda} > 0\}$
11: **return** $\mathbf{w}_{\lambda^*}$

[13]To be clear, we are not claiming the original mean-variance approach of Markowitz is popular. Indeed it's well known that parameter estimation issues render Markowitz useless in practice. Developments which build on Markowitz such as Black-Litterman, robust mean-variance etc. are popular, however, and they too take a mean-variance perspective.

Remark 4.1. *Note that λ^* in line 10 can be computed using the Monte Carlo samples of $(\boldsymbol{\delta}, G^{(r')})$ that are inputs to the algorithm. In this case, the computation of $\mathbb{P}\{Y_{\mathbf{w}_\lambda} > 0\}$ does not rely on any normal approximation. Alternatively, λ^* could also be estimated via the assumption that each $Y_{\mathbf{w}_\lambda}$ is approximately normally distributed. We also note that if it turns out that $\lambda^* \approx 0$, then the optimization will basically seek to maximize $\mathbf{w}^\top \boldsymbol{\mu}_{\boldsymbol{\delta}}$, thereby suggesting there is little value to be gained from modeling opponents.*

One difficulty that might arise with the mean-variance approach is if the distribution of the $Y_{\mathbf{w}}$'s display a significant skew. While we have seen no evidence[14] of this when $\boldsymbol{\delta}$ is assumed multivariate normally distributed, we might see such a skew if we assumed a distribution for $\boldsymbol{\delta}$ which also had a skew. A significant skew in the $Y_{\mathbf{w}}$'s could then result in us mistakenly seeking a portfolio with a small variance or vice versa. To see this, consider a double-up contest where the top 50% of contestants earn a reward. If the $Y_{\mathbf{w}}$'s display a significant right skew, then their medians will be less than their means. It's then possible there exists a $\mathbf{w}$ such $\mu_{Y_{\mathbf{w}}} \geq 0$ but that $\text{median}(Y_{\mathbf{w}'}) < 0$ for all $\mathbf{w}' \in \mathbb{W}$. In that event, the condition of the if statement on line 1 of Algorithm 2 will be satisfied and so we end up seeking a team with a large mean and a small variance. It's possible, however, that we should be seeking a team with a large mean and a large variance since $\text{median}(Y_{\mathbf{w}'}) < 0$ for all $\mathbf{w}' \in \mathbb{W}$. Note that such a mistake might occur because the reward cutoff of the contest is determined by the median and not the mean, which is what we use in Algorithm 2. Of course this issue doesn't arise if the $Y_{\mathbf{w}}$'s are normal since then the means and medians coincide. An easy solution to this problem is to simply ignore the if-else statements in Algorithm 2 and consider both possibilities. This results in Algorithm 3, which is presented below.

Algorithm 3 Adjusted Optimization for the Double-Up Problem with a Single Entry

Require: $\mathbb{W}$, Λ, $\boldsymbol{\mu}_{\boldsymbol{\delta}}$, $\boldsymbol{\Sigma}_{\boldsymbol{\delta}}$, $\mu_{G^{(r')}}$, $\sigma^2_{G^{(r')}}$, $\boldsymbol{\sigma}_{\boldsymbol{\delta}, G^{(r')}}$ and Monte Carlo samples of $(\boldsymbol{\delta}, G^{(r')})$

for all $\lambda \in \Lambda$ **do**

$\quad \mathbf{w}_{\lambda^-} = \underset{\mathbf{w} \in \mathbb{W},\, \mu_{Y_{\mathbf{w}}} \geq 0}{\text{argmax}} \left\{ \mu_{Y_{\mathbf{w}}} - \lambda \sigma^2_{Y_{\mathbf{w}}} \right\}$

$\quad \mathbf{w}_{\lambda^+} = \underset{\mathbf{w} \in \mathbb{W}}{\text{argmax}} \left\{ \mu_{Y_{\mathbf{w}}} + \lambda \sigma^2_{Y_{\mathbf{w}}} \right\}$

end for

return $\mathbf{w}^* = \underset{\mathbf{w} \in \{\mathbf{w}_{\lambda^-}, \mathbf{w}_{\lambda^+} : \lambda \in \Lambda\}}{\text{argmax}} \mathbb{P}\{Y_{\mathbf{w}} > 0\}$

4.2.1. Generating Monte Carlo Samples

In order to execute Algorithm 2, we must first compute the inputs $\mu_{G^{(r')}}$, $\sigma^2_{G^{(r')}}$ and $\boldsymbol{\sigma}_{\boldsymbol{\delta}, G^{(r')}}$ as defined above. These quantities can be estimated off-line via Monte Carlo simulation as they do not depend on our portfolio choice $\mathbf{w}$. We simply note here that the Monte Carlo can be performed relatively efficiently using results from the theory of order statistics. The specific details can be found in Appendix A.

4.2.2. Solving the Binary Quadratic Programs

The optimization problems in lines 3 and 7 of Algorithm 2 require the solution of binary quadratic programs (BQPs). In our numerical experiments of Sections 6 and 7, we solved these BQPs using Gurobi's [20] default BQP solver although the specific algorithm used by Gurobi was not clear from the online documentation. (We do note in passing, however, that it is straightforward to transform a BQP into an equivalent binary program (BP) at the cost of adding $O(P^2)$ binary variables and $O(P^2)$ linear constraints.)

[14]In unreported experiments, we found $Y_{\mathbf{w}}$ to be unimodal and very well approximated by a normal distribution for a test-set of $\mathbf{w}$'s when $\boldsymbol{\delta}$ was also normally distributed.

4.3. The Double-Up Problem with Multiple Entries

We now briefly consider the case where we can submit a fixed but finite number of $N > 1$ entries to the double-up contest. In the case of a risk-neutral DFS player, it seems intuitively clear that if $O \to \infty$ so that $N/O \to 0$, then a *replication* strategy is an optimal or near-optimal strategy. In particular, under such a replication strategy the DFS player should solve for the $N = 1$ optimal portfolio and then submit N copies of this portfolio to the contest. To gain some intuition for this observation, consider a double-up contest where the top 50% of entries double their money and let $\mathbf{w}^*$ be the optimal entry for the $N = 1$ problem. Now consider the $N = 2$ case with O large and suppose we submit two copies of $\mathbf{w}^*$. Given the optimality of $\mathbf{w}^*$ for the $N = 1$ problem, submitting two copies of it (for the $N = 2$ problem) can only be sub-optimal to the extent that the second copy of $\mathbf{w}^*$ competes with the first copy. But such competition can only occur to the extent that $\mathbf{w}^*$ is the entry defining the boundary cutoff $G^{(r')}$. But this event will (in general) occur with vanishingly small probability in the limit as $O \to \infty$.

Even when O is not large, we suspect the replication strategy will still be close to optimal. The key issue would be the probability of having a portfolio exactly at the cutoff $G^{(r')}$ between receiving and not receiving the cash payoff. While we can derive conditions guaranteeing the optimality or near-optimality of replication for double-up contests, these conditions are not easily expressed in terms of the *observable* parameters of the contest. This is not surprising since we know the double-up payoff structure can be viewed as a special case of the top-heavy payoff structure and certainly (as we shall see) replication is not optimal for top-heavy contests.

There is a simple test we can deploy, however, to check if replication is indeed optimal in any given double-up contest. To see this let $\mathcal{R}(N \times \mathbf{w}^*)$ denote the expected reward when we replicate $\mathbf{w}^*$ (the $N = 1$ optimal entry) N times. Using Monte-Carlo we can easily check to see whether or not $\mathcal{R}(N \times \mathbf{w}^*) \approx N\mathcal{R}(\mathbf{w}^*)$. If this is the case then we know that replication of $\mathbf{w}^*$ is optimal because it must be the case that the expected reward of any portfolio of N entries is less than or equal to $N\mathcal{R}(\mathbf{w}^*)$. In fact this is what we observed in various simulation experiments calibrated to the real-world contests of Section 6. Connecting to our discussion above, this suggests that the probability of $\mathbf{w}^*$ being at or very close to the cutoff $G^{(r')}$ was essentially zero in these numerical experiments.

Finally, if it transpires for a given double-up contest that $\mathcal{R}(N \times \mathbf{w}^*) < N\mathcal{R}(\mathbf{w}^*)$ then rather than simply enforcing replication, we note it is easy to define[15] a greedy algorithm analogous to Algorithm 6 in Section 5.2 where replication is not *imposed* a priori on the portfolio of N entries. We note replication may still be optimal in that case, however, in which case the greedy algorithm is likely to return a replication portfolio.

5. Solving the Top-Heavy Problem

We can now extend the analysis we developed for the double-up problem in Section 4 to tackle the more interesting top-heavy problem. We consider first the single-entry case where $N = 1$. In that case, the problem in (2) simplifies to solving

$$\max_{\mathbf{w} \in \mathbb{W}} \sum_{d=1}^{D} (R_d - R_{d+1})\, \mathbb{P}\left\{\mathbf{w}^\top \boldsymbol{\delta} > G^{(r'_d)}(\mathbf{W}_{\text{op}}, \boldsymbol{\delta})\right\}, \tag{12}$$

where $r'_d := O + 1 - r_d$. Following the development in Section 4.2, we can define $Y^d_{\mathbf{w}} := \mathbf{w}^\top \boldsymbol{\delta} - G^{(r'_d)}$ and define

$$\mu_{Y^d_{\mathbf{w}}} := \mathbf{w}^\top \boldsymbol{\mu}_{\boldsymbol{\delta}} - \mu_{G^{(r'_d)}} \tag{13}$$

$$\sigma^2_{Y^d_{\mathbf{w}}} := \mathbf{w}^\top \boldsymbol{\Sigma}_{\boldsymbol{\delta}} \mathbf{w} + \sigma^2_{G^{(r'_d)}} - 2\mathbf{w}^\top \boldsymbol{\sigma}_{\boldsymbol{\delta}, G^{(r'_d)}} \tag{14}$$

[15] The theory that we will develop in Sections 5.1 and 5.2.2 also applies to double-up contests and so a greedy algorithm for double-up contests analogous to Algorithm 6 in Section 5.2 would have the same theoretical support.

where $\mu_{G^{(r'_d)}} := \mathrm{E}\left[G^{(r'_d)}\right]$, $\sigma^2_{G^{(r'_d)}} := \mathrm{Var}\left(G^{(r'_d)}\right)$ and $\boldsymbol{\sigma}_{\boldsymbol{\delta},G^{(r'_d)}}$ is a $P \times 1$ vector with p^{th} component equal to $\mathrm{Cov}(\delta_p, G^{(r'_d)})$. Following our mean-variance approach, we can now approximate (12) as

$$\max_{\mathbf{w} \in \mathbb{W}} \sum_{d=1}^{D} (R_d - R_{d+1}) \left(1 - \Phi\left(-\frac{\mu_{Y^d_{\mathbf{w}}}}{\sigma_{Y^d_{\mathbf{w}}}}\right)\right). \tag{15}$$

Before proceeding, we need to make two additional assumptions, which we will state formally.

Assumption 5.1. *$\mu_{Y^d_{\mathbf{w}}} < 0$ for $d = 1, \ldots, D$ and for all $\mathbf{w} \in \mathbb{W}$.*

Assumption 5.1 can be interpreted as stating that, in expectation, the points total of our optimal portfolio will not be sufficient to achieve the minimum payout R_D. In option-pricing terminology, we are therefore assuming our optimal portfolio is "out-of-the-money". This is a very reasonable assumption to make for top-heavy contests where it is often the case that only the top 20% or so of entries earn a cash payout. In numerical experiments, our model often predicts that our optimal portfolio will (in expectation) be at or around the top 20^{th} percentile. The assumption therefore may break down if payoffs extend beyond the top 20% of entries. Nonetheless, the payoff sizes around the 20^{th} percentile are very small and almost negligible. Indeed within our model, most of the expected profit and loss (P&L) comes from the top few percentiles and $\mu_{Y^d_{\mathbf{w}}} < 0$ is certainly true for these values of d. Finally, we note the well-known general tendency of models to over-estimate the performance of an optimally chosen quantity (in this case our portfolio). We therefore anticipate that our optimal portfolio will not quite achieve (in expectation) the top 20^{th} percentile and may well be out of the money for all payoff percentiles as assumed in Assumption 5.1.

Given Assumption 5.1, it follows that each of the arguments $-\mu_{Y^d_{\mathbf{w}}}/\sigma_{Y^d_{\mathbf{w}}}$ to the normal CDF terms in (15) is *positive*. Given the objective in (15) is to maximize, it is also clear that for a fixed value of $\mathbf{w}^\top \boldsymbol{\mu}_{\boldsymbol{\delta}}$ in (13), we would like the standard deviation $\sigma_{Y^d_{\mathbf{w}}}$ to be as large as possible. Unfortunately, the third term, $2\mathbf{w}^\top \boldsymbol{\sigma}_{\boldsymbol{\delta},G^{(r'_d)}}$, on the r.h.s. of (14) suggests that $\mathbf{w}$ impacts the variance by a quantity that depends on d. Fortunately, however, we found this dependence on d to be very small in our numerical experiments with real-world DFS top-heavy contests. Specifically, we found these covariance terms to be very close to each other for values of d corresponding to the top 20 percentiles and in particular for the top few percentiles. We now formalize this observation via the following assumption.

Assumption 5.2. *$Cov\left(\delta_p, G^{(r'_d)}\right) = Cov\left(\delta_p, G^{(r'_{d'})}\right)$ for all $d, d' = 1, \ldots, D$ and for all $p \in \{1, \ldots, P\}$.*

Further support for Assumption 5.2 is provided by the following simple proposition.

Proposition 5.1. *Suppose the $\mathbf{w}_o$'s are IID given $\mathbf{p}$ and D is finite. Then, in the limit as $O \to \infty$, we have*

$$Cov\left(\delta_p, G^{(r'_d)}\right) = Cov\left(\delta_p, G^{(r'_{d'})}\right) \text{ for all } d, d' = 1, \ldots, D \tag{16}$$

for any $p \in \{1, \ldots, P\}$.

Proof. First note that the number of feasible lineups is finite and so any $\mathbf{w}_o$ which is selected with strictly positive probability will be chosen infinitely often as $O \to \infty$. In particular, the top team will be chosen infinitely often and so it follows that the top D teams will be identical for any finite D and any realisation of $(\boldsymbol{\delta}, \mathbf{p})$. It therefore follows that conditioned on $(\boldsymbol{\delta}, \mathbf{p})$, $G^{(r'_d)} = G^{(r'_{d'})}$ w.p. 1 in the limit as $O \to \infty$. (16) will then follow from a simple interchange of limit and expectation, which can easily be justified assuming $\boldsymbol{\delta}$ is integrable. □

In many of the contests we participated in, we saw values of $O \approx 200,000$ which, while large, is actually quite small relative to the total number of feasible lineups. As such, we do not expect to see the top D teams being identical in practice or even to see much if any repetition among them. Nonetheless, we do expect to see sizeable overlaps in these top teams, especially for the very highest ranks, which are our ultimate target given the lop-sided reward structure of typical top-heavy contests. It was no surprise then that in our numerical experiments we observed a very weak dependence of $\mathrm{Cov}\left(\delta_p, G^{(r'_d)}\right)$ on d as stated earlier.

We therefore proceed to take Assumption 5.2 as given. It is then clear from (14) that the impact of $\mathbf{w}$ on $\sigma^2_{Y^d_{\mathbf{w}}}$ does not depend on d. Given the preceding arguments, it follows that for any fixed value of $\mathbf{w}^\top \boldsymbol{\mu}_{\boldsymbol{\delta}}$, we would like to make $\mathbf{w}^\top \boldsymbol{\Sigma}_{\boldsymbol{\delta}} \mathbf{w} - 2\mathbf{w}^\top \boldsymbol{\sigma}_{\boldsymbol{\delta}, G^{(r'_d)}}$ (the terms from (14) that depend on $\mathbf{w}$) as large as possible. We are therefore in the situation of part (i) of Proposition 4.1 and so we have a simple algorithm for approximately solving the top-heavy problem. This is given in Algorithm 4 below where we omit the dependence on d of those terms that are assumed (by Assumption 5.2) to not vary with d.

Algorithm 4 Optimization Algorithm for the Top-Heavy Problem with a Single Entry

Require: $\mathbb{W}$, Λ, $\boldsymbol{\mu}_{\boldsymbol{\delta}}$, $\boldsymbol{\Sigma}_{\boldsymbol{\delta}}$, $\boldsymbol{\sigma}_{\boldsymbol{\delta}, G^{(r')}}$ and Monte Carlo samples of $(\boldsymbol{\delta}, G^{(r'_d)})$ for all $d = 1, \ldots, D$

1: **for all** $\lambda \in \Lambda$ **do**

2: $\quad \mathbf{w}_\lambda = \underset{\mathbf{w} \in \mathbb{W}}{\mathrm{argmax}} \left\{ \mathbf{w}^\top \boldsymbol{\mu}_{\boldsymbol{\delta}} + \lambda \left(\mathbf{w}^\top \boldsymbol{\Sigma}_{\boldsymbol{\delta}} \mathbf{w} - 2\mathbf{w}^\top \boldsymbol{\sigma}_{\boldsymbol{\delta}, G^{(r')}} \right) \right\}$

3: **end for**

4: $\lambda^* = \underset{\lambda \in \Lambda}{\mathrm{argmax}} \ \sum_{d=1}^{D} (R_d - R_{d+1}) \, \mathbb{P}\left\{ \mathbf{w}_\lambda^\top \boldsymbol{\delta} > G^{(r'_d)}(\mathbf{W}_{\mathrm{op}}, \boldsymbol{\delta}) \right\}$

5: **return** $\mathbf{w}_{\lambda^*}$

As with Algorithm 2, the $\mathbf{w}_\lambda$'s are computed by solving BQPs and the optimal λ^* from line 4 can then be determined via the Monte Carlo samples that were used as inputs.

5.1. An Aside on Parimutuel Betting Markets

Our next goal is to understand how to extend Algorithm 4 to the case of multiple entries, i.e., the case where $N > 1$. In order to tackle this problem, we will first consider the setting of parimutuel betting markets, which can be viewed as a special case of our top-heavy DFS contests. Parimutuel betting is widespread in the horse-racing industry and has often been studied in the economics literature [4, 33, 39, 43] with the goal of studying the efficient markets hypothesis and the investment behavior of individuals in a simple and well-defined real-world setting. Our goal here is to use the simplified setting of parimutuel betting to gain some insight into the structure of the optimal strategy for constructing multiple entries in a top-heavy DFS contest. The results we establish here are straightforward to obtain but are new to the best of our knowledge.

Consider then a horse race where there are H horses running and where there will be a single winner so there is no possibility of a tie. Each wager is for \$1 and we have \$$N$ to wager. We let n_h denote the number of wagers, i.e., dollars, that we place on horse h. It therefore follows that $\sum_{h=1}^{H} n_h = N$ and we use $(n_1, n_2, n_3, \ldots, n_H)$ to denote the allocation of our N wagers. We let $q_h > 0$ denote the probability that horse h wins the race. We assume there are a total of O wagers made by our opponents so that $\sum_{h=1}^{H} O_h = O$, where O_h is the number of opponent wagers on horse h. We assume[16] the O_h's are deterministic and known. The total dollar value of the wagers is then $O + N$ and w.l.o.g. we assume the cut or "vig" taken by the race-track is zero. To make clear the connection between parimutuel and DFS contests, we can equate each horse to a feasible team in DFS.

5.1.1. Parimutuel Winner-Takes-All Contests

In a parimutuel winner-takes-all (WTA) contest, the players that pick the winning horse share the total value wagered. In particular, if horse h wins, then our winnings are $(O + N) n_h / (O_h + n_h)$ so that our share is proportional to the number of wagers we placed on h. If the winning horse is picked by no one, then we assume that none of the contestants receives a payoff. This is in contrast to the DFS setting where

[16] We could model the O_h's as being stochastic but this makes the analysis unnecessarily complicated.

the reward $O+N$ would be allocated to the highest ranked team that was submitted to the contest. This difference is quite significant and we will return to it later. For now, we note that it results in what we refer to as *reward independence* whereby the expected reward we earn from our wagers on horse h does not depend on $n_{h'}$ for any $h' \neq h$.

Definition 5.1. *Suppose our current portfolio of wagers has $n_h = k$ with at least as yet one "unassigned" wager. Let μ_h^{k+1} denote the* expected gain *we obtain from assigning this wager to horse h.*

Reward independence allows us to easily compute μ_h^{k+1}. In particular, we obtain

$$\mu_h^{k+1} = \underbrace{\frac{(k+1)q_h(O+N)}{O_h+k+1}}_{\text{"after"}} - \underbrace{\frac{kq_h(O+N)}{O_h+k}}_{\text{"before"}} = \frac{q_h(O+N)}{O_h+k} \times \frac{O_h}{O_h+k+1} \tag{17}$$

It follows immediately from (17) that μ_h^k is strictly decreasing in k for $k = 1, \ldots, N$ and for all horses h. We refer to this as the *saturation effect*. W.l.o.g., we assume hereafter that the horses have been sorted in decreasing order of the μ_h^1's so that $\mu_1^1 \geq \mu_2^1 \geq \ldots \geq \mu_H^1$. This ordering and the saturation effect then imply the following partial ordering of the μ_h^k's:

$$\begin{array}{ccccccc}
\mu_1^1 & \geq & \mu_2^1 & \geq & \ldots & \geq & \mu_H^1 \\
\text{IV} & & \text{IV} & & \ldots & & \text{IV} \\
\mu_1^2 & & \mu_2^2 & & \ldots & & \mu_H^2 \\
\text{IV} & & \text{IV} & & \ldots & & \text{IV} \\
\vdots & & \vdots & & \vdots & & \vdots \\
\text{IV} & & \text{IV} & & \ldots & & \text{IV} \\
\mu_1^N & & \mu_2^N & & \ldots & & \mu_H^N.
\end{array}$$

This partial ordering suggests an approach for allocating the N wagers. We start by allocating the first wager to the first horse. (This is optimal in the $N = 1$ case due to the presumed ordering of the horses.) We then consider allocating the second wager to either the first horse (and thereby replicating the first wager) or to the second horse. Because of the partial ordering, this must be optimal for the $N = 2$ case. Suppose the optimal choice was to allocate the second wager to the second horse. Then the third wager should be allocated to either one of the first two horses (thereby replicating an earlier wager) or to the third horse. In contrast, if the optimal choice for the second wager was to replicate the first wager and place it on the first horse, then only the first and second horses need be considered for the third wager. These observations all follow from the partial ordering of the expected gains and they lead immediately to Algorithm 5, which handles the case of general N. It is a greedy algorithm where each successive wager is placed on the horse with the highest expected gain given all previous wagers. The following proposition asserts the optimality of Algorithm 5. A proof can be found in Appendix B.1.

Proposition 5.2. *Algorithm 5 returns an optimal wager portfolio for the parimutuel WTA contest.*

A natural question that arises when solving the $N > 1$ problem is whether to replicate or diversify our wagers. Some insight into this issue can be provided in the $N = 2$ case. From Proposition 5.2, we know the optimal portfolio of wagers $(n_1, n_2, n_3, \ldots, n_H)$ is of the form $(2, 0, 0, \ldots, 0)$ or $(1, 1, 0, \ldots, 0)$. A simple calculation that compares the expected values of these portfolios then implies

$$(n_1, n_2, n_3, \ldots, n_H) = \begin{cases} (2, 0, 0, \ldots, 0) & \text{if } \frac{\mu_1^1}{\mu_2^1} > \frac{O_1+2}{O_1} \\ (1, 1, 0, \ldots, 0) & \text{otherwise.} \end{cases} \tag{18}$$

Algorithm 5 Greedy Algorithm for Constructing a Portfolio of N Horses for Parimutuel WTA Contests

Require: $\{\mu_h^k : 1 \le h \le H, 1 \le k \le N\}$, N

$n_h = 0$ for all $h = 1, \ldots, H$ % initialize
$n_1 = 1$ % assign first wager to horse 1
for $j = 2 : N$ **do**
  $\mathbb{A} = \{(h, n_h + 1) : n_h > 0\} \cup \{(h, 1) : n_h = 0,\ n_{h-1} > 0\}$ % next wager will be a replication or first horse
  % in ordering that has not yet been wagered upon
  $h^* = \underset{\{h : (h,k) \in \mathbb{A}\}}{\text{argmax}}\ \mu_h^k$ % horse in $\mathbb{A}$ with highest expected gain
  $n_{h^*} = n_{h^*} + 1$ % fill entry j with horse h^*
end for
return $(n_1, n_2, n_3, \ldots, n_H)$

We see from the condition in (18) that diversification becomes relatively more favorable when O_1 is small so that horse 1 is not very popular among opponents. Our expected gain from replicating our wager on this horse declines as O_1 decreases. For example, if $O_1 = 0$, then we would have made $O + N$ if this horse won and the expected gain from replicating our wager on this horse equals 0. Diversification (by applying our second wager on horse 2) is clearly optimal in this case and this is reflected by the fact that $\mu_1^1 / \mu_2^1 > (O_1 + 2)/O_1 = \infty$ can never be satisfied.

In contrast, replication becomes relatively more favorable when O_1 is large and horse 1 is therefore very popular among opponents. This horse has a good chance of winning the race (since it has the highest μ_1^1) and by replicating our wager on it we can almost double our share of the total reward should the horse win. This follows because replicating our wager on horse 1 increases our total expected reward from $(O + N)/(O_1 + 1)$ to $(O + N)2/(O_1 + 2)$, which is an approximate doubling when O_1 is large. This must be close to optimal given that it was optimal to place our initial wager on horse 1 in the first place. Indeed this is reflected in the condition $\mu_1^1 / \mu_2^1 > (O_1 + 2)/O_1$ from (18), which will typically be satisfied when O_1 is large since it is always the case that $\mu_1^1 / \mu_2^1 \ge 1$.

It is perhaps worth mentioning at this point that unlike the parimutuel setting, there will typically be far more feasible teams than contestants in the DFS setting. This will be the case even for contests with several hundred thousand contestants. As such, in DFS contests we are invariably in the setting of small O_h's, which results in diversification being favored.

5.1.2. Extension to Parimutuel Top-Heavy Contests

The previous analysis for parimutuel WTA contests can be easily extended to more general parimutuel top-heavy contests. Suppose the horse that places d^{th} in the race carries a reward R_d for $d = 1, \ldots, D \le H$. This reward is then allocated to all contestants who placed wagers on this horse. Again, we assume that if no wagers were placed on it, then the reward is not allocated. We let $q_h^d := \mathbb{P}\{\text{horse } h \text{ places } d^{th} \text{ in race}\}$ and then update our expression for the expected gain μ_h^{k+1}. A simple calculation leads to

$$\mu_h^{k+1} = \frac{q_h^1 R_1 + q_h^2 R_2 + \ldots + q_h^D R_D}{O_h + k} \times \frac{O_h}{O_h + k + 1}.$$

To maintain consistency with our earlier WTA setting, we can assume $\sum_{d=1}^{D} R_d = O + N$. Everything now goes through as before. In particular, Algorithm 5 still applies as does the proof of Proposition 5.2, which guarantees its optimality. (We note that this also applies to double-up style parimutuel contests by simply assuming that for $d = 1, \ldots, D$ we have $R_d = R$, a constant.)

5.1.3. Difference Between Parimutuel and DFS Contests

The key difference between DFS contests and our parimutuel setup is that in the DFS contests, the reward R_d is allocated to the submitted entry that has the d^{th} highest ranking among the *submitted* entries. As such, the prize is *always* awarded for each $d = 1, \dots, D$. To make the distinction concrete, suppose[17] there are 10 billion feasible teams ("horses") in a given DFS contest with 500,000 entries and a WTA payoff structure. In this case, at most 0.005% of the feasible teams will have had wagers placed on them and so it's very unlikely[18] that the ex-post best team will have received a wager. In our parimutuel setup, the $O + N$ would simply not be allocated in that case. It is allocated in the DFS contest, however, and is allocated to the best performing team among the teams that were wagered upon. This might appear like a minor distinction but it is significant. In particular, reward independence no longer holds. To see this, consider a team that we have wagered upon and suppose it is ex-post the third ranked team out of the 10 billion possible entries. Again assuming a WTA structure, then that wager will win the reward of $O+N$ only if there were no wagers placed by anyone else and ourselves in particular, on the first two horses. Our expected gain from the wager therefore depends on the other wagers we have placed. Because reward independence no longer holds, it means Proposition 5.2 no longer holds even with updated μ_h^ks.

Nonetheless, it is easy to see this loss of reward independence points towards a strategy of even greater diversification than that provided[19] by Algorithm 5. To see this, consider the following stylized setting. Suppose the space of feasible teams for the DFS contest can be partitioned into M "clusters" where M is "large". The clustering is such that the fantasy points scores of teams in the same cluster are strongly positively correlated (owing to significant player overlap in these teams) while teams in different clusters are only weakly correlated. Suppose cluster 1 is ex-ante the "best" cluster in that the teams in cluster 1 have the highest expected reward. Clearly, in the $N = 1$ case, it would be optimal to wager on the best team in cluster 1. In the $N = 2$ case, however, it may not be optimal to place the second wager on a team from cluster 1 even if this team has the second highest expected reward when considered by itself. This is because in some sense, the first wager "covers" cluster 1. To see this, suppose none of our opponents wagered on a team from cluster 1 and that ex-post, the best team was another team from cluster 1. While we did not wager on the ex-post best team, neither did anyone else and as we were the only contestant to wager on a team from cluster 1, there's a good chance our team will win the reward of $O + N$ (assuming again a WTA structure) due to the strong positive correlation among teams within a cluster. It therefore may make more sense to select a team from another cluster for our second wager.

5.2. The Top-Heavy DFS Problem with Multiple Entries

We return now to the more general top-heavy DFS problem where we must submit N entries to the contest. We repeat again the problem formulation from Section 2.2:

$$\max_{\mathbf{W} \in \mathbb{W}^N} \sum_{i=1}^{N} \sum_{d=1}^{D} (R_d - R_{d+1}) \, \mathbb{P}\left\{ \mathbf{w}_i^\top \boldsymbol{\delta} > G_{-i}^{(r'_d)}(\mathbf{W}_{-i}, \mathbf{W}_{\mathrm{op}}, \boldsymbol{\delta}) \right\}, \tag{19}$$

where $r'_d \coloneqq O + N - r_d$, $G_{-i}^{(r)}$ is the r^{th} order statistic of $\{G_o\}_{o=1}^{O} \cup \{F_j\}_{j=1}^{N} \setminus F_i$ and $\mathbf{W}_{-i} \coloneqq \mathbf{W} \setminus \mathbf{w}_i$. Our initial algorithm for solving (19) is a greedy-style algorithm that we formally state in Algorithm 6 below. To be concise, we use $\mathcal{R}(\mathbf{W})$ to denote the expected reward corresponding to our portfolio $\mathbf{W}$ of possibly

[17]To give these numbers some perspective, the typical top-heavy DFS contest that we entered had 24 NFL teams playing in a series of 12 games. We calculated the number of feasible entries for these contests to be approx. 2×10^{13} of which approx. 7×10^{10} utilized 99% of the budget. (In our experience, the vast majority of DFS contestants like to use $> 98\%$ of their budget when constructing their entries.)

[18]In Section 6, we describe our results from playing various DFS contests during the 2017 NFL regular season. In the top-heavy contests of each of the 17 weeks of the season, we found that the ex-post best performing team was *not* wagered upon!

[19]Algorithm 5 can yield portfolios anywhere on the spectrum from complete replication to complete diversification but, as mentioned earlier, the O_h's tend to be very small and often 0 in DFS contests and this strongly encourages diversification.

multiple entries. Trivially, the analytical expression[20] of $\mathscr{R}(\mathbf{W})$ is identical to the objective function in (19) if we replace N by $|\mathbf{W}|$, the number of entries in $\mathbf{W}$.

Algorithm 6 Top-Heavy Optimization for N Entries (with Backtracking)

Require: $\mathbb{W}$, N, γ, Λ, $\boldsymbol{\mu_\delta}$, $\boldsymbol{\Sigma_\delta}$, $\boldsymbol{\sigma}_{\boldsymbol{\delta},G^{(r')}}$ and Monte Carlo samples of $(\boldsymbol{\delta}, G^{(r'_d)})$ for all $d = 1, \dots, D$

1: $\mathbf{W}^* = \varnothing$

2: **for all** $i = 1, \dots, N$ **do**

3:     **for all** $\lambda \in \Lambda$ **do**

4:         $\mathbf{w}_\lambda = \underset{\mathbf{w}\in\mathbb{W}}{\operatorname{argmax}} \left\{ \mathbf{w}^\top \boldsymbol{\mu_\delta} + \lambda \left(\mathbf{w}^\top \boldsymbol{\Sigma_\delta} \mathbf{w} - 2\mathbf{w}^\top \boldsymbol{\sigma}_{\boldsymbol{\delta},G^{(r')}} \right) \right\}$

5:     **end for**

6:     $\lambda^* = \underset{\lambda\in\Lambda}{\operatorname{argmax}}\ \mathscr{R}(\mathbf{W}^* \cup \mathbf{w}_\lambda)$ % pick λ corresponding to biggest value-add

7:     $\mathbf{w}_i^* = \underset{\mathbf{w}\in\{\mathbf{w}_1^*,\dots,\mathbf{w}_{i-1}^*,\mathbf{w}_{\lambda^*}\}}{\max} \mathscr{R}(\mathbf{W}^* \cup \mathbf{w})$ % best addition from $\{\mathbf{w}_1^*, \dots, \mathbf{w}_{i-1}^*, \mathbf{w}_{\lambda^*}\}$ to $\mathbf{W}^*$

8:     $\mathbf{W}^* = \mathbf{W}^* \cup \{\mathbf{w}_i^*\}$

9:     $\mathbb{W} = \mathbb{W} \cap \{\mathbf{w} : \mathbf{w}^\top \mathbf{w}_i^* \le \gamma\}$ % add diversification constraint for next *candidate* entry

10: **end for**

11: **return** $\mathbf{W}^*$

Several comments are now in order. We first note that Algorithm 6, which reduces to Algorithm 4 when $N = 1$, is modeled on Algorithm 5 from the parimutuel setting. To see this, first note that the constraint $\mathbf{w}^\top \mathbf{w}_i^* \le \gamma$ from line 9 restricts the next candidate entry $\mathbf{w}$ to have less than or equal to γ players in common with the previously selected entries. Recalling that C is the number of players in a DFS entry, it therefore follows that if we set $\gamma \ge C$, then the constraint $\mathbf{w}^\top \mathbf{w}_i^* \le \gamma$ is never binding. But if we set $\gamma < C$ (which is what we ultimately recommend), then the candidate entry $\mathbf{w}_{\lambda^*}$ from iteration i will always be a new entry, i.e., an entry not represented in the current portfolio $\{\mathbf{w}_1^*, \dots, \mathbf{w}_{i-1}^*\}$. As such, the set $\{\mathbf{w}_1^*, \dots, \mathbf{w}_{i-1}^*, \mathbf{w}_{\lambda^*}\}$ in line 7 is analogous to the set $\mathbb{A}$ in Algorithm 5 and so our definition of $\mathbf{w}_i^*$ from line 7 corresponds to the definition of h^* in line 6 of Algorithm 5.

There are also some important differences, however. In Algorithm 5, we identify the horse who will *add* the most to the portfolio. In the DFS setting, however, that is difficult. In particular, in iteration i of Algorithm 6, given the current portfolio $\mathbf{W}$ consisting of $i - 1$ entries, it is a non-trivial task to identify the entry $\widehat{\mathbf{w}}_i$ that will add the most to $\mathbf{W}$ in terms of expected reward. The reason is that it is not necessarily true that $\widehat{\mathbf{w}}_i$ will lie on the "efficient frontier" constructed in lines 3 to 5 of Algorithm 6. Hence, even though our optimization in line 7 of Algorithm 6 identifies the highest value-add entry in the set $\{\mathbf{w}_1^*, \dots, \mathbf{w}_{i-1}^*, \mathbf{w}_{\lambda^*}\}$, it is possible and indeed very likely that $\widehat{\mathbf{w}}_i$ does not belong to $\{\mathbf{w}_1^*, \dots, \mathbf{w}_{i-1}^*, \mathbf{w}_{\lambda^*}\}$ to begin with. In fact, when $\gamma = C$, we expect that the candidate entry $\mathbf{w}_{\lambda^*}$ will often coincide with a previously chosen entry, i.e., an entry from $\{\mathbf{w}_1^*, \dots, \mathbf{w}_{i-1}^*\}$. Indeed this is what we observed in our numerical experiments where we typically found just ≈ 10 unique entries when $N = 50$. But this is simply a reflection of our failure to find $\widehat{\mathbf{w}}_i$.

In order to force the algorithm to find an entry with a higher value-add, we allowed for more diversification by setting $\gamma < C$ in Algorithm 6. (It's important to note that we used diversification only to find a better choice of $\widehat{\mathbf{w}}_i$. Diversification, however, was not forced upon the final portfolio since line 7 allows replication of previously chosen entries.) We observed considerably more diversification and a *much* higher expected reward for the final portfolio. In particular, the expected reward almost doubled and we typically observed

[20] Note that the $\mathscr{R}$ appearing in Algorithm 6 is really an estimated version of the reward since we can only evaluate it using the Monte-Carlo samples. On a related note we mention that samples of $(\boldsymbol{\delta}, G^{(r')})$ for some additional ranks r' besides the r'_d's will be required in order to properly estimate $\mathscr{R}$. For example, suppose $D = N = 2$ with $r_1 = 1$ and $r_2 = 20$. Then we will also need samples corresponding to the 19^{th} rank since if our first entry comes 5^{th} say then our second entry will only be among the top 20 if it's among the top 19 of our opponents' entries.

25+, 35+, and 40+ unique entries when we set $\gamma = C - 1$, $C - 2$, and $C - 3$, respectively.

5.2.1. Forcing Diversification

Recall from Section 5.1.3 that we do not have reward independence in DFS contests. This is why even an idealized greedy algorithm (where we could find $\widehat{\mathbf{w}}_i$) would not[21] be optimal in general. This is in contrast to our parimutuel setup and led to us arguing that even more diversification might be called for in the DFS setting. An easy way to test this is to set $\gamma < C$ and to simply add entry $\mathbf{w}_{\lambda^*}$ to the portfolio $\mathbf{W}$ without considering replicating one of the previously chosen entries. This then results in full diversification and the selection of N distinct entries. We present this approach as Algorithm 7, where the only difference from Algorithm 6 is the removal of the "backtracking" step, that is, line 7 in Algorithm 6. In all of our numerical experiments, we found that $\gamma = C - 3 = 6$ was an optimal choice in both Algorithms 6 and 7 in that it led to final portfolios with the highest expected value in each case. We also found that for any fixed value of γ, the portfolio resulting from Algorithm 7 was approximately 5% to 20% better (in expected value terms) than the portfolio resulting from Algorithm 6. In light of our earlier comments, this was not very surprising and so Algorithm 7 is our preferred algorithm and the one we used in our numerical experiments of later sections.

Algorithm 7 Top-Heavy Optimization for N Entries (without Backtracking)

Require: $\mathbb{W}$, N, γ, Λ, $\boldsymbol{\mu}_{\boldsymbol{\delta}}$, $\boldsymbol{\Sigma}_{\boldsymbol{\delta}}$, $\boldsymbol{\sigma}_{\boldsymbol{\delta},G^{(r')}}$ and Monte Carlo samples of $(\boldsymbol{\delta}, G^{(r'_d)})$ for all $d = 1, \ldots, D$

1: $\mathbf{W}^* = \varnothing$
2: **for all** $i = 1, \ldots, N$ **do**
3:   **for all** $\lambda \in \Lambda$ **do**
4:     $\mathbf{w}_\lambda = \underset{\mathbf{w} \in \mathbb{W}}{\operatorname{argmax}} \left\{ \mathbf{w}^\top \boldsymbol{\mu}_{\boldsymbol{\delta}} + \lambda \left(\mathbf{w}^\top \boldsymbol{\Sigma}_{\boldsymbol{\delta}} \mathbf{w} - 2\mathbf{w}^\top \boldsymbol{\sigma}_{\boldsymbol{\delta},G^{(r')}} \right) \right\}$
5:   **end for**
6:   $\lambda^* = \underset{\lambda \in \Lambda}{\operatorname{argmax}}\, \mathscr{R}(\mathbf{W}^* \cup \mathbf{w}_\lambda)$ % pick λ corresponding to biggest value-add
7:   $\mathbf{W}^* = \mathbf{W}^* \cup \{\mathbf{w}_{\lambda^*}\}$
8:   $\mathbb{W} = \mathbb{W} \cap \{\mathbf{w} : \mathbf{w}^\top \mathbf{w}_{\lambda^*} \leq \gamma\}$ % add diversification constraint for next entry
9: **end for**
10: **return** $\mathbf{W}^*$

While we have taken N as given up to this point, it is perhaps worth mentioning that one can always use Algorithm 7 to determine an optimal value of N. Specifically, we can continue to increase N until the expected P&L contribution from the next entry goes negative or below some pre-specified threshold. We also mention that it is straightforward to add additional linear constraints to $\mathbb{W}$ if further or different forms of diversification are desired. Finally, we note it's easy to estimate the expected P&L of any portfolio of entries via Monte Carlo simulation.

5.2.2. An Aside on the Submodularity of the Top-Heavy Objective

In Appendix B.2 we show that the objective function for the top-heavy problem is a monotone sub-modular function. This allows us to provide some additional theoretical support for Algorithms 6 and 7 via a classical result [30] on the maximization of monotone submodular functions. This result states that the greedy algorithm will return a portfolio of entries whose expected value is greater than or equal to $1 - 1/e \approx 63.2\%$ of the expected value of the optimal portfolio. The greedy algorithm, however, assumes that the entry which provides the maximum *gain* to the current portfolio (denoted by $\widehat{\mathbf{w}}_i$ in our earlier discussion) is added at each iteration. Unfortunately (and as discussed above), we can only hope to find a good candidate entry for

[21] But see Section 5.2.2 below. If we could find $\widehat{\mathbf{w}}_i$, then a performance guarantee on the resulting greedy algorithm could be given.

adding to the portfolio in Algorithms 6 and 7. There is of course also the added assumption of using the mean-variance criterion to solve for $\mathbf{w}_{\lambda^*}$ in each iteration of Algorithms 6 and 7 and so this also dilutes the support from the classical result in [30].

As an aside, we note the greedy algorithm proposed by Hunter et al. [22] was also motivated by submodularity considerations but their focus was on maximizing the probability of winning a WTA contest whereas our focus is on maximizing the expected reward in general top-heavy contests.

6. Numerical Experiments

We participated in real-world DFS contests on FanDuel during the 2017 NFL regular season, which consisted of 17 weeks. Each week, we participated in three contests: top-heavy, quintuple-up and double-up. The cost per entry was \$1 in top-heavy and \$2 in both quintuple-up and double-up contests. The number of opponents O was approximately $200,000$, $10,000$ and $30,000$ for the three contests, respectively, with these numbers varying by around 10% from week-to-week. The payoff in the top-heavy contest[22] for rank 1 was approx. \$5,000, for rank 2 it was approx. \$2,500 and then it declined quickly to approx. \$100 for rank 30. The lowest winning rank was around 50,000, with a payoff of \$2.

We used two different models for each contest: our strategic model and a benchmark model. To be clear, for all top-heavy contests, our strategic model was Algorithm 7 with $\gamma = 6$. Our strategic model for the double-up and quintuple-up contests was also Algorithm 7 with[23] $\gamma = 6$ but lines 3 to 5 replaced by lines 1 to 9 of Algorithm 2 and with the understanding that the expected reward function $\mathscr{R}(\cdot)$ corresponds to the double-up / quintuple-up contest. The second model is a *benchmark* model that does not model opponents and hence is not strategic. For each model, we submitted $N = 50$, 25 and 10 entries to top-heavy, quintuple-up and double-up contests, respectively each week. Other details regarding our model inputs such as $\boldsymbol{\mu_\delta}$, $\boldsymbol{\Sigma_\delta}$, stacking probability q, diversification parameter γ, and the budget lower bound B_{lb} are discussed in Appendix C along with the specifications of the hardware and software we use to solve the BQPs.

6.1. Benchmark Models

Our two benchmark models do not model opponents and in fact, they (implicitly) assume the benchmarks $G^{(r')}$ or $G^{(r'_d)}$ are deterministic.

6.1.1. Benchmark Model 1 (For Double-Up Contests)

To optimize in the $N = 1$ case, our first benchmark model simply maximizes the expected points total subject to the feasibility constraints on the portfolio. The resulting optimization model is a binary program (BP):

$$\max_{\mathbf{w}\in\mathbb{W}} \mathbf{w}^\top \boldsymbol{\mu_\delta}. \tag{20}$$

For $N > 1$ (which is the case in our numerical experiments), we employ the greedy diversification strategy discussed in Section 5.2 but suitably adapted for the case where we do not model opponents. In particular, when optimizing over the i^{th} entry, we add the constraints that ensure the i^{th} entry can not have more than γ athletes in common with any of the previous $i - 1$ entries. We use this benchmark model for the double-up contest because, according to our calibrated model, we are comfortably in the case (ii) scenario of Proposition 4.1 where, other things being equal, we prefer less variance to more variance.

[22] We note that there are other top-heavy contests with even more competitors and payoff structures that are even more "top-heavy". For example, a regular NFL contest on FanDuel often has approximately $400,000$ entries with a top payoff of \$250,000 to \$1,000,000. Payoffs then decline quickly to approx. \$500 for the 50^{th} rank. Top-heavy contests are therefore extremely popular and hence are our principal focus in this paper.

[23] The reason for doing so in the double-up / quintuple-up contests was simply to reduce the variance of our P&L albeit at the cost of a (hopefully slightly) smaller expected P&L. This is discussed further in Section 6.2.

6.1.2. Benchmark Model 2 (For Top-Heavy and Quintuple-Up Contests)

The second benchmark model is similar to the first and indeed the objective functions are identical. The only difference is that we add a stacking constraint to force the model to pick the QB and main WR from the same team. We denote this constraint as "QB-WR". Mathematically, the resulting BP for $N = 1$ is:

$$\max_{\mathbf{w} \in \mathbb{W}, \text{QB-WR}} \mathbf{w}^\top \boldsymbol{\mu}_{\boldsymbol{\delta}}. \tag{21}$$

Again for $N > 1$, we employ a suitably adapted version of the greedy diversification strategy from Section 5.2, i.e., the i^{th} entry can not have more than γ athletes in common with any of the previous $i - 1$ entries. As discussed in Section 3.2, the purpose of the stacking constraint is to increase the portfolio's variance. This is because we are invariably "out-of-the-money" in these contests as we noted in Section 5 and so variance is preferred all other things, that is, expected number of points, being equal. We note this model is very similar to the model proposed by Hunter et al. [22] for hockey contests. They presented several variations of their model typically along the lines of including more stacking (or anti-stacking[24]) constraints, e.g. choosing athletes from exactly 3 teams to increase portfolio variance. We note that we could easily construct and back-test other similar benchmark strategies as well but for the purposes of our experiments, the two benchmarks above seemed reasonable points of comparison.

6.2. Main Results

We now discuss the P&L-related results for the strategic and benchmark models across the three contest structures for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season. Table 1 and Figure 2 display the cumulative realized P&L for both models across the three contest structures during the season. The strategic portfolio has outperformed the benchmark portfolio since inception in the top-heavy series of contests. The strategic portfolio has earned a cumulative profit of \$280.74, which is over 3 times the realized P&L of the benchmark portfolio. Moreover, the maximum cumulative loss, that is, the max shortfall, for the strategic portfolio is just \$18.5. In addition, the small initial investment of \$50 plus two additional investments of \$18.5 and \$7.26 (total of \$75.76) have been sufficient to fund the strategic portfolio throughout the season. This suggests a profit of \$280.74 on an investment of \$75.76, that is, a return of over 350% in just 17 weeks. In contrast, the benchmark portfolio needed much more capital than the initial investment of \$50. If we account for this additional required capital, then the benchmark portfolio has earned a return of less than 50% in 17 weeks. Note that given the so-called house-edge of approximately 15%, both models have performed considerably better than the average portfolio which would have lost $\approx 17 \times 15\% \times 50 = \$127.5$ across the 17 weeks.

With regards to the quintuple-up series, the strategic model was better until the end of week 6 but since then the benchmark portfolio has outperformed it. We note, however, that the difference in the cumulative P&L between the two models at the end of the season $(20 - (-40) = 60)$ could easily be wiped out in just one week's contest as we can see when we look at the relative performances of the two strategies in week 7, for example.

We are confident that the realized P&L to-date for each contest series is actually conservative and that superior performance (in expectation) could easily be attained. There are at least three reasons for this. First, we used off-the-shelf estimates of the input parameters $\boldsymbol{\mu}_{\boldsymbol{\delta}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\delta}}$, which are clearly vital to the optimization model. Moreover, we obtained the $\boldsymbol{\mu}_{\boldsymbol{\delta}}$ estimate a day before the actual NFL games started and mostly ignored the developments in the last few hours preceding the games, which can be very important in football. For example, in week 7, the main RB of the Jacksonville Jaguars (Leonard Fournette) was

[24] An example of an anti-stacking constraint in hockey is that the goalie of team A cannot be selected if the attacker of team B was selected and teams A and B are playing each other in the series of games underlying the DFS contest in question. Such an anti-stacking constraint is also designed to increase variance by avoiding athletes whose fantasy points would naturally be negatively correlated.

questionable to play. Accordingly, their second main RB (Chris Ivory) was expected to play more time on the field. However, our $\boldsymbol{\mu_\delta}$ estimate did not reflect this new information. Our estimate projected 17.27 and 6.78 fantasy points for Fournette and Ivory, respectively. Moreover, since FanDuel sets the price of the athletes a few days before the games take place, Fournette was priced at 9000 and Ivory at 5900. There was a clear benefit of leveraging this information as Fournette was over-priced and Ivory was under-priced. In fact, our opponents exploited this opportunity as around 60% of them (in double-up) picked Ivory. A proactive user would have updated his $\boldsymbol{\mu_\delta}$ estimate following such news. In fact, the so-called sharks do react to such last-minute information [31], meaning that we were at a disadvantage by not doing so.

For another example, consider Devin Funchess, a wide-receiver (WR) for the Carolina Panthers. During the course of the season, Funchess was usually the main WR for Carolina but in week 16 he was expected to be only the second or third WR and in fact Damiere Byrd was expected to be the main WR. This was late developing news, however, and our $\boldsymbol{\mu_\delta}$ estimate did not reflect this. Moreover, Byrd was priced at 4900 while Funchess was priced at 7000 and so Byrd was clearly under-priced relative to Funchess. In the week 16 game itself, Byrd scored 9.6 points while Funchess scored only 2.6 points. Because of our failure to respond to this late developing news and update our parameters, it transpired that 52 of our entries picked Funchess. We observed (after the fact) many similar situations during the course of the season and there is no doubt that we could have constructed superior portfolios had we been more pro-active in monitoring these developments and updating parameters accordingly.

The second reason is simply a variance issue in that a large number of DFS contests (and certainly much greater than 17) will be required to fully establish the outperformance of the strategic model in general. In fact, we believe the variance of the cumulative P&L is particularly high for NFL DFS contests. There are several reasons for this. Certainly, the individual performance of an NFL player in a given week will have quite a high variance due to the large roster size[25] as well as the relatively high probability of injury. This is in contrast to other DFS sports where there is considerably more certainty over the playing time of each athlete. To give but one example, in week 5 we witnessed a series of injuries that impacted many of our submitted portfolios (both strategic and benchmark). Devante Parker (Miami Dolphins) was injured in the first quarter but was picked by 56 of our entries. Charles Clay (Buffalo Bills) and Sterling Shepard (NY Giants) were injured before halftime, affecting 70 and 4 entries, respectively. Bilal Powell (NY Jets) and Travis Kelce (Kansas City Chiefs) left the field close to the halftime, impacting 44 and 25 entries, respectively. Furthermore, the NFL season consists of just 16 games per team whereas teams in sports such as basketball, ice hockey and baseball play 82, 82 and 162 games, respectively, per season. As a result, the cumulative P&L from playing DFS contests over the course of an NFL season will have a very high variance relative to these other sports. This high variance of NFL-based fantasy sports has been noted by other researchers including for example Clair and Letscher [9]. We also suspect that Hunter et al. [22] focused on ice hockey and baseball (and avoided NFL) for precisely this reason.

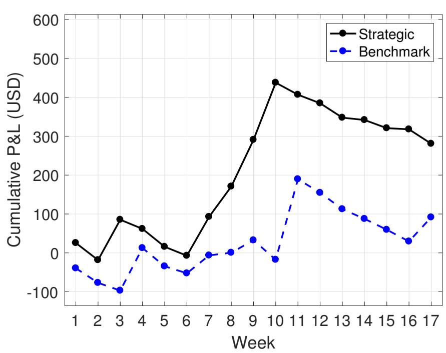
The third reason applies specifically to the quintuple-up contests. In our strategic model for quintuple-up, there is a possibility of incorrectly minimizing portfolio variance when we should in fact be maximizing it (along with expected number of points of course). Proposition 4.1 leads us to try and increase variance if $\mu_{\mathbf{w}} < 0$ for all $\mathbf{w} \in \mathbb{W}$ and to try and decrease variance otherwise. But $\mu_{\mathbf{w}}$ must be estimated via Monte Carlo and is of course also model-dependent. As such, if we estimate a maximal value of $\mu_{\mathbf{w}} \approx 0$, it is quite possible we will err and increase variance when we should decrease it and vice versa. We suspect this may have occurred occasionally with the quintuple-up contests where we often obtained an estimate of $\mu_{\mathbf{w}}$ that was close to zero. This of course is also related to the median versus mean issue we mentioned immediately after Algorithm 2. We note that one potential approach to solving this problem would have been to use Algorithm 3 instead of Algorithm 2. We note that the benchmark portfolio is always long expected points and variance of points.

Figure 3 displays the in-model P&L distribution for the diversification strategy from Section 5.2 for both

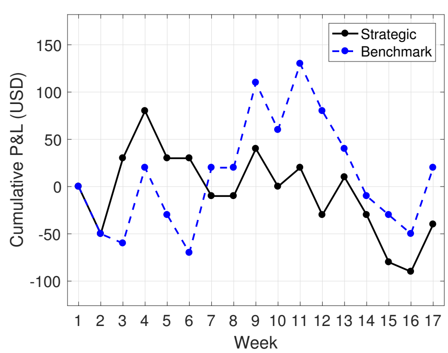
[25]There are more than 45 athletes on a roster but only 11 on the field at any one time.

Table 1: **Cumulative realized dollar P&L** for the strategic and benchmark models across the three contest structures for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season. We invested \$50 per week per model in top-heavy series with each entry costing \$1. In quintuple-up the numbers were \$50 per week per model with each entry costing \$2 and in double-up we invested \$20 per week per model with each entry costing \$2. (We were unable to participate in the quintuple-up contest in week 1 due to logistical reasons.)

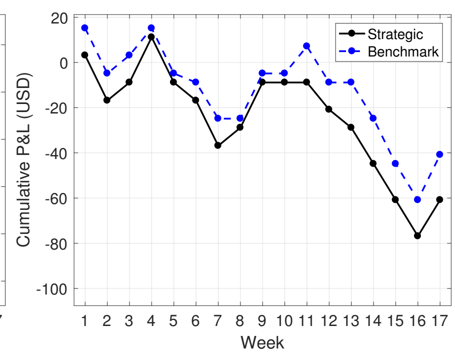
	Top-heavy		**Quintuple-up**		**Double-up**	
Week	Strategic	Benchmark	Strategic	Benchmark	Strategic	Benchmark
1	25.5	−39.5	-	-	3.13	15.13
2	−18.5	−77	−50	−50	−16.87	−4.87
3	85.24	−97	30	−60	−8.87	3.13
4	61.74	12.5	80	20	11.13	15.13
5	15.74	−34.5	30	−30	−8.87	−4.87
6	−7.26	−52.5	30	−70	−16.87	−8.87
7	92.74	−6.5	−10	20	−36.87	−24.87
8	170.74	0.5	−10	20	−28.87	−24.87
9	290.74	32.5	40	110	−8.87	−4.87
10	437.74	−17.5	0	60	−8.87	−4.87
11	406.74	189.5	20	130	−8.87	7.13
12	384.74	154.5	−30	80	−20.87	−8.87
13	347.74	112.5	10	40	−28.87	−8.87
14	341.74	87.5	−30	−10	−44.87	−24.87
15	320.74	59.5	−80	−30	−60.87	−44.87
16	317.74	29.5	−90	−50	−76.87	−60.87
17	280.74	91.5	−40	20	−60.87	−40.87



(a) Top-heavy



(b) Quintuple-up



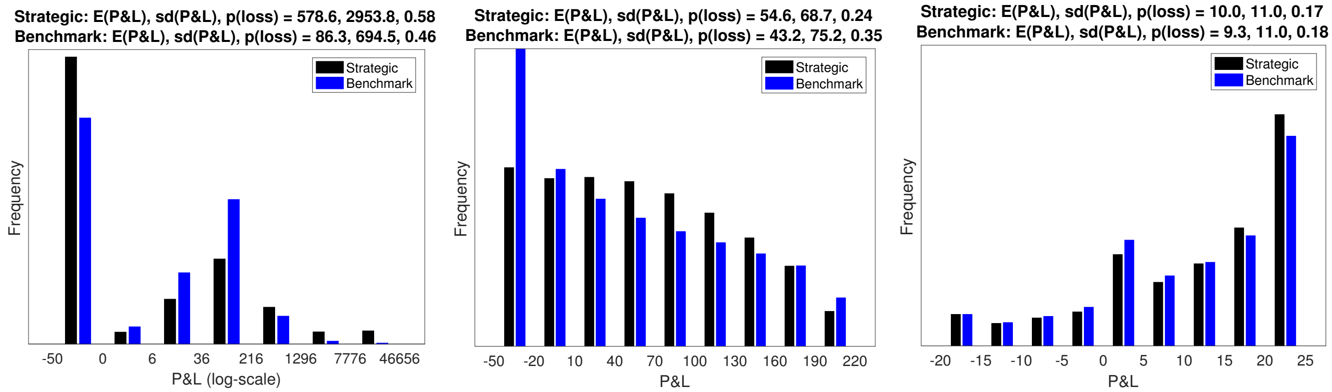
(c) Double-up

Figure 2: Cumulative realized dollar P&L for the strategic and benchmark models across the three contest structures for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season.

strategic and benchmark portfolios in week 10[26] contests. For the strategic portfolio, we use Algorithm 7 as explained in the beginning of Section 6 and for the benchmark portfolio, we use the procedure outlined in Section 6.1. We note this P&L distribution is as determined by our model with the continued assumption of the multivariate normal distribution for $\boldsymbol{\delta}$ as well as the Dirichlet-multinomial model for opponents' portfolio selections. The strategic model dominates the benchmark model in terms of expected profit. In the top-heavy contest, the expected profit of the strategic portfolio is over 5 times that of the benchmark portfolio.

[26]Other weeks have similar results as shown in Figure 5.

The gain is not as drastic in the quintuple-up and double-up contests. The substantial gain in top-heavy seems to come from the fact that the strategic portfolio has considerably more mass in the right-tail. Note this leads to the higher standard deviation of the top-heavy strategic portfolio[27].



(a) Top-heavy (b) Quintuple-up (c) Double-up

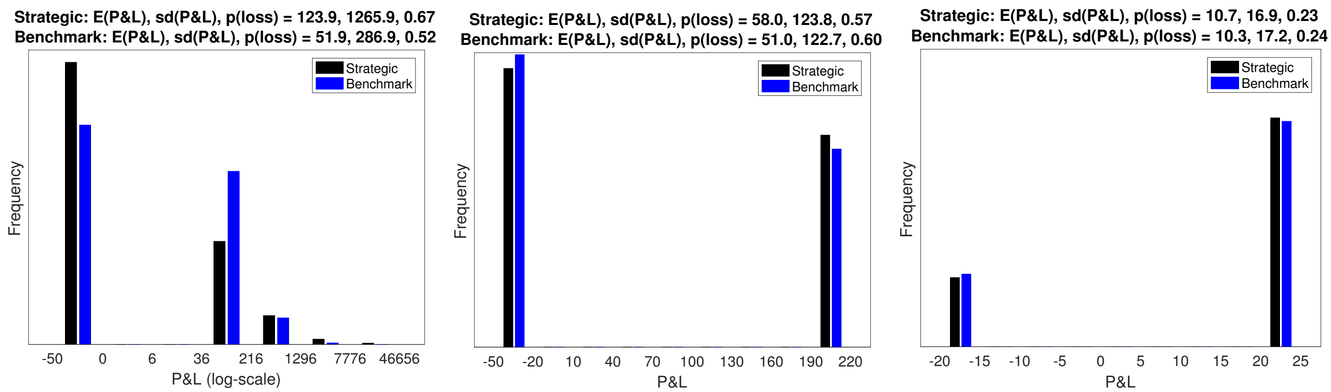
Figure 3: P&L distribution for the diversification strategy for the strategic and benchmark portfolios for week 10 contests of the 2017 NFL season. Recall $N = 50$, 25 and 10 for top-heavy, quintuple-up and double-up, respectively. The three metrics at the top of each image are the expected P&L, the standard deviation of the P&L and the probability of loss, that is, $\mathbb{P}(\text{P\&L} < 0)$.

Figure 4 is similar to Figure 3 except it is based upon using the replication strategy from Section 4.3 instead of the diversification strategy. We note the strategic model continues to have a higher expected P&L than the benchmark model. The main observation here is that the expected P&L drops considerably when we go from the diversification strategy to the replication strategy for top-heavy. This is consistent with our analysis from Section 5.1 on parimutuel betting as well as our discussion surrounding Algorithms 6 and 7 in Section 5.2. In contrast, the P&L increases for both quintuple-up and double-up when we employ the replication strategy. Again, this is consistent with our earlier argument in favor of replication for double-up style contests. In our numerical experiments, however, we used the diversification strategy for both double-up and quintuple-up contests. This was only because of the variance issue highlighted earlier and our desire to use a strategy which had a considerably smaller standard deviation (while ceding only a small amount of expected P&L). As can be seen from Figures 3 and 4, the diversification strategy has (as expected) a smaller expected P&L as well as a smaller probability of loss.

Figure 5 displays the realized and expected P&Ls. For both strategic and benchmark models and all three contests, the expected profit is greater than the realized profit. This is perhaps not too surprising given the bias that results from optimizing within a model. In top-heavy, however, the realized P&L is within one standard deviation of the expected P&L although this is not the case for the quintuple- and double-up contests. As discussed above, we believe our realized results are conservative and that a more proactive user of these strategies who makes a more determined effort to estimate $\boldsymbol{\mu_\delta}$ and $\boldsymbol{\Sigma_\delta}$ and responds to relevant news breaking just before the games can do considerably better. Despite this potential for improvement, the strategic model has performed very well overall. The small loses from the double-up and quintuple-up contests have been comfortably offset by the gains in the top-heavy contests. As we noted earlier, the return on investment in top-heavy is over 350% for a seventeen week period although we do acknowledge there is considerable variance in this number as evidenced by Figure 5(a).

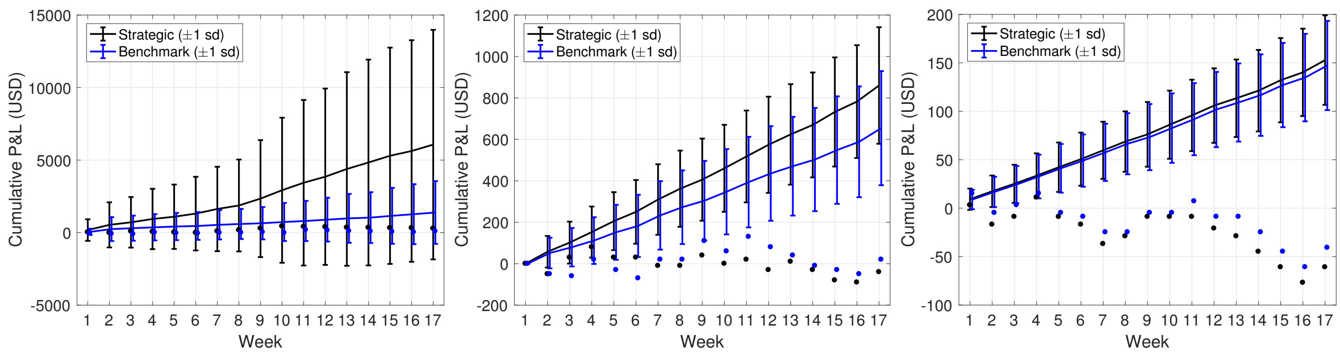
More granular results are presented in Appendix D. For example, in that appendix we show the perfor-

[27]The high standard deviation in the top-heavy strategic portfolio should be seen as a pro instead of a con, since it is mostly coming from the right-tail of the P&L distribution.



(a) Top-heavy (b) Quintuple-up (c) Double-up

Figure 4: P&L distribution for the replication strategy for the strategic and benchmark portfolios for week 10 contests of the 2017 NFL season. Recall $N = 50$, 25 and 10 for top-heavy, quintuple-up and double-up, respectively. The three metrics at the top of each image are the expected P&L, the standard deviation of the P&L and the probability of loss, that is, $\mathbb{P}(\text{P\&L} < 0)$.



(a) Top-heavy (b) Quintuple-up (c) Double-up

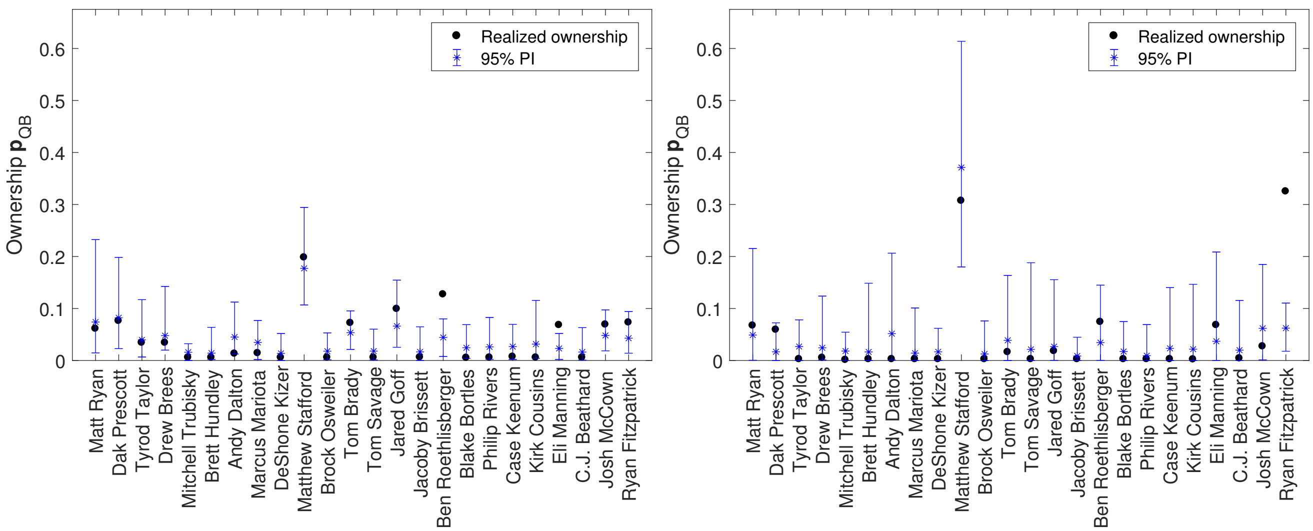
Figure 5: Predicted and realized cumulative P&L for the strategic and benchmark models across the three contest structures for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season. The realized cumulative P&Ls are displayed as points.

mance of each week's best entry for the strategic and benchmark models corresponding to the top-heavy contests for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season. We also present various statistics of interest, e.g., expected fantasy points, standard deviation, λ^* for the first optimal entry in each week for the strategic and benchmark models. We observe there that λ^* is closer to zero for the double-up and quintuple-up contests thereby indicating (see Remark 4.1) that the value of modeling opponents is much greater for top-heavy contests. Finally, Appendix D also contains some additional anecdotes describing situations where our strategic model went against the "crowd" and was successful in doing so.

6.3. Dirichlet Regression Results

Before concluding this section, we shed light on the performance of our Dirichlet-multinomial data generating process for modeling team selections of opponents and the corresponding Dirichlet regression introduced in Section 3 in terms of how well they predict the marginals $\mathbf{p}_{\text{QB}}, \ldots, \mathbf{p}_{\text{D}}$ and how well they predict the benchmark fantasy points $G^{(r')}$ (double-up) and $G^{(r'_d)}$ for $d = 1, \ldots, D$ (top-heavy). Our Dirichlet regression models used the features described in (5) and we validated this choice of features by evaluating its goodness-of-fit and comparing its out-of-sample performance against two "simpler" variations of the Dirichlet regression model. Specific details are deferred to Appendix E. In this subsection we focus instead on the key results and anecdotes we witnessed during the 2017-18 NFL season.

In Figure 6, we show the performance of our approach in terms of predicting the QB marginals $\mathbf{p}_{\text{QB}}$ for the top-heavy and double-up contests[28] in week 10 of the 2017 NFL season. First, we observe that in both top-heavy and double-up contests, our model correctly forecasted one of the top-picked QBs in week 10, namely Matthew Stafford. Second, we observe that our 95% prediction intervals (PI) contain around 95% of the realizations. This speaks to the predictive power of our statistical model. Of course, we expect roughly 5% of the realizations to lie outside the 95% intervals and we do indeed see this in our results. For example, in Figure 6, out of a total of 24 QBs, the number of QBs that lie outside the intervals for top-heavy and double-up equal 2 and 1, respectively.



(a) Top-heavy

(b) Double-up

Figure 6: Predicted and realized QB ownerships ($\mathbf{p}_{\text{QB}}$) for week 10 contests of the 2017 NFL season.

Of course, we did not do as well across all seventeen weeks as we did in week 10 but in general, our 95% prediction intervals contained 95% of the realizations. Over the course of the season, we did witness instances where our models under-predicted or over-predicted the ownerships by a relatively large margin. See Ryan Fitzpatrick in Figure 6(b), for example. Accordingly, there is room for improvement, specifically in the quality of features provided to our Dirichlet regression. Retrospectively speaking, including a feature capturing the "momentum" of athletes, that is, how well they performed in the previous few weeks, would have been beneficial in terms of predicting opponents' behavior. This statement is supported by multiple cases we noticed in the 2017 season. To give but one example, in week 9, Ezekiel Elliott (Dallas Cowboys) was picked by around 80% of our opponents in double-up but our 95% interval predicted 0% to 10%. It turns out that Elliott had performed extremely well in the two weeks prior to week 9. In fact, he was the top-scoring RB in week 7 and the second highest scoring RB in week 8.

We also would expect a significant improvement in predicting the player selections of our opponents if we were more proactive in responding to late developing news as discussed in Section 6.2. Such late developing

[28] We do not present Dirichlet regression results corresponding to quintuple-up contests for brevity. We note that the results in quintuple-up are very similar.

news would typically impact our estimate of $\boldsymbol{\mu_\delta}$ which in turn would change both our optimal portfolios as well as our opponents' team selections. Continuing on with the week 7 Fournette-Ivory case study from Section 6.2, due to our low estimate of the expected points of Ivory, we predicted his ownership to be below 5% with high probability (in top-heavy). In reality, around 15% of fantasy players in the top-heavy contest picked Ivory, which aligns with the sequence of events we discussed earlier. If we had updated our expected points estimate corresponding to Ivory to a value of 15 (from 6.78)[29], we would have fared better. This is illustrated in Figure 7(a) where we plot our original predictions ("before") in blue and updated predictions ("after")[30] in red. It's clear that our original over-prediction of the ownership of Le'Veon Bell can be largely attributed to our stale $\boldsymbol{\mu_\delta}$ estimate of Ivory. As a side note, we can also observe that both our "before" and "after" predictions in Figure 7(a) under-predict the ownership of Adrian Peterson (first point on the x-axis). We believe the reason for this is "momentum" (a feature we omitted from our Dirichlet regressions) as Peterson scored over 25 fantasy points in week 6 but was expected to score only 7 points, making him the RB with the highest points to cost ratio (among all week 6 RBs) that week.

An interesting illustration of the importance of good features can be found in Figure 7(b) where we display the positional marginals for QBs in week 12's double-up contest. We clearly over-predicted Tom Brady's ownership and under-predicted Russell Wilson's ownership. Perhaps the main reason for this was the point estimate $\mathbf{f}$ provided by FantasyPros (29.5% for Brady and 20.9% for Wilson) which was a feature in our Dirichlet regression. FantasyPros therefore severely overestimated the ownership of Brady and underestimated the ownership of Wilson and our regression model followed suit. However, it is well known in football that Tom Brady (arguably the greatest QB of all time) and the New England Patriots generally perform very poorly in Miami where his team were playing in week 12. It is no surprise then that the realized ownership of Tom Brady that week was very low. Unfortunately FantasyPros did not account for this in their prediction and so none of our features captured this well known Tom Brady - Miami issue. To confirm that it was indeed the FantasyPros point estimate that skewed our predictions we re-ran the regression after deducting 11% from Brady's FantasyPros' estimate (making it 18.5%) and adding it to Wilson's estimate. The resulting fit is displayed in red in Figure 7(b) and it's clear that it does a much better job of predicting the realized ownerships.

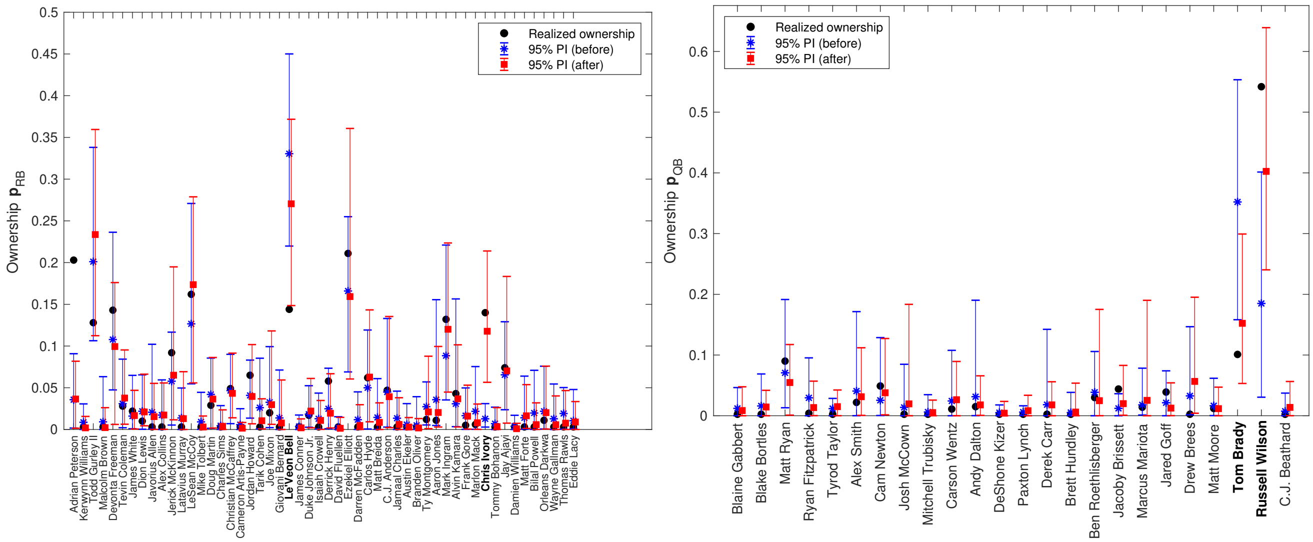
In Figure 8(a), we plot the realized fantasy points total against the rank r_d in the top-heavy contest of week 10. We also show our 95% prediction intervals for these totals as well as our 95% prediction intervals *conditional* on the realized value of $\boldsymbol{\delta}$. These conditional prediction intervals provide a better approach to evaluate the quality of our Dirichlet-multinomial model for $\mathbf{W}_{\text{op}}$ as they depend only on our model for $\mathbf{W}_{\text{op}}$. Not surprisingly, the interval widths shrink considerably when we condition on $\boldsymbol{\delta}$ and it is clear from the figure that we do an excellent job in week 10. In Figure 8(b), we display the results for the double-up contests across the entire 17 weeks of the 2017 NFL season. While our model appears to perform well overall, there were some weeks where the realized points total was perhaps 3 conditional standard deviations away from the conditional mean. This largely reflects the issues outlined in our earlier discussions, in particular the need to better monitor player developments in the day and hours immediately preceding the NFL games.

7. The Value of Modeling Opponents, Insider Trading, and Collusion

In the numerical results of Section 6, we found that modeling opponents' behavior can significantly increase the expected P&L from participating in top-heavy DFS contests and we explore it in more depth in Section 7.1. In Section 7.2, motivated by the issue of insider trading in fantasy sports we described in Section 1, we evaluate how much a fantasy player gains by having access to inside information. Finally, in Section 7.3, we analyze the value of collusion in fantasy sports, that is, how much does a fantasy player gain by strategically partnering with other fantasy players and submitting more portfolios than allowed. In each of

[29] An estimate of 15 for the expected points of a "main" RB is quite reasonable.

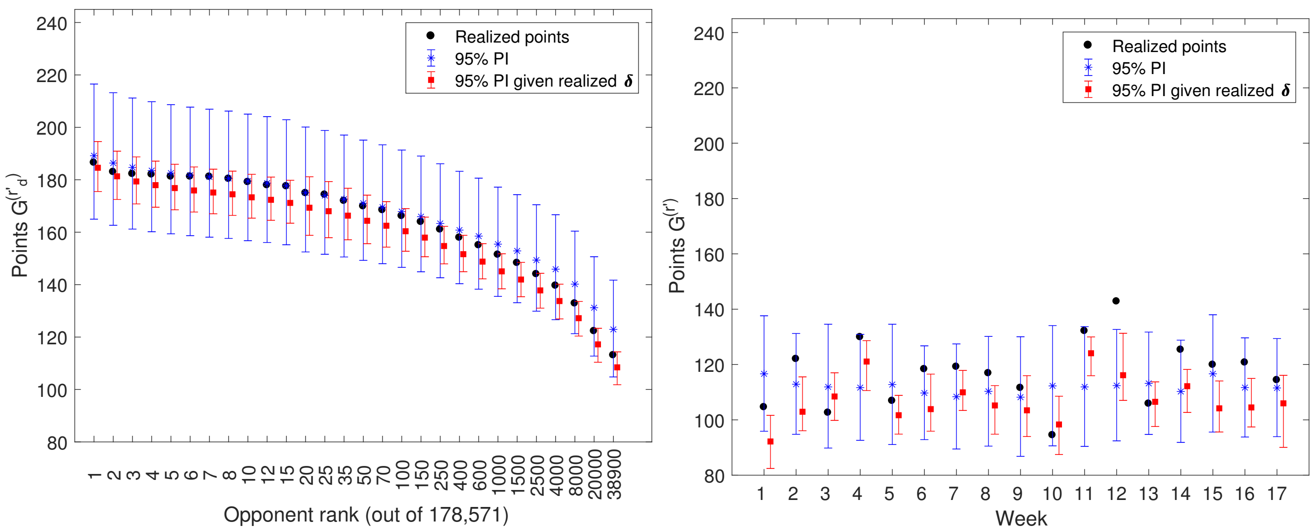
[30] For our updated predictions, we did not include the FantasyPros point estimate feature $\mathbf{f}$ in our Dirichlet regression model since we do not know how FantasyPros would have updated their estimate.



(a) Top-heavy week 7 RBs

(b) Double-up week 12 QBs

Figure 7: Highlighting instances where the Dirichlet regression either under-predicted or over-predicted ownerships of some athletes ("before") and what would have happened ("after") if we had (a) reacted to breaking news or (b) access to better quality features that accounted for historical factors such as Brady's poor track record in Miami.



(a) Top-heavy

(b) Double-up

Figure 8: Predicted and realized portfolio points total of various opponent ranks for (a) the top-heavy contest in week 10 and (b) all weeks of the double-up series during the 2017 NFL season. For double-up, the rank of interest for each week was around 13,000 and the number of opponents was around 30,000.

these experiments, we employ the same algorithms as we did for the numerical experiments of Section 6.

7.1. The Value of Modeling Opponents

As we saw in Figures 3 and 4, the value of modeling opponents is clearly contest-dependent. Indeed our model, which explicitly models opponents, has a much bigger edge (in terms of expected P&L) over the benchmark model in the top-heavy contest[31] as compared to the double-up and quintuple-up contests. But the value of modeling opponents also depends on how *accurately* we model their behavior. On this latter point, it is of interest to consider:

(a) How much do we gain (with respect to the benchmark model) if we use a deterministic $\mathbf{p} = (\mathbf{p}_{\text{QB}}, \ldots, \mathbf{p}_{\text{D}})$? For example, in the NFL contests, we could set $\mathbf{p}$ equal to the values predicted by the FantasyPros website.

(b) How much additional value is there if instead we assume $(\mathbf{p}_{\text{QB}}, \ldots, \mathbf{p}_{\text{D}}) \sim (\text{Dir}(\boldsymbol{\alpha}_{\text{QB}}), \ldots, \text{Dir}(\boldsymbol{\alpha}_{\text{D}}))$ as in Algorithm 1 but now $\boldsymbol{\alpha}_{\text{QB}}, \ldots, \boldsymbol{\alpha}_{\text{D}}$ only depend on the first two features stated in Equation (5), that is, the constant feature and the estimate of $\mathbf{p}$ that we obtain from the FantasyPros website?

(c) Finally, how much additional value is there to be gained by assuming the model of Algorithm 1 where $\boldsymbol{\alpha}_{\text{QB}}, \ldots, \boldsymbol{\alpha}_{\text{D}}$ is allowed to depend on any and all relevant features?

To answer these questions, we computed the optimal portfolios for each of the three cases described above (and for the benchmark model) and also the corresponding expected P&Ls by assuming case (c) to be the ground truth. We did this for all three contest structures for each of the 17 weeks in the 2017 NFL regular season. All the parameter values such as N and γ were as in Section 6. We found the value of modeling opponents accurately to be most valuable in the top-heavy contests. In particular, the total expected P&L (over 17 weeks) in the top-heavy series was approximately \$1,400, \$5,400, \$5,800, and \$6,000 for the benchmark model, case (a), case (b), and case (c), respectively. Accordingly, even though the deterministic model for $\mathbf{p}$ (case (a)) explains most of the gain in expected P&L we reap by being strategic, there is approximately an additional 10% reward we receive by modeling the opponents more precisely (cases (b) and (c)). It is worth emphasizing, however, that this 10% additional gain depends on our "ground truth" model. For example, if we had assumed some other ground truth where $\mathbf{p}$ was more predictable given additional and better chosen features, then there might be more to gain in moving from case (a) to case (c).

7.2. The Value of Insider Trading

A question that is somewhat dual to the first question concerns the issue of insider trading and the value of information. This question received considerable attention in 2015 [12, 13] when a DraftKings employee was accused of using data from DraftKings contests to enter a FanDuel DFS contest in the same week and win \$350,000. Without addressing the specific nature of insider trading in that case, we pose several questions:

(i) How much does the insider gain if he knows the true positional marginals $\mathbf{p} = (\mathbf{p}_{\text{QB}}, \ldots, \mathbf{p}_{\text{D}})$?

(ii) How much does the insider gain if he knows the entries of all contestants, that is, $\mathbf{W}_{\text{op}}$? In that case, the only uncertainty in the system is the performance vector $\boldsymbol{\delta}$ of the real-world athletes. (Note that the problem of computing an optimal portfolio given full knowledge of $\mathbf{W}_{\text{op}}$ is straightforward in our framework.)

To state these questions more formally, we note that the optimal expected P&L for a portfolio consisting of N entries satisfies

$$\max_{\mathbf{W} \in \mathbb{W}^N} \left\{ \mathop{\mathbb{E}}_{\mathbf{p}} \left[\mathop{\mathbb{E}}_{\boldsymbol{\delta}, \mathbf{W}_{\text{op}}} \left[\text{Reward}(\mathbf{W}, \boldsymbol{\delta}, \mathbf{W}_{\text{op}}) \mid \mathbf{p} \right] \right] \right\} \tag{22}$$

where $\text{Reward}(\mathbf{W}, \boldsymbol{\delta}, \mathbf{W}_{\text{op}})$ denotes the P&L function which is easy to compute given $\mathbf{W}$, $\boldsymbol{\delta}$, and $\mathbf{W}_{\text{op}}$. The answer to question (i) is then given by the difference between (22) and

$$\mathop{\mathbb{E}}_{\mathbf{p}} \left[\max_{\mathbf{W} \in \mathbb{W}^N} \left\{ \mathop{\mathbb{E}}_{\boldsymbol{\delta}, \mathbf{W}_{\text{op}}} \left[\text{Reward}(\mathbf{W}, \boldsymbol{\delta}, \mathbf{W}_{\text{op}}) \mid \mathbf{p} \right] \right\} \right]. \tag{23}$$

[31] This is discussed in more detail in Appendix B.3. The reason for the relative importance of modeling opponents is largely due to the importance of selecting entries with both a high variance and expectation in top-heavy contests.

Similarly, the answer to question (ii) is given by the difference between (22) and

$$\mathop{\mathbb{E}}_{\mathbf{p},\mathbf{W}_{\text{op}}}\left[\max_{\mathbf{W}\in\mathbb{W}^N}\left\{\mathop{\mathbb{E}}_{\boldsymbol{\delta}}\left[\text{Reward}(\mathbf{W},\boldsymbol{\delta},\mathbf{W}_{\text{op}})\mid\mathbf{p},\mathbf{W}_{\text{op}}\right]\right\}\right]. \tag{24}$$

However, computing both (23) and (24) is computationally expensive since the optimization occurs inside the expectation over high-dimensional random variables and hence many expensive optimizations would be required. Though one could perform such computations on an HPC cluster over an extended period of time, we instead designed less demanding but nonetheless informative experiments to evaluate the value of insider trading. In particular, we ran the following two experiments for all three contest structures across all 17 weeks of the 2017 NFL season:

- **Experiment 1**: We first compute the optimal portfolio for each week conditional on knowing the realized $\mathbf{p}$. We call this portfolio the *insider portfolio*. We then compare the expected P&L of the insider portfolio with the optimal strategic non-insider portfolio that we submitted to the real-world contests. (We assume the ground truth in the P&L computations to be the realized marginals $\mathbf{p}$ together with the same stacking parameters from Section 6.)
- **Experiment 2**: This is similar to Experiment 1 but we now replace $\mathbf{p}$ with $\mathbf{W}_{\text{op}}$. However, we do not have access to the realized values of $\mathbf{W}_{\text{op}}$ during the NFL season. Instead, for each week we sample one realization of $\mathbf{W}_{\text{op}}$ using the realized $\mathbf{p}$ (with the same stacking parameters from Section 6) and treat the sampled $\mathbf{W}_{\text{op}}$ as the realized value. We then compute the optimal portfolio (the *insider portfolio*) for each week conditional on knowing the realized $\mathbf{W}_{\text{op}}$ and compare the expected P&Ls of the insider portfolio with the strategic non-insider optimal portfolio assuming the ground truth in P&L computations to be the realized $\mathbf{W}_{\text{op}}$.

It is worth emphasizing that in both Experiments 1 and 2, we are taking expectations over $\boldsymbol{\delta}$, the performance vector of the underlying NFL athletes. As such, we are averaging over the largest source of uncertainty in the system.

In Experiment 1, we found the insider to have an edge (in terms of total expected P&L across the season) of around 20%, 1%, and 2% in top-heavy, quintuple-up[32], and double-up contests respectively over the (strategic) non-insider. In Figure 9, we compare the weekly expected top-heavy P&L of the insider and (strategic) non-insider portfolios and observe that the weekly increase varies from 1% (week 6) to 50% (week 16). As one would expect, the insider portfolio's P&L dominates that of the non-insider's. Of course, the insider will have an even greater edge over a non-insider who is not strategic as we have already seen in Section 6 that the strategic non-insider has roughly five times the expected P&L of the non-strategic non-insider in top-heavy contests. Compared to this approximately 500% difference between the benchmark and strategic players, the additional 20% increase in expected P&L gained via insider trading seems modest. This modest increase is due in part to how well our Dirichlet regression model allows the (strategic) non-insider to estimate the positional marginals $\mathbf{p}$. Accordingly, the value of inside information depends on how well the non-insider can predict opponents' behavior. In particular, the more sophisticated the non-insider is, then the less value there is to having inside information.

In Experiment 2, we found the insider's edge to be similar to that of Experiment 1. Intuitively, one would expect the edge to be bigger in Experiment 2 due to the insider having the more granular information

[32] As expected, the benefits of insider trading were much greater in top-heavy contests than in the double- and quintuple-up contests where we expected the benefits to be quite small. It is quite interesting, however, to see that the observed benefits in quintuple-up (1%) were less than the observed benefits in double-up (2%). We suspect this may be related to the same issue with quintuple-up that we identified earlier in Section 6.2, namely the issue that arises when the maximal value of $\mu_{\mathbf{w}} \approx 0$. In this case, the optimal value of λ in Algorithm 2 will be close to 0. Indeed this is what we observed in Table 4. As a result (and this should be clear from the expression for $\sigma^2_{Y_{\mathbf{w}}}$ in (11) together with lines 3 and 7 of Algorithm 2) the only benefit to inside information in this case is in estimating $\mu_{Y_{\mathbf{w}}}$.

of $\mathbf{W}_{\text{op}}$. Noting that the variance of $G^{(r')} \mid (\boldsymbol{\delta}, \mathbf{p})$ goes to zero as the number of opponents O goes to infinity, however, we can conclude that the additional value of seeing the realized $\mathbf{W}_{\text{op}}$ over and beyond the value of seeing the realized $\mathbf{p}$ should[33] be small when O is large. Given that the contests we participated in had large O, this observation supports our results from Experiment 2.

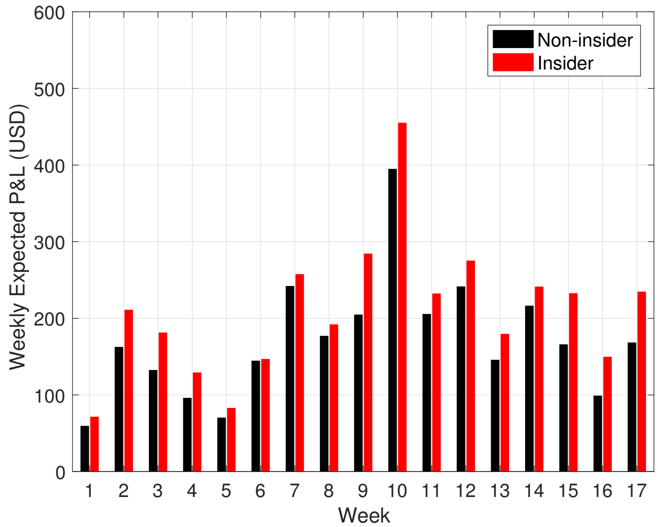


Figure 9: Weekly expected dollar P&L for the strategic model ($N = 50$) with and without inside information $\mathbf{p}$ in the top-heavy series.

7.3. The Value of Collusion

In addition to the insider trading controversy, the subject of collusion in fantasy sports contests has also received considerable attention. In one suspected case, two brothers were suspected of colluding when one of them won 1 million dollars [7, 35] in one of DraftKings' "Fantasy Football Millionaire" contests, a particularly top-heavy contest where just the first few places earn most of the total payoff. Collusion refers to the situation where two or more DFS players form (unbeknownst to the contest organizers) a strategic partnership and agree to pool their winnings. Maintaining separate accounts allows the partnership to submit $N_{\text{collude}} \times E_{\max}$ entries to a given contest where N_{collude} is the number of players in the partnership and $E_{\max}$ is the maximum number of entries permitted per player. Collusion can be beneficial in top-heavy contests as it allows the colluding players to avoid substantial overlap (and therefore achieve greater diversification) in their portfolios thereby increasing the probability that the partnership will win a large payout.

We will assume that the N_{collude} players will construct a single portfolio of $N_{\text{collude}} \times E_{\max}$ entries when they collude. This portfolio can be constructed using Algorithm 7 from Section 5.2 with $N = N_{\text{collude}} \times E_{\max}$. This portfolio can then be separated into N_{collude} separate sub-portfolios each consisting of $E_{\max}$ entries and each colluding player can then submit one of these sub-portfolios as his official submission.

In order to estimate the benefits of collusion, it is first necessary to understand the behavior of the colluding players when they are unable to collude. Many different behaviors are of course possible but it seems reasonable to assume that potentially colluding players are sophisticated and understand how to construct good portfolios. We therefore assume[34] that each of the potentially colluding players has access to

[33]But note we are assuming here that the dependence structure between the positional marginals in $\mathbf{p}$ is known regardless of whether we only see $\mathbf{p}$ or $\mathbf{W}_{\text{op}}$.

[34]To the extent that our framework is a good framework for constructing DFS portfolios (which we believe to be the case!), then this might overstate the value of collusion as most colluding players will not have access to such a framework. Nonetheless, we can use this framework to consider just how beneficial colluding *might* be.

the modeling framework outlined in this paper and that as a result, each one submits *identical* portfolios of E_{max} entries. This portfolio is constructed using the same approach from Section 5.2. While this assumption is stylized and not realistic in practice, it does allow us to compute an upper bound on how beneficial colluding might be. Specifically, we can easily estimate and compare the expectations and standard deviations of the profits for the colluding and non-colluding portfolios in order to estimate the potential benefits of colluding. We would argue that the difference in expected values provides an upper bound on the value of colluding since in practice non-colluders are very unlikely to choose identical or even near-identical portfolios.

Before describing our numerical experiments, it is worthwhile noting that the results of Section 6.2 and specifically, Figures 3(a) and 4(a), can be used to estimate the benefits of colluding in week 10 top-heavy[35] contests of the 2017 NFL season if $E_{\text{max}} = 1$ and $N_{\text{collude}} = 50$. We see from Figure 3(a) that collusion in this case results in an estimated expected profit of 578.6 with a standard deviation of 2,953.8. In contrast, we can see from Figure 4(a) that the non-colluding portfolio has an expected profit of 123.9 with a standard deviation of 1,265.9. In this case, the colluding portfolio has an expected profit that is almost 5 times the expected profit of the non-colluding portfolio. It may appear this gain is coming at a cost, namely a higher standard deviation, but we note the higher standard deviation is entirely due to increased dispersion on the *right-hand-side* of the probability distribution. This is clear from Figures 3(a) and 4(a). Indeed we note that the probability of loss is 0.58 in Figure 3(a) (collusion) and increases to 0.67 in Figure 4(a) (non-collusion). This increased standard deviation can therefore hardly be considered a cost of collusion.

We also performed a more formal experiment to evaluate the value of collusion in top-heavy contests. We assumed the larger value of $E_{\text{max}} = 50$ which is quite common in practice and then varied the number of colluders so that N_{collude} ranged from 1 to 5. To be clear, the non-colluding portfolio comprised 50 strategic entries replicated N_{collude} times whereas the colluding portfolio consisted of $N_{\text{collude}} \times 50$ strategic entries. In Table 2, we compare the performances of the colluding and non-colluding portfolios over the 2017 NFL season in terms of the total expected dollar P&L, the average weekly Sortino ratio, and the average weekly probability of loss over the 17 weeks of the 2017 NFL season. To be clear, both portfolios were constructed for each week using our calibrated model for that specific week. The expected P&L for the week was then computed by averaging (via Monte Carlo) over $\boldsymbol{\delta}$ and $\mathbf{W}_{\text{op}}$ where samples of $(\boldsymbol{\delta}, \mathbf{W}_{\text{op}})$ were generated using the same[36] calibrated model. In particular, the *realized* $(\boldsymbol{\delta}, \mathbf{W}_{\text{op}})$'s across the 17 weeks played no role in the experiment.

The colluding portfolio clearly dominates the non-colluding portfolio across the three metrics and for all values of N_{collude}. For example, collusion among 5 sophisticated fantasy players can increase the expected P&L for the 17-week season by 44%, increase the average weekly Sortino ratio by 63%, and decrease the average weekly loss probability by 8%. It is also clear from these numbers that collusion also results in a decreased downside risk (square root of $\mathbb{E}[\text{P\&L}^2 \times \mathbb{1}_{\{\text{P\&L} \leq T\}}]$) since the percentage increase in the Sortino ratio is more than the percentage increase in the expected P&L. Accordingly, collusion results in a win-win situation by increasing the expected P&L and decreasing the downside risk simultaneously, which demonstrates that collusion can be surprisingly valuable in top-heavy DFS contests.

Of course the benefits from collusion are not as great as those from week 10 reported above when $E_{\text{max}} = 1$ and $N_{\text{collude}} = 50$. This is because it is intuitively clear that these benefits, while positive, are a decreasing function of E_{max} all other things being equal. For example, in the extreme case where $E_{\text{max}} = \infty$, there are clearly no benefits to colluding. In practice, we suspect the gains from collusion are much smaller for risk-neutral players since it is extremely unlikely that non-colluders would ever choose identical or near-identical portfolios as we have assumed here.

[35] Not surprisingly we do not see any benefits to collusion in the double-up or quintuple-up contest here and indeed as pointed out earlier, we expect replication (which corresponds to non-collusion in the setting considered here) to be very close to optimal.

[36] Both colluding and non-colluding portfolios then benefitted in this experiment from the fact that the assumed model was indeed the correct model. We are interested in the *difference* in performances of the two portfolios, however, and so the bias that results from assuming the players know the true model should be relatively small.

Table 2: Total expected dollar P&L (over 17 weeks), average weekly Sortino ratio and average weekly probability of loss related to the top-heavy contests for both the non-colluding ("NC") and colluding ("C") portfolios with $E_{\max} = 50$ and $N_{\text{collude}} \in \{1, \ldots, 5\}$. The average weekly Sortino ratio is simply the average of the weekly Sortino ratios, SR_i for $i = 1, \ldots, 17$. Specifically $\text{SR}_i := (\mathbb{E}[\text{P\&L}_i] - T)/\text{DR}_i$ where $\mathbb{E}[\text{P\&L}_i]$ denotes the expected P&L for week i, T denotes the target P&L which we set to 0, and $\text{DR}_i := \sqrt{\mathbb{E}[\text{P\&L}_i^2 \times \mathbb{1}_{\{\text{P\&L}_i \leq T\}}]}$ denotes the downside risk for week i. (The expected P&L is rounded to the nearest integer whereas the Sortino ratio and probability of loss are rounded to two decimal places.)

	Expected P&L (USD)			**Sortino Ratio**			**Probability of Loss**		
N_{collude}	NC	C	Increase	NC	C	Increase	NC	C	Decrease
1	6,053	6,053	0%	14.60	14.60	0%	0.49	0.49	0%
2	9,057	10,240	13%	11.02	13.24	20%	0.49	0.47	4%
3	10,975	13,776	26%	8.96	12.32	37%	0.49	0.46	6%
4	12,411	16,883	36%	7.64	11.56	51%	0.49	0.46	7%
5	13,632	19,677	44%	6.75	10.99	63%	0.49	0.45	8%

8. Conclusions and Further Research

In this paper, we have developed a new framework for constructing portfolios for both double-up and top-heavy DFS contests. Our methodology explicitly accounts for the behavior of DFS opponents and leverages mean-variance theory (for the outperformance of stochastic benchmarks) to develop a tractable algorithm that requires solving a series of binary quadratic programs. Following Hunter et al. [22], we also provide a tractable greedy algorithm for handling the multiple entry, i.e., $N > 1$, case for top-heavy style contests. This is in contrast to the replication approach we advocate for double-up style contests. Moreover, our greedy algorithm (or simple variations of it) can be justified theoretically via the results we developed on parimutuel betting as well as the classic result of Nemhauser et al. [30] on the performance of an idealized greedy algorithm for submodular maximization.

There are many potential directions for future research. We could back-test other benchmark strategies as well as refine our own preferred strategies. It would also be interesting to further develop our modeling and estimation approach for a random opponent's portfolio $\mathbf{w}_o$. We assumed in Section 3 that we had sufficient data to estimate the positional marginals of $\mathbf{w}_o$ and we would like to explore other features that might be useful in the Dirichlet regression to better estimate these marginals. We would also like to explore other copula models for splicing these marginals together to construct the joint distribution of $\mathbf{w}_o$. It is not clear, however, whether we could ever obtain a rich enough data-set to estimate other copulas sufficiently accurately.

While NFL contests are among the most popular DFS contests, the season is quite short with only 17 rounds of games. Moreover, as mentioned in Section 6.2, the individual performance of an NFL player in a given week has quite a high variance, potentially causing the cumulative P&L in football DFS contests to be high relative to other sports such as basketball, ice hockey and baseball. For these reasons and others, it would be interesting to apply our modeling framework to DFS contests in these other sports. It would also be interesting to use domain knowledge of these other sports to actively update estimates of $\boldsymbol{\mu_\delta}$ and $\boldsymbol{\Sigma_\delta}$ as the round of games approaches. This is something we did not do in the current NFL season. Indeed we recorded many instances when it would have been possible to avoid certain athletes in our DFS entries had we used up-to-date information that was available before the games in question and before our entries needed to be submitted. As a result, we believe the net positive P&L achieved by our models is very encouraging and can easily be improved (in expectation) by more active monitoring of the athletes.

Other directions for future research include the development of very fast re-optimization procedures / heuristics that could be performed on an already optimized portfolio of N entries when new information regarding player injuries, availability, weather etc. become known in the hours (and indeed minutes) before the portfolio of entries must be submitted to the DFS contest. As discussed in Section 6, such late-breaking developments are common-place and in order to extract the full benefit of the modeling framework presented here, it is important that such developments be reflected in updated parameter estimates which in turn calls for re-optimizing the entries. Of course, it would be desirable to re-optimize the entire portfolio in such circumstances but given time constraints, it may be necessary to make do with simple but fast heuristic updates. For the same reason, it would also be of interest to pursue more efficient Monte Carlo strategies for estimating the inputs $\mu_{G^{(r')}}$, $\sigma^2_{G^{(r')}}$ and $\boldsymbol{\sigma}_{\boldsymbol{\delta},G^{(r')}}$ that are required for the various algorithms we proposed. While we did make use of results from the theory of order statistics to develop our Monte Carlo algorithm, it should be possible to develop considerably more efficient algorithms to do this. In the case of top-heavy contests, for example, the moments corresponding to the top order statistics are particularly important and it may be possible to design importance-sampling or other variance reduction algorithms to quickly estimate them.

Finally, we briefly mention the area of mean-field games. In our modeling of opponents, we did not assume they were strategic although we did note how some strategic modeling along the lines of stacking to increase portfolio variance could be accommodated. If we allowed some opponents to be fully strategic, then we are in a game-theoretic setting. Such games would most likely be impossible to solve. Even refinements such as mean-field games (where we let $O \to \infty$ in some appropriate fashion) would still likely be intractable, especially given the discreteness of the problem (binary decision variables) and portfolio constraints. But it may be possible to solve very stylized versions of these DFS games where it is possible to purchase or sell short fractional amounts of athletes. There has been some success in solving mean-field games in the literature on parimutuel betting [4] in horse-racing and it may be possible to do likewise here for very stylized versions of DFS contests.

We hope to pursue some of these directions in future research.

References

1. Anderton, K. 2016. FanDuel And DraftKings Are Dominating The Daily Fantasy Sports Market. https://www.forbes.com/sites/kevinanderton/2016/11/30/fanduel-and-draftkings-are-dominating-the-daily-fantasy-sports-market-infographic/#2979acb87c4f. [Accessed: 26-Feb-2017].

2. Asmussen, S. and P.W. Glynn. 2007. *Stochastic simulation: algorithms and analysis*. Vol. 57. Springer Science & Business Media.

3. Bales, J. 2016. Pairing a QB with his Receiver(s). https://rotogrinders.com/articles/pairing-a-qb-with-his-receiver-s-481544. [Accessed: 26-Feb-2017].

4. Bayraktar, E. and A. Munk. 2017. High-Roller Impact: A Large Generalized Game Model of Parimutuel Wagering. *Market Microstructure and Liquidity*. 3(01) 1750006.

5. Becker, A. and X.A. Sun. 2016. An analytical approach for fantasy football draft and lineup management. *Journal of Quantitative Analysis in Sports*. 12(1) 17-30.

6. Bergman, D. and J. Imbrogno. 2017. Surviving a National Football League Survivor Pool. *Operations Research*. 65(5) 1343-1354.

7. Brown, L. 2016. DraftKings investigating potential collusion in millionaire contest. http://larrybrownsports.com/sports-business/draftkings-investigating-collusion-millionaire-contest/325182. [Accessed: 29-March-2018].

8. Cook, J.D. 2010. Determining distribution parameters from quantiles.

9. Clair, B. and D. Letscher. 2007. Optimal strategies for sports betting pools. *Operations Research*. 55(6) 1163-1177.

10. David, H.A. and H.N. Nagaraja. 1981. *Order statistics*. Wiley Online Library.

11. Davis, A. 2017. Data-Driven Portfolios Power 'Home-Run' Exits in MIT Study. https://www.wsj.com/articles/data-driven-portfolios-power-home-run-exits-in-mit-study-1502105402. [Accessed: 1-Dec-2017].

12. Drape, J. and J. Williams. 2015a. In Fantasy Sports, Signs of Insiders' Edge. https://www.nytimes.com/2015/10/12/sports/fantasy-sports-draftkings-fanduel-insiders-edge-football.html. [Accessed: 26-Feb-2017].

13. Drape, J. and J. Williams. 2015b. Scandal Erupts in Unregulated World of Fantasy Sports. https://www.nytimes.com/2015/10/06/sports/fanduel-draftkings-fantasy-employees-bet-rivals.html. [Accessed: 26-Feb-2017].

14. FanDuel. 2016. Rules & Scoring. https://www.fanduel.com/rules. [Accessed: 26-Feb-2017].

15. FantasyPros. 2017. https://www.fantasypros.com. [Accessed: 30-Aug-2017].

16. Fry, M.J., A.W. Lundberg, and J.W. Ohlmann. 2007. A player selection heuristic for a sports league draft. *Journal of Quantitative Analysis in Sports*. 3(2).

17. FSTA. 2015. Industry demographics. http://fsta.org/research/industry-demographics/. [Accessed: 26-Feb-2017].

18. Gelman, A., H.S. Stern, J.B. Carlin, D.B. Dunson, A. Vehtari and D.B. Rubin. 2013. *Bayesian data analysis*. Chapman and Hall/CRC.

19. Gibbs J. 2017. Week 1 FanDuel NFL Tournament Pivots. https://www.numberfire.com/nfl/lists/16195/week-1-fanduel-nfl-tournament-pivots. [Accessed: 1-Dec-2017].

20. Gurobi Optimization, Inc. 2016. Gurobi Optimizer Reference Manual. http://www.gurobi.com.

21. Harwell, D. 2015. Why you (probably) won't win money playing DraftKings, FanDuel. http://www.dailyherald.com/article/20151012/business/151019683/. [Accessed: 26-July-2016].

22. Hunter, D.S., J.P. Vielma, and T. Zaman. 2016. Picking winners using integer programming. *arXiv preprint arXiv:1604.01455*.

23. Johnson, A. 2015. After Ferocious Battle, New York Legalizes DraftKings, FanDuel. https://www.nbcnews.com/news/us-news/after-ferocious-battle-new-york-legalizes-draftkings-fanduel-n622606. [Accessed: 16-April-2019].

24. Kaplan, E.H. and S.J. Garstka. 2001. March madness and the office pool. *Management Science*. 47(3) 369-382.

25. Kolodny, L. 2015. Fantasy Sports Create Billion-Dollar Startups. https://www.wsj.com/articles/fantasy-sports-create-billion-dollar-startups-1436846402. [Accessed: 26-Feb-2017].

26. Miller, E., D. Singer. 2015. For daily fantasy-sports operators, the curse of too much skill. https://www.mckinsey.com/~/media/McKinsey/Industries/Media%20and%20Entertainment/Our%20Insights/For%20daily%20fantasy%20sports%20operators%20the%20curse%20of%20too%20much%20skill/For%20daily%20fantasysports%20operators%20the%20curse%20of%20too%20much%20skill.ashx. [Accessed: 17-Apr-2019].

27. Morton, D.P., E. Popova, I. Popova, M. Zhong. 2003. Optimizing benchmark-based utility functions. *Bulletin of the Czech Econometric Society*. 10(18).

28. Mulshine, M. 2015. How one man made hundreds of thousands of dollars playing daily fantasy sports. http://www.businessinsider.com/draft-kings-jonathan-bales-a-day-in-the-life-2015-10. [Accessed: 26-Feb-2017].

29. Nelsen, R.B. 2007. *An introduction to copulas*. Springer Science & Business Media.

30. Nemhauser, G.L., L.A. Wolsey, M.L. Fisher. 1978. An analysis of approximations for maximizing submodular set functions-I. *Mathematical programming*. 14(1) 265-294. Springer.

31. Nickish, K. 2015. Meet A Bostonian Who's Made $3 Million This Year Playing Daily Fantasy Sports. http://www.wbur.org/news/2015/11/23/dfs-power-player-profile. [Accessed: 26-Feb-2017].

32. O'Keeffe, K. 2015. Daily Fantasy-Sports Operators Await Reality Check. https://www.wsj.com/articles/daily-fantasy-sports-operators-await-reality-check-1441835630. [Accessed: 26-Feb-2017].

33. Plott, C.R., J. Wit, W.C. Yang. 2003. Parimutuel betting markets as information aggregation devices: experimental results. *Economic Theory*. 22 311–351.

34. Pramuk, J. 2015. Former pro poker player makes a living on fantasy sports. http://www.cnbc.com/2015/10/06/former-pro-poker-player-makes-a-living-on-fantasy-sports.html. [Accessed: 26-Feb-2017].

35. Reagan, B. 2016. DraftKings Investigating Potential Collusion in $1 Million Contest. https://www.wsj.com/articles/draftkings-investigating-potential-collusion-in-1-million-contest-1475091553. [Accessed: 29-March-2018].

36. RotoViz. 2017. http://rotoviz.com. [Accessed: 30-Aug-2017].

37. Sklar, M. 1959. Fonctions de repartition an dimensions et leurs marges. *Publ. inst. statist. univ. Paris.* 8 229-231.

38. Stan Development Team. 2017. *RStan: the R interface to Stan.* R package version 2.16.2. http://mc-stan.org.

39. Terrell, D., A. Farmer. 1996. Optimal Betting and Efficiency in Parimutuel Betting Markets with Information Costs. *The Economic Journal.* Vol. 106, No. 437, 846–858.

40. Wong, K. 2015. The Fantasy Sports Industry, by the Numbers. http://www.nbcnews.com/business/business-news/fantasy-sports-industry-numbers-n439536. [Accessed: 26-Feb-2017].

41. Woodward, C. 2015. Top fantasy sports player uses software, analytics to reap millions. https://www.bostonglobe.com/business/2015/12/23/from-boston-penthouse-world-best-fantasy-player-plunges-into-startup-world/QHNpLhOO3QMyUDqTd4t27N/story.html. [Accessed: 26-Feb-2017].

42. Woodward, C. 2016. DraftKings, FanDuel collected $3b in entry fees last year, analyst says. https://www.bostonglobe.com/business/2016/02/29/fantasy-sports-industry-hits-amid-legal-questions-analyst-says/NKw364kiLjv8XcD54vRr4H/story.html. [Accessed: 26-Feb-2017].

43. Thaler, R.H., W.T. Ziemba. 1988. Anomalies: Parimutuel Betting Markets: Racetracks and Lotteries, *Journal of Economic Perspectives.* 2(2) 161–174.

A. Efficient Sampling of Order Statistic Moments

Monte Carlo simulation is required to generate samples of $(\boldsymbol{\delta}, G^{(r')})$. These samples are required to:

1. Estimate the input parameters $(\mu_{G^{(r')}}, \boldsymbol{\sigma}_{\boldsymbol{\delta},G^{(r')}})$ that are required by the various algorithms in Sections 4, 5 and 7.
2. Estimate the expected payoff from a given entry in the various algorithms, e.g. line 4 in Algorithm 4.
3. Estimate the P&L distribution for a given portfolio of entries using samples of $(\boldsymbol{\delta}, G^{(r')})$.

Recalling $G_o = \mathbf{w}_o^\top \boldsymbol{\delta}$ is the fantasy points score of the o^{th} opponent, we first note the G_o's, $o = 1, \ldots, O$, are IID given $(\boldsymbol{\delta}, \mathbf{p})$ where $\mathbf{p}$ denotes the multinomial probability vectors for the positional marginals as discussed in Section 3. This then suggests the following algorithm for obtaining independent samples of $(\boldsymbol{\delta}, G^{(r')})$:

1. Generate[37] $\boldsymbol{\delta} \sim \mathrm{N}(\boldsymbol{\mu}_{\boldsymbol{\delta}}, \boldsymbol{\Sigma}_{\boldsymbol{\delta}})$ and $(\mathbf{p}, \mathbf{W}_{\text{op}})$ using Algorithm 1 where $\mathbf{p} := (\mathbf{p}_{\text{QB}}, \ldots, \mathbf{p}_{\text{D}})$ and $\mathbf{W}_{\text{op}} = \{\mathbf{w}_o\}_{o=1}^{O}$.
2. Compute $G_o := \mathbf{w}_o^\top \boldsymbol{\delta}$ for $o = 1, \ldots, O$.
3. Order the G_o's.
4. Return $(\boldsymbol{\delta}, G^{(r')})$.

While all of the contests that we participated in had relatively large values of O, it is worth noting there are also some very interesting DFS contests with small values of O that may range[38] in value from $O = 1$ to $O = 1000$. These small-O contests often have very high entry fees with correspondingly high payoffs and there is therefore considerable interest in them. At this point we simply note that (based on unreported numerical experiments) the algorithm described above seems quite adequate for handling small-O contests. Of course, if we planned to participate in small-O contests and also be able to quickly respond to developing news in the hours and minutes before the games, then it may well be necessary to develop a more efficient Monte Carlo algorithm. This of course is also true for the large-O algorithm we develop below.

A.1. Efficient Monte Carlo when O is Large

When O is large, e.g. when $O = 500,000$ which is often the case in practice, the algorithm above is too computationally expensive and so a more efficient algorithm is required. Recalling that the conditional random variables $G_o \mid (\boldsymbol{\delta}, \mathbf{p})$ are IID for $o = 1, \ldots, O$, it follows [10] from the theory of order statistics that $G^{(r')} \mid (\boldsymbol{\delta}, \mathbf{p})$ satisfies

$$G^{(qO)} \mid (\boldsymbol{\delta}, \mathbf{p}) \xrightarrow{p} F^{-1}_{G|(\boldsymbol{\delta},\mathbf{p})}(q) \qquad \text{as} \;\; O \to \infty \tag{25}$$

where $q \in (0,1)$ and "$\xrightarrow{p}$" denotes convergence in probability. In large-O contests, we can use the result in (25) by simply setting $G^{(r')} = F^{-1}_{G|(\boldsymbol{\delta},\mathbf{p})}\left(\frac{r'}{O}\right)$. Of course in practice we do not know the CDF $F_{G|(\boldsymbol{\delta},\mathbf{p})}$ and so we will have to estimate it as part of our algorithm. The key observation now is that even if the DFS contest in question has say 500,000 contestants, we can estimate $F_{G|(\boldsymbol{\delta},\mathbf{p})}$ with potentially far fewer samples. Our algorithm for generating Monte Carlo samples of $(\boldsymbol{\delta}, G^{(r')})$ therefore proceeds as follows:

1. Generate $\boldsymbol{\delta} \sim \mathrm{N}(\boldsymbol{\mu}_{\boldsymbol{\delta}}, \boldsymbol{\Sigma}_{\boldsymbol{\delta}})$ and $(\mathbf{p}, \mathbf{W}_{\text{op}})$ using Algorithm 1 (or the stacking variant of it) where $\mathbf{p} := (\mathbf{p}_{\text{QB}}, \ldots, \mathbf{p}_{\text{D}})$ and $\mathbf{W}_{\text{op}} = \{\mathbf{w}_o\}_{o=1}^{O}$.
2. Compute $G_o := \mathbf{w}_o^\top \boldsymbol{\delta}$ for $o = 1, \ldots, O$.
3. Use the G_o's to construct $\widehat{F}_{G\,|\,(\boldsymbol{\delta},\mathbf{p})}(\cdot)$.

[37] We note that the normal assumption for $\boldsymbol{\delta}$ is not necessary and any multivariate distribution with mean vector $\boldsymbol{\mu}_{\boldsymbol{\delta}}$ and variance-covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\delta}}$ could also be used.

[38] All of the contests that we participated in during the 2017 NFL season had values of O that exceeded 8,000. That said, the cutoff between small O and large O is entirely subjective and indeed we could also add a third category – namely moderate-O contests. These contests might refer to contests with values of O ranging from $O = 500$ to $O = 5,000$.

4. Set $G^{(r')} = \widehat{F}^{-1}_{G\,|\,(\boldsymbol{\delta},\mathbf{p})}\left(\frac{r'}{O}\right)$.
5. Return $(\boldsymbol{\delta}, G^{(r')})$.

Note in this algorithm O now represents the number of Monte Carlo samples we use to estimate $F_{G|(\boldsymbol{\delta},\mathbf{p})}$ rather than the number of contestants in a given DFS contest. One issue still remains with this new algorithm, however. Consider for example the case where $r' = O+N-1$ (corresponding to the # 1 ranked opponent) in a top-heavy contest with say 100,000 contestants. This corresponds to the quantile $q = 1-10^{-5}$ and according to line 4 of the algorithm we can generate a sample of $G^{(O+N-1)}$ by setting it equal to $\widehat{F}^{-1}_{G|(\boldsymbol{\delta},\mathbf{p})}\left(1-10^{-5}\right)$. We cannot hope to estimate $\widehat{F}^{-1}_{G|(\boldsymbol{\delta},\mathbf{p})}\left(1-10^{-5}\right)$ with just a moderate number of samples from line 2 of the algorithm, however, and this of course also applies to the values of r' corresponding to the # 2 ranked opponents, the # 3 ranked opponent etc.

We overcome this challenge as follows. We set O to a moderate value, e.g. $O = 10,000$, and then estimate the conditional CDF $\widehat{F}_{G|(\boldsymbol{\delta},\mathbf{p})}(\cdot)$ with the empirical CDF of those O samples from line 2 of the algorithm. For r' values that are not deep in the tail, we use $\widehat{F}^{-1}_{G|(\boldsymbol{\delta},\mathbf{p})}(\cdot)$ to sample $G^{(r')}$. For r' values that are deep in the right tail (corresponding to the largest payoffs), however, we will use an approximation based on the normal distribution. Specifically, we choose the mean and variance of the normal distribution so that it has the same 99.0^{th} and 99.5^{th} percentiles as $\widehat{F}_{G|(\boldsymbol{\delta},\mathbf{p})}(\cdot)$; see [8]. We then use this normal distribution in place of $\widehat{F}$ in line 4 of the algorithm for values of r' that correspond to extreme percentiles.

Further efficiencies were obtained through the use of *splitting*. The high-level idea behind splitting is as follows. If a system is dependent on two random variables and it takes more time to sample the second variable but the first variable influences the system more, then one should generate multiple samples of the first variable for each sample of the second variable. In our context, $\mathbf{W}_{\text{op}}$ takes more time to sample but $\boldsymbol{\delta}$ appears to influence $G^{(r')}$ more. Accordingly, in our experiments we implemented splitting[39] with a ratio of 50:1 so that for each sample of $\mathbf{W}_{\text{op}}$ we generated 50 samples of $\boldsymbol{\delta}$.

B. Technical Details for Top-Heavy Formulation with $N > 1$

B.1. Parimutuel Betting Markets

Proof of Proposition 5.2

Let $\{n_h\}_{h=1}^{H}$ denote the output of Algorithm 5 and define $\mathbb{H} := \{(h,k) : 1 \le h \le H,\ 1 \le k \le n_h\}$. Note that $|\mathbb{H}| = N$. The expected reward of this wager allocation is $\sum_{(h,k)\in\mathbb{H}} \mu_h^k$. Let $\mathbb{H}^c$ denote the complement of $\mathbb{H}$ so that $\mathbb{H}^c := \{(h,k) : 1 \le h \le H,\ 1 \le k,\ (h,k) \notin \mathbb{H}\}$. By construction of our greedy algorithm (which follows the partial ordering described after (17)) we have

$$\mu_h^k \ge \mu_{h'}^{k'} \text{ for all } (h,k) \in \mathbb{H} \text{ and for all } (h',k') \in \mathbb{H}^c. \tag{26}$$

Consider now any alternative wager allocation $\{n_h^{\text{alt}}\}_{h=1}^{H}$ where $\sum_{h=1}^{H} n_h^{\text{alt}} = N$. Define $\mathbb{H}^{\text{alt}} := \{(h,k) : 1 \le h \le H,\ 1 \le k \le n_h^{\text{alt}}\}$ and note that $\mathbb{H}^{\text{alt}} = \mathbb{H}_1 \cup \mathbb{H}_2$ where $\mathbb{H}_1 := \mathbb{H}^{\text{alt}} \cap \mathbb{H}$ and $\mathbb{H}_2 := \mathbb{H}^{\text{alt}} \cap \mathbb{H}^c$. Since $\mathbb{H}_1 \cap \mathbb{H}_2 = \varnothing$ the expected reward of the alternative wager allocation can be written as

$$\begin{aligned}
\sum_{(h,k)\in\mathbb{H}^{\text{alt}}} \mu_h^k &= \sum_{(h,k)\in\mathbb{H}_1} \mu_h^k + \sum_{(h',k')\in\mathbb{H}_2} \mu_{h'}^{k'} \\
&= \sum_{(h,k)\in\mathbb{H}} \mu_h^k - \left(\sum_{(h,k)\in\mathbb{H}\setminus\mathbb{H}_1} \mu_h^k - \sum_{(h',k')\in\mathbb{H}_2} \mu_{h'}^{k'} \right) \qquad (27) \\
&\le \sum_{(h,k)\in\mathbb{H}} \mu_h^k. \qquad (28)
\end{aligned}$$

[39] We empirically tested several split ratios and found a ratio of 50:1 to perform best. See Chapter V of Asmussen and Glynn [2] for further details on splitting.

where (28) follows because the term in parentheses in (27) is non-negative which itself follows from (26) and the fact that $|\mathbb{H} \setminus \mathbb{H}_1| = |\mathbb{H}_2|$. (To see that $|\mathbb{H} \setminus \mathbb{H}_1| = |\mathbb{H}_2|$ observe that $|\mathbb{H} \setminus \mathbb{H}_1| = N - |\mathbb{H}_1|$ and $\mathbb{H}_2 = \mathbb{H}^{\text{alt}} \setminus \mathbb{H}_1$ so that $|\mathbb{H}_2| = |\mathbb{H}^{\text{alt}} \setminus \mathbb{H}_1| = N - |\mathbb{H}_1|$.) The result now follows. $\square$

B.2. Submodularity of Top-Heavy Objective Function

We first recall the definition of a submodular function.

Definition B.1. *Let Ω be a finite set and suppose $f : 2^\Omega \to \mathbb{R}$ is a function where 2^Ω denotes the power set of Ω. Suppose $f(X \cup \{x\}) - f(X) \geq f(Y \cup \{x\}) - f(Y)$ for every $X, Y \subseteq \Omega$ with $X \subseteq Y$ and every $x \in \Omega$. Then, f is submodular.*

We also need the following definition.

Definition B.2. *A $f : 2^\Omega \to \mathbb{R}$ is monotonic if $f(X) \leq f(Y)$ whenever $X \subseteq Y$.*

A classic result of Nemhauser et al. [30] for the maximization of monotonic submodular functions subject to a cardinality constraint is that the greedy algorithm returns a solution whose objective is within $1 - 1/e$ of the optimal objective.

Returning now to the top-heavy DFS contest, consider a given $(\boldsymbol{\delta}, \mathbf{W}_{\text{op}})$ realization and let $\mathbf{W}_1$ and $\mathbf{W}_2$ be two arbitrary choices of our portfolios of team selections with $\mathbf{W}_1 \subseteq \mathbf{W}_2 \subseteq \mathbb{W}$. Letting $\mathbf{w} \in \mathbb{W}$ denote any feasible team, we then have

$$f(\mathbf{W}_1 \cup \{\mathbf{w}\}) - f(\mathbf{W}_1) \geq f(\mathbf{W}_2 \cup \{\mathbf{w}\}) - f(\mathbf{W}_2) \tag{29}$$

where f is the total reward earned from the contest conditional on $(\boldsymbol{\delta}, \mathbf{W}_{\text{op}})$. It should be clear that (29) follows because $\mathbf{W}_1 \subseteq \mathbf{W}_2$ and so the gain from adding $\mathbf{w}$ to portfolio $\mathbf{W}_1$ (the l.h.s. of (29)) must be greater than the gain from adding $\mathbf{w}$ to portfolio $\mathbf{W}_2$ (the r.h.s. of (29)). Intuitively speaking, $\mathbf{W}_1$ competes less than $\mathbf{W}_2$ with $\mathbf{w}$. It follows then that the top-heavy reward is submodular for any given realization of $(\boldsymbol{\delta}, \mathbf{W}_{\text{op}})$ and since the expectation of a submodular function is submodular, it also follows that the objective function for the DFS contest is submodular. Since monotonicity follows trivially[40], the result of [30] applies and we are guaranteed that a greedy algorithm for the top-heavy contest with N entries performs within $1 - 1/e$ of optimality.

Unfortunately, we are unable to implement the greedy algorithm of [30] for the reasons outlined in Section 5.2.2. We also note that we did not use the top-heavy payoff structure of the contests to establish submodularity. It should therefore be clear the expected reward function for double-up contests is also submodular. This observation is less relevant for double-up contests, however, since it is intuitively clear that (barring pathological cases) when O is large, then replication is the optimal strategy. Indeed, as discussed in Section 4.3, we could also easily run a version of Algorithm 6 for double-up contests to confirm this.

B.3. Why Skill Matters More for Top-Heavy Contests

In the numerical experiments that we reported in Section 6, the performance of our strategic model was better in top-heavy contests than in double-up contests. While we would be reluctant to draw too many conclusions from this observation given the high variance of NFL games and the relatively few games in an NFL season, we do nonetheless believe skill is more important for top-heavy contests than for double-up contests. In fact, our numerical experiments also point to this. In particular, we report the optimal values of λ in Table 4 in Appendix D for the top-heavy, double-up and quintuple-up contests of Section 6. We see there that the top-heavy contests have a considerably higher value of λ^* than the double-up and quintuple-up

[40] We are assuming the decision to submit the N entries has already been made and so the cost of these entries is a sunk cost and excluded from the optimization.

contests whose values of λ^* are close to zero. This points to the fact that variance is important for top-heavy contests and that it plays a much smaller role for double-up contests. Moreover, because the variance of the fantasy points total includes the covariance term $-2\mathbf{w}^\top \boldsymbol{\sigma}_{\boldsymbol{\delta},G^{(r'_d)}}$ (see (14) for example), we know that our ability to estimate $\boldsymbol{\sigma}_{\boldsymbol{\delta},G^{(r'_d)}}$ is very important in determining the optimal entry $\mathbf{w}^*$ for top-heavy contests. This was not the case for the double-up or quintuple-up contests we played, however, since λ^* was close to zero for them. This then can be viewed as a "structural" explanation for why top-heavy contests are more amenable to skill than double-up contests. (Of course, if the cutoff point for rewards in double-up contests was very high, e.g. the top 5% or 1% of entries, then we'd expect variance to start playing a more important role for these contests as well.)

We can provide some further intuition regarding the relative importance of skill for top-heavy contests. Consider for example a stylized setting where there are two classes of DFS players: skilled and unskilled. We assume:

- A skilled player's fantasy points total is always greater than an unskilled player's fantasy points total.
- If two players are unskilled, then each one will outscore the other 50% of the time.
- Likewise, if two players are skilled, then each one will outscore the other 50% of the time.

Let α denote the proportion of opponents that are skilled and suppose $N = 1$ so we just submit a single entry. Moreover, an entry costs \$1 and all entry fees are returned as rewards, i.e., the organizers do not take a cut. Let μ_α^{w} and μ_α^{d} denote our expected rewards in the DFS winner-takes-all (WTA) and double-up contests, respectively. A double-up contest winner receives a reward of 2 while the WTA winner receives $O+1$ (assuming no ties) where as usual O denotes the number of opponents. Assuming we are skillful, then it's easy to see that

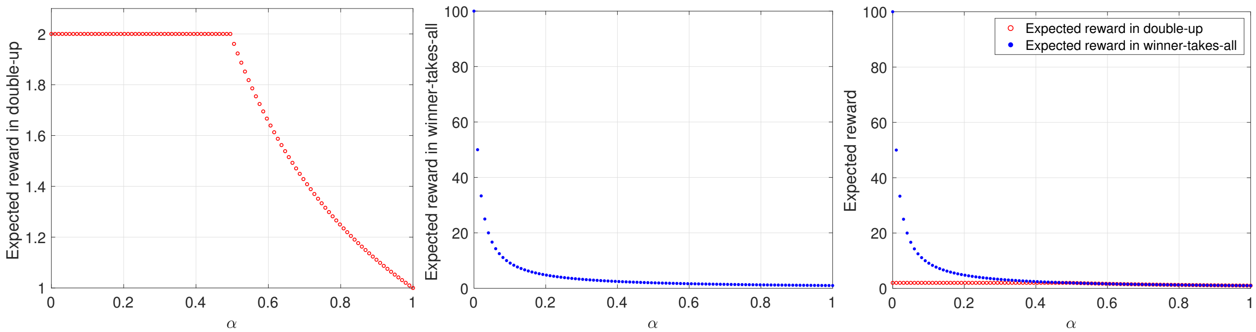
$$\mu_\alpha^{\mathrm{d}} = \min\left\{ \underbrace{2}_{\text{when } \alpha < 0.5}, \frac{O+1}{\alpha O + 1} \right\} \qquad \text{and} \qquad \mu_\alpha^{\mathrm{w}} = \frac{O+1}{\alpha O + 1}. \tag{30}$$

It follows immediately from (30) that $\mu_\alpha^{\mathrm{w}} \geq \mu_\alpha^{\mathrm{d}}$ for all $\alpha \in [0,1]$ so that a skilled player should prefer WTA over double-up. It also follows that $\mu_\alpha^{\mathrm{w}} - \mu_\alpha^{\mathrm{d}}$ is monotonically decreasing in α so that the preference of a skilled player for WTA is stronger when α is small. These results are not at all surprising of course: they follow because the reward in double-up contests is upper-bounded by 2 whereas there is no such bound in WTA contests. While this setting is very simple, it is straightforward to generalize it to top-heavy contests and more than one skill class. Figure 10 shows the extent of the preference for skilled players for WTA contests as a function of α.

C. Parameters and Other Inputs for the Numerical Experiments of Section 6

Our models rely on the following five input "parameters": the expected fantasy points of the real-world athletes $\boldsymbol{\mu}_{\boldsymbol{\delta}}$, the corresponding variance-covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\delta}}$, the stacking probability q from Section 3.2, the diversification parameter γ from Section 5.2 and the lower bound on the budget for accepting an opponent's portfolio B_{lb} from Section 3.3.

We obtain the estimate of $\boldsymbol{\mu}_{\boldsymbol{\delta}}$ from FantasyPros [15]. This estimate is specific to each week's games and we normally obtained it a day before the NFL games were played. We decompose the variance-covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\delta}}$ into the correlation matrix $\boldsymbol{\rho}_{\boldsymbol{\delta}}$ and the standard deviations of the individual athletes $\boldsymbol{\sigma}_{\boldsymbol{\delta}} \in \mathbb{R}^P$. The estimate of $\boldsymbol{\rho}_{\boldsymbol{\delta}}$ was obtained from RotoViz [36] and $\boldsymbol{\sigma}_{\boldsymbol{\delta}}$ is estimated using the realized $\boldsymbol{\delta}$ values from the 2016 and 2017 seasons. In particular, RotoViz provides correlations pegged to positions. For instance, using historical data, RotoViz has estimated the average correlation between the kicker of a team and the defense of the *opposing* team to be −0.50 and the average correlation between the kicker of a team and the defense of the *same* team to be 0.35. These estimates are not specific to any teams or athletes but are averages. (As



(a) Double-up (b) Winner-takes-all (c) Double-up and winner-takes-all

Figure 10: Expected reward in double-up and WTA as a function of α when there are $O = 99$ opponents. Clearly, $\mu_\alpha^{\mathrm{w}} - \mu_\alpha^{\mathrm{d}} \geq 0$ for all α and the difference is monotonically decreasing in α. Note the y-axis in Figure (a) is on a different scale to Figures (b) and (c).

a sanity check, we verified that the resulting correlation matrix $\boldsymbol{\rho_\delta}$ is positive semi-definite.) Hence, $\boldsymbol{\rho_\delta}$ does not change from week to week whereas $\boldsymbol{\sigma_\delta}$ is updated weekly using the realized $\boldsymbol{\delta}$ from the previous week.

It was also necessary to assume a distribution for $\boldsymbol{\delta}$ as we needed to generate samples of this random vector. We therefore simply assumed that $\boldsymbol{\delta} \sim \mathrm{MVN}_P(\boldsymbol{\mu_\delta}, \boldsymbol{\Sigma_\delta})$ where MVN_P denotes the P-dimensional multivariate normal distribution. Other distributions may have worked just as well (or better) as long as they had the same first and second moments, that is, the same $\boldsymbol{\mu_\delta}$ and $\boldsymbol{\Sigma_\delta}$.

We also needed the input features $\mathbf{X}$ and the realized $\mathbf{p}$ values for the Dirichlet regressions. Such data is available on the internet. For example, the $\mathbf{f}$ feature (point estimate of $\mathbf{p}$) was available at the FantasyPros website and FanDuel contains the cost vector $\mathbf{c}$ (before a contest starts) and the realized positional marginals $\mathbf{p}$ (after a contest is over). We note that accessing the positional marginals data at FantasyPros required us to create an account and pay for a six-month subscription costing \$65.94.

For the stacking probability q, we first note that we expect it to be contest-specific as we anticipate more stacking to occur in top-heavy style contests where variance is relatively more important than in double-up contests. Accordingly, we empirically checked the proportion of opponents who stacked using data[41] from the 2016-17 season for each contest-type. We then calibrated q to ensure that our Dirichlet-multinomial model for generating opponents implied the same proportion (on average). We estimated q to be 0.35, 0.25 and 0.20 for top-heavy, quintuple-up and double-up contests, respectively. In principle, one can perform out-of-sample testing to pick the "best" q in order to avoid in-sample over-fitting. However, given we are estimating a one-dimensional parameter using a reasonably large (and random) dataset, over-fitting was not our concern.

We set $B_{\mathrm{lb}} = 0.99B$ using a straightforward moment matching technique. In particular, we observed that most of our opponents used 100% of the budget and the average budget usage was around 99.5%. Using our Dirichlet-multinomial model for generating opponent portfolios, we simply calibrated B_{lb} so that the average budget usage was approximately 99.5%, which resulted in an estimate of $B_{\mathrm{lb}} = 0.99B$.

We used $\gamma = 6$ for the strategic and benchmark models across all contests since we found this value of γ to produce a near maximum within-model expected P&L. We note that the sensitivty of the expected P&L with respect to γ (around $\gamma = 6$) is relatively low in all contest types for both strategic and benchmark

[41] Because of the user interface of FanDuel.com, collecting entry-level data on each opponent was very challenging and had to be done manually. Instead of collecting data for each entry (which would have been too time consuming), we therefore collected data on 300 entries for each reward structure type. We also ensured the 300 entries were spread out in terms of their ranks so that they formed a representative sample of the entire population. For each contest type, we then estimated q by inspecting the 300 data-points and checking to see whether or not stacking of the QB and main WR was present.

portfolios. For instance, in the top-heavy contest with $N = 50$, the average weekly expected P&L (averaged over 17 weeks of the 2017-18 NFL season) for the strategic portfolio equals USD 342, 357, and 344 for γ equals 5, 6, and 7, respectively. Futhermore, if we allow γ to vary from week-to-week, i.e., in week t, pick $\gamma_t \in \{5, 6, 7\}$ that results in the maximum expected P&L in week t, then the average weekly expected P&L changes to USD 358 (an increase of only 1). This indicates the robustness of setting $\gamma = 6$.

We used a data sample from DFS contests in the 2016 NFL season to select an appropriate choice for Λ (the grid of λ values required for our optimization algorithms). In all of our contests, we set $\Lambda = (0.00, 0.01, \dots, 0.20)$. We could of course have reduced the computational burden by allowing Λ to be contest-specific. For example, in the case of quintuple-up contests, a choice of $\Lambda = (0.00, 0.01, \dots, 0.05)$ would probably have sufficed since λ^* for quintuple-up was usually close to zero as discussed in Appendix D.

All of our experiments were performed on a shared high-performance computing (HPC) cluster with 2.6 GHz Intel E5 processor cores. Each week, we first estimated the parameters $\mu_{G^{(r')}}$, $\sigma^2_{G^{(r')}}$ and $\boldsymbol{\sigma}_{\boldsymbol{\delta}, G^{(r')}}$ (as required by Algorithms 2 and 7) via Monte Carlo simulation. We typically ran the Monte-Carlo for one hour each week on just a single core and this was sufficient to obtain very accurate estimates of the parameters. We note there is considerable scope here for developing more sophisticated variance reduction algorithms which could prove very useful in practical settings when portfolios need to be re-optimized when significant late-breaking news arrives. In addition, it would of course also be easy to parallelize the Monte-Carlo by sharing the work across multiple cores.

The BQPs were solved using Gurobi's [20] default BQP solver and all problem instances were successfully solved to optimality with the required computation time varying with P (the number of real-world athletes), λ (see Algorithms 2 and 7) and the contest structure (double-up, quintuple-up or top-heavy). A typical BQP problem instance took anywhere from a fraction of a second to a few hundred seconds to solve. It was possible to parallelize with respect to λ and so we used 4 cores for double-up and quintuple-up contests and 8 cores for the top-heavy contests where the BQPs required more time to solve. Our experiments required up to 8 GB of RAM for double-up and quintuple-up contests and up to 16 GB of RAM for top-heavy contests.

D. Additional Results from the 2017 NFL Season

In Table 3, we display the performance of each week's best realized entry (out of the 50 that were submitted) for the strategic and benchmark models corresponding to the top-heavy contests for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season. Perhaps the most notable feature of Table 3 is the variability of our highest rank entry from week to week. This reflects the considerable uncertainty that is inherent to these contests. While the best strategic entry did well, we are confident that it could do much better (at least in expectation) by being more vigilant in updating parameter and feature estimates each week. In Table 4, we present various statistics of interest for the ex-ante optimal entry $\mathbf{w}_1^*$ of the strategic model across all three reward structures for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season. It is interesting to note that none of the numbers vary much from week to week. It is also interesting to see how the top-heavy entry $\mathbf{w}_1^*$ has a lower mean and higher standard deviation than the corresponding entries in the double-up and quintuple-up contests. This is not surprising and is reflected by the fact that the top-heavy contests have a higher value of λ^* than the double-up and quintuple-up contests. This is as expected since variance is clearly more desirable in top-heavy contests and the optimization over λ recognizes this. It is also interesting to see that λ^* for the quintuple-up contests is approximately 0.

Table 5 below shows the same results as Table 4 except this time for the benchmark portfolios. It is interesting to see how similar[42] the statistics are for all three contest-types in Table 5. Indeed these statistics are similar to the statistics for the double-up and quintuple-up strategic portfolios in Table 4 which is not

[42] Recall the benchmark quintuple-up model enforces stacking and hence, the optimal strategic quintuple-up entry might differ from the optimal benchmark portfolio even if $\lambda^* = 0$. There is no enforced stacking in the benchmark model for double-up and so if $\lambda^* = 0$ for the strategic double-up model, then the two entries will coincide and have same expected fantasy points.

Table 3: Performance of each week's best realized entry for the strategic and benchmark models corresponding to the top-heavy contests for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season.

	Total #	Rank		Percentile		Reward (USD)	
Week	**of entries**	Strategic	Benchmark	Strategic	Benchmark	Strategic	Benchmark
1	235,294	2,851	40,421	1.21%	17.18%	8	1.5
2	235,294	46,909	26,728	19.94%	11.36%	1.5	2
3	235,294	429	5,715	0.18%	2.43%	25	5
4	238,095	2,566	864	1.08%	0.36%	8	15
5	208,333	46,709	24,695	22.42%	11.85%	2	3
6	208,333	10,466	59,45	5.02%	2.85%	4	5
7	208,333	139	647	0.07%	0.31%	20	10
8	208,333	550	5,767	0.26%	2.77%	10	5
9	178,571	138	3,103	0.08%	1.74%	25	5
10	178,571	211	60,938	0.12%	34.13%	20	0
11	178,571	7,480	24	4.19%	0.01%	4	100
12	148,809	4,301	11,994	2.89%	8.06%	4	3
13	190,476	8,263	6,759	4.34%	3.55%	4	4
14	166,666	5,503	8,566	3.30%	5.14%	4	3
15	142,857	5,189	7,601	3.63%	5.32%	4	3
16	142,857	1,424	6,835	1.00%	4.78%	6	3
17	142,857	11,920	87	8.34%	0.06%	3	30

Table 4: Various statistics of interest for the ex-ante optimal entry $\mathbf{w}_1^*$ of the strategic model across all three reward structures for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season. Mean and StDev refer to the expected fantasy points and its standard deviation. (We were unable to participate in the quintuple-up contest in week 1 due to logistical reasons.)

	Top-heavy			Quintuple-up			Double-up		
Week	Mean	StDev	λ^*	Mean	StDev	λ^*	Mean	StDev	λ^*
1	124.45	24.76	0.03	-	-	-	127.35	20.00	0.00
2	120.70	26.94	0.05	124.22	23.69	0.01	123.58	21.50	0.01
3	115.08	27.54	0.04	121.15	22.34	0.00	120.54	21.03	0.01
4	114.18	27.54	0.04	121.85	21.67	0.00	121.85	21.67	0.00
5	115.48	23.65	0.05	123.22	20.49	0.00	123.22	20.49	0.00
6	106.45	27.96	0.05	118.82	21.37	0.00	118.28	17.53	0.02
7	108.69	29.82	0.06	120.53	22.08	0.00	119.34	21.02	0.02
8	107.61	28.26	0.04	120.73	20.22	0.00	120.61	20.11	0.01
9	105.16	28.52	0.05	116.42	21.83	0.01	115.94	19.48	0.01
10	110.25	28.99	0.05	123.36	21.49	0.00	122.74	19.42	0.02
11	107.79	29.43	0.04	123.28	20.88	0.00	122.44	19.88	0.02
12	117.60	25.47	0.03	124.90	19.72	0.00	124.90	19.72	0.00
13	116.70	29.30	0.03	123.10	22.65	0.00	122.20	19.44	0.02
14	111.50	28.15	0.04	119.70	20.68	0.00	119.40	19.33	0.01
15	116.80	27.79	0.06	129.30	19.30	0.00	129.30	19.30	0.00
16	117.60	26.38	0.04	122.20	23.40	0.01	120.90	16.96	0.02
17	110.70	27.68	0.07	126.80	19.02	0.00	126.40	17.29	0.01
Average	**113.34**	**27.54**	**0.05**	**122.47**	**21.30**	**0.00**	**122.29**	**19.66**	**0.01**

Table 5: Mean fantasy points and its standard deviation for the first optimal entry $\mathbf{w}_1^*$ of the benchmark model across all three reward structures for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season. (We were unable to participate in the quintuple-up contest in week 1 due to logistical reasons.)

	Top-heavy		**Quintuple-up**		**Double-up**	
Week	Mean	StDev	Mean	StDev	Mean	StDev
1	125.71	19.28	-	-	127.35	20.00
2	122.65	23.70	122.65	23.70	124.24	23.19
3	117.28	23.53	117.28	23.53	121.15	22.34
4	118.07	21.67	118.07	21.67	121.85	21.67
5	120.17	20.16	120.17	20.16	123.22	20.49
6	116.71	18.97	116.71	18.97	118.82	21.37
7	118.35	22.37	118.35	22.37	120.53	22.08
8	119.12	20.04	119.12	20.04	120.73	20.22
9	113.60	21.78	113.60	21.78	116.51	21.40
10	121.85	22.80	121.85	22.80	123.36	21.49
11	121.00	21.94	121.00	21.94	123.28	20.88
12	121.40	22.03	121.40	22.03	124.90	19.72
13	120.40	23.39	120.40	23.39	123.10	22.65
14	117.80	20.81	117.80	20.81	119.70	20.68
15	126.40	20.96	126.40	20.96	129.30	19.30
16	119.90	22.10	119.90	22.10	122.30	22.70
17	125.30	18.88	125.30	18.88	126.80	19.02
Average	**120.34**	**21.44**	**120.00**	**21.57**	**122.77**	**21.13**

surprising because the value of λ^* in those contests was close to 0. (We know that when λ^* is close to 0, then there is less value to being able to model opponents accurately. This merely reinforces the view that our strategic model adds more value in top-heavy style contests.) In Table 6, we present some information on the QB selected by the best performing entry of the strategic and benchmark models in the top-heavy contests for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season. It's clear that, on average, the strategic model picks less popular QBs than the benchmark model - an average $\mathbf{p}_{\mathrm{QB}}$ of 6.88% for the strategic model versus an average of 12.74% for the benchmark model. In addition, QBs picked by the strategic model cost less (approx. 3% lower on average) and have lower expected fantasy points (approx. 9% lower on average) than the QBs picked by the benchmark model. To put the cost numbers in perspective, the budget that was available for entry was set by the contest organizers at $B = 60,000$.

We end this appendix with two anecdotes highlighting top-heavy contests where our strategic portfolio went against the "crowd" and was successful in doing so. In week 3, our strategic model selected an entry that consisted of some crowd favorites, in particular Tom Brady (QB) and A.J. Green as one of the three WRs. The entry also included four underdog picks from the Minnesota Vikings: two WRs (S. Diggs and A. Thielen), the kicker K. Forbath and the defense. Each of these four picks were expected to be chosen by less than 5% of our opponents and by choosing four players from the same team, the entry was stacked which resulted in a reasonably high variance. The Minnesota Vikings ended up having a good game, winning 34-17 at home against Tampa Bay. Our entry ended up ranking 429^{th} out of 235,294 entries in total. In contrast, none of the benchmark entries were similar to this team. While some of them picked Thielen, none of them picked Diggs, Forbath or the Vikings defense and so there was no strategic stacking.

Another such example can be found in week 10. One of the entries selected by the strategic model included an underdog QB (C. Keenum) again from the Minnesota Vikings. Keenum was predicted to be chosen by fewer than 3% of our opponents. This could be explained by his low δ, i.e., his low expected fantasy points, and his low expected return. Choosing Keenum was quite a bold choice since the QB position is particularly important as QBs typically have the highest expected points and expected returns among all positions. In that particular week, Matthew Stafford was predicted to be the most popular QB with approx. 25% of opponents expected to pick him; see Figure 6(a). In addition to Keenum, our strategic entry was also

Table 6: Characteristics of the QB picked by the best performing entry of the strategic and benchmark models in the top-heavy contests for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season.

	Strategic				**Benchmark**			
Week	QB	$\mathbf{p}_{\text{QB}}$	Cost	$\boldsymbol{\mu}_{\boldsymbol{\delta}}$	QB	$\mathbf{p}_{\text{QB}}$	Cost	$\boldsymbol{\mu}_{\boldsymbol{\delta}}$
1	D. Carr (OAK)	9.80%	7700	18.63	R. Wilson (SEA)	7.30%	8000	20.23
2	A. Rodgers (GB)	9.80%	9100	26.19	M. Ryan (ATL)	10.90%	8200	24.28
3	T. Brady (NE)	7.20%	9400	20.72	A. Dalton (CIN)	2.40%	6800	15.85
4	R. Wilson (SEA)	13.20%	7900	21.19	A. Dalton (CIN)	3.50%	7100	17.19
5	J. Brissett (IND)	4.40%	7000	15.79	A. Rodgers (GB)	18.10%	9500	24.67
6	D. Carr (OAK)	1.10%	7500	16.48	D. Watson (HOU)	29.70%	7900	20.76
7	D. Brees (NO)	8.50%	8300	22.75	D. Brees (NO)	8.50%	8300	22.75
8	D. Watson (HOU)	3.70%	8000	17.30	A. Dalton (CIN)	9.70%	7600	19.02
9	R. Wilson (SEA)	16.30%	8500	24.52	R. Wilson (SEA)	16.30%	8500	24.52
10	C. Keenum (MIN)	0.70%	6800	15.50	B. Roethlisberger (PIT)	12.70%	7600	18.53
11	C. Keenum (MIN)	2.80%	7300	15.26	T. Brady (NE)	20.50%	8600	24.60
12	M. Ryan (ATL)	8.80%	7600	19.20	M. Ryan (ATL)	8.80%	7600	19.20
13	C. Keenum (MIN)	6.50%	7600	17.80	R. Wilson (SEA)	8.50%	8200	21.90
14	C. Keenum (MIN)	3.40%	7500	17.20	P. Rivers (LAC)	12.80%	8100	20.30
15	C. Keenum (MIN)	8.00%	7400	18.30	B. Roethlisberger (PIT)	13.70%	8000	21.10
16	A. Smith (KC)	7.50%	7800	19.20	C. Newton (CAR)	24.00%	8300	22.30
17	M. Ryan (ATL)	5.20%	7400	18.40	P. Rivers (LAC)	9.10%	8300	19.90
Average		**6.88%**	**7812**	**19.08**		**12.74%**	**8035**	**21.01**

stacked with 2 WRs (A. Thielen and S. Diggs) and the kicker (K. Forbath) all chosen from the Vikings and all predicted to be chosen by only approx. 5% of opponents. In the NFL game itself, the Vikings won 38-30 away to the Redskins with Keenum, Thielen, Diggs and Forbath scoring 26.06, 26.6, 15.8 and 10 fantasy points, respectively. Our entry ended up ranking 211^{th} out of 178,571 entries. In contrast, all 50 benchmark entries chose Ben Roethlisberger as the QB, who was considerably more popular than Keenum.

E. Model Checking and Goodness-of-Fit of the Dirichlet Regressions

In our numerical experiments of Section 6, we used the Dirichlet regression model of Section 3.1 to predict our opponents' behavior and in particular, the positional marginals $\mathbf{p}_{\text{QB}}, \ldots, \mathbf{p}_{\text{D}}$ of the players in our opponents' team selections. In Section 6.3, we discussed the performance of these Dirichlet regressions but not in a systematic fashion. In this appendix, we revisit this issue and evaluate the goodness-of-fit of our particular model and benchmark it against two simpler variations using the data from the 2017-18 NFL season. We first state the three variations of the Dirichlet regression model that we considered.

1. Variation 1 has 2 features: the player cost vector $\mathbf{c}$ and the expected points vector $\boldsymbol{\mu}_{\boldsymbol{\delta}}$.
2. Variation 2 has 1 feature: the point estimate $\mathbf{f}$ of the positional marginals from FantasyPros.
3. Variation 3 has 3 features: the player cost vector $\mathbf{c}$, the expected points vector $\boldsymbol{\mu}_{\boldsymbol{\delta}}$, and the point estimate $\mathbf{f}$ of the positional marginals from FantasyPros.

All three variations also include the constant vector $\mathbf{1}$ as an intercept. Variation 3 is therefore the model proposed in (5) and used in the numerical experiments of Section 6. Clearly, variations 1 and 2 are simpler versions of variation 3. We also note that we scaled the features to ensure they were roughly on the same scale as this helped `STAN` in fitting the model. In particular, we divided the costs by 10,000 and divided the expected points by 25 so that all the feature values were typically in $[0, 2]$.

Before proceeding, we note our Dirichlet regression model is very simple and that modern software packages such as `STAN` can fit such models within seconds. One could therefore easily include additional features as well as interaction / higher-order terms with the goal of increasing the predictive power of the model. Our goal here was not to find the best set of features, however, but simply to find features that explain the data reasonably well. As we will show later in this appendix, the three variations listed above all explain the data quite well while variation 3 performed best in the cross-validation tests. These results justified the use of variation 3 in our numerical experiments in Section 6. The plots and discussion in Section 6.3 also provide further support for the model. That said, the anecdotes from Section 6.3 (which are reflected in the results in Appendix E.2 below) suggest how the performance of the model could have been significantly improved had we focussed more on the correctness of the features particularly in the light of new player developments before the games.

Finally, we note that the model checking of Appendix E.2 and the cross-validation results of Appendix E.3 are standard Bayesian techniques and are discussed for example in Gelman et al. [18]. We used data from the 2017-18 NFL season for both of these tasks.

E.1. Data Collection

We were able to obtain complete data on the features $\{\mathbf{f}_{\mathrm{QB},t}, \mathbf{c}_{\mathrm{QB},t}, \boldsymbol{\mu}_{\mathrm{QB},t}\}_{t=1}^{T}$ where $T = 17$ was the number of weeks in the season. There was a minor issue with obtaining the realized positional marginals and to explain this issue, we will focus on the QB position whose realized marginals are $\{\mathbf{p}_{\mathrm{QB},t}\}_{t=1}^{T-1}$. Consider now week t with $\mathbf{p}_{\mathrm{QB},t} = \{p_{\mathrm{QB},t}^{k}\}_{k=1}^{P_{\mathrm{QB}}}$ and note that $\sum_{k=1}^{P_{\mathrm{QB}}} p_{\mathrm{QB},t}^{k} = 1$. If there were O opponents in a given contest in week t, then we would like to inspect their lineups to determine $\mathbf{p}_{\mathrm{QB},t}$. Unfortunately, there was no way to do this in an automated fashion and so we had to resort to sampling their lineups. Fortunately, however, if a particular QB appeared in a lineup, then the web-site listed the realized positional marginal for that particular QB in the underlying DFS contest. See Figure 11. As a result, it would only be necessary to sample lineups until each QB appeared at least once. Unfortunately, some QBs were selected very rarely if at all and sampling sufficient lineups to find them proved too time consuming. Instead, we typically sampled approx. 100 lineups each week and we let $\mathbb{C}_{\mathrm{QB},t} \subseteq \{1,\ldots,P_{\mathrm{QB}}\}$ denote the set of QBs for which we collect the marginal in week t. Since $\mathbb{C}_{\mathrm{QB},t}$ is a subset of $\{1,\ldots,P_{\mathrm{QB}}\}$, it follows that $\sum_{k\in\mathbb{C}_{\mathrm{QB},t}} p_{\mathrm{QB},t}^{k} \le 1$. Typical values of $\sum_{k\in\mathbb{C}_{\mathrm{QB},t}} p_{\mathrm{QB},t}^{k}$ were 95% to 99%.

We defined the vector of *collected* marginals for week t as $\hat{\mathbf{p}}_{\mathrm{QB},t} := [p_{\mathrm{QB},t}^{k}]_{k\in\mathbb{C}_{\mathrm{QB},t}}$ and the corresponding vector of FantasyPros estimates $\hat{\mathbf{f}}_{\mathrm{QB},t} := [f_{\mathrm{QB},t}^{k}]_{k\in\mathbb{C}_{\mathrm{QB},t}}$. Both of these vectors are then re-scaled so that they sum to 1. We similarly define $\hat{\mathbf{c}}_{\mathrm{QB},t} := [c_{\mathrm{QB},t}^{k}]_{k\in\mathbb{C}_{\mathrm{QB},t}}$ and $\hat{\boldsymbol{\mu}}_{\mathrm{QB},t} := [\mu_{\mathrm{QB},t}^{k}]_{k\in\mathbb{C}_{\mathrm{QB},t}}$ and then use these features to fit the three Dirichlet regression models using non-informative priors for $\boldsymbol{\beta}_{\mathrm{QB}}$; see (6). While this data collection procedure might introduce some bias in the estimation of $\boldsymbol{\beta}_{\mathrm{QB}}$, we expect this to be a second order effect as $\sum_{k\in\mathbb{C}_{\mathrm{QB},t}} p_{\mathrm{QB},t}^{k}$ and $\sum_{k\in\mathbb{C}_{\mathrm{QB},t}} \hat{\mathbf{f}}_{\mathrm{QB},t}$ were (before scaling) always close to 1. That said, further investigation of this may be worth pursuing.

E.2. Model Checking and Posterior Predictive Checks

The purpose of model checking and posterior predictive checks is to obtain a general idea of how well the model in question explains the data and what its weaknesses are. It may be viewed as a form of checking for internal consistency; see [18]. For each possible tuple of the form (model variation, reward structure, position), we fit a Bayesian Dirichlet regression model with `STAN`. We ran each of 4 MCMC chains for 1,000 iterations and then discarded the first 500 iterations from each one. This left us with 2,000 posterior samples of the corresponding β parameter. All of our $\hat{R}$ values[43] were between 1.00 and 1.05, indicating the MCMC chains had mixed well.

Marginal Posterior Predictive Checks

For each of the 2,000 posterior samples of β, we generated a sample of all of the positional marginals using the appropriate Dirichlet distribution and then used these samples to construct 95% posterior intervals for

[43] $\hat{R}$ is a commonly used metric to determine how well the Markov chains have mixed. The closer $\hat{R}$ is to 1 the better and a common rule of thumb is that an $\hat{R}$ between 1.00 and 1.05 indicates that the chains have mixed sufficiently well.

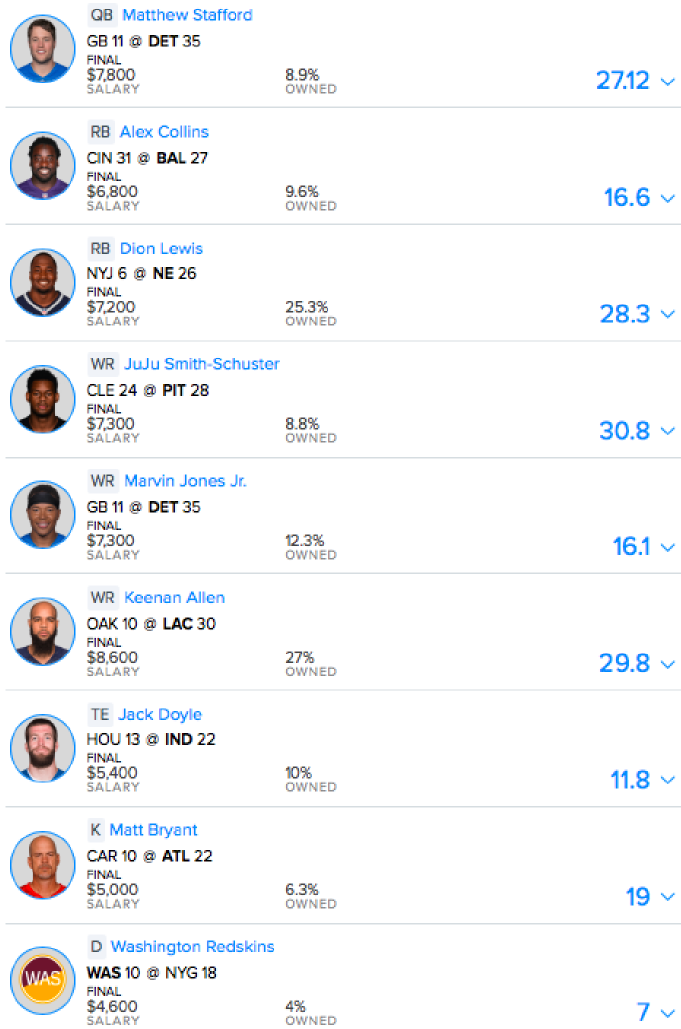


Figure 11: Screenshot of a web-page from `FanDuel.com` when we click on an opponent lineup. The lineup has 9 athletes. For each athlete selected in the lineup we can observe the realized positional marginal of that athlete in the underlying contest. For example in the contest corresponding to this screenshot the realized positional marginal of Matthew Stafford equals 8.9%.

the marginals. We then computed the proportion of times the 95% posterior intervals contain the true realized values. For each (variation, reward structure, position) tuple, we computed a summary statistic as follows.

As before, we will use the QB position to explain our procedure. There were $T = 17$ weeks and for each week t, there were $P_{\mathrm{QB},t}$ QBs available for selection. Hence there were $\sum_{t=1}^{T} P_{\mathrm{QB},t}$ QB "instances" in total and for each such instance, we know the true realized marginal from the real-world data that we used to fit the model. We also have the posterior samples for that instance and therefore a 95% posterior interval for it. If the realized marginal is in the 95% posterior interval, then we assign the instance the value 1. Otherwise we assign it the value 0. The summary statistic is then the average of these binary indicators over all $\sum_{t=1}^{T} P_{\mathrm{QB},t}$ instances. The summary statistic for each combination of model variation and reward structures is shown in Table 7 for the QB, RB, and WR positions[44]. The three model variations seem to pass this check (at least at the aggregate level) as each of the summary statistics lie between 93% and 97%.

[44]The results are similar for the other positions and are not shown for the sake of brevity.

Table 7: Posterior predictive test summary statistic for each variation (denoted by V1, V2, and V3) of the Dirichlet regression model across all reward structures corresponding to the QB, RB, and WR positions.

	QB			**RB**			**WR**		
	V1	V2	V3	V1	V2	V3	V1	V2	V3
Top-heavy	0.96	0.97	0.96	0.95	0.97	0.95	0.96	0.96	0.96
Quintuple-up	0.95	0.93	0.95	0.95	0.95	0.95	0.94	0.95	0.95
Double-up	0.93	0.93	0.94	0.94	0.95	0.95	0.94	0.94	0.94

Most-Picked Athlete Predictive Checks

We also computed predictive checks of the test quantity "most-picked athlete" for each combination of model variation, reward structure, position, and week. To see how these p-values were computed, consider a specific combination where the position (as usual) is the QB and the week is week t. Each posterior sample of $\mathbf{p}_{\text{QB},t}$ has a maximum value corresponding to the most popular QB that week in that posterior sample. We take all of these maximum values and compute the percentage of them that exceeded the *realized*[45] maximum. Ideally, the resulting percentile should be away from the extremes, i.e., 0 and 1. They are reported in Tables 8 (top-heavy) and 9 (double-up) below for the QB, RB and WR positions. We omitted the other positions and quintuple-up contests for the sake of brevity. Highlighted instances correspond to percentiles less than 2.5% (blue) or greater than 97.5% (red). While we would expect to see extreme values approx. 5% of the time even if the model in question was correct, we see such extreme values approx. 12.5% of the time for the top-heavy contests and 19.5% of the time for the double-up contests. While variation 3 does perform the best of the models on this test, there is clearly some room for improvement here. It is interesting to note that in the double-up contests, the extreme values (when they occur) are almost invariably on the low end, i.e., less than 2.5%. This means that in these instances, the Dirichlet regression model is predicting that the most popular player in the given position will be considerably less popular among opponents than the realized most popular player in that position.

There are two obvious directions for improving the model performance in light of these results. First of all, we have outlined in Section 6 some of our occasional failures to obtain accurate data for the features or to adjust features to account for relevant information that would be known to most DFS players and in particular, our DFS opponents. For example, in Section 6.3, we discussed the failure of the FantasyPros feature $\mathbf{f}$ to account for Russell Wilson's popularity in week 12 double-up – in part because it also failed to account for Tom Brady's well-known difficulties with playing in Miami. As depicted in Figure 7(b), Wilson's realized ownership that week was over 50% and so this was the realized maximum in week 12 for the QB position in double-up contests. Given our feature values that week and in particular the point estimate $\mathbf{f}$, our fitted models were generally unable to produce such a high value in the posterior samples of the most-picked QB. As a result, we observed the low values that we see for the QB position in week 12 in Table 9. In contrast, we can see from Figure 6 that we had no difficulty in predicting the popularity of the most popular QB in the week 10 double-up and top-heavy contests. It is not surprising then to see that the week 10 QB results in Tables 8 and 9 are not[46] extreme. It is no surprise then that our explanation for the less than perfect results here are explained by a combination of occasionally inaccurate feature data as well as possibly missing other useful features, e.g. momentum. It should be clear, however, that these are issues with the features and occasional accuracy of the features rather than a problem with our Dirichlet regression, which certainly seems to be the right way to model this problem.

[45] The realized maximum is the percentage of people who picked the most popular QB that week in that contest. So for example, if Matthew Stafford was the most popular QB in the week 10 top-heavy contest with 23% of contestants picking him, then the realized maximum for that week was 23%.

[46] While it is clear from Figure 6(b) that we severely underestimated the ownership of Ryan Fitzpatrick in week 10 double-up, this isn't reflected in Table 9 because we are focusing on the most popular player in that table and we did predict correctly a very popular player that week, i.e., Matthew Stafford.

Table 8: Bayesian p-values for the test statistic "most-picked athlete" for each variation of the Dirichlet regression model and each week corresponding to the QB, RB, and WR positions in the top-heavy reward structure.

	QB			**RB**			**WR**		
Week	V1	V2	V3	V1	V2	V3	V1	V2	V3
1	0.77	0.88	0.77	**0.98**	0.95	**0.99**	0.90	0.86	0.90
2	0.92	0.95	0.94	0.46	0.48	0.48	0.37	0.71	0.46
3	0.37	0.91	0.46	0.34	0.90	0.57	0.27	0.84	0.35
4	0.22	0.34	0.21	0.61	0.90	0.74	0.15	0.33	0.22
5	0.78	0.95	0.87	0.28	0.87	0.40	0.14	0.81	0.20
6	**0.01**	**0.02**	**0.01**	0.61	0.64	0.65	0.04	0.44	0.07
7	0.74	0.36	0.71	0.92	0.96	0.96	0.44	0.93	0.60
8	**0.00**	**0.02**	**0.00**	**0.98**	**0.98**	**0.99**	0.14	0.13	0.21
9	0.87	0.87	0.91	0.20	0.32	0.24	0.49	0.91	0.57
10	0.09	0.87	0.14	0.65	0.27	0.70	**1.00**	0.44	**1.00**
11	0.68	0.73	0.75	0.15	0.17	0.17	0.80	0.95	0.80
12	0.74	0.80	0.82	0.84	**1.00**	0.89	0.94	0.84	0.89
13	0.30	0.51	0.39	0.43	0.83	0.44	0.06	0.16	0.06
14	0.07	0.21	0.09	0.72	0.63	0.70	0.12	**0.98**	0.16
15	0.96	0.96	0.96	0.80	0.89	0.88	**1.00**	**1.00**	**1.00**
16	0.03	0.22	0.04	0.11	0.33	0.11	0.35	0.77	0.46
17	0.78	0.78	0.77	**0.99**	0.29	0.97	0.09	0.07	0.09

Table 9: Bayesian p-values for the test statistic "most-picked athlete" for each variation of the Dirichlet regression model and each week corresponding to the QB, RB, and WR positions in the double-up reward structure.

	QB			**RB**			**WR**		
Week	V1	V2	V3	V1	V2	V3	V1	V2	V3
1	0.41	0.49	0.40	0.07	0.54	0.59	**0.02**	0.08	0.07
2	0.15	0.40	0.35	0.04	**0.02**	0.03	**0.01**	0.08	0.06
3	0.12	0.22	0.11	0.05	0.72	0.65	**0.00**	0.09	0.07
4	0.18	0.15	0.18	0.03	0.15	0.11	0.07	0.13	0.11
5	0.03	0.21	0.06	0.03	0.65	0.39	0.15	0.67	0.58
6	**0.00**	**0.00**	**0.00**	0.36	0.45	0.33	**0.02**	0.12	0.04
7	0.64	0.55	0.53	0.09	0.24	0.15	0.32	0.73	0.47
8	0.25	0.18	0.27	0.29	0.46	0.48	0.03	**0.02**	**0.02**
9	0.80	0.66	0.86	0.05	0.04	0.05	0.19	0.25	0.24
10	0.27	0.70	0.54	0.03	0.03	0.05	0.58	0.11	0.13
11	**0.01**	**0.00**	**0.01**	**0.02**	**0.02**	**0.02**	0.36	0.36	0.32
12	**0.01**	0.06	0.06	0.07	0.93	0.88	0.24	0.21	0.22
13	0.04	0.11	0.07	0.09	0.37	0.27	0.04	0.08	0.07
14	0.08	0.09	0.06	**0.01**	**0.01**	**0.01**	**0.01**	0.47	0.25
15	0.41	0.43	0.56	0.09	0.44	0.44	0.97	**1.00**	**1.00**
16	**0.00**	**0.00**	**0.00**	**0.02**	0.05	0.03	0.32	0.58	0.43
17	0.30	0.22	0.28	0.14	0.05	0.13	**0.00**	**0.01**	**0.00**

Table 10: Comparing the three variations of the Dirichlet regression model using normalized cross-validation scores for each position and each reward structure.

	Top-heavy			Quintuple-up			Double-up		
Position	V1	V2	V3	V1	V2	V3	V1	V2	V3
D	**1.0000**	0.9376	0.9822	**1.0000**	0.9910	0.9998	0.9961	0.9895	**1.0000**
K	0.9700	0.9565	**1.0000**	0.9747	0.9843	**1.0000**	0.9649	0.9863	**1.0000**
QB	0.9998	0.9299	**1.0000**	0.9852	0.9808	**1.0000**	0.9708	0.9792	**1.0000**
RB	0.9978	0.9260	**1.0000**	0.9737	0.9878	**1.0000**	0.9607	0.9977	**1.0000**
TE	0.9983	0.9023	**1.0000**	0.9693	0.9655	**1.0000**	0.9744	0.9682	**1.0000**
WR	**1.0000**	0.9593	0.9911	0.9870	**1.0000**	0.9832	0.9806	**1.0000**	0.9887

E.3. Cross-Validation

In order to compare the models in terms of out-of-sample performance, we perform leave-one-out cross-validation. For each combination of (model variation, reward structure, position), we do the following. We pick 16 weeks (the training set) out of the 17 available. We then fit the model on the data from those 16 weeks and use it to generate posterior samples of β. We then compute the log-likelihood on the data for the holdout week (the test set). We repeat this 17 times, each time with a different holdout week, and sum the 17 log-likelihoods to get a "raw" cross-validation score. See Chapter 7 (page 175) of Gelman et al. [18] for further details.

The results are displayed in Table 10 for all positions across all variations and reward structures. In the table we report a "normalized" cross-validation score to make the results easier to interpret. Consider the QB position and the top-heavy reward structure for example. The "raw" cross-validation scores are 742.71, 690.79, and 742.88 for variations 1, 2, and 3, respectively and we normalize these scores by simply dividing across by their maximum. Hence, a normalized score of 1 denotes the "winner" and this winner is displayed in bold font in Table 10. Variation 3 clearly performs the best among the three models.