Robust Daily Fantasy Sports: Maximizing Reward via the Robust Optimization Paradigm

Dubem Mbeledogu

1. Introduction

Fantasy sports are games where participants get to play as the owner/manager of a sports team. As the manager, the participant gets to assemble a virtual version of a sports team composed of proxies of real players. A player's statistical performance in real life games or matches is used to calculate the number of fantasy points that are generated for the fantasy team. The sum of all the fantasy points generated by all the selected players on the team is the total team score. The total team score is compared against the team scores of other participants and those with the highest scores are the winners. For daily fantasy football, the player assembly process is done by "purchasing" players that each have a cost. Typically, the better the player, the more they cost, and that value is set by the operator of the fantasy sports league (DraftKings, FanDuel, etc.). There is a limited budget for the participant acting as the manager so the decision on which players can be drafted is constrained.

If a participant would like to "win" at daily fantasy football by scoring more fantasy points than other participants, it seems that there are two parts of the problem to engage. The first would be to accurately estimate the number of fantasy points each player will score. If a participant knows the number of fantasy points a player will score, they are at an advantage to select the players that will score more than the ones selected by other participants. There is plenty of work in this space, however the focus of this work is on the second part of the problem. The second part of the problem is to determine the best possible team given the number of fantasy points a participant thinks each player will score and the budget constraint the participant faces. There is a simple mixed integer linear programming (MILP) formulation of this problem that is common amongst online optimizers. That formulation is:

$$\max_{\boldsymbol{x}} \boldsymbol{p}^T \boldsymbol{x} \tag{1}$$

Subject to:

$$\sum_{i \in \mathcal{QB}} x_i = 1 \tag{2}$$

$$\sum_{i \in \mathcal{WR}} x_i \geq 3 \tag{3}$$

$$\sum_{i \in \mathcal{RB}} x_i \geq 2 \tag{4}$$

$$\sum_{i \in \mathcal{TE}} x_i \geq 1 \tag{5}$$

$$\sum_{i \in \mathcal{DST}} x_i = 1 \tag{6}$$

$$\sum_{i\in\mathcal{P}} x_i = 9 \tag{7}$$

$$\boldsymbol{c}^T\boldsymbol{x} \leq Budget \tag{8}$$

Where:
$\mathcal{P}$, the set of all eligible players,
$\mathcal{QB}, \mathcal{WR}, \mathcal{RB}, \mathcal{TE}, \mathcal{DST}$ are the sets of all eligible quarterbacks, wide receivers, running backs, tight ends, and defense/special teams
$\boldsymbol{p} \in \mathbb{R}^{|\mathcal{P}|}$, the vector of projected points each player will score
$\boldsymbol{c} \in \mathbb{R}_{+}^{|\mathcal{P}|}$, the vector of costs for each player,
$\boldsymbol{x} \in \{0,1\}^{|\mathcal{P}|}$, the vector of selections for the lineup

Equation (1) maximizes the number of projected points the lineup will score. Equations (2) through (7) ensure that the lineup is proper; for DraftKings that's one QB, at least 3 wide receivers, 2 running backs, and 1 TE, one defense/special teams, and a flex player who is a WR, RB or TE. Equation (8) is the budget constraint which is $50,000 for DraftKings. There are two issues with this formulation. The first is that it may not be required to score the most points. There are some game types where this is the case, one of which is 50/50, where the participants who score in the top 50 percent receive 1.8x the money they put in and the bottom half loses their money. For this game, all that is needed is to score *enough* points, which fundamentally changes lineup selection strategy. The second issue is that estimates for fantasy points are not accurate. The projected points R^2 for the data used in this study is 0.62. The data was provided by DFSForecast.com [1] who's won the DraftKing's Milly Maker competition several times, so the projections must be above average. Given the competition type and the uncertainty around player performance, it is possible that a lineup that will score fewer points on average will still receive a payout more often. Figure 1 details that scenario where lineup 1 scores more points on average than lineup 2 however, the variance on the projection errors of lineup 1 might be wide enough that lineup 2 is expected to win money more often.

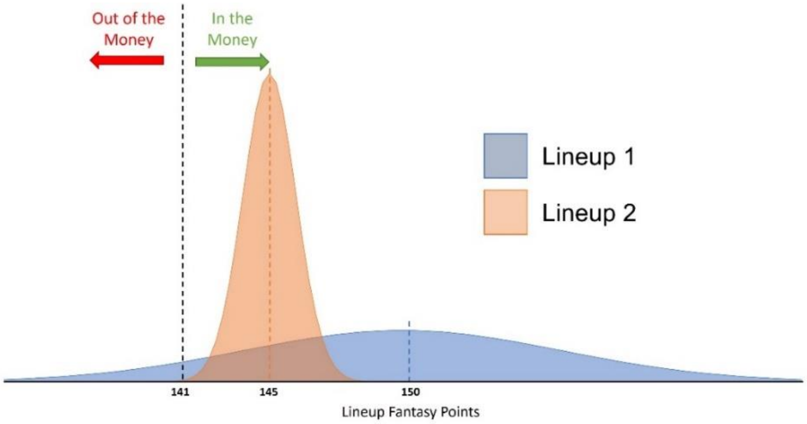


Figure 1

Expert heuristics for 50/50 address the issues with selecting the lineup that maximizes average points and these heuristics produce a distribution like lineup 1. Ryan Chase from Daily Fantasy Café says that the two most important overarching 50/50 themes are to "minimize risk" and "aim for good, not great". He says that "sacrificing a bit of ceiling when selecting one player over another is advisable if your lower ceiling player carries a higher floor" [2]. This matches the idea of selecting a lineup that has a lower variance, minimizing risk, even if that means the average points scored is not as high.

The mathematical formulation of the ideas from Figure 1 and from the 50/50 expert heuristics is to maximize the expected payout, which for 50/50 is to push as much of the point probability distribution above the payout line as possible. There is an optimization paradigm, stochastic programming, which directly solves this problem and excellent work has been done by Sarah Newell at Kansas State University on the topic [3]. Stochastic programming addresses the true problem but comes with its own disadvantages which include the difficulty in specifying the expectation function. For example, in the prior mentioned work, a normal distribution is in part specified by its standard deviation which cannot be calculated linearly. Extra specification must be made to approximate the lineup standard deviation. Secondly, excellent work done by Haugh and Singhal [4] directly addresses the idea of maximizing the probability of performing above a payout line with direct application to the 50/50 competition. Their formulation, while very sophisticated, is a binary quadratic problem that requires minutes of solve time on a high-performance computing cluster for which the average DFS participant or even DFS professional may not have access to. Given these difficulties, a simpler interpretation of this problem is to maximize the worst-case scenario given the uncertainty around points scored. The paradigm of maximizing the worst-case scenario is called Robust Optimization. The benefit of using robust optimization is the reduction in assumptions made about the performance distribution as well as the tractability and interpretability of the final solution. The rest of this work focuses on formulating the robust problem for daily fantasy football.

2. Methodology

2.1 Introducing the Robust Constraint

Because the constraints enforced by equations (2)-(8) must hold true for any lineup, they will be dropped for the remainder of the derivation as they remain untouched. As determined in the introduction, the goal is to maximize the worst-case scenario of *actual* points scored. Actual points can be thought of as the projected points a player might score plus some error on the model used to project those points. The problem can then be formulated as:

$$\max_{\boldsymbol{x},\theta} \theta \tag{9}$$

Subject to:

$$\theta \le (\boldsymbol{p}+\boldsymbol{e})^T\boldsymbol{x} \quad \forall \boldsymbol{e} \in \mathcal{U} \tag{10}$$

Where:
$\theta \in \mathbb{R}$, dummy scalar variable
$\boldsymbol{e} \in \mathbb{R}^{|\mathcal{P}|}$, the vector of errors between the projections and the actual player performance
$\mathcal{U}$, the uncertainty set. The possible set of errors that may occur

The right-hand side of the constraint in equation (10) looks like the objective in equation (1) except now it includes the errors on the model. Rather than having the right-hand side of equation (10) be the objective, it is shifted to the constraints by adding a "dummy" variable θ, and by maximizing over the dummy. Because the objective is to maximize the dummy variable and it is upper bounded by the right-hand side of equation (10), this formulation is no different than if the right-hand side of equation (10) were the objective. Now the constraint is made “robust” to the uncertainty in model error by including the stipulation that the constraint must hold for all errors in the uncertainty set. Because the constraint is an upper bound, if the dummy variable meets the constraint in the worst-case error scenario, it will meet the constraint for any error scenario. This is how the “worst-case” scenario is maximized. The robust constraint can then be rewritten:

$$\theta \leq (\boldsymbol{p} + \boldsymbol{e})^T \boldsymbol{x} \quad \forall \boldsymbol{e} \in \mathcal{U} \tag{11}$$

$$\theta \leq \min_{\boldsymbol{e} \in \mathcal{U}} (\boldsymbol{p} + \boldsymbol{e})^T \boldsymbol{x} \tag{12}$$

$$\theta \leq \boldsymbol{p}^T \boldsymbol{x} + \min_{\boldsymbol{e} \in \mathcal{U}} \boldsymbol{e}^T \boldsymbol{x} \tag{13}$$

The focus then becomes to solve the minimization problem inside the constraint. To do this, an uncertainty set must be defined that will capture the worst-case error scenario without being too conservative.

2.2 Defining the Uncertainty Set

Vector norms are common uncertainty sets due to their symmetry and tractable solution when involved in the inner minimization problem. Vector norms are used for this problem and are defined as:

$$\|\boldsymbol{e}\|_p = \left(\sum_i |e_i|^p \right)^{1/p} \tag{14}$$

The uncertainty set using vector norms could then be defined as:

$$\|\boldsymbol{e}\|_p \leq \rho \tag{15}$$

Where:
$p \in [1, \infty)$, the uncertainty set shape
$\rho \in \mathbb{R}^+$, the uncertainty set size

When p is 1, the set is called a polygon set; when p is 2, it is called a ball set; when p approaches infinity, it is called a box set. The shaded regions of figure 2 show the area that is covered by p-norm uncertainty sets. The figures show that this definition for the uncertainty set implies that the errors are symmetrically distributed around 0. This is true for most models for which the errors are typically normally distributed around 0 with some variance or $e \sim \mathcal{N}(0, \sigma_e^2)$. This holds true for the model predictions supplied by DFSforecast.com as shown in appendix A. This definition of uncertainty set also implies that that the errors are not correlated which does not hold true. One could reason that if a quarterback performs better than predicted, most likely a receiver will perform better than predicted. See figure 3 which shows WR vs. QB error for the dataset used in this study with the uncertainty region overlaid. For the box set, the corners in the second and fourth quadrants have almost no points in them. For the polygonal set, the top and bottom corners have almost no points in them. Due to the correlation, these standard p-norm sets will include regions where there are no errors, giving an estimate for the worst-case scenario that is too conservative. The vector norm needs to be adjusted to account for the correlation.

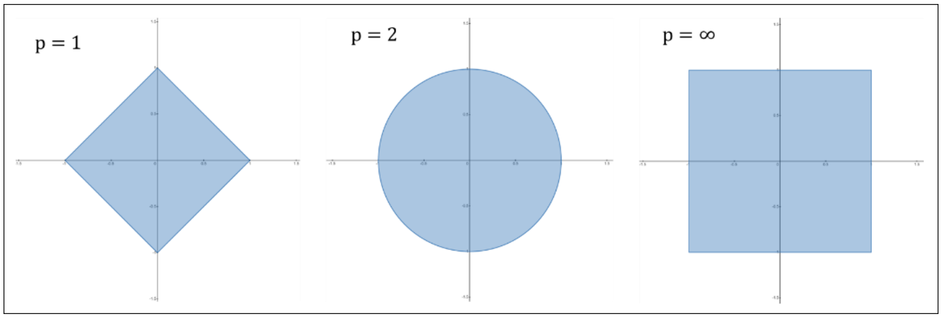


Figure 2: Uncertainty set shapes (polygon-left, ball-middle, box-right)

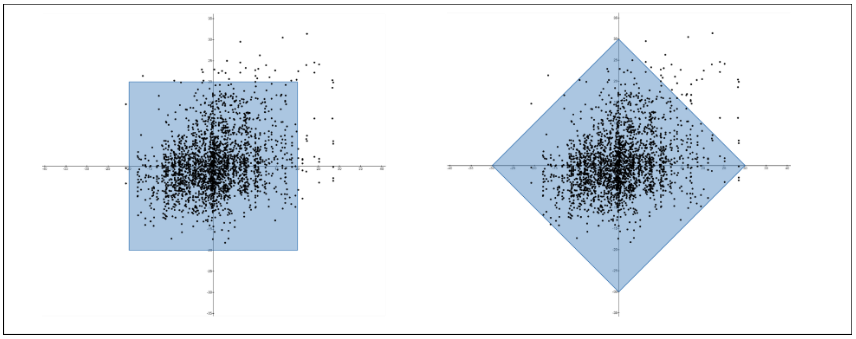


Figure 3: WR vs QB error with uncertainty sets (box-left, polygon-right)

2.2.1 Data Whitening

There is a technique used to incorporate correlation in uncertainty sets. The technique is borrowed from data whitening and is used in other robust uncertainty set definitions by the likes of Yuan, Li and Huang [5]. Data whitening is the process of transforming correlated data with unequal variances into uncorrelated data with equal, unit variances. Data whitening applies to this problem as the output of the vector norm is uncorrelated with equal variances but the input, in this case the error vector, is correlated with unequal variances.

The whitening matrix ($\boldsymbol{W} \in \mathbb{R}^{nxn}$) is one such that when multiplied by some arbitrary data matrix ($\boldsymbol{M} \in \mathbb{R}^{nxm}$) with n features and m observations, the resulting matrix ($\boldsymbol{Y} \in \mathbb{R}^{nxm}$) yields an identity ($\boldsymbol{I} \in \mathbb{R}^{nxn}$) covariance matrix. Then for a centered dataset (mean of each feature is 0) the covariance can be estimated as:

$$\boldsymbol{\Sigma_M} = \frac{\boldsymbol{M}\boldsymbol{M}^T}{m} \tag{16}$$

It follows that:

$$\boldsymbol{Y} = \boldsymbol{W}\boldsymbol{M} \tag{17}$$

$$\boldsymbol{\Sigma_Y} = \frac{\boldsymbol{YY}^T}{m} = \frac{(\boldsymbol{WM})(\boldsymbol{WM})^T}{m} = \frac{\boldsymbol{WMM}^T\boldsymbol{W}^T}{m} = \boldsymbol{W\Sigma_M W}^T = \boldsymbol{I} \tag{18}$$

$$\boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{\Sigma}_{\boldsymbol{M}}^{-1} \tag{19}$$

There are many whitening matrices that can make the equality in equation (19) true. It's clear that the Cholesky decomposition of the inverse covariance matrix of $\boldsymbol{M}$ works. The inverse square root of the covariance matrix of $\boldsymbol{M}$ works as well and it will be used for the rest of this problem. For this problem, the whitening matrix is $\boldsymbol{\Sigma}_{\boldsymbol{e}}^{-1/2}$or the inverse square root of the error covariance matrix.

The new uncertainty set is now defined:

$$\left\| \boldsymbol{\Sigma}_{\boldsymbol{e}}^{-1/2}\boldsymbol{e} \right\|_p \leq \rho \tag{20}$$

This definition will shift the uncertainty set to account for the correlations of the errors. Figure 4 below contains WR vs QB error with the new uncertainty sets overlaid in green. The regions in the second and fourth quadrants with low point density have reduced area when correlations are accounted for. For the polygon uncertainty set, the edges in the second and fourth quadrants are pulled in towards the origin. For the box uncertainty set, the corners in the second and fourth quadrants are pulled in towards the origin.

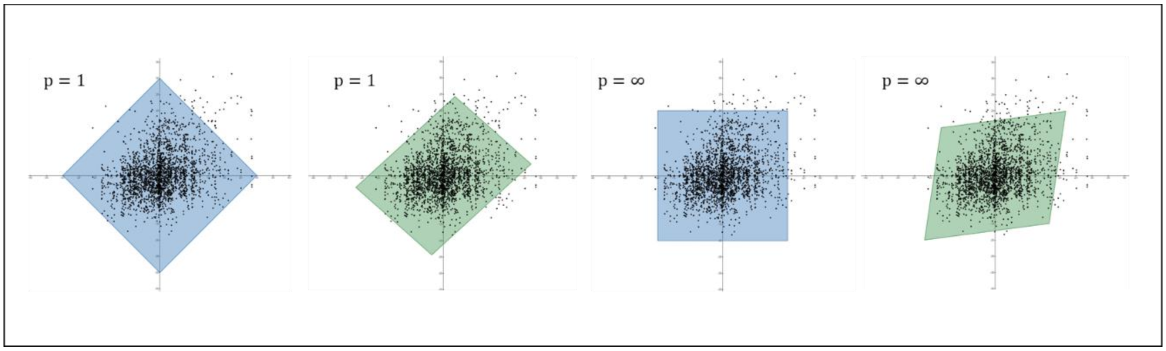
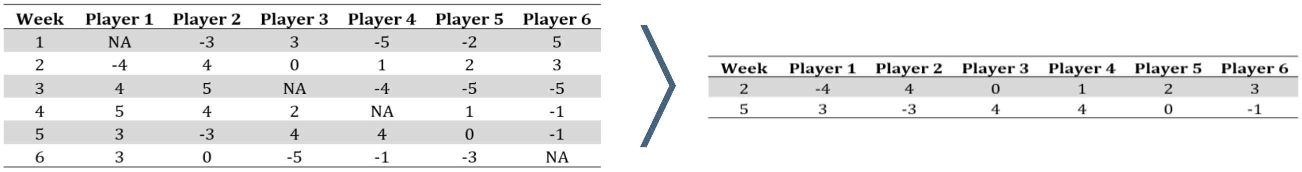


Figure 4: WR vs QB error with uncertainty sets (polygon-left, box-right, no correlation-blue, w/correlation-green)

2.2.2 Initially Generating the Covariance Matrix

It appears that the covariance matrix can be estimated directly using equation (16) which is implemented in many computing packages like NumPy. The issue is that every player may not play in every game. There may be some missing errors. To remedy this, players who miss more than 20% of games are removed from the set of players that will be evaluated. Even after this, some players may still miss a game. If equation (16) is to be used, every week or row where at least one player didn't play must be removed. Because players may miss different games, very few datapoints may remain and the estimate of the covariance matrix will be weak. See figure 5.



Week	Player 1	Player 2	Player 3	Player 4	Player 5	Player 6
1	NA	-3	3	-5	-2	5
2	-4	4	0	1	2	3
3	4	5	NA	-4	-5	-5
4	5	4	2	NA	1	-1
5	3	-3	4	4	0	-1
6	3	0	-5	-1	-3	NA

Week	Player 1	Player 2	Player 3	Player 4	Player 5	Player 6
2	-4	4	0	1	2	3
5	3	-3	4	4	0	-1

Figure 5: Error data frame reduction from missing data

To reduce the loss in data, pairwise covariances are calculated. For players that do not play in the same game, it's assumed that they do not affect each other and their covariances are set to 0. For players on the same team, their pairwise covariances are calculated normally. Errors for players on opposing teams are correlated for the data provided by DFSForecast.com as shown in Appendix B. To capture this, the covariance between a player and the opposing *positions* over the previous weeks were calculated. For the week a lineup is to be optimized, the pairwise covariance for opposing players is the minimum of the magnitude of the covariances between their positions. For example, if Aaron Rodgers is playing the Chicago Bears in week 13, an error covariance is calculated between Aaron Rodgers and opposing defenses from the previous 12 weeks and an error covariance is calculated between the Bears defense and opposing QB's from the previous 12 weeks. The final pairwise covariance used for Aaron Rodgers/Bears defense is the minimum of the magnitude between the two. This method was chosen because these covariances are the least sure estimates so their effect on the solution should be minimized.

2.2.3 Nearest Correlation Matrix

Nick Higham, Richardson Professor of Applied Mathematics at the University of Manchester, states that "symmetric positive definiteness is arguably one of the highest mathematical accolades to which a matrix can aspire" [6]. This special quality of being positive semi-definite (PSD) is required of a matrix to take its square root and return real values. The square root of the covariance matrix must return real values for this problem so it must also be PSD. Covariance matrices generated using equation (16) are PSD by definition but, because the covariance matrix generated by the previous section was done pairwise, it may not actually be PSD. The generated covariance matrix must be forced to be PSD to continue with the analysis. The goal is then to manipulate the values of the covariance matrix as minimally as possible such that it becomes PSD. To do this, rather than generating a covariance matrix, initially pairwise *correlations* are calculated, and the nearest possible correlation matrix is found. Nearest_correlation.py developed by Mike Croucher [7] was used for this task which implements Nick Higham's nearest correlation algorithm [8]. The algorithm works by using Dykstra's alternating projections algorithm for projecting a point onto the intersection of two convex sets. In the case of the nearest correlation problem, that is the projection, in the Frobenius norm, of the initial, pairwise correlation matrix onto the intersection of the set of symmetric PSD matrices and the set of symmetric matrices with diagonal 1. The nearest correlation is defined in equations (21) & (22).

$$\min_{\boldsymbol{R_e}} \|\boldsymbol{X} - \boldsymbol{R_e}\|_F \tag{21}$$

Subject to:

$$\boldsymbol{R_e} \in \mathcal{S}, \mathcal{V} \tag{22}$$

Where:
$\boldsymbol{X}$, the initial, pairwise correlation matrix of the errors
$\boldsymbol{R_e}$, the nearest correlation matrix of the errors to be calculated
$\mathcal{S}$, the set of all symmetric PSD matrices

$\mathcal{V}$, the set of all symmetric matrices with diagonal 1

The resulting nearest correlation matrix is then converted back to a covariance matrix by using the error standard deviation for each player:

$$\boldsymbol{D_e} = diag([\sigma_{e1}, \sigma_{e2}, \cdots, \sigma_{e|\mathcal{P}|}]) \tag{23}$$

$$\boldsymbol{\Sigma_e} = \boldsymbol{D_e R_e D_e} \tag{24}$$

Where:
σ_{e_i} is the standard deviation of the errors for player i

This method was chosen rather than directly finding the nearest covariance matrix because the variances are the surest values in the matrix that are being estimated. It is preferred to fix the variances and slightly alter the covariances which are already minimized to subdue their impact. The final algorithm to generate the covariance matrix to be used in the uncertainty set definition is:

Algorithm 1 Final Covariance Matrix generation for Week W

```
Initialize player/player error correlation matrix $\boldsymbol{R_e}$ of eligible players as identity matrix $I$
Initialize player error standard deviation matrix $\boldsymbol{D_e}$ of eligible players with zeros
Initialize player/opposing position error correlation matrix $\boldsymbol{Q_e}$ with zeros
for each player $i$ in 1 to $|\mathcal{P}|$
    $D_{e_{ii}} \leftarrow \sigma_{e_i}$ #Calculate player error standard deviations
    for each position j in 1 to |Positions|
        $Q_{e_{ij}} \leftarrow q_{e_i e_{position_j}}$ #Calculate player i /opposing position j error correlation
    end for
end for
for player $i$ in 1 to $|\mathcal{P}|$
    for player $j$ in $i + 1$ to $|\mathcal{P}|$
        if player $j$ is on the same team as player $i$
            $R_{e_{ij}}, R_{e_{ji}} \leftarrow r_{e_i e_j}$ #Player correlation
        end if
        if player $j$ is on the opposing team to player $i$ for week W
            $R_{e_{ij}}, R_{e_{ji}} \leftarrow \min\,(abs(q_{e_i e_{position_j}}), abs(q_{e_j e_{position_i}}))$ #Opposing position correlation
        end if
    end for
end for
$\boldsymbol{R_e} \leftarrow nearest\_correlation(\boldsymbol{R_e})$
$\boldsymbol{\Sigma_e} \leftarrow \boldsymbol{D_e R_e D_e}$
```

2.3 Inner minimization solution

Focus is returned to the inner minimization problem from equation (13). The inner minimization problem for which will now be called the primal problem, is now formulated as a conic programming problem:

$$\min_{\boldsymbol{e}} \boldsymbol{x}^T \boldsymbol{e} \tag{25}$$

Subject to:
$$(\boldsymbol{e}, \rho) \in \mathcal{K} = \left\{ (\boldsymbol{e}, \rho) \mid \left\| \boldsymbol{\Sigma}_{\boldsymbol{e}}^{-1/2} \boldsymbol{e} \right\|_p \leq \rho \right\} \tag{26}$$

The conic constraint is replaced with a dual constraint and Lagrangian variables $\boldsymbol{s} \in \mathbb{R}^{|P|}$ and $t \in \mathbb{R}$ are added resulting in:

$$\mathcal{L}(\boldsymbol{e}, \boldsymbol{s}, t) = \boldsymbol{x}^T \boldsymbol{e} - \boldsymbol{s}^{\mathrm{T}} \boldsymbol{\Sigma}_{\boldsymbol{e}}^{-1/2} \boldsymbol{e} - t\rho \tag{27}$$

Subject to:
$$(\boldsymbol{s}, t) \in \mathcal{K}^* = \{ (\boldsymbol{s}, t) \mid \|\boldsymbol{s}\|_q \leq t \} \; where \; \frac{1}{p} + \frac{1}{q} = 1 \tag{28}$$

The Lagrangian is first minimized over the errors then maximized over the dual variables. If strong duality holds (which it does here), the objective of the final maximization problem, which will now be called the dual problem, will be equal to the objective of the primal problem. Because the dual problem is a lower bound to the primal problem and the case of interest is when they are equal, only the case where a finite objective is obtained is considered.

$$g(\boldsymbol{s}, t) = \min_{\boldsymbol{e}} \mathcal{L}(\boldsymbol{e}, \boldsymbol{s}, t) = \min_{\boldsymbol{e}} \left[\left(\boldsymbol{x}^T - \boldsymbol{s}^{\mathrm{T}} \boldsymbol{\Sigma}_{\boldsymbol{e}}^{-1/2} \right) \boldsymbol{e} - t\rho \right] = \begin{cases} -t\rho & \boldsymbol{x} = \boldsymbol{\Sigma}_{\boldsymbol{e}}^{-1/2} \boldsymbol{s} \\ -\infty & otw \end{cases} \tag{29}$$

Because only the finite case is considered, the problem becomes:

$$\max_{\boldsymbol{s}, t} g(\boldsymbol{s}, t) = \max_{\boldsymbol{s}, t} -t\rho \tag{30}$$

Subject to:
$$\boldsymbol{x} = \boldsymbol{\Sigma}_{\boldsymbol{e}}^{-1/2} \boldsymbol{s} \tag{31}$$

$$\|\boldsymbol{s}\|_q \leq t \tag{32}$$

To maximize the objective function, the smallest value for t must be selected and since t is lower bounded by $\|\boldsymbol{s}\|_q$, it can be replaced by $\|\boldsymbol{s}\|_q$ in the objective. Then $\boldsymbol{s}$ is constrained by equality so it can be replaced with $\boldsymbol{\Sigma}_{\boldsymbol{e}}^{1/2} \boldsymbol{x}$. The dual problem simplifies to:

$$\max_{\boldsymbol{s}, t} -t\rho = \max_{\boldsymbol{s}, t} -\rho \|\boldsymbol{s}\|_q \tag{33}$$

Subject to:
$$\boldsymbol{s} = \boldsymbol{\Sigma}_{\boldsymbol{e}}^{1/2} \boldsymbol{x} \tag{34}$$

$$\Rightarrow \max_{\boldsymbol{s}, t} -\rho \left\| \boldsymbol{\Sigma}_{\boldsymbol{e}}^{1/2} \boldsymbol{x} \right\|_q = -\rho \left\| \boldsymbol{\Sigma}_{\boldsymbol{e}}^{1/2} \boldsymbol{x} \right\|_q \tag{35}$$

The solution to the dual problem (right-hand side of equation (35)), can now replace the primal problem (minimization term in equation (13)) in the initial robust constraint. The dummy variable can be removed, and the right-hand side of the constraint can be shifted back to the objective function. The objective function looks like MILP but now there is an extra term subtracted. This can be thought of as the “safety factor” from the effect of the errors. The higher the variance of the errors, the bigger the penalty of the worst-case scenario.

$$\theta \leq \boldsymbol{p}^T \boldsymbol{x} - \rho \left\| \boldsymbol{\Sigma}_{\boldsymbol{e}}^{1/2} \boldsymbol{x} \right\|_q \tag{36}$$

$$\max_{\boldsymbol{x}} \boldsymbol{p}^T \boldsymbol{x} - \rho \left\| \boldsymbol{\Sigma}_{\boldsymbol{e}}^{1/2} \boldsymbol{x} \right\|_q \tag{37}$$

For proper choice of uncertainty set shape, equation (37) can be formulated as a MILP, retaining its tractability compared to equation (1). For a box uncertainty set ($q = 1$), the objective becomes:

$$\max_{\boldsymbol{x}} \boldsymbol{p}^T \boldsymbol{x} - \rho \sum_{i=1}^{|\mathcal{P}|} \left| \Sigma_e^{1/2} x \right|_i \tag{38}$$

Which can then be linearly formulated as:

$$\max_{\boldsymbol{x},\boldsymbol{z}} \boldsymbol{p}^T \boldsymbol{x} - \rho \sum_{i=1}^{|\mathcal{P}|} z_i \tag{39}$$

Subject to:

$$-\boldsymbol{z} \leq \boldsymbol{\Sigma}_{\boldsymbol{e}}^{1/2} \boldsymbol{x} \leq \boldsymbol{z} \tag{40}$$

Where:
$\boldsymbol{z} \in \mathbb{R}^{|\mathcal{P}|}$

For a polygonal uncertainty set ($q = \infty$), the objective becomes:

$$\max_{\boldsymbol{x}} \left\{ \boldsymbol{p}^T \boldsymbol{x} - \rho \max_{1 \leq i \leq |\mathcal{P}|} \left| \Sigma_e^{1/2} x \right|_i \right\} \tag{41}$$

Which can then be linearly formulated as:

$$\max_{\boldsymbol{x},z} \boldsymbol{p}^T \boldsymbol{x} - \rho z \tag{42}$$

Subject to:

$$z \geq \left(\Sigma_e^{1/2} x \right)_i \quad \forall i \; in \; P \tag{43}$$

$$z \geq -\left(\Sigma_e^{1/2} x \right)_i \quad \forall i \; in \; P \tag{44}$$

Where:
$z \in \mathbb{R}$

This formulation is not only advantageous because of the tractability of the problem and simplicity of interpretation, but also because it’s completely model dependent. The uncertainty that is being accounted for is on the errors in the predictive model that generates the projections. This solution can be tailor-made for any predictive model.

3 Results

A simulation of 10,000 outcomes was run for the 2021-2022 season for which DFSforecast.com provided projections. The “true” error covariance matrix for all players was calculated using all 18 weeks of data and algorithm 1. It was assumed that the errors are distributed according to a multivariate normal $\boldsymbol{e}\sim\mathcal{N}(0,\boldsymbol{\Sigma}_{\boldsymbol{e}-true})$. Then both the standard MILP and the robust formulation were used to select lineups for weeks 10-18, starting from week 10 so that there is adequate data to estimate the error covariance matrix for the robust formulation. It should be noted that the robust formulation could be used on earlier weeks if errors from the previous season are available. Since projections were only provided for the 2021-2022 season, the earlier games in the season could not be run. Some players such as rookies for the current season or players who switched teams will not have representative data from the previous season. To start the season, rookies would take on the average variance for their position and 0 covariance with other players. For players who switched teams, it may be useful to use their variance from the previous season and their player-position covariances from the previous season as proxies for the current player-player covariances.

The goal of this formulation is to maximize the worst-case scenario, so the 1 percentile outcome of actual points scored from the simulations was recorded in tables 1 and 2. Performance for this set of projections was also compared against empirical payout lines of DraftKings competitions from the 2021 season, collected from fantasycruncher.com [9]. There are tens of 50/50 competitions, usually with 100 competitors each, that happen every week and the average payout line from the performances each week were used as the payout line for this study. The box uncertainty set probability distribution was used for the probability of payout in table 3. The simulations were run in jupyter notebooks using Gurobi as the solver on an Intel core i7 10th gen processor. The optimizations took fractions of a second to solve.

Table 1: Robust Optimization simulation results using polygon uncertainty set

Polygon									
Week	**Mean**				**1%**				$\boldsymbol{\rho}$
	MILP	**Robust**	**Change**	**%Change**	**MILP**	**Robust**	**Change**	**%Change**	
10	151.3	151.3	0.0	0.0%	100.3	100.3	0.0	0.0%	NA
11	151	145.3	-5.7	-3.8%	97.3	97.9	0.6	0.6%	2
12	139.1	137.8	-1.3	-0.9%	89.2	89.3	0.1	0.1%	2
13	153.8	141.8	-12	-7.8%	100.8	104	3.2	3.2%	5
14	146.3	140.1	-6.2	-4.2%	89.1	96	6.9	7.7%	5
15	143.2	136.3	-6.9	-4.8%	85.4	91.1	5.7	6.7%	5
16	148.5	141.8	-6.7	-4.5%	97.6	109	11.4	11.7%	5
17	144.5	144.5	0.0	0.0%	101.1	101.1	0.0	0.0%	NA
18	140.2	138.3	-1.9	-1.4%	86.3	90.9	4.6	5.3%	2

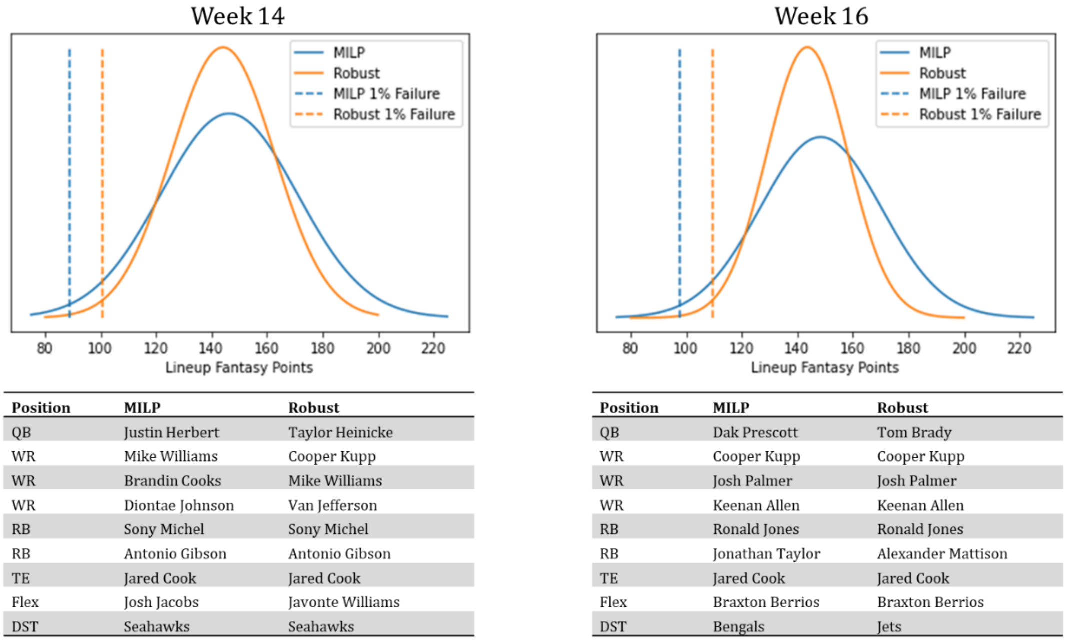
Table 2: Robust Optimization simulation results using box uncertainty set

Box									
Week	Mean				1%				ρ
	MILP	Robust	Change	%Change	MILP	Robust	Change	%Change	
10	151.3	151.3	0.0	0.0%	100.3	100.3	0.0	0.0%	NA
11	151	151	0.0	0.0%	97.3	97.3	0.0	0.0%	NA
12	139.1	130.5	-8.6	-6.2%	89.2	91.1	1.9	2.1%	0.25
13	153.8	145	-8.8	-5.7%	100.8	104.6	3.8	3.8%	0.25
14	146.3	144.1	-2.2	-1.5%	89.1	100.9	11.8	13.2%	0.1
15	143.2	137.3	-5.9	-4.1%	85.4	89.4	4	4.7%	0.35
16	148.5	143.6	-4.9	-3.3%	97.6	109.6	12	12.3%	0.35
17	144.5	137	-7.5	-5.2%	101.1	105.7	4.6	4.5%	0.25
18	140.2	138.8	-1.4	-1.0%	86.3	91.6	5.3	6.1%	0.1

Table 3: Box performance against empirical payout lines

Week	Payout Line	MILP – Probability of payout	Robust – Probability of payout
12	98.78	97%	97%
13	173.75	19%	5%
14	128.34	77%	80%
15	123.52	79%	75%
16	133.46	75%	77%
17	134.39	71%	58%
18	120.79	80%	81%

For weeks 10/17 in the polygon uncertainty set and weeks 10/11 for the box uncertainty set, there was no increase in the 1% actual points outcome. This would be interpreted as the lineup that scores the most points on average *also* has the maximum worst-case scenario. This can happen depending on the model used to generate the projections. There are several weeks where the worst-case scenario increased so much so that the robust lineup would've been preferred empirically over the MILP, those being weeks 14, 16, and 18. Figure 6 shows the distributions of points scored between MILP and robust for the box uncertainty set for weeks 14 and 16.



Position	MILP	Robust
QB	Justin Herbert	Taylor Heinicke
WR	Mike Williams	Cooper Kupp
WR	Brandin Cooks	Mike Williams
WR	Diontae Johnson	Van Jefferson
RB	Sony Michel	Sony Michel
RB	Antonio Gibson	Antonio Gibson
TE	Jared Cook	Jared Cook
Flex	Josh Jacobs	Javonte Williams
DST	Seahawks	Seahawks

Position	MILP	Robust
QB	Dak Prescott	Tom Brady
WR	Cooper Kupp	Cooper Kupp
WR	Josh Palmer	Josh Palmer
WR	Keenan Allen	Keenan Allen
RB	Ronald Jones	Ronald Jones
RB	Jonathan Taylor	Alexander Mattison
TE	Jared Cook	Jared Cook
Flex	Braxton Berrios	Braxton Berrios
DST	Bengals	Jets

Figure 6: Weeks 14 and 16 lineup comparisons. MILP in blue, Robust with box uncertainty in orange

The robust formulation will prefer players who score highly on average but who also have less variance on the errors of their past projections as they're less likely to have bad outcomes. This formulation also favors pairs of players with negative covariances. The negative covariances reduce the overall variance of the lineup. Appendix C & D show the lineup covariance matrices. An example of the robust lineup preferences at play is the choice of Tom Brady over Dak Prescott in week 16. The selection comes from the negative covariance between Tom Brady and running back, Ronald Jones. For this specific model, when Tom Brady performs worse than expected, Ronald Jones tends to perform better than expected which provides a hedge against the two players performing under expectation at the same time. Dak Prescot was projected to score 22.14 points that week while Tom Brady was projected to score only 20.79 points at a more expensive cost, however this formulation has decided to accept that difference for the gain in worst-case outcome. This formulation provides optimal decision making for tradeoffs like this.

Lastly, it seems that across almost all weeks, the box uncertainty set improves the 1% outcome better than the polygon uncertainty set. This most likely comes from the polygon uncertainty set having a worst-case scenario that is too conservative; it accounts for scenarios that do not occur. Referring to figure 4, the corners of the polygon uncertainty set, even after accounting for correlated errors, still have lower point density. The box uncertainty sets naturally fits to the correlated errors as its corners are oriented in the direction of the correlation while the polygon's flat edges are oriented in the direction of the correlation. This general phenomenon most likely extends to higher dimensional space.

4 Discussion

4.1 Considerations

This formulation is not without its flaws. There is an underlying assumption that the predictive model being used to project the points has greater than 50% of its probability density above the payout line when using the basic MILP to generate lineups. In other words, the predictive model needs to be good enough that the MILP generated lineups have a positive expected value on the earnings. If that's not the case, the robust formulation will shift more probability density to a lower number of points, exacerbating the low expected value issue. In this scenario, it may be interesting to favor higher variance in the errors rather than punish it. This will widen the probability distribution and shift more probability density to a higher number of points. A simple change can be made to equation (37) to accomplish this:

$$\max_{\boldsymbol{x}} \boldsymbol{p}^T \boldsymbol{x} + \rho \left\| \boldsymbol{\Sigma}_{\boldsymbol{e}}^{1/2} \boldsymbol{x} \right\|_q \tag{45}$$

Although equation (45) isn't rigorously derived nor does it simplify as easily to something like equations (39) or (42), it can be interpreted similarly to equation (37). The "safety" term is now additive and can be thought of as a "surprise" term. This will favor players who not only are projected to score highly, but also have a higher variance in their errors and have a higher covariance with teammates and opposing positions. This logic is already standard to most daily fantasy football experts who purport that one should "stack" QB's and WR's. Stack means to select QB's and WR's from the same team for your lineup.

There is a secondary assumption being made which is that the payout line for a game like 50/50 is independent of lineup performance. This may not be true. A scenario could happen where there are only two lineups to select from. One that has the "MILP" distribution and one that has the "robust" distribution. 50% of participants could select the robust lineup while the remainder select the MILP distribution. Since there are only two outcomes, MILP or robust, the payout line becomes the result of the robust performance. The probability that the robust lineup receives a payout is simply the probability that the robust lineup produces more points than the MILP lineup. If it is assumed that the player errors are normally distributed, then the lineup performances are normally distributed and the difference between the lineup performances will be normally distributed as well with mean of $\mu_{Robust} - \mu_{MILP}$ and variance $\sigma_{Robust}^2 + \sigma_{MILP}^2 - 2\sigma_{MILP-Robust}$. The probability that the difference is greater than 0 will be less than 50% because the mean is less than 0 and the distribution is symmetrical. Under this scenario, it never makes sense to select the robust lineup no matter how much the worst-case scenario is improved. It seems to be a fair assumption that the payout line doesn't swing much with differences in the robust lineup performance because there tend to be many lineups that are very different. Pro Football Focus projects ownership percentages, which are the percent of total lineups including player X. For week 10 of the 2022 season, Pro Football Focus projects that ~170 players will be used across lineups, the vast majority of which have 10% usage or less [10]. In other words, there are many different lineups and there are few players being shared across those lineups. There seems to be enough unique lineups that a large swing in robust lineup performance probably won't also swing the payout line.

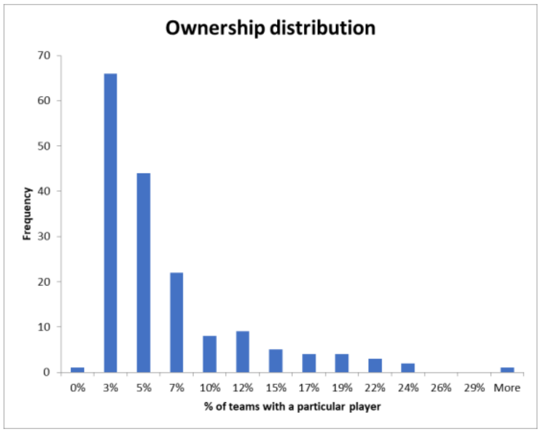


Figure 7: Histogram of player usage for week 10 of the 2022 season projected by Pro Football Focus

It should be noted that this formulation given its benefits and flaws should be used *in conjunction with* MILP and domain knowledge. It is an extra tool to simply and tractably understand the tradeoffs of risk and reward in the 50/50 competition.

4.2 Next Steps

The uncertainty sets defined in this work are tractable and fit well for the problem, however they are not the only uncertainty sets. There is an uncertainty set definition for robust discrete optimization proposed by Bertsimas and Sim [11] that defines the worst-case scenario as the worst subset of selections that could be made. In the case of daily fantasy football, this would be the subset of the selected lineup that when underperforming, impacts the lineup most negatively. The problem would be formulated as:

$$\max_{\boldsymbol{x}} \left\{ \boldsymbol{p}^T \boldsymbol{x} + \min_{\{\mathcal{Z} | \mathcal{Z} \subseteq \mathcal{P}, |\mathcal{Z}| \leq \Gamma\}} \sum_{j \in \mathcal{Z}} e_j x_j \right\} \tag{46}$$

Where:
$\mathcal{Z}$, the subset of the players $\mathcal{P}$
Γ, the number of players allowed to deviate from their projected performance

The first term in the objective of equation (46) is the same as the objective in equation (1). The second term defines the most negative impact of realizing the errors of a subset of the lineup. This formulation is fundamentally different than the one used for this work and could be useful as it doesn't assume that the worst case is happening for all players at the same time.

5 Conclusion

Depending on the daily fantasy football game type, it may not be the best strategy to select a lineup that maximizes the average point total. It is always the goal to maximize expected payout however, it's not always tractable to solve this problem. This work uses the robust optimization paradigm to maximize the worst-case scenario which may be advantageous for game types like 50/50. Robust optimization is interpretable, tractable and, under the formulation presented by this work, adaptable to any model used to generate projections. For the projections provided, the robust scenario selected lineups that increased the worst-case scenario so much so that it would have been the preferred lineup over standard MILP. The work will be continued by experimenting with other uncertainty sets. Furthermore, the work will be continued in other sports like basketball that have more games so that empirical performance of this formulation can be measured.

References

[1] DFSForecast. [Online]. Available: https://dfsforecast.com/

[2] R. Chase, 'Head-to-Head and 50/50 Strategies', Daily Fantasy Café, https://www.dailyfantasycafe.com/academy/graduate/head-to-head-50-50-strategy

[3] S. Newell, 'Optimizing daily fantasy sports contests through stochastic integer programming', Dept. Industrial and Manufacturing Systems Engineering, MS thesis, Kansas State University, Manhattan, KS, 2017

[4] M. B. Haugh & S. Singal, 'How to Play Strategically in Fantasy Sports (and Win)', 2018.

[5] Y. Yuan, Z. Li, & B. Huang, 'Robust optimization under correlated uncertainty: Formulations and computational study', *Computers & Chemical Engineering*, vol. 85, pp. 58–71, 2016.

[6] N. J. Higham, 'Computing a nearest symmetric positive semidefinite matrix', *Linear Algebra and its Applications*, vol. 103, pp. 103–118, 1988.

[7] M. Croucher, nearest_correlation, GitHub repository, https://github.com/mikecroucher/nearest_correlation, 2014

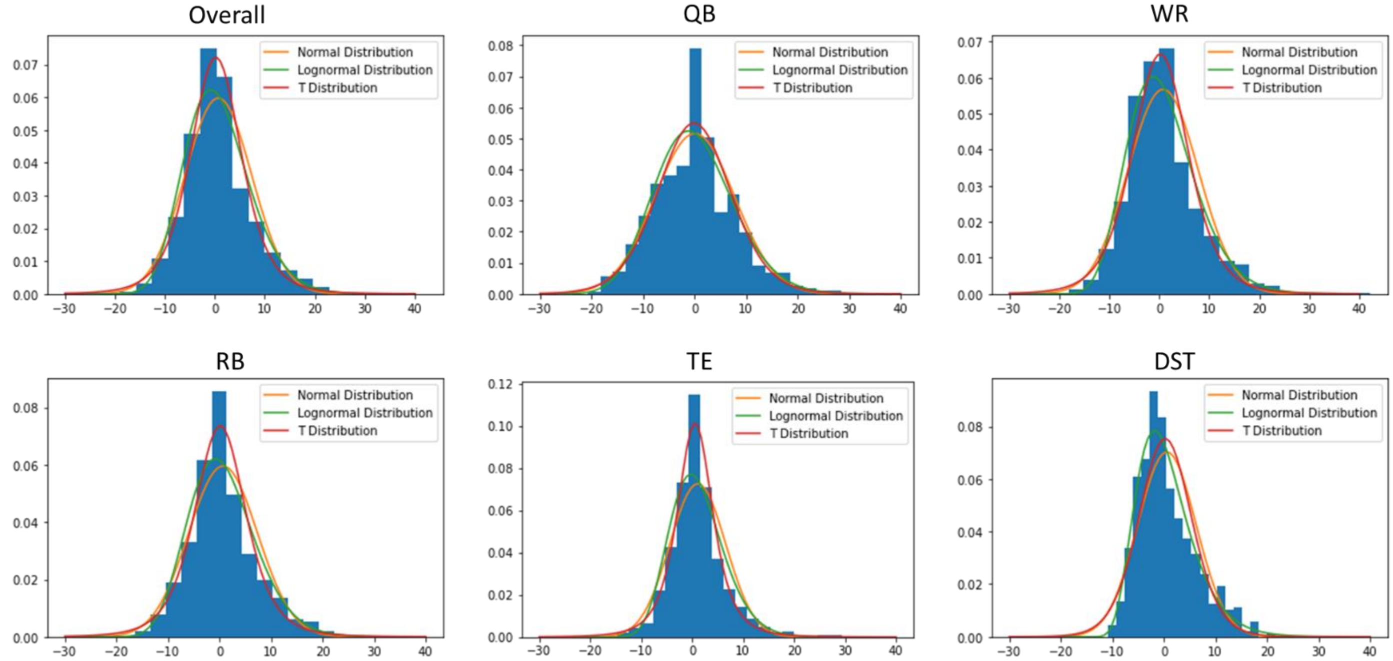
[8] N. J. Higham, 'Computing the nearest correlation matrix—a problem from finance', *IMA Journal of Numerical Analysis*, vol. 22, no. 3, pp. 329–343, July 2002.

[9] Fantasy Cruncher. [Online]. Available: https://www.fantasycruncher.com/contest-links/NFL/

[10] Pro Football Focus. [Online]. Available: https://www.pff.com/dfs/ownership/

[11] D. Bertsimas & M. Sim, 'Robust discrete optimization and network flows', *Mathematical Programming*, vol. 98, pp. 49–71, September 2003.

Appendix A: Error Distributions



Appendix B: Positional Correlations

Same Team	QB	WR	RB	TE	DST
QB	1	0.27	0.07	0.26	-0.1
WR	0.27	1	-0.01	0	-0.05
RB	0.07	-0.01	1	0	0.02
TE	0.26	0	0	1	-0.1
DST	-0.1	-0.05	0.02	-0.1	1

Opposing Team	QB	WR	RB	TE	DST
QB	0.22	0.09	0.07	0.1	-0.35
WR	0.09	0.04	0.01	0.06	-0.12
RB	0.07	0.01	0	0.05	-0.16
TE	0.1	0.06	0.05	0.02	-0.1
DST	-0.35	-0.12	0.02	-0.1	-0.31

Appendix C: Week 14 Covariance Matrices

MILP	Sony Michel	Jared Cook	Justin Herbert	Mike Williams	Seahawks	Antonio Gibson	Diontae Johnson	Josh Jacobs	Brandin Cooks
Sony Michel	27.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Jared Cook	0.0	24.2	-8.8	-27.3	0.0	0.0	0.0	0.0	0.0
Justin Herbert	0.0	-8.8	114.3	87.1	0.0	0.0	0.0	0.0	0.0
Mike Williams	0.0	-27.3	87.1	165.5	0.0	0.0	0.0	0.0	0.0
Seahawks	0.0	0.0	0.0	0.0	9.9	0.0	0.0	0.0	-6.4
Antonio Gibson	0.0	0.0	0.0	0.0	0.0	50.5	0.0	0.0	0.0
Diontae Johnson	0.0	0.0	0.0	0.0	0.0	0.0	36.8	0.0	0.0
Josh Jacobs	0.0	0.0	0.0	0.0	0.0	0.0	0.0	45.8	0.0
Brandin Cooks	0.0	0.0	0.0	0.0	-6.4	0.0	0.0	0.0	58.8

Poly	Sony Michel	Jared Cook	Ezekiel Elliott	Seahawks	Antonio Gibson	Taylor Heinicke	Chase Claypool	Diontae Johnson	Hunter Renfrow
Sony Michel	27.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Jared Cook	0.0	24.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Ezekiel Elliott	0.0	0.0	65.0	0.0	-9.9	-7.7	0.0	0.0	0.0
Seahawks	0.0	0.0	0.0	9.9	0.0	0.0	0.0	0.0	0.0
Antonio Gibson	0.0	0.0	-9.9	0.0	50.5	-15.4	0.0	0.0	0.0
Taylor Heinicke	0.0	0.0	-7.7	0.0	-15.4	44.8	0.0	0.0	0.0
Chase Claypool	0.0	0.0	0.0	0.0	0.0	0.0	40.4	-5.2	0.0
Diontae Johnson	0.0	0.0	0.0	0.0	0.0	0.0	-5.2	36.8	0.0
Hunter Renfrow	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	61.5

Box	Cooper Kupp	Sony Michel	Van Jefferson	Jared Cook	Mike Williams	Seahawks	Antonio Gibson	Taylor Heinicke	Javonte Williams

Cooper Kupp	82.9	-6.2	-21.3	0.0	0.0	0.0	0.0	0.0	0.0
Sony Michel	-6.2	27.9	-19.1	0.0	0.0	0.0	0.0	0.0	0.0
Van Jefferson	-21.3	-19.1	35.1	0.0	0.0	0.0	0.0	0.0	0.0
Jared Cook	0.0	0.0	0.0	24.2	-27.3	0.0	0.0	0.0	0.0
Mike Williams	0.0	0.0	0.0	-27.3	165.5	0.0	0.0	0.0	0.0
Seahawks	0.0	0.0	0.0	0.0	0.0	9.9	0.0	0.0	0.0
Antonio Gibson	0.0	0.0	0.0	0.0	0.0	0.0	50.5	-15.4	0.0
Taylor Heinicke	0.0	0.0	0.0	0.0	0.0	0.0	-15.4	44.8	0.0
Javonte Williams	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	41.0

Appendix D: Week 16 Covariance Matrices

MILP	Cooper Kupp	Jared Cook	Josh Palmer	Keenan Allen	Dak Prescott	Jonathan Taylor	Ronald Jones	Bengals	Braxton Berrios
Cooper Kupp	84.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Jared Cook	0.0	20.9	-4.4	-5.5	0.0	0.0	0.0	0.0	0.0
Josh Palmer	0.0	-4.4	11.5	-2.5	0.0	0.0	0.0	0.0	0.0
Keenan Allen	0.0	-5.5	-2.5	21.6	0.0	0.0	0.0	0.0	0.0
Dak Prescott	0.0	0.0	0.0	0.0	139.8	0.0	0.0	0.0	0.0
Jonathan Taylor	0.0	0.0	0.0	0.0	0.0	136.2	0.0	0.0	0.0
Ronald Jones	0.0	0.0	0.0	0.0	0.0	0.0	17.1	0.0	0.0
Bengals	0.0	0.0	0.0	0.0	0.0	0.0	0.0	23.2	0.0
Braxton Berrios	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	15.8

Poly	Cooper Kupp	Van Jefferson	Jared Cook	Josh Palmer	Keenan Allen	Ronald Jones	Alexander Mattison	Kirk Cousins	Bengals
Cooper Kupp	84.8	-22.1	0.0	0.0	0.0	0.0	-10.6	-12.2	0.0
Van Jefferson	-22.1	32.2	0.0	0.0	0.0	0.0	-6.6	7.4	0.0
Jared Cook	0.0	0.0	20.9	-4.4	-5.5	0.0	0.0	0.0	0.0
Josh Palmer	0.0	0.0	-4.4	11.5	-2.5	0.0	0.0	0.0	0.0
Keenan Allen	0.0	0.0	-5.5	-2.5	21.6	0.0	0.0	0.0	0.0
Ronald Jones	0.0	0.0	0.0	0.0	0.0	17.1	0.0	0.0	0.0
Alexander Mattison	-10.6	-6.6	0.0	0.0	0.0	0.0	24.2	-7.5	0.0
Kirk Cousins	-12.2	7.4	0.0	0.0	0.0	0.0	-7.5	48.3	0.0
Bengals	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	23.2

Box	Cooper Kupp	Jared Cook	Josh Palmer	Keenan Allen	Ronald Jones	Tom Brady	Alexander Mattison	Braxton Berrios	Jets
Cooper Kupp	84.8	0.0	0.0	0.0	0.0	0.0	-10.6	0.0	0.0
Jared Cook	0.0	20.9	-4.4	-5.5	0.0	0.0	0.0	0.0	0.0
Josh Palmer	0.0	-4.4	11.5	-2.5	0.0	0.0	0.0	0.0	0.0
Keenan Allen	0.0	-5.5	-2.5	21.6	0.0	0.0	0.0	0.0	0.0
Ronald Jones	0.0	0.0	0.0	0.0	17.1	-27.1	0.0	0.0	0.0
Tom Brady	0.0	0.0	0.0	0.0	-27.1	94.4	0.0	0.0	0.0
Alexander Mattison	-10.6	0.0	0.0	0.0	0.0	0.0	24.2	0.0	0.0
Braxton Berrios	0.0	0.0	0.0	0.0	0.0	0.0	0.0	15.8	-0.7
Jets	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.7	15.4