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

April 17, 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 the literature on 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. 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 (i) insider trading and (ii) collusion whereby a number of DFS players combine to construct a single portfolio of entries to a given contest. We demonstrate the value of our framework by applying it to both double-up and top-heavy DFS contests in the 2017 NFL season.
1. Introduction

Daily Fantasy Sports (DFS) has become a multi-billion dollar industry \cite{27, 23, 32, 1, 34} with millions of annual users \cite{17, 32}. 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 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 \cite{27, 23}. Approximately 80% of DFS players have been classified as *minnows* \cite{28} 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 \cite{28, 33, 25, 20} 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 portfolios then earn a monetary reward which depends on the specific payoff structure, e.g. *double-up* or *top-heavy*, of the DFS contest.

Multiple papers have already been written on the topic of fantasy sports. For example, Fry, Lundberg, and Ohlmann \cite{16} and Becker and Sun \cite{5} develop models for season-long fantasy contests while Bergman and Imbrogno \cite{6} propose strategies for the survivor pool contest which is also a season long event. Several 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 \cite{22} and Clair and Letscher \cite{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. \cite{21} 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. \cite{11}, and the authors report earning significant sums in real fantasy sports contests based on the National Hockey League (NHL) and baseball.

There are several directions for potential improvement, however, and they are the focus of the work in this paper. First, Hunter et al. \cite{21} 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 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

\footnote{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.}
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 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 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 colluding in DFS contests. Specifically, we consider the case where a number $N_{collude}$ of DFS players combine to construct a single portfolio of
$N_{\text{collude}} \times E_{\text{max}}$ entries for a given contest where $E_{\text{max}}$ is the maximum number of permitted entries per DFS player. We show that the benefits 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. Finally, we mention in passing that our model adds to the ongoing “skill-versus-luck” debate surrounding online gambling\(^2\) in the United States. It should be clear from our framework and the corresponding statistical estimation problems that high levels of “skill” are required to play fantasy sports successfully.

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 and 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 here. We conclude in Section 8 where we also discuss some directions for ongoing and future research. Some technical details and results are deferred to Appendix A.

2. Problem Formulation

We assume there are a total of $P$ athletes / real-word players whose performance, $\delta \in \mathbb{R}^P$, in a given round of games is random. We assume that $\delta$ has mean vector $\mu_\delta$ and variance-covariance matrix $\Sigma_\delta$. Our decision in the fantasy sports competition is to choose a portfolio $w \in \{0, 1\}^P$ of athletes. Typically, there are many constraints on $w$. 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 $w$ can then be formulated as

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

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 $w \in \{0, 1\}^P$ 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 $W_o = \{w_o\}_{o=1}^{O}$ to denote their portfolios with each $w_o \in \mathbb{W}$.

Once the round of NFL games has taken place, we get to observe the realized performances $\delta$ of the $P$ NFL athletes. Our portfolio then realizes a points total of $F := w^T \delta$ whereas our opponents’ realized points totals are $G_o := w_o^T \delta$ for $o = 1, \ldots, O$. All portfolios are then ranked according to their points total and the

\(^2\)Currently, online gambling is essentially illegal in the U.S. with exceptions made for some activities such as fantasy sports on the basis that there is some skill associated with them. Nonetheless this continues to be a contentious issue and some U.S. states currently outlaw the playing of fantasy sports. A quick internet search on the topic will reveal just how topical this debate is.
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 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

\[
\max_{\mathbf{w} \in \mathcal{W}} \mathbb{P}\left\{ \mathbf{w}^T \delta > G^{(r)}(\mathbf{W}_{op}, \delta) \right\},
\]

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 \( \mathbf{W}_{op} \) of our \( O \) opponents and the performance vector \( \delta \) 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\(^4\) for the possibility of submitting \( N > 1 \) entries to the contest. We use \( \mathbf{W} = \{\mathbf{w}_j\}_{j=1}^{N} \) to denote these entries and \( F_i := \mathbf{w}_i^T \delta \) 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

\[
\max_{\mathbf{w} \in \mathcal{W}} \sum_{i=1}^{N} \sum_{d=1}^{D} (R_d - R_{d+1}) \mathbb{P}\left\{ \mathbf{w}_i^T \delta > G_{-i}^{(r'_d)}(\mathbf{W}_{-i}, \mathbf{W}_{op}, \delta) \right\}
\]

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 \( \mathbf{W}_{-i} := \mathbf{W} \setminus \mathbf{w}_i \). We note that the top-heavy payoff structure is our main concern in this paper. That said, it should be clear that the

\(^3\)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 \( E_{max} = 150 \).

\(^4\)Our reasons for this will be clear later in Section 5.
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 \( W_{\text{op}} \).

3. Modeling Opponents’ Team Selections

A key aspect of our modeling approach is that there is value to modeling our opponents’ portfolio choices, \( W_{\text{op}} \). This is in direct contrast to the work of Hunter et al. [21] 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\(^5\) 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 \( W_{\text{op}} \). This is due to the well-known fact that some athletes are (often considerably) more 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 \( W_{\text{op}} \) 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 [18].

If we are to exploit our opponents’ team selections, then we must be able to predict \( W_{\text{op}} \) reasonably accurately. Indeed it is worth emphasizing that \( W_{\text{op}} \) is not observed before the contest and so we must make do with predicting it. In most fantasy sports contests, however, it is possible to learn about \( W_{\text{op}} \) after the contest is over and the winners have been announced. We therefore assume we have access to partial data on \( W_{\text{op}} \) from previous contests. In particular, we make the reasonable assumption that we have access to sufficient data to estimate the 1-dimensional marginal distributions\(^6\) of \( w_o \in \{0,1\}^P \), a random opponent’s portfolio. We will also assume we have access to other observable features, e.g. expected NFL player performance \( \mu_k \), home or away indicators, quality of opposing teams etc. from these previous contests. We will use this information to build a Dirichlet regression model for estimating the marginal distributions of \( w_o \). Towards this end, we must first describe our regression model and then in Section 3.2, we put everything together to (implicitly) define the distribution of \( W_{\text{op}} \).

3.1. Dirichlet Regression

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). Consider now the marginal distribution of the QB selections in \( w_o \). To begin, we assume there is a total of \( N_{\text{QB}} \) QB’s available for selection and we number\(^7\) them from \( k = 1 \) to \( k = N_{\text{QB}} \). We assume a Dirichlet-multinomial data generating process for a random opponent’s QB selection. Specifically, we assume:

- \( p_{\text{QB}} \sim \text{Dir}(\alpha_{\text{QB}}) \) where \( \text{Dir}(\alpha_{\text{QB}}) \) denotes the Dirichlet distribution with parameter vector \( \alpha_{\text{QB}} \).
- A random opponent then selects QB \( k \) with probability \( p_{\text{QB}}^k \) for \( k = 1, \ldots, N_{\text{QB}} \).

\(^5\)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.

\(^6\)Sometimes we can only get to observe \( W_{\text{op}} \) after a contest via web-scraping, something that is not very desirable and that is frowned upon by some sites.

\(^7\)Recall that \( w_o \in \{0,1\}^P \) where \( P \) is the total number of NFL athletes available for selection. We can assume then that the first \( N_{\text{QB}} < P \) components of \( w_o \) correspond to the \( N_{\text{QB}} \) QB’s and so a feasible portfolio will set exactly one of these components to 1 with the remaining \( N_{\text{QB}} - 1 \) components set to 0. The QB component set to 1 corresponds of course to the QB that is selected in the portfolio.
Note \( p_{QB} := \{p_{QB}^k\}_{k=1}^{N_{QB}} \) lies on the unit simplex in \( \mathbb{R}^{N_{QB}} \) and therefore defines a probability distribution over the available quarterbacks. It is important to note that the selection probabilities \( p_{QB} \) are not known in advance of the DFS contest. Moreover, they do not appear to be perfectly predictable and so we have to explicitly model\(^8\) their randomness. We do this by assuming instead that the parameter vector \( \alpha_{QB} \in \mathbb{R}^{N_{QB}} \) is predictable. In particular, we assume
\[
\alpha_{QB} = \exp(X_{QB} \beta_{QB})
\]
where \( \beta_{QB} \) is a vector of parameters that we must estimate and \( X_{QB} \) is a matrix (containing \( N_{QB} \) rows) of observable independent variables that are related to the specific features of the NFL games and QB’s underlying the DFS contest. To be clear, the exponential function in the r.h.s. of (3) is actually an \( N_{QB} \times 1 \) vector of exponentials.

For example, in a DFS contest for week \( t \) we might assume
\[
\alpha_{QB,t} = \exp(\beta_{QB}^0 \mathbf{1} + \beta_{QB}^1 \mathbf{f}_{QB,t} + \beta_{QB}^2 \mathbf{c}_{QB,t} + \beta_{QB}^3 \mathbf{\mu}_{QB,t})
\]
where \( \mathbf{f}_{QB,t} \) is an estimate of \( p_{QB} \) for week \( t \) that we can obtain from the FantasyPros website [15], \( \mathbf{c}_{QB,t} \) are the (appropriately scaled) week \( t \) costs of the QB’s in the contest, and \( \mathbf{\mu}_{QB,t} \) is an (appropriately scaled) sub-vector of \( \mathbf{\mu}_F \) for week \( t \) whose components correspond to the QB positions in \( \mathbf{\mu}_F \). Other features are of course also possible. For example, we might also want to include expected returns \( \mathbf{\mu}_{QB,t}/\mathbf{c}_{QB,t} \) (where division is understood to be component-wise), home-away indicators, quality of opponents etc. as features. That said, these latter features should already be accounted for by the features included in (4).

We can estimate the \( \beta_{QB} \) vector by fitting a Bayesian Dirichlet regression. Assuming we have data from weeks \( t = 1 \) to \( t = T - 1 \) and a flat prior on \( \beta_{QB} \), then the posterior satisfies
\[
p(\beta_{QB} \mid \{\alpha_{QB,t}\}_{t=1}^{T-1}) \propto \frac{p(\beta_{QB})}{\alpha_{QB}} p(\alpha_{QB}) \prod_{t=1}^{T-1} \text{Dir}(\alpha_{QB,t} | \alpha_{QB})
\]
\[
\propto \prod_{t=1}^{T-1} \frac{1}{\text{B}(\alpha_{QB,t})} \prod_{k=1}^{N_{QB}} \left( \mathbf{p}_{QB,k,t} \right)^{\alpha_{QB,t,k} - 1}
\]
where \( \text{B}(\alpha_{QB,t}) \) is the normalization factor for the Dirichlet distribution. We fit this model using the Bayesian software package STAN [31].

3.2. Generating Random Opponents’ Portfolios
Suppose now that the Dirichlet regression model has been fit for each position, that is, the \( \alpha_{QB}, \alpha_{RB}, \alpha_{TE} \) etc. vectors have all been estimated. The question arises as to how we can use them to define a random opponent’s portfolio \( \mathbf{w}_o \). We note that the fitted Dirichlet regression models provide a straightforward mechanism for defining the \textit{positional marginals} of \( \mathbf{w}_o \). For example, let \( \mathbf{w}_o^{QB} \) denote the \( N_{QB} \times 1 \) sub-vector of \( \mathbf{w}_o \) corresponding to the QB’s available for selection. We can easily generate a sample of this vector by:

(i) First drawing a sample \( \mathbf{p}_{QB} \) from the Dir(\( \alpha_{QB} \)) distribution.

(ii) Then drawing a single sample from the Mult(\( \mathbf{p}_{QB} \)) distribution, that is, multinomial distribution with parameter vector \( \mathbf{p}_{QB} \).

\(^8\)In initial unreported experiments, we assumed \( \mathbf{p}_{QB} \) was fixed and known but this led to over-certainty and poor performance of the resulting portfolios.
(iii) This draw then defines our chosen QB, that is, it sets one component of $w^\text{QB}_o$ to 1 with the others being set to 0.

The obvious approach to defining the distribution of $w_o$ is to assume that each positional marginal vector is independent and therefore drawn according to the three steps outlined above with the obvious understanding that $\alpha_{\text{RB}}$ would be used for the RB positions, $\alpha_{\text{TE}}$ for the TE position etc. This marginal independence assumption then defines the distribution of a potential $w_o$. There is no guarantee, however, that the resulting portfolio is feasible, that is, $w_o \in \mathbb{W}$. We therefore use an accept-reject approach whereby potential portfolios $w_o$ are generated according to the steps outlined above and are only accepted if they are feasible, that is, if $w_o \in \mathbb{W}$. In fact, we impose one further condition: we insist that an accepted $w_o$ 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 his budget unspent. This is purely a behavioral requirement and so we insist the cost of an accepted $w_o$ satisfy $c^\top w_o \geq B_{lb}$ for some lower bound $B_{lb} \leq B$ that we get to choose. Algorithm 1 below describes how to generate $O$ random opponents’ portfolios $W_{op}$ and it therefore (implicitly) defines the distribution of $W_{op}$. (We use Mult($\cdot$, 1) to denote a single draw from the multinomial distribution.)

**Algorithm 1 Sampling $O$ Opponent Portfolios**

Require: $(\beta_{\text{QB}}, \ldots, \beta_{D}), (X_{\text{QB}}, \ldots, X_{D}), c, B_{lb}$

1: $(\alpha_{\text{QB}}, \ldots, \alpha_{D}) = (\exp(X_{\text{QB}} \beta_{\text{QB}}), \exp(X_{D} \beta_{D}))$

2: $(p_{\text{QB}}, \ldots, p_{D}) \sim (\text{Dir}(\alpha_{\text{QB}}), \ldots, \text{Dir}(\alpha_{D}))$

3: for $o = 1 : O$ do

4: reject = true

5: while reject do

6: $(k_{\text{QB}}, \ldots, k_{D}) \sim (\text{Mult}(p_{\text{QB}}, 1), \ldots, \text{Mult}(p_{D}, 1))$

7: Let $w_o$ denote the portfolio corresponding to $(k_{\text{QB}}, \ldots, k_{D})$

8: if $w_o \in \mathbb{W}$ and $c^\top w_o \geq B_{lb}$ then

9: reject = false

10: end if

11: end while

12: end for

13: return $W_{op} = \{w_o\}_o^O$

Algorithm 1 describes the case where the positional marginals within each opponent portfolio $w_o$ are independent. It is well known, however, that some sophisticated DFS players intentionally select their portfolios so that the positional marginals of $w_o$ are *not* independent. We are referring here to the phenomenon of so-called *stacking* where a portfolio is chosen with a view to maximizing its points variance (as well as its points mean). For example, if a DFS player chooses a QB from a particular team, then it is quite common for the DFS player to also choose the WR from the same team [3]. This is because the realized points of a QB and WR from the same team tend to be strongly correlated and picking the two together will result in a fantasy portfolio with a higher variance (than would otherwise be the case if the chosen QB and WR were from different NFL teams).

We mention here that it is easy to adapt Algorithm 1 to model this stacking behavior and indeed we do this in the numerical experiments of Section 6. For example, once the QB has been chosen, we could assume that the WR from the same team is then chosen with some known probability $q$. Otherwise, the WR is chosen as before, that is, as a draw from the relevant multinomial distribution.
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 some results from mean-variance optimization and in particular, the problem of maximizing the probability of exceeding a stochastic benchmark.

4.1. Mean Variance Optimization and Outperforming Stochastic Benchmarks

We consider 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 \( \xi = (\xi_1, \ldots, \xi_P) \) is realized. Let \( \mu_\xi \) and \( \Sigma_\xi \) denote the mean return vector and variance-covariance matrix, respectively, of \( \xi \). The goal is then to construct a portfolio \( w = (w_1, \ldots, w_P) \) with random return \( R_w = w^T \xi \) that maximizes the probability of exceeding a random benchmark \( R_b \). Mathematically, we wish to solve

\[
\max_{w \in W} \mathbb{P}(R_w - R_b \geq 0) \tag{5}
\]

where \( W \) includes the budget constraint \( w^T 1 = 1 \) as well as any other linear constraints we wish to impose. We assume that \( R_w - R_b \) has a multivariate normal distribution so that

\[
R_w - R_b \sim N(\mu_w, \sigma_w^2)
\]

for some \( \mu_w \) and \( \sigma_w^2 \) that depend on the portfolio \( w \). Then, the problem in (5) amounts to solving

\[
\max_{w \in W} 1 - \Phi\left( \frac{-\mu_w}{\sigma_w} \right) \tag{6}
\]

where \( \Phi(\cdot) \) denotes the standard normal CDF. Let \( w^* \) be the optimal solution to (6). The following result is adapted from Morton et al. [24] and follows from the representation in (6).

Proposition 4.1. (i) Suppose \( \mu_w < 0 \) for all \( w \in W \). Then

\[
w^* \in \left\{ w(\lambda) : w(\lambda) \in \arg \max_{w \in W} (\mu_w + \lambda \sigma_w^2), \ \lambda \geq 0 \right\}. \tag{7}
\]

(ii) Suppose \( \mu_w \geq 0 \) for some \( w \in W \). Then

\[
w^* \in \left\{ w(\lambda) : w(\lambda) \in \arg \max_{w \in W, \mu_w \geq 0} (\mu_w - \lambda \sigma_w^2), \ \lambda \geq 0 \right\}. \tag{8}
\]

so that \( w^* \) is mean-variance efficient.

Proposition 4.1 is useful because it allows us to solve the problem in (6) efficiently. In particular, we determine which of the two cases from the proposition applies. This can be done when \( W \) is polyhedral by simply solving a linear program that maximizes \( \mu_w \) (which is affine in \( w \)) over \( w \in W \). If the optimal mean is negative, then we are in case (i); otherwise we are in case (ii). We then form a grid \( \Lambda \) of possible \( \lambda \) values and for each \( \lambda \in \Lambda \), we solve the appropriate quadratic optimization problem (defining \( w(\lambda) \)) from (7) or (8) and then choose the value of \( \lambda \) that yields the largest objective in (5) or (6). See Algorithm 2 in Section 4.2 below for when we apply these results to our double-up problem.

---

9 The material and results in this subsection follow Morton et al. [24] 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.

10 If the benchmark \( R_b \) is deterministic, then \( \mu_w := w^T \mu_\xi - R_b \) and \( \sigma_w^2 := w^T \Sigma_\xi w \).
4.2. The Double-Up Problem

Recall now the double-up problem as formulated in (1). We define $Y_w := \mathbf{w}^T \delta - G(r')$ and assume\footnote{We justified the normal approximation for $Y_w$ via Monte Carlo simulation. Specifically, we generated many samples of $Y_w$ corresponding to various portfolios $\mathbf{w}$. These portfolios included $\mathbf{w}^*$ as well as highly ranked portfolios from DFS contests in the current NFL season. In each case, we found the distribution of $Y_w$ to be unimodal and approximately symmetric about its mean. Regardless, we note that the mean-variance framework we develop here can still be used even if the assumption of $Y_w$ being normally distributed was not fully justified. Moreover, we are free in lines 10 and 4 of Algorithms 2 and 3, respectively, to use a more accurate probability distribution if we so desire.} that $Y_w \sim N(\mu_{Y_w}, \sigma_{Y_w}^2)$ where

$$
\mu_{Y_w} := \mathbf{w}^T \mu_{\delta} - \mu_{G(r')}
$$

$$
\sigma_{Y_w}^2 := \mathbf{w}^T \Sigma_{\delta} \mathbf{w} + \sigma_{G(r')}^2 + 2\mathbf{w}^T \sigma_{\delta, G(r')} \lambda
$$

(9)

where $\mu_{G(r')} = \mathbb{E}[G(r')]$,  $\sigma_{G(r')}^2 = \text{Var}(G(r'))$ and $\sigma_{\delta, G(r')}$ is a $P \times 1$ vector with $p^{th}$ component equal to $\text{Cov}(\delta_p, G(r'))$. The solution of our double-up problem now follows immediately from Proposition 4.1 and the following discussion. Our procedure is stated formally in Algorithm 2 below.

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

**Require:** $\mathcal{W}$, $\Lambda$, $\mu_{\delta}$, $\Sigma_{\delta}$, $\mu_{G(r')}$, $\sigma_{G(r')}^2$, $\sigma_{\delta, G(r')}$ and Monte Carlo samples of $(\delta, G(r'))$

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

**Remark 4.1.** Note that $\lambda^*$ in line 10 can be computed using the Monte Carlo samples of $(\delta, G(r'))$ that are inputs to the algorithm. Alternatively, $\lambda^*$ could also be computed using the normal approximation assumption.

**Generating Monte Carlo Samples**

In order to execute Algorithm 2, we must first compute the inputs $\mu_{G(r')}$, $\sigma_{G(r')}^2$ and $\sigma_{\delta, G(r')}$ 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.

**Solving the Binary Quadratic Programs**

The optimization problems in lines 3 and 7 of Algorithm 2 require the solution of binary quadratic programs (BQP’s). In our numerical experiments of Sections 6 and 7, we solved these BQP’s using Gurobi’s [19] 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.)
4.3. The Double-Up Problem with Multiple Entries

We briefly consider now 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 should be intuitively clear\(^\text{12}\) that if \( O \to \infty \) so that \( N/O \to 0 \), then the optimal strategy is a *replication* strategy. In particular, the DFS player should solve for the \( N = 1 \) optimal portfolio and then submit \( N \) copies of this portfolio to the contest. Even when \( O \) is not large, we suspect the replication strategy may 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.

Finally, we note that a risk-averse DFS player would never (ignoring certain pathological cases) want to use a replication strategy and would prefer instead a more diversified portfolio of entries. Such a diversified portfolio can easily be constructed using the heuristics we outline in Section 5.1 for the case of top-heavy contests.

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

\[
\max_{\text{w} \in \mathbb{W}} \sum_{d=1}^{D} (R_d - R_{d+1}) \mathbb{P} \left\{ w^\top \delta > G(r'_d) (W_{\text{op}}, \delta) \right\}, \tag{10}
\]

where \( r'_d = O + 1 - r_d \). Following the development in Section 4.2, we can define \( Y_d^w := w^\top \delta - G(r'_d) \) and assume \( Y_d^w \sim N(\mu_{Y_d^w}, \sigma_{Y_d^w}^2) \) with

\[
\mu_{Y_d^w} := w^\top \mu_{G(r'_d)} \tag{11}
\]

\[
\sigma_{Y_d^w}^2 := w^\top \Sigma_{\delta} w + \sigma_{G(r'_d)}^2 - 2w^\top \sigma_{\delta, G(r'_d)} \tag{12}
\]

where \( \mu_{G(r'_d)} = \mathbb{E}[G(r'_d)] \), \( \sigma_{G(r'_d)}^2 = \text{Var}(G(r'_d)) \) and \( \sigma_{\delta, G(r'_d)} \) is a \( P \times 1 \) vector with \( p \)-th component equal to \( \text{Cov}(\delta_p, G(r'_d)) \). We can now write (10) as

\[
\max_{\text{w} \in \mathbb{W}} \sum_{d=1}^{D} (R_d - R_{d+1}) \left( 1 - \Phi \left( \frac{-\mu_{Y_d^w}}{\sigma_{Y_d^w}} \right) \right). \tag{13}
\]

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

**Assumption 5.1.** \( \mu_{Y_d^w} < 0 \) for \( d = 1, \ldots, D \) and for all \( w \in \mathbb{W} \).

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\% percentile. The assumption therefore may break down if payoffs extend beyond the top 20\% of entries. Nonetheless, the payoff sizes around the 20\% percentile are very small and almost negligible. Indeed within our model, most of the expected P&L comes from the top few percentiles and \( \mu_{Y_d^w} < 0 \) is certainly true for these values of \( d \). Finally, we note the well-known general tendency of models

\(^{12}\)Certainly, we would need to impose some technical conditions on \( \{w_o\}_{o=1}^O \) to make this claim rigorous. We will not discuss such technical conditions, however, since top-heavy contests are our main focus in this paper and such a replication result does not hold for these contests. See Section 5 for further discussion on this issue.
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 20th 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 \(-\mu_{Y,d}/\sigma_{Y,d}\) to the normal CDF terms in (13) are positive. Given the objective in (13) is to maximize, it is also clear that for a fixed value of \(w^T\mu_d\) in (11), we would like the standard deviation \(\sigma_{Y,d}\) to be as large as possible. Unfortunately, the third term, \(2w^T\sigma_d G(r_{d})\), on the r.h.s. of (12) suggests that \(w\) impacts the variance by a quantity that depends on \(d\). The following assumption enables us to circumvent this problem.

**Assumption 5.2.** \(\text{Cov}(\delta_p, G'(r_{d})) = \text{Cov}(\delta_p, G'(r'_{d}))\) for all \(d, d' = 1, \ldots, D\).

Assumption 5.2 seems very reasonable based on our numerical experiments with real-world DFS top-heavy contests. In these contests, 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 therefore proceed to take Assumption 5.2 as given. It is then clear from (12) that the impact of \(w\) on \(\sigma_{Y,d}^2\) does not depend on \(d\).

Given the preceding arguments, it should be clear that for any fixed value of \(w^T\mu_d\), we would like to make \(w^T\Sigma_d w - 2w^T\sigma_{d} G'(r_{d})\) (the terms from (12) that depend on \(w\)) as large as possible. We are therefore in the situation of part (i) of Proposition 4.1 and we have a simple algorithm for solving the top-heavy problem. This is given in Algorithm 3 below where we omit the dependence on \(d\) of those terms that are assumed (by Assumption 5.2) to not vary with \(d\).

**Algorithm 3 Optimization Algorithm for the Top-Heavy Problem with a Single Entry**

Require: \(W, \Lambda, \mu_d, \Sigma_{d}, \sigma_{d} G'(r_{d})\) and Monte Carlo samples of \((\delta, G'(r_{d}))\) for all \(d = 1, \ldots, D\)

1: for all \(\lambda \in \Lambda\) do
2: \(w_{\lambda} = \arg\max_{w \in W} \left\{ w^T \mu_d + \lambda \left( w^T \Sigma_d w - 2w^T \sigma_{d} G'(r_{d}) \right) \right\} \)
3: end for
4: \(\lambda^{\ast} = \arg\max_{\lambda \in \Lambda} \sum_{d=1}^{D} \left( R_{d} - R_{d+1} \right) \mathbb{P} \left\{ w^T \delta > G'(r_{d})(W_{op}, \delta) \right\} \)
5: return \(w_{\lambda^{\ast}}\)

As with Algorithm 2, the \(w_{\lambda}\)'s are computed by solving BQP's and the optimal \(\lambda^{\ast}\) from line 4 can then be determined via the Monte Carlo samples that were used as inputs. Alternatively, we could also compute \(\lambda^{\ast}\) in line 4 using the normal approximation assumption along with Assumptions 5.1 and 5.2.

**5.1. The Top-Heavy Problem with Multiple Entries**

Consider now the more general top-heavy problem where we must submit \(N\) entries or portfolios. We repeat again the problem formulation from Section 2.2:

\[
\max_{W \in \mathbb{W}^N} \sum_{i=1}^{N} \sum_{d=1}^{D} (R_{d} - R_{d+1}) \mathbb{P} \left\{ w_i^T \delta > G'(r_{d})(W_{-i}, W_{op}, \delta) \right\},
\]

where \(r_{d}' = O + N - r_{d},\ G_{-i}'(r)\) is the \(r^{th}\) order statistic of \(\{G_{d}\}_{d=1}^{O} \cup \{F_{j}\}_{j=1}^{N} \setminus F_{i}\) and \(W_{-i} := W \setminus w_i\). The first thing to note is that, unlike the double-up format, there is no reason to believe that a portfolio replication strategy should ever be optimal here. To see this, consider the winner-takes-all contest where only the entry with the most fantasy points wins. This is clearly a special-case\footnote{The winner-takes-all contest can also be viewed as a pathological case of the double-up format where replication is definitely not an optimal strategy for the reasons outlined in this subsection.} of the top-heavy format. Under some mild...
assumptions, e.g. the probability of a winning tie being very small, it is clear that the goal is to maximize the probability of having a winning entry. This probability can only be increased by diversifying and so replication is (in general) strictly suboptimal.

For more general top-heavy contests that are not winner-takes-all contests, it is not clear whether or not some form of replication should take place. For example, consider the case where $N = 2$ and we are extremely skilled relative to all of our opponents. If the contest pays out to the top 20% of entries say, and $O$ is very large, then it may well be the case that submitting the same optimal entry twice is the best strategy. In general, solving the multiple entry top-heavy problem is difficult and so we propose using a greedy approach to tackle it. In particular, we will follow\footnote{Hunter et al. \cite{hun13} provide some performance guarantees in their setting for the greedy approach in the winner-takes-all setting. Those guarantees would also apply here.} the greedy approach of Hunter et al. \cite{hun13} and also impose constraints on our entries that ensure a diversified portfolio of entries. In particular, we:

(i) Solve for the optimal $N = 1$ entry. Call this entry $w_i^*$. 
(ii) For $i = 2, \ldots, N$, we again solve the $N = 1$ problem but with a new feasibility set $\mathcal{W}_i$ defined as

$$
\mathcal{W}_i := \mathcal{W} \cap \{w_i : w_i^T w_j^* \leq \gamma \ \forall j = 1, \ldots, i-1\}
$$

where $\gamma$ is a pre-specified parameter satisfying $\gamma \leq C$. Let $w_i^*$ denote the optimal solution. Our portfolio of $N$ entries would then consist of $w_1^*, \ldots, w_N^*$. Note that the effect of $\gamma$ is to ensure that portfolio $i$ can not have more than $\gamma$ athletes in common with each of the previous $i - 1$ portfolios. We mention in passing that in our numerical results of Section 6, we found this diversification heuristic for top-heavy contests to have a considerably higher expected P&L than the replication strategy, which simply submits $N$ copies of the optimal portfolio for the single-entry problem. Not surprisingly given our discussion in Section 4.3, this was not the case for our double-up contests. We also note it is straightforward to add additional linear constraints to $\mathcal{W}_i$ if further or different forms of diversification are desired.

**Remark 5.1.** While we have taken $N$ as given up to this point, it is perhaps worth mentioning that one can always use this heuristic to determine an “optimal” value of $N$. Specifically, we can continue to increase $N$ until the expected P&L contribution from the next entry (determined by our heuristic above) goes negative or below some pre-specified threshold. We also note that it is easy to estimate the P&L of any portfolio of entries via Monte Carlo simulation.

### 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\footnote{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 50th rank. Top-heavy contests are therefore extremely popular and hence are our principal focus in this paper.} 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. The first model (which we used across all contests) is the *strategic* model that we introduced in Section 4 for double-up / quintuple-up contests and in Section 5 for top-heavy contests. For all contests, we used the diversification heuristic\footnote{This makes sense as previously discussed for top-heavy contests. 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 6.3.} instead of the replication.
strategy for $N > 1$. 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.

**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.

**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):

$$\max_{w \in W} w^T \mu_δ.$$  \hspace{1cm} (15)

For $N > 1$ (which is the case in our numerical experiments), we employ the greedy heuristic discussed in Section 5.1 but suitably adapted for the case where we do not model opponents. 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.

**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\(^{17}\) from the same team. We denote this constraint as “QB-WR”. Mathematically, the resulting BP for $N = 1$ is:

$$\max_{w \in W, QB-WR} w^T \mu_δ.$$  \hspace{1cm} (16)

Again for $N > 1$, we employ a suitably adapted version of the greedy heuristic from Section 5.1. 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. [21] for hockey contests. They presented several variations of their model typically along the lines of including more stacking (or anti-stacking\(^{18}\)) 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. Parameters**

Our models rely on the following five input “parameters”\(^{19}\): the expected fantasy points of the real-world athletes $\mu_δ$, the corresponding variance-covariance matrix $\Sigma_δ$, the stacking probability $q$ from Section 3.2, the

---

\(^{17}\)By “main” WR of a team, we refer to the WR with the highest expected points among all the WRs in the same team.

\(^{18}\)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.

\(^{19}\)In addition to these five input parameters, we need the input features $X$ and the realized $p$ values for the Dirichlet regression. Such data is available on the internet. For example, the $f$ feature (point estimate of $p$) is available at the FantasyPros website and FanDuel contains the cost vector $c$ (before a contest starts) and the realized positional marginals $p$ (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.
diversification parameter \( \gamma \) from Section 5.1 and the lower bound on the budget for accepting an opponent’s portfolio \( B_n \) from Algorithm 1.

We obtain the estimate of \( \mu_\delta \) 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 \( \Sigma_\delta \) into the correlation matrix \( \rho_\delta \) and the standard deviations of the individual athletes \( \sigma_\delta \in \mathbb{R}^P \). The estimate of \( \rho_\delta \) is obtained from RotoViz [30] and \( \sigma_\delta \) is estimated using the realized \( \delta \) values from the 2016 and 2017 seasons. Note that \( \rho_\delta \) does not change from week to week whereas \( \sigma_\delta \) is updated weekly using the realized \( \delta \) from the previous week. It was also necessary to assume a distribution for \( \delta \) as we needed to generate samples of this random vector. We therefore simply assumed that \( \delta \sim \text{MVN}_P (\mu_\delta, \Sigma_\delta) \) where \( \text{MVN}_P \) 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 \( \mu_\delta \) and \( \Sigma_\delta \).

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 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. Finally, we set\(^{20} \gamma = 6 \) for the strategic and benchmark models across all contests and \( B_n = 0.99B \) (based on some empirical data we collected on opponents’ portfolio selections).

We used a data sample from DFS contests in the 2016 NFL season to select an appropriate choice for \( \Lambda \) (the grid of \( \lambda \) values required for Algorithm 2). In all of our contests, we set \( \Lambda = (0.00, 0.01, \ldots, 0.20) \). We could of course have reduced the computational burden by allowing \( \Lambda \) to be contest-specific. For example, in the case of quintuple-up contests, a choice of \( \Lambda = (0.00, 0.01, \ldots, 0.05) \) would probably have sufficed since \( \lambda^* \) for quintuple-up was usually close to zero.

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 \( \mu_{G(\cdot')} \), \( \sigma^2_{G(\cdot')} \) and \( \sigma_{\delta, G(\cdot')} \) (as required by Algorithms 2 and 3) 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 BQP’s were solved using Gurobi’s [19] 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), \( \lambda \) (see Algorithm 2) 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 \( \lambda \) and so we used 4 cores for double-up and quintuple-up contests and 8 cores for the top-heavy contests where the BQP’s 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.

6.3. 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

\(^{20}\text{We found this value of } \gamma \text{ to produce a near maximum within-model expected P&L.}\)
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 $\mu_\delta$ and $\Sigma_\delta$, which are clearly vital to the optimization model. Moreover, we obtained the $\mu_\delta$ 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 questionable to play. Accordingly, their second main RB (Chris Ivory) was expected to play more time on the field. However, our $\mu_\delta$ 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 $\mu_\delta$ estimate following such news. In fact, the so-called sharks do react to such last-minute information [26], 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 $\mu_\delta$ 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, 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)

\[21\text{There are more than 45 athletes on a roster but only 11 on the field at any one time.}\]
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. [21] focused on ice hockey and baseball (and avoided NFL) for precisely this reason.

The third reason applies specifically to the quintuple-up contests and is quite an interesting case. 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 \( \mu^w < 0 \) for all \( w \in W \) and to try and decrease variance otherwise. But \( \mu^w \) must be estimated via Monte Carlo and is of course also model-dependent. As such, if we estimate a maximal value of \( \mu^w \approx 0 \), it is quite possible we will err and increase variance when we should decrease it and vice versa. We suspect this occurred on regular basis with the quintuple-up contests, where we often obtained an estimate of \( \mu^w \) that was close to zero. We note that the benchmark portfolio is always long expected points and variance of points. This problem should be easy to address by introducing a threshold value of \( \mu^w \) (presumably greater than 0) below which we are led to increase variance. One could back-test to find a good choice of this threshold.

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.)

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

Figure 3 displays the in-model P&L distribution for the diversification strategy from Section 5.1 for both strategic and benchmark portfolios in week 10\(^2\) contests. We note this P&L distribution is as determined by our model with the continued assumption of the multivariate normal distribution for \( \delta \) as well as the Dirichlet-
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.

The 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. 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.

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, \( P(\text{P&L} < 0) \).

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 heuristic to the replication strategy for top-heavy. This is consistent with the winner-takes-all story we discussed in Section 5.1, where we argued that replication was in general

\[23\] 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.
clearly suboptimal. 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 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, $P(P&L < 0)$.

Figure 5 displays the realized and expected P&L’s. 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 $\mu$ and $\Sigma$ 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 losses 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).

6.4. 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 $p_{QB}, \ldots, p_{D}$ and how well they predict the benchmark fantasy points $G^{(r')} (\text{double-up})$ and $G^{(r'd)}$ for $d = 1, \ldots, D$ (top-heavy).

In Figure 6, we show the performance of our approach in terms of predicting the QB marginals $p_{QB}$ for the top-heavy and double-up contests\footnote{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.} 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,
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&L’s are displayed as points.

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. 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 marginals 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 6.3. Such late developing news would typically impact our estimate of $\mu_\delta$ which in turn would change both our optimal portfolios as well as our opponents’ team selections.

In Figure 7(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 $\delta$. These conditional prediction intervals provide a better approach to evaluate the quality of our Dirichlet-multinomial model for $W_{op}$ as they depend only on our model for $W_{op}$. Not surprisingly, the interval widths shrink considerably when we condition on $\delta$ and it is clear from the figure that we do an excellent job in week 10. In Figure 7(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 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.
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 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 \( p = (p_{QB}, \ldots, p_D) \)? For example, in the NFL contests, we could set \( p \) equal to the values predicted by the FantasyPros website.

(b) How much additional value is there if instead we assume \( (p_{QB}, \ldots, p_D) \sim (\text{Dir}(\alpha_{QB}), \ldots, \text{Dir}(\alpha_D)) \) as in Algorithm 1 but now \( \alpha_{QB}, \ldots, \alpha_D \) only depend on the first two features stated in Equation (4), that is, the constant feature and the estimate of \( p \) 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 \( \alpha_{QB}, \ldots, \alpha_D \) 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&L’s 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 \( \gamma \) 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 \( p \) (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 is depends on our “ground truth” model. For example, if we had assumed some other ground truth where \( p \) 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 \( p = (p_{QB}, \ldots, p_D) \)?

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

\[
\max_{W \in \mathcal{W}} \left\{ \mathbb{E}_p \left[ \mathbb{E}_{\delta, W_{op}} \left[ \text{Reward}(W, \delta, W_{op}) \mid p \right] \right] \right\} \quad \text{(17)}
\]

where \( \text{Reward}(W, \delta, W_{op}) \) denotes the P&L function which is easy to compute given \( W, \delta, \) and \( W_{op} \). The answer to question (i) is then given by the difference between (17) and

\[
\mathbb{E}_p \left[ \max_{W \in \mathcal{W}} \left\{ \mathbb{E}_{\delta, W_{op}} \left[ \text{Reward}(W, \delta, W_{op}) \mid p \right] \right\} \right]. \quad \text{(18)}
\]
Similarly, the answer to question (ii) is given by the difference between (17) and
\[ E_{p,W_{op}} \left[ \max_{W \in W^N} \left\{ E_\delta \left[ \text{Reward}(W, \delta, W_{op}) \mid p, W_{op} \right] \right\} \right]. \]  

(19)

However, computing both (18) and (19) 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 \( p \). 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 \( p \) together with the same stacking parameters from Section 6.)

- **Experiment 2**: This is similar to Experiment 1 but we now replace \( p \) with \( W_{op} \). However, we do not have access to the realized values of \( W_{op} \) during the NFL season. Instead, for each week we sample one realization of \( W_{op} \) using the realized \( p \) (with the same stacking parameters from Section 6) and treat the sampled \( W_{op} \) as the realized value. We then compute the optimal portfolio (the **insider portfolio**) for each week conditional on knowing the realized \( W_{op} \) and compare the expected P&L’s of the insider portfolio with the strategic non-insider optimal portfolio assuming the ground truth in P&L computations to be the realized \( W_{op} \).

It is worth emphasizing that in both Experiments 1 and 2 we are taking expectations over \( \delta \), the performance vector of the underlying NFL players. 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, and double-up contests respectively over the (strategic) non-insider. In Figure 8, 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 \( p \). 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 of \( W_{op} \). Noting that the variance of \( G^{(r)} \mid (\delta, p) \) goes\(^\text{26}\) to zero as the number of opponents \( O \) goes to infinity,

\[^{25}\text{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 6.3, namely the issue that arises when the maximal value of \( \mu_w \approx 0 \). In this case, the optimal value of \( \lambda \) in Algorithm 2 will be close to 0. As a result (and this should be clear from the expression for \( \sigma_{Y_w}^2 \) in (9) together with lines 3 and 7 of Algorithm 2) the only benefit to inside information in this case is in estimating \( \mu_{Y_w} \).}\]

\[^{26}\text{See Appendix A.1.}\]
however, we can conclude that the additional value of seeing the realized $W_{op}$ over and beyond the value of seeing the realized $p$ should \(^{27}\) 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 8: Weekly expected dollar P&L for the strategic model ($N = 50$) with and without inside information $p$ in the top-heavy series.](image)

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 \([7, 29]\) 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 $N_{collude} \times E_{max}$ 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 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 $N_{collude} \times E_{max}$ entries when they collude. This portfolio can be constructed using the approach outlined in Section 5.1 with $N = N_{collude} \times E_{max}$. 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 \(^{28}\) that each of the potentially colluding players has access to the modeling framework outlined in this paper and that as a result, each one submits identical

\(^{27}\) But note we are assuming here that the dependence structure between the positional marginals in $p$ is known regardless of whether we only see $p$ or $W_{op}$.

\(^{28}\) 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.
portfolios of $E_{\text{max}}$ entries. This portfolio is again constructed using the approach outlined in Section 5.1. While this assumption is stylized and not realistic in practice, it does allow us to get a sense of just 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.

Before describing our numerical experiments, it is worthwhile noting that the results of Section 6.3 and specifically, Figures 3(a) and 4(a), can be used to estimate the benefits of colluding in week 10 top-heavy 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 $N_{\text{collude}}$ ranged from 1 to 5. To be clear, the non-colluding portfolio comprised 50 strategic entries replicated $N_{\text{collude}}$ 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 $\delta$ and $W_{\text{op}}$, where samples of $(\delta, W_{\text{op}})$ were generated using the same calibrated model. In particular, the realized $(\delta, W_{\text{op}})$'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 $N_{\text{collude}}$. 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 $E[P&L^2 \times \mathbb{1}_{(P&L \leq T)}]$) 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 $E_{\text{max}}$ all other things being equal. For example, in the extreme case where $E_{\text{max}} = \infty$, there are clearly no benefits to colluding.

### 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 results from assuming the players know the true model should be relatively small.

---

29 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.

30 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 $E_{\text{max}} = 50$ 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 $SR_i := (E[P&L_i] - T)/DR_i$ where $E[P&L_i]$ denotes the expected P&L for week $i$, $T$ denotes the target P&L which we set to 0, and $DR_i := \sqrt{E[P&L_i^2] \times \text{I}_{\{P&L_i \leq T\}}}$ 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.)

<table>
<thead>
<tr>
<th>$N_{\text{collude}}$</th>
<th>Expected P&amp;L (USD)</th>
<th>Sortino Ratio</th>
<th>Probability of Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NC</td>
<td>C</td>
<td>Increase</td>
</tr>
<tr>
<td>1</td>
<td>6,053</td>
<td>6,053</td>
<td>0%</td>
</tr>
<tr>
<td>2</td>
<td>9,057</td>
<td>10,240</td>
<td>13%</td>
</tr>
<tr>
<td>3</td>
<td>10,975</td>
<td>13,776</td>
<td>26%</td>
</tr>
<tr>
<td>4</td>
<td>12,411</td>
<td>16,883</td>
<td>36%</td>
</tr>
<tr>
<td>5</td>
<td>13,632</td>
<td>19,677</td>
<td>44%</td>
</tr>
</tbody>
</table>

that requires solving a series of binary quadratic programs. Following Hunter et al. [21], we also provide a tractable greedy heuristic for handling the multiple entry, that is, $N > 1$, case for top-heavy style contests. This is in contrast to the replication approach we advocate for double-up style contests.

There are many potential directions for future research. The data we are currently gathering for these numerical experiments can be used to back-test other benchmark strategies as well as refining our own preferred strategies. For example, in Section 6.3, we discussed the possibility our strategic model might be incorrectly minimizing portfolio variance in quintuple-up. A more robust approach is therefore called for and it would be interesting to optimize over the threshold value of $\mu_w$ below which we are led to increase variance.

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 6.3, 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 $\mu_\delta$ and $\Sigma_\delta$ 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.

It would also be interesting to further develop our modeling and estimation approach for a random opponent’s portfolio $w_o$. We assumed in Section 3 that we had sufficient data to estimate the positional marginals of $w_o$. 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 approaches to splicing these marginals together to construct $w_o$. (We only discussed splicing them together independently or using a somewhat ad-hoc stacking approach but to do so more systematically we would need to estimate the relevant copula. It is not clear, however, whether we could ever obtain a rich enough data-set to estimate such a copula accurately.)

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 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 $\sigma_{\delta G(r')}$ that are required for Algorithms 2 and 3. 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


A. Appendix: Efficient Sampling Of Order Statistic Moments

In the double-up problem, there are three tasks that rely on Monte Carlo simulation. They are:

1. Estimating the input parameters\(^{31}\) \((\mu_{G^{(r')}}, \sigma_{G^{(r')}})\) that are required for lines 3 and 7 of Algorithm 2.
2. Generating samples of \((\delta, G^{(r')})\) to execute line 10 of Algorithm 2. As discussed in Remark 4.1, however, we could also execute line 10 using the normal approximation for \(Y_w\).
3. Estimating the P&L distribution for a given portfolio of entries using samples of \((\delta, G^{(r')})\). This task can also be performed using the normal approximation and indeed it is not actually necessary for the purpose of constructing the optimal portfolio of entries.

Recalling \(G_o = w_o \delta\) is the fantasy points score of the \(o^{th}\) opponent, we first note the \(G_o\)’s, \(o = 1, \ldots, O\), are IID given \((\delta, p)\) where \(p\) 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 \((\delta, G^{(r')})\):

1. Generate\(^{32}\) \(\delta \sim N(\mu_\delta, \Sigma_\delta)\) and \((p, W_{op})\) using Algorithm 1 (or the stacking variant of it) where \(p := (p_{QB}, \ldots, p_O)\) and \(W_{op} = \{w_o\}_{o=1}^O\).
2. Compute \(G_o := w_o \delta\) for \(o = 1, \ldots, O\).
3. Order the \(G_o\)’s.
4. Return \((\delta, G^{(r')})\).

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\(^{33}\) 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 \(G_o \mid (\delta, p)\) are IID for \(o = 1, \ldots, O\), it follows\(^{10}\) from the theory of order statistics that \(G^{(r')} \mid (\delta, p)\) satisfies

\[
G^{(q_0)} \mid (\delta, p) \xrightarrow{\text{P}} F^{-1}_{G(\delta, p)}(q) \quad \text{as} \quad O \to \infty
\]

(20)

where \(q \in (0, 1)\) and “\(\xrightarrow{\text{P}}\)” denotes convergence in probability. In large-\(O\) contests, we can use the result in (20) by simply setting \(G^{(r')} = F^{-1}_{G(\delta, p)}\left(\frac{r'}{O}\right)\). Of course in practice we do not know the CDF \(F_{G(\delta, p)}\) 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 \(F_{G(\delta, p)}\) with potentially far fewer samples. Our algorithm for generating Monte Carlo samples of \((\delta, G^{(r')})\) therefore proceeds as follows:

\(^{31}\)We do not include the parameter \(\sigma^2_G\) here since it does not impact the optimal solution to the problems in lines 3 and 7 of Algorithm 2. We also mention here that we focus on the double-up problem w.l.o.g. since we have already shown that the top-heavy problem can be solved using the double-up machinery under Assumptions 5.1 and 5.2.

\(^{32}\)We note that the normal assumption for \(\delta\) is not necessary and any multivariate distribution with mean vector \(\mu_\delta\) and variance-covariance matrix \(\Sigma_\delta\) could also be used.

\(^{33}\)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 \(O = 5,000\).
1. Generate $\delta \sim N(\mu_\delta, \Sigma_\delta)$ and $(p, W_{op})$ using Algorithm 1 (or the stacking variant of it) where $p := (p_{QB}, \ldots, p_D)$ and $W_{op} = \{w_o\}_{o=1}^O$.

2. Compute $G_o := w_o^\top \delta$ for $o = 1, \ldots, O$.

3. Use the $G$’s to construct $\tilde{F}_{G(\delta, p)}(\cdot)$.

4. Set $G(r^\prime) = \tilde{F}_{G(\delta, p)}^{-1}\left(\frac{r^\prime}{O}\right)$.

5. Return $(\delta, G(r^\prime))$.

Note in this algorithm $O$ now represents the number of Monte Carlo samples we use to estimate $F_{G(\delta, p)}$ 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 $r^\prime = O + N - 1$ (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 $\tilde{F}_{G(\delta, p)}^{-1}\left(1 - 10^{-5}\right)$. We cannot hope to estimate $\tilde{F}_{G(\delta, p)}^{-1}\left(1 - 10^{-5}\right)$ with just a moderate number of samples from line 2 of the algorithm, however, and this of course also applies to the values of $r^\prime$ 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 $\tilde{F}_{G(\delta, p)}(\cdot)$ with the empirical CDF of those $O$ samples from line 2 of the algorithm. For $r^\prime$ values that are not deep in the tail, we use $\tilde{F}_{G(\delta, p)}^{-1}(\cdot)$ to sample $G(r^\prime)$. For $r^\prime$ 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 99.0th and 99.5th percentiles as $\tilde{F}_{G(\delta, p)}(\cdot)$; see [8]. We then use this normal distribution in place of $\tilde{F}$ in line 4 of the algorithm for values of $r^\prime$ 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, $W_{op}$ takes more time to sample but $\delta$ appears to influence $G(r^\prime)$ more. Accordingly, in our experiments we implemented splitting with a ratio of 50:1 so that for each sample of $W_{op}$ we generated 50 samples of $\delta$.

---

34 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.