Team 6
Fantasy Football Team Optimization

Aditya Joglekar
Aditya Rathi
Jaydutt Kulkarni
Jeffrey Wang

Mechanical Engineering
Engineering Optimization - 24785

December 10 2021

1 Abstract

In Fantasy Premier League, individuals manage their own squad of players, with points based on real-life player performance. Managers select a squad of players within a set budget and attempt to predict the optimal squad to maximize their team's performance. We have taken existing Expected Value(EV) data from past performance data, and implemented techniques from Engineering Optimization to create a model that predicts optimal squad performance. The results that we got by solving the NLP version of the problem are in-line with expectations from real-life results. From this model, we have provided users with a squad which is most likely to outperform squads created by other users

2 Introduction

Fantasy Premier League is a game which allows individuals to run their own, virtual team. Fantasy Premier League players (also known as managers) select 15 players from real Premier League teams and then "play" them over the course of a season to get points in-game. Points are based off real-life performance; a player who plays better in real-life will produce more points in their fantasy performance. Points are based off a number of statistics, such as goals scored, goals against (for defenders and goalkeepers) and assists. In order to have the best fantasy team, fantasy players must choose the squad which produces the most points collectively. The caveat, however, is that managers must not exceed a standard budget of \$100 million. Each premier league player is given an associated cost which fluctuates over the season, tied to overall player popularity in fantasy teams. Players who consistently produce higher point performances naturally cost more than players who do not perform as well. Thus, the goal of a fantasy manager is to maximize the Expected Value of their squad by selecting players that will outperform any other feasible combination of players by providing the most points (not accounting for real-life variance).

Managers must select their squad of 15 from 4 different player positions: 2 goalkeepers, 5 defenders, 5 midfielders, and 3 forwards. To clarify between fantasy teams and real Premier League teams, we will refer to a fantasy team as a squad, and Premier League team (such as Liverpool) as a club. As an additional wrinkle, managers must select their starting 11 from their overall squad of 15. As players who perform better on average are inherently valued higher, the main objective of the manager is to find hidden value and maximize the performance per unit price. Some managers may elect to spend most of their budget on a few top-tier players and fill the remainder of their squad with average players, while others may try to spread the budget evenly between their squad.

While better premier league players may on average perform better in Fantasy Premier League, individual game performance is often much more nuanced. Team opponents, player health and fitness, and team strategy all may play roles in determining a player's fantasy impact. Fantasy managers need to weigh these criteria to formulate their own predictions about player performance. To help predict a player's Expected Value, managers can leverage tools which can take this data and more to provide the Expected Value for a week. Based on these Expected Values along with the players' associated costs, managers can try to maximize their squad performance.

Our objective is to create a mathematical model which can provide a user with a prediction for overall squads which will maximize their performance. In the following sections, we will detail our proposed process as well as dive deeper into the various constraints and mechanisms of the problem.

3 Problem Statement

The objective of our optimization problem is to maximize the Expected Value scored by our Fantasy Premier League team while keeping in line with the numerous constraints introduced by the problem. First and foremost is the constraint of the total cost of obtaining said team which should be less than or equal to \$100 Million. Secondly the total number of players in the team should be 11 in the starting lineup and 4 on the bench for a total of 15 players in the squad. Within these players there are further constraints based on the role of each player. The squad must have exactly 2 goalkeepers, 5 defenders, 5 midfielders and 3 forwards. Additionally, within the starting 11, there must be between 3 and 5 defenders, between 2 and 5 midfielders,

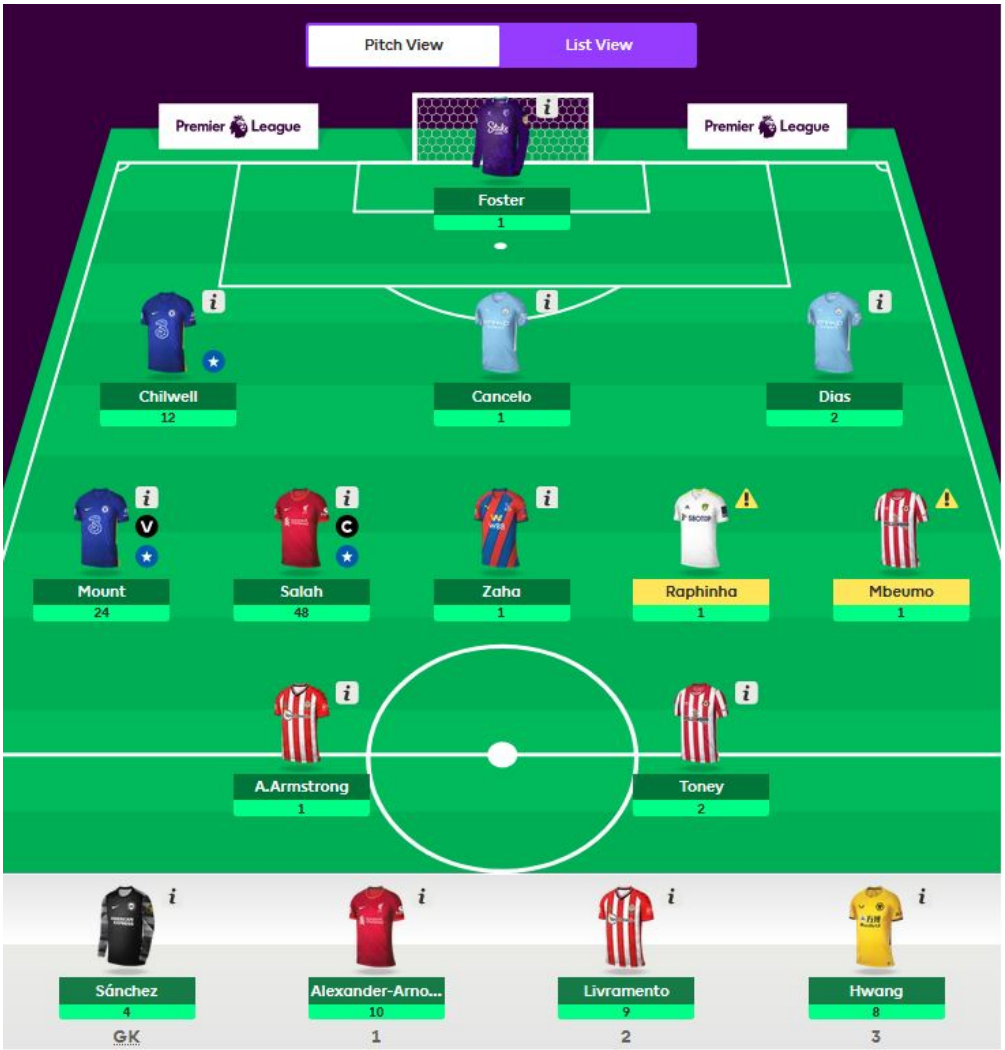

Figure 1: Sample Fantasy Premier League lineup

and between 1 and 3 forwards such that the total number of players within the starting 11 is maintained. Finally, there can be at most of 3 players from a single club (from which we pick our players) in our fantasy team. In most cases, a team's points will be the sum of the points scored by each member of his starting 11 squad. In some circumstances, there may be a scenario where a starter does not end up playing real-life and thus scores zero points. In this case, a member of the bench will be automatically substituted in place of said player [3].

Our objective function uses weight matrices to convert the player selection into Expected Value. These weight matrices represent the characteristics of each player but in a team level fashion i.e. each entry is the mean value of the players from one club. We will assume that the number of players within a position on our squad can take continuous values within the ranges specified above.

Mathematical Model

Linear Programming

Initially we set up the linear programming model to understand the various constraints and to set up our player variables

Minimize the negative of the expected Value

$$\underset{x_1, x_2 \in \mathbb{R}^{80}}{minimize} \;\; f(x_1, x_2) = -\sum_{i,j} W_{i,j}(rx_{1_{i,j}} + (1-r)x_{2_{i,j}})$$

where $\mathbf{X}_{i,j} = \begin{bmatrix} \mathbf{x_1}_{i,j} \\ \mathbf{x_2}_{i,j} \end{bmatrix}$

Subject to:

Max budget = \$100 Million:

$$g_1(\mathbf{X}) = \sum(x_{1_{i,j}} c_{i,j} + x_{2_{i,j}} c_{i,j}) - b_s \leq 0$$

Total Number of Players in Squad:

$$h_1(\mathbf{X}) = \sum_i \sum_j (\mathbf{x_{1_{i,j}}} + \mathbf{x_{2_{i,j}}}) - 15 = 0$$

Total Number of Players in Starting 11:

$$h_2(\mathbf{X}) = \sum_i \sum_j \mathbf{x_{1_{i,j}}} - 11 = 0$$

Number of Goalkeepers in Squad:

$$h_3(\mathbf{X}) = \sum(x_{1_{i,1}} + x_{2_{i,1}}) - 2 = 0$$

Number of Defenders in the squad:

$$h_4(\mathbf{X}) = \sum(x_{1_{i,2}} + x_{2_{i,2}}) - 5 = 0$$

Number of Midfielders in the squad:

$$h_5(\mathbf{X}) = \sum(x_{1_{i,3}} + x_{2_{i,3}}) - 5 = 0$$

Number of Forwards in the squad:

$$h_6(\mathbf{X}) = \sum(x_{1_{i,4}} + x_{2_{i,4}}) - 3 = 0$$

Maximum number Defenders in the starting 11:

$$g_2(\mathbf{X}) = \sum x_{1_{i,2}} - 5 \leq 0$$

Minimum number of Defenders in the starting 11:

$$g_3(\mathbf{X}) = -\sum x_{1_{i,2}} + 3 \leq 0$$

Maximum number of Midfielders in the starting 11:

$$g_4(\mathbf{X}) = \sum x_{1_{i,3}} - 5 \leq 0$$

Minimum number of Midfielders in the starting 11:

$$g_5(\mathbf{X}) = -\sum x_{1_{i,3}} + 2 \leq 0$$

Maximum number of Forwards in the starting 11:

$$g_6(\mathbf{X}) = \sum x_{1_{i,4}} - 3 \leq 0$$

Minimum number of Forwards in starting 11:

$$g_7(\mathbf{X}) = -\sum x_{1_{i,4}} + 1 \leq 0$$

Number of Goalkeepers in the starting 11:

$$h_7(\mathbf{X}) = \sum x_{1_{i,4}} - 1 = 0$$

Maximum number of players in the squad from each club:

$$g_k(\mathbf{X}) = \sum_j x_{1_{k-7,j}} - 3 \leq 0 \;\forall\; k \in [8, 27]$$

Finally table 1 introduces all of the variables, symbols, and parameters for the linear problem defined.

	Description	Value	Units
$\mathbf{x_{1_{i,j}}}$	Starting Lineup; i^{th} club; j^{th} position	-	-
$\mathbf{x_{2_{i,j}}}$	Bench Lineup; i^{th} club; j^{th} position	-	-
b_s	Starting Budget	100 Million	$
$\mathbf{W_{i,j}}$	Weight matrix to convert to Expected Value per team per position	-	-
$\mathbf{c_{i,j}}$	Cost matrix to convert to money spent per team per position	-	$
r	Probability of a starting lineup player playing a match	0.9	-
j=1	Goalkeeper	-	-
j=2	Defender	-	-
j=3	Midfielder	-	-
j=4	Forward	-	-
i	Club Number	-	-

Table 1: Symbols and Parameters

Non-Linear Programming

For the Non-linear case, the probabilities that a given player will play on the field was converted into a variable. This is represented by the expected minutes per position per club i.e. X_{min}. X_{min} is the number of minutes (averaged per club per position) that a player plays on the field, converted into probabilistic values between 0 and 1. Thus players with a higher X_{min} value tend to play more and hence have a higher probability of playing any given game. By optimizing the value of X_{min} we hope to create a better FPL squad than the competition. Adding X_{min} will make the model more realistic which in turn will allow the solvers to find a more optimal solution

Thus the new objective function becomes

$$\underset{\mathbf{x_1}, \mathbf{x_2}, \mathbf{X_{min}} \in \mathbb{R}^{80}}{minimize} \; f(\mathbf{x_1}, \mathbf{x_2}, \mathbf{X_{min}}) = -\sum_{i,j} W_{i,j} \left(X_{min_{i,j}} x_{1_{i,j}} + \left(1 - \frac{\sum \mathbf{X_{min}} \mathbf{x_1}}{11}\right) X_{min_{i,j}} x_{2_{i,j}} \right)$$

Subject to:

Minimum value of X_{min}:

$$g_{28}(\mathbf{X_{min}}) = -\mathbf{X_{min}} \leq 0$$

Maximum value of X_{min}:

$$g_{29}(\mathbf{X_{min}}) = \mathbf{X_{min}} - 1 \leq 0$$

Maximum number of bench players:

$$g_{30}(\mathbf{x_1}, \mathbf{X_{min}}) = -\sum \mathbf{X_{min}} \mathbf{x_1} + 7 \leq 0$$

	Description	Value	Units
$\mathbf{X_{min}}$	Expected minutes per position per club	-	-

Table 2: NLP symbols and parameters

Parameter	Base Value	Sensitivity Values	Unit
b_s	100	80-130	Million $
r	0.9	0.5-1	-

Table 3: Sensitivity of Parameters

Sensitivity of Parameters

Table 3 documents the sensitivity of the parameters present in our problem. We have varied these parameters within practically acceptable ranges for FPL to characterize their effect on the Expected Value of our team.

Model Improvements

After carefully reviewing the optimization checklist from the textbook by Dr. Papalambros we found some improvements that can be made to the problem at hand. First and foremost is the computational complexity of the problem because of the non-linearity and the presence of multiple local minimas. Hence it takes a long time for all of the solvers to find a solution. So according to the optimization checklist we can look into creating meta-models for the expected Value at predetermined intervals. Curve fitting on this data will drastically improve the computational performance allowing a larger set of parameters to be tested. Secondly, since the model is algebraic we plan to replace the finite differentiation with actual gradients to make the model more robust. Lastly, a new model at a higher hierarchy level can be made to increase decision-making options and thereby likely reach to a more optimal solution.

4 Analysis of Problem Statement

A few modeling assumptions were made that were necessary to make the problem solvable within the given time-frame. Red cards (player being sent-off the pitch due to dangerous play, or denying obvious goal-scoring chances via illegal means) have not been considered while modeling the $\mathbf{X_{min}}$ variable, since there is a possibility that the minutes played by all players in a particular club (in real-life) do not add up to 990.

The following concepts have also not been incorporated into the optimization model for the sake of limiting the complexity of the problem:

1. Multiple Match Week Optimization: The impact of transfers (transferring players in and out of the FPL team) across multiple weeks. Currently, the optimization model only looks at a 1-week window.
2. Chips: The impact of utilizing chips such as Wildcard, Free-Hit, Bench Boost, and Triple Captain, on the optimal solution. These chips help boost the user's score for a given Match Week.
3. Price changes: Incorporate a dynamic player price change model, which is in line with the cost constraints that would change depending on the team value for a given Match Week.

Natural vs Practical Constraints

The rules of the real-life game of football are considered as the natural constraints. These include the total number of players that can play on the field, the total minutes that can be played by a particular team in a football match, and the minimum number of players per position. The practical constraints are the constraints implemented in Fantasy Premier League for making the game more engaging, which include the maximum amount spent on the players and the maximum number of players in the squad from each club.

Problem Class

The first part of the project focuses on a reduced-LP version of the problem. This is because the objective function and all the constraints are simply an affine combination of variables.
The second and major part focuses on the NLP version of the problem, where we have the variable associated with probability of players playing ($\mathbf{X_{min}}$) getting multiplied with the squad selection variables ($\mathbf{X_1}$ and $\mathbf{X_2}$ in the objective function.

Continuity and Smoothness

Since none of the variables are discrete, and are in $\mathbb{R}^n$, there are no discontinuities in any of the functions. Also, since the LP version is simply an affine combination, and the NLP version of the objective function is a hyperbola, there are no functions with derivative discontinuities.

Convexity

For the LP version, it is convex by default (linear problem).
For the NLP version, the objective function is non-convex, when looking at the function without any practical constraints (values need to be greater than or equal to zero). This can also be figured out by looking at the Hessian of the objective function, which is indefinite by nature. However, the objective function is convex in the tangent subspace, i.e., where all variables ≥ 0. This implies that the problem itself is non-convex, even though all inequality constraints, and equality constraints are linear.

Undefined Regions

The objective function does not have any regions where it or its gradients are undefined.

Size

In the LP problem, we have 160 decision variables, 7 equality constraints and 27 inequality constraints.
In the NLP problem, we have 240 decision variables, 7 equality constraints and 30 inequality constraints.

5 Optimization Study

Numerical method and software used

For the base linear programming case, we used both Excel and MATLAB to find the solution to the problem. Excel offers both the simplex and the GRG algorithms for solving the programming problem while MATLAB offers a host of numerical NLP solvers such as SQP, Active-Set and Interior Point. This offered us the opportunity to compare the performance of Non-Linear algorithms with simplex, of which the latter can exploit the structure of the problem to be much more efficient and robust.

MATLAB was used for solving the formulated NLP problem, and its reduced LP version as well. It had an advantage over Excel of ability to optimize more than 200 variables, which was required as our optimization problem has 240 variables. The fmincon algorithm in MATLAB was used for optimization. The SQP, Active-Set and Interior Point algorithms were tried in fmincon. SQP showed the best results. It generated different local minima depending on starting point. For values closer to zero and below zero as initialization, SQP converged to a sub-optimal solutions with an objective function value in the range of -55 to -57. For values closer to one and above one as initialization, SQP found a global optimum solution of -59.7.

Solution to base case and Interpretation

Linear Problem

We set up our linear programming case as shown in Fig. 2. The first four columns represent the weight matrix W and the next four represent the cost matrix c. Finally the decision variables are shown in the pink cells with the first set representing the starting lineup and the next set representing the bench players. Using simplex we were able to find a global solution with the objective function taking a value of -55.28. Using GRG initially the algorithm was unable to converge to this solution because the second order derivative is zero and hence the algorithm takes much longer to converge. However, after tuning the convergence criteria and the maximum allowed number of iterations we were able to get GRG to converge to the same solution as simplex.

Club\Position	GK	DEF	MID	FWD	Cost	GK	DEF	MID	FWD	Starting XI	GK	DEF	MID	FWD	Bench	GK	DEF	MID	FWD
Arsenal	3.243033708	1.840396574	3.550886855	3.905475614		4.7	4.566666667	5.488889	9.2		0	0	0	0		0	0	0	0
Newcastle	3.699310345	3.218392006	3.815373269	4.83505564		4.45	4.371428571	5.0375	6.3		0	0	0	0		0	0	0	0
Aston Villa	3.791123596	3.196238826	3.989889949	4.685785714		4.75	4.742857143	5.3125	7.5		0	0	0	0		0	0	0	0
Man City	4.649662921	5.288873906	6.180533746	5.67875		6	5.571428571	7.966667	8.7		0	0	3	0		0	0	0	0
Brighton	3.658264286	3.379831637	3.845475774	4.972257353		4.25	4.571428571	5.4125	5.4		0	0	0	0		0	0	0	0
Brentford	3.69	3.485677554	4.113698273	5.191058824		4.5	23.48	5.145455	6		0	0	0	0		0	0	0	0
Burnley	3.671797753	3.388845396	3.55256337	5.165016234		5.4	4.528571429	5.2	5.625		0	0	0	0		0	0	0	0
Chelsea	4.063295455	4.353844362	4.758248707	5.276131579		6.2	5.5875	6.154545	10.05		0	3	0	0		0	0	0	0
West Ham	3.706179775	3.454973957	4.20773224	4.547142857		5	4.757142857	5.51	8.2		0	0	0	0		0	0	0	0
Crystal Palace	3.833595506	3.239142342	4.184599096	4.52039839		4.5	4.385714286	5.433333	6.2		0	0	0	0		0	0	0	0
Norwich	3.433146067	2.79636895	3.808929706	4.450676161		4.5	4.275	4.788889	5.366667		0	0	0	0		0	0	0	0
Everton	3.170224719	1.652191982	2.98960775	3.586475904		5	4.728571429	5.044444	7.133333		0	0	0	0		0	0	0	0
Leeds	3.343146067	2.460544432	3.494466041	4.203355263		5	4.416666667	5.118182	5.4		0	0	0	0		0	0	0	0
Leicester	3.363370787	2.706212593	3.577522863	4.288758295		5	4.825	5.544444	8.3		0	0	0	0		0	0	0	0
Liverpool	3.999886364	4.35103803	5.03450532	5.94		6	5.9	6.89	4.9		0	0	1	0.516176		0	0	0	0
Man Utd	4.003448276	4.134971171	4.914251823	6.434006969		5	16.85	7.188889	10.4		0.516176	0	0	2.483824		0	0	0	0
Southampton	3.92258427	3.697386378	4.279946315	5.064724346		4.6	4.516666667	5.28	5.9		0.483824	0	0	0		1	0	1	0
Spurs	3.836629213	3.881101727	4.539949627	6.015176471		5.4	4.7	6.25	12.2		0	0	0	0		0	2	0	0
Watford	3.241011236	2.225191319	3.336603218	3.950533077		4.1	4.3375	5.066667	5.3		0	0	0	0		0	0	0	0
Wolves	3.636818182	3.168868621	3.929109928	4.605077922		5	4.55	5.383333	6.4		0	0	0	0		0	0	0	0

Figure 2: Solution to the linear problem using Excel

Starting Players						Bench Players			
Club\Position	GK	DEF	MID	FWD	Club\Position	GK	DEF	MID	FWD
Arsenal	0	0	0	0	Arsenal	0	0	9.02E-04	0
Newcastle	0	0	0	0	Newcastle	0.0093	0	0.0049	0
Aston Villa	0	0	0	0	Aston Villa	0	0	0	0
Man City	0	0.2827	2.7173	0	Man City	0	0	0	0
Brighton	0	0	0	0	Brighton	0.9896	0	0	0
Brentford	0	0	0	0	Brentford	0	0	0.02	0
Burnley	0	0	0	0	Burnley	0	0	0	0
Chelsea	0.3901	1.5613	0	0	Chelsea	0	0	0	0
West Ham	0	0	0	0	West Ham	0	0	0	0
Crystal Palace	0	0	0	0	Crystal Palace	0	0	0	0
Norwich	0	0	0	0	Norwich	0	0	0	0
Everton	0	0	0	0	Everton	0	0	0	0
Leeds	0	0	0	0	Leeds	0	0	0	0
Leicester	0	0	0	0	Leicester	0	0	0	0
Liverpool	0	1.1561	1.2827	0.549	Liverpool	0	0	0	0
Man Utd	0.549	0	0	2.451	Man Utd	0	0	0	0
Southampton	0.0609	0	0	0	Southampton	0.0012	0.8617	0.9742	0
Spurs	0	0	0	0	Spurs	0	1.1383	0	0
Watford	0	0	0	0	Watford	0	0	0	0
Wolves	0	0	0	0	Wolves	0	0	0	0

Figure 3: Linear Problem Solution

MATLAB gives -55.2 as the objective function value for the linear problem.As seen in Fig. 3, the solution's starting lineup includes more number of players from high average expected score value teams like Manchester City, Liverpool and Manchester United. Moreover, some nuances like Manchester City midfielders have really high expected score values are taken into consideration by the model and most midfielders in the output fantasy team are from Manchester City. The bench players are from lesser known clubs with average expected Value values on the lower side. This is due to the cost constraint restriction taken into account by the model.

Starting Players

Club\Position	GK	DEF	MID	FWD
Arsenal	0	0	0	0
Newcastle	0	0	0	0
Aston Villa	0	0	0	0
Man City	0	0.9334	2.0666	0
Brighton	0	0	0	0
Brentford	0	0	0	0
Burnley	0	0	0	0
Chelsea	1	1.4806	0	0
West Ham	0	0	0	0
Crystal Palace	0	0	0	0
Norwich	0	0	0	0
Everton	0	0	0	0
Leeds	0	0	0	0
Leicester	0	0	0	0
Liverpool	0	0.586	1.9334	0.29
Man Utd	0	0	0	2.71
Southampton	0	0	0	0
Spurs	0	0	0	0
Watford	0	0	0	0
Wolves	0	0	0	0

Bench Players

Club\Position	GK	DEF	MID	FWD
Arsenal	0	0	0	0
Newcastle	0	0	0	0
Aston Villa	0	0	0	0
Man City	0	0	0	0
Brighton	0	0	0	0
Brentford	0	0	0	0
Burnley	0	0	0	0
Chelsea	0	0	0	0
West Ham	0	0	0	0
Crystal Palace	0	0	0	0
Norwich	0	1.3552	1	0
Everton	0	0	0	0
Leeds	0	0	0	0
Leicester	0	0	0	0
Liverpool	0	0	0	0
Man Utd	0	0	0	0
Southampton	0	0	0	0
Spurs	0	0	0	0
Watford	1	0.6448	0	0
Wolves	0	0	0	0

Xmin

Club\Position	GK	DEF	MID	FWD
Arsenal	0.9989	0.9974	0.9794	0.9828
Newcastle	0.9809	0.9763	0.982	0.9918
Aston Villa	0.9818	0.976	0.9837	0.9904
Man City	0.9878	1	1	1
Brighton	0.9805	0.9778	0.9823	0.9932
Brentford	0.9808	0.9788	0.9849	0.8488
Burnley	0.9806	0.9779	0.9794	0.9973
Chelsea	1	1	0.9769	0.2577
West Ham	0.9809	0.9786	0.9988	0.989
Crystal Palace	0.9913	0.9765	0.9855	0.9888
Norwich	0.9783	0.9723	0.9821	0.9881
Everton	0.9995	0.9991	1	0.9797
Leeds	0.9771	0.9688	0.9789	0.9857
Leicester	0.9776	0.9712	0.9797	0.9865
Liverpool	0.9286	1	1	1
Man Utd	0.7795	0.9944	0.9613	1
Southampton	0.9251	0.9987	0.9994	0.9941
Spurs	0.988	0.9998	0.9868	0.9637
Watford	0.9992	0.9667	0.9767	0.9833
Wolves	0.9891	0.9757	0.9831	0.9896

Figure 4: Non-linear Problem Solution

Termination	*Exitflag*: 1 *Termination message*: Local minimum found that satisfies the constraints. Optimization completed because the objective function is non-decreasing in feasible directions, to within the value of the optimality tolerance, and constraints are satisfied to within the value of the constraint tolerance. The optimizer found a local minimum for the given starting point.
Local Optimality	The KKT conditions are satisfied. The output given by the fmincon function states that first order optimality conditions are satisfied and also that a local minimum has been found. Thus, second order optimality conditions are satisfied as well.
Global Optimality	Different starting points give different solutions. Initialization with -10, -5, -1, 0, random initialization from 0 to 1, 1, 5 and 10 has been tried. As stated previously, the solutions found with initialization from 1 and above, and sometimes in random initialization from 0 to 1, are always better than solutions found with 0 as initialization. The given solution can be considered a global optimum.
Uniqueness	There are more than one local minimum present. The other local minima are sub-optimal solutions of the problem where the distribution of players is not the best for maximizing the expected value. The mentioned global optimum solution is found only with the given initialization values of ones. However, there are multiple solutions in the vicinity(10e-2) of this global optimum.

Table 4: Analysis of Solution

Non-Linear Problem

For the Non-Linear Problem, Matlab's SQP algorithm was used. The solution obtained is shown in Fig. 4 and as expected is better than the the LP case due to the increased flexibility. The Expected Value obtained from the NLP case is -59.7 compared to -55.2 for LP. Obtaining this solution required a lot of tuning with the solver options to ensure consistent convergence of the problem. Similar to the LP case, starting players are from teams with high average Expected Values. Most of the resulting X_{min} values obtained are closer to 1 hinting towards the fact that the starting lineup has a much higher effect on the overall Expected Value of the squad compared to the bench players. Hence, to maximize the Expected Value focus should be on the starting 11 compared to the bench players.

Physical sense of the solution

Our fantasy team is chosen by the algorithm to optimize the Expected Value. Partly, as expected, we have a large number of players from the best clubs in the premier league as they have high weight values while taking into account the cost constraint. Fig. 5 represents a possible team that our algorithm has predicted which was obtained by rounding off non-integer values to their closest integers. Furthermore, because the value of X_{min} tends to 1, it implies that by purchasing the best players, real-life uncertainties such as such as player injuries are mitigated enough to not warrant spending much on the substitute players.

Lagrange Multipliers

The lagrange multipliers for all of the active constraints are documented in table 5. All other constraints have lagrange multiplier of zero as they are not active.

Constraint	Lagrange Multiplier	Type of Constraint
Total number of players in squad	-92592	Active
Total number of starting players	3.7416	Active
Number of goalkeepers in squad	92592	Active
Number of defenders in squad	92592	Active
Number of midfielders in squad	92592	Active
Number of forwards in squad	92592	Active
Number of starting goalkeepers	-0.9547	Active
Budget	0.0861	Active
Minimum number of defenders starting lineup	0.6279	Active
Maximum number of forwards starting lineup	1.1028	Active
Chelsea Goalkeeper Starting Probability	2.9678	Active
Manchester City Defenders Starting Probability	3.9142	Active
Chelsea Defenders Starting Probability	4.8243	Active
Liverpool Defenders Starting Probability	1.9076	Active
Manchester City Midfielders Starting Probability	10.5084	Active
Liverpool Midfielders Starting Probability	7.6158	Active
Liverpool Forwards Starting Probability	1.4047	Active
Manchester United Forwards Starting Probability	14.4675	Active
Manchester City Maximum Players	0.9721	Active

Table 5: Lagrange Multipliers

6 Sensitivity Analysis

Taking the solution with $r = 0.9$, i.e. $X_{min} = 0.9$ in LP, we can use the calculated Lagrange multipliers to perform a sensitivity analysis on our results. The results are shown on here in completeness. Note that Lagrange multipliers for the r values are excluded, as these are overall considered as part of the parametric study and not a decision variable. In Table 6 we can see that most of our constraints are inactive.

Of the 27 inequality constraints, only a handful are active and significant to the solution. Some of the more important Lagrange multipliers include the budget, minimum and maximum number of players at a certain position, and a few limitations on high-performing team numbers. For the budget, there is a 0.06 expected point increase per dollar increase in budget. This is a fairly low number, showing that the other constraints have a bigger impact in limiting the Expected Value than the budget constraint. At 0.48 more Expected Value per one fewer defender in starting lineup allowed, and 0.82 more Expected Value per one additional forward in the starting lineup allowed, these active constraints show the domination of forwards / attackers in producing expected value for our team's performance. This shows that the limitation of the makeup of our starting roster is limiting the performance of our team, which would be further increased by

Figure 5: Solution (Best Possible Fantasy Premier League Team)

swapping some defenders for forwards. In terms of constraints for maximums of player limits per club, we can see high multipliers for teams such as Manchester City (0.99 more Expected Value per one additional player allowed from the club), Liverpool (0.11 more Expected Value per additional player), and Manchester United (0.28 more Expected Value per additional player). As these teams consist of some of the highest performing players, the limitation of 3 players maximum per club is putting a hard limit on the maximum contributions by players of these clubs, limiting our overall performance.

One value which seems to be more reasonable is the 2.96 Expected Point increase by increasing our starting lineup by one player. Of course, this is impossible due to the rules of the game, but this represents the marginal benefit of one additional player's contributions. Interestingly, there is a -0.77 multiplier for the number of starting goalkeepers–this number likely indicates that the addition of another goalkeeper into the starting lineup (and subtraction of another player in his place) would also decrease the Value, meaning that goalkeepers contribute less than other players.

Parametric Study

In order to determine how our model reacts to variations in expected minutes of specific team player, we designed a parametric study. By varying our expected minutes probability matrix from values of 0.5 to 1, with a step size of 0.01, we were able to determine the effect of this parameter on our optimal solution. In order to conduct this, we created new equality constraints setting each of the X_{min} values of our matrix to their respective values.

$$h_k(\mathbf{X_{min}}) = \mathbf{x_{m,i,j}} - r \ \forall \ k \in [8, 88] \qquad \text{where, } r = 0.5 : 0.01 : 1$$

In order to simplify operation, the parametric study was run with fixed x_0 consisting of all ones for the $\mathbf{x_1}$, $\mathbf{x_2}$, and $\mathbf{X_{min}}$ matrices. We ran fmincon with SQP, and the maximum function evals and iterations both

Inequality constraints	Lagrange Multipliers	Inequality Constraints	Lagrange Multiplers
Budget	0.06	Chelsea Max. Players	0.00
Min. Defenders Starting Lineup	0.48	West Ham Max. Players	0.00
Max. Forwards Starting Lineup	0.82	Crystal Palace Max. Players	0.00
Manchester City Max. Players	0.99	Norwich City Max. Players	0.00
Liverpool Max. Players	0.11	Everton Max. Players	0.00
Manchester United Max. Players	0.28	Leeds Max. Players	0.00
Arsenal Max. Players	0.00	Leicester Max. Players	0.00
Newcastle United Max. Players	0.00	Southampton Max. Players	0.00
Aston Villa Max. Players	0.00	Tottenham Max. Players	0.00
Brighton Max. Players	0.00	Watford Max. Players	0.00
Brentford Max. Players	0.00	Wolves Max. Players	0.00
Burnley Max. Players	0.00		

Table 6: Lagrange multipliers for practical constraints

set to infinity. Below are our results showing the Expected Value for a team based on a set X_{min} value:

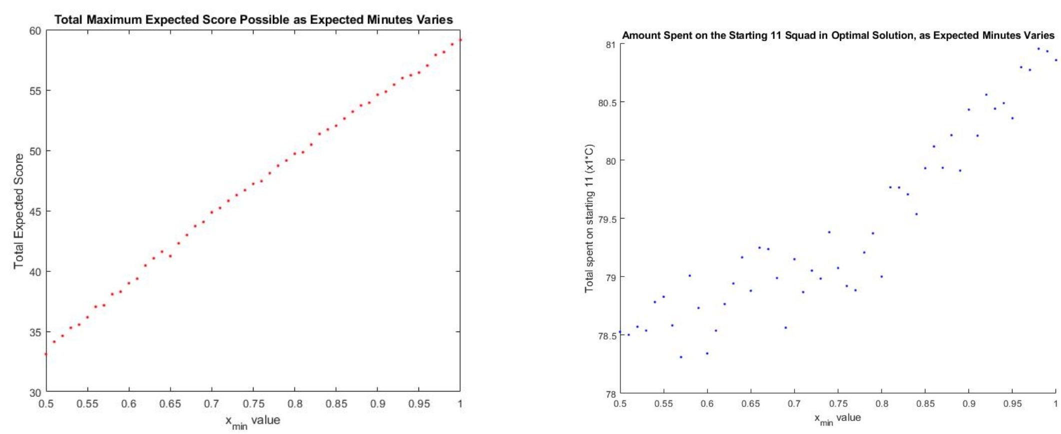

Figure 6: A: Expected Value vs X_{min} 1 B: Budget for starting 11 vs X_{min}

As seen in Fig. 6A, the Expected Value increases as the expected minutes of our players increases. This is a logical conclusion, as the more minutes we expect our players to play, the more chances they will have to score points for our team. The relationship is very linear–as the expected minutes increase, the expected points increases almost proportionally. For 0.01 increase in the value of r, we get 0.51 Expected Value from it. This is a direct result of the overall weighting of which the X_{min} matrix has on our objective function. More interesting that this linear relationship, however, is the percentage of our budget spent on our starting squad.

As seen in Fig. 6B, as the probability of our players playing decreases, the optimal solution tends towards spending more money on our bench lineup (and therefore, less money on our starting lineup). While this difference is not much (around 78.5 for an X_{min} probability of 0.5 vs. around 81 for a probability of 1), this small change represents a significant increase in the bench budget, rising from \$19 to \$21.5, an increase of 13%. Most importantly, this shows that we cannot simply maximize starting 11 performance (our x_1), and must also balance out a stronger bench. The actual X_{min} in real life may vary from a variety of factors, but this allows users of this tool to set an optimal lineup, given a user-estimated X_{min} probability model. Especially in times where players cannot play matches due to COVID, injuries, or manager decisions, this analysis provides us more flexibility to maximize expected performance. To further analyze the difference in results, we can take a look at the breakdown in out X matrices. In comparing the solutions between $r = 0.5$ and $r = 0.9$, we notice that there are some significant differences in which player selection leads to the

	X1				X2			
	Goalies	Defenders	Midfielders	Forwards	Goalies	Defenders	Midfielders	Forwards
Arsenal	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Newcastle	0.00	0.00	0.00	0.00	0.07	0.00	0.00	0.00
Aston Villa	0.01	0.00	0.00	0.00	**0.12**	0.00	0.00	0.00
Man City	**0.33**	**0.59**	**1.75**	0.00	0.00	**0.19**	**0.14**	0.00
Brighton	0.00	0.00	0.00	0.00	0.05	0.00	0.00	0.00
Brentford	0.00	0.00	0.00	0.00	0.05	0.00	0.01	0.00
Burnley	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Chelsea	**0.23**	**1.38**	**0.52**	0.00	0.04	**0.57**	**0.26**	0.00
West Ham	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00
Crystal Palace	0.04	0.00	0.00	0.00	**0.21**	0.00	0.02	0.00
Norwich	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Everton	0.00	0.00	0.00	0.00	0.00	**0.62**	0.00	0.00
Leeds	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Leicester	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Liverpool	0.00	**0.93**	**0.94**	**0.92**	0.00	**0.14**	0.08	0.00
Man Utd	**0.18**	0.00	**0.78**	**1.75**	0.09	0.00	**0.19**	0.00
Southampton	**0.21**	0.00	0.00	0.00	**0.30**	**0.15**	**0.11**	0.00
Spurs	0.00	0.10	0.00	**0.34**	0.07	**0.33**	**0.16**	0.00
Watford	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Wolves	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Figure 7: $x_\dagger$ for the optimal solution at $r = 0.5$

	X1				X2			
	Goalies	Defenders	Midfielders	Forwards	Goalies	Defenders	Midfielders	Forwards
Arsenal	0.00	0.00	0.00	0.00	0.06	0.00	0.00	0.00
Newcastle	0.00	0.00	0.00	0.00	0.07	0.02	0.01	0.00
Aston Villa	0.00	0.00	0.00	0.00	0.06	0.00	0.01	0.00
Man City	0.00	**0.80**	**2.20**	0.00	0.00	0.00	0.00	0.00
Brighton	0.00	0.00	0.00	0.00	0.08	0.05	0.00	0.00
Brentford	0.00	0.00	0.00	0.00	0.05	0.00	0.06	0.00
Burnley	0.00	0.00	0.00	0.00	0.00	0.06	0.00	0.00
Chelsea	**0.62**	**1.25**	0.06	0.00	0.00	**0.31**	**0.15**	0.00
West Ham	0.00	0.00	0.00	0.00	0.00	0.05	0.03	0.00
Crystal Palace	0.00	0.00	0.00	0.00	**0.12**	0.02	0.04	0.00
Norwich	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00
Everton	0.00	0.00	0.00	0.00	**0.24**	**1.01**	**0.55**	0.00
Leeds	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Leicester	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Liverpool	0.07	**0.95**	**1.28**	**0.65**	0.00	0.04	0.00	0.00
Man Utd	**0.20**	0.00	**0.46**	**2.35**	0.00	0.00	0.00	0.00
Southampton	**0.11**	0.00	0.00	0.00	**0.14**	**0.20**	0.09	0.00
Spurs	0.00	0.00	0.00	0.00	0.00	**0.25**	0.04	0.00
Watford	0.00	0.00	0.00	0.00	**0.18**	0.00	0.00	0.00
Wolves	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Figure 8: $x_\dagger$ for the optimal solution at $r = 0.9$

optimal solution. In Fig. 7 we see higher values in several categories, notably Chelsea starting midfielders, Manchester City starting goalkeepers, and Spurs forwards. We also see the selection of increased defenders and midfielders from Manchester City on our bench squad. Manchester City defenders and midfielders are above average, and thus increasing our selection of these in our bench squad highlights the increased potential from our bench players.

7 Conclusions

The goal of this project was to try and find an EV(Expected Value)-optimal solution for a Fantasy Premier League squad, which will provide the user an advantage in the game. This solution should adhere to the numerous constraints provided by the game itself, such as budget, number of players that can be selected from a club, and the number of players per position. The model of this problem was broken down into two phases, i.e., Linear Programming and Non-Linear Programming.

In the LP phase, we simplified the problem by assuming a static value for the probabilities of FPL assets playing in real-life. Then by using Simplex and GRG algorithm, this version of the problem was solved to find a common minima. This was done in both Microsoft Excel as well as MATLAB for verification purposes. Once we were sure that the model produces the correct results, this was extended to introduce further complexity in the problem, i.e., by having variable probabilities instead of static, scalar multiples.

In the NLP version of the problem, we used a host of techniques such as SQP, Active-Set, and Interior

Point algorithms. By varying function parameters of the solver, fmincon, we went through a lot of cases where we got different exitflags based on function parameter values and starting points. In some cases, the algorithms were allowed to run overnight by removing constraints on maximum number of iterations. All possible combinations were tried until we reached a solution for which multiple algorithms had converged, which turned out to be a better solution than the static LP problem, as expected.

The given solution can be interpreted as shown in Fig. 5. It suggests that the goalkeeper and defenders should be selected that play against the clubs with the weakest attacks (in real life), which will increase the probability of them not conceding a goal and thus earning more points. In the case of midfielders and forwards, the optimal solution suggests that those playing against the weakest defences should be selected for the given Match week, which increases the odds of them scoring/assisting a goal and thus earning points. The objective function value itself is consistent with the scale of points scored in FPL as well.

References

[1] T. Matthews, S. Ramchurn, and G. Chalkiadakis, "Competing with humans at fantasy football: Team formation in large partially-observable domains," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 26, 2012.

[2] B. K. Kristiansen, A. Gupta, and W. Eilertsen, "Developing a forecast-based optimization model for fantasy premier league," M.S. thesis, NTNU, 2018.

[3] P. League. "Fantasy premier league rules." (), [Online]. Available: `https://fantasy.premierleague.com/help/rules`.

Appendix 1: Code

```
    clc
clear all
close all
syms  x1 x2 [20 4]
syms x1_flat x2_flat c_flat w_flat X_min  [1 80]
X = [x1;x2];

%% Setup LP
W = readmatrix('Exp_Score.csv');
C = readmatrix('Cost.csv');

W_flat = reshape(W,1,[]);
C_flat = reshape(C,1,[]);



% f = -sum((0.9*W_flat.*x1_flat+0.1*W_flat.*x2_flat),'all');
budget = 100;
numgoalkeepers = 2;
numdefenders = 5;
nummidfielders = 5;
numforwards = 3;
numplayers = 15;
numstarters = 11;

minstartingdefenders = 3;
minstartingmidfielders = 2;
minstartingforwards = 1;
numstartinggoalkeepers = 1;

g1 = sum(x1_flat.*C_flat + x2_flat.*C_flat)-budget;
h1 = sum(x1_flat)+sum(x2_flat)-numplayers;
h2 = sum(x1_flat, 'all')-numstarters;
h3 = sum(x1_flat(20*0+1:20*0+20))+sum(x2_flat(20*0+1:20*0+20))-numgoalkeepers;
h4 = sum(x1_flat(20*1+1:20*1+20))+sum(x2_flat(20*1+1:20*1+20))-numdefenders;
h5 = sum(x1_flat(20*2+1:20*2+20))+sum(x2_flat(20*2+1:20*2+20))-nummidfielders;
h6 = sum(x1_flat(20*3+1:20*3+20))+sum(x2_flat(20*3+1:20*3+20))-numforwards;
h7 = sum(x1_flat(20*0+1:20*0+20))-numstartinggoalkeepers;
g2 = sum(x1_flat(20*1+1:20*1+20))-numdefenders;
g3 = -sum(x1_flat(20*1+1:20*1+20))+minstartingdefenders;
g4 = sum(x1_flat(20*2+1:20*2+20))-nummidfielders;
g5 = -sum(x1_flat(20*2+1:20*2+20))+minstartingmidfielders;
g6 = sum(x1_flat(20*3+1:20*3+20))-numforwards;
g7 = -sum(x1_flat(20*3+1:20*3+20))+minstartingforwards;




for i = 1:20
gplayerlims(i) = sum(x1_flat([i,20+i,40+i,60+i])) + sum(x2_flat([i,20+i,40+i,60+i])) - 3;
end
```

```
% g= [g1;g2;g3;g4;g5;g6;g7;gplayerlims'];
% h = [h1;h2;h3;h4;h5;h6;h7];
%
%
%  F =  matlabFunction(f,'vars',{[x1_flat,x2_flat]});
%  G = matlabFunction(g,'vars',{[x1_flat,x2_flat]});
%  H = matlabFunction(h,'vars',{[x1_flat,x2_flat]});

%  nonlinfc= @(in1)deal(G(in1),H(in1));



 %% Run Different Scenarios
%  x0 = [zeros([20,4]); zeros( [20,4])];
%  opts = optimset('Algorithm','sqp','Display','off');
%
%  runfmincon(F,x0, nonlinfc, opts);
%
%   x0 = [zeros([20,4]); zeros( [20,4])];
%  opts = optimset('Algorithm','active-set','Display','off');
%
%
%  runfmincon(F,x0, nonlinfc, opts);
%
%
%
%   x0 = [zeros([20,4]); zeros( [20,4])];
%  opts = optimset('Algorithm','interior-point','Display','off');
%
%  runfmincon(F,x0, nonlinfc, opts);
%
%   x0 = [ones([20,4]); ones( [20,4])];
%  opts = optimset('Algorithm','sqp','Display','off');
%
%  runfmincon(F,x0, nonlinfc, opts);
%
%   x0 = [ones([20,4]); ones( [20,4])];
%  opts = optimset('Algorithm','active-set','Display','off');
%
%
%  runfmincon(F,x0, nonlinfc, opts);
%
%
%
%   x0 = [ones([20,4]); ones( [20,4])];
%  opts = optimset('Algorithm','interior-point','Display','off');
%  x0_flat = reshape(x0,1,[]);
%
%  runfmincon(F,x0_flat, nonlinfc, opts);
%
%
%
%
```

```
%
%
%    x0 = [zeros([20,4]); zeros( [20,4])];
%  opts = optimset('Algorithm','sqp','Display','off','MaxFunEvals',inf,'MaxIter',inf);
%  runfmincon(F,x0, nonlinfc, opts);
%     x0 = [zeros([20,4]); zeros( [20,4])];
%  opts = optimset('Algorithm','active-set','Display','off','MaxFunEvals',inf,'MaxIter',inf);
%  runfmincon(F,x0, nonlinfc, opts);
%    x0 = [zeros([20,4]); zeros( [20,4])];
%  opts = optimset('Algorithm','interior-point','Display','off','MaxFunEvals',inf, 'MaxIter',inf);
%  runfmincon(F,x0, nonlinfc, opts);


 %% NLP
% s = sum((1-(X_min.*x1_flat/11)),'all')
f = -sum(X_min.*W_flat.*x1_flat,'all')- sum((1-((sum(X_min.*x1_flat,'all'))/11))*X_min.*W_flat.*x2_flat





for i = 1:80
% h8(i) =X_min(i) -numstartinggoalkeepers;
% h9(i) = sum(X_min([i,20+i,40+i,60+i]));
% g8(i) = X_min(20*1+i)-numdefenders;
% g9(i) = -X_min(20*1+i)+minstartingdefenders;
% g10(i) = X_min(20*2+i)-nummidfielders;
% g11(i) = -X_min(20*2+i)+minstartingmidfielders;
% g12(i) = X_min(20*3+i)-numforwards;
% g13(i) = -X_min(20*3+i)+minstartingforwards;
g8(i) = -X_min(i);
g9(i) = X_min(i)-1;
end
g10 = 1-((sum(X_min.*x1_flat,'all'))/11) - (4/11);





g= [g1;g2;g3;g4;g5;g6;g7;g8';g9';g10;gplayerlims'];
h = [h1;h2;h3;h4;h5;h6;h7];


 F =  matlabFunction(f,'vars',{[x1_flat,x2_flat,X_min]});
 G = matlabFunction(g,'vars',{[x1_flat,x2_flat,X_min]});
 H = matlabFunction(h,'vars',{[x1_flat,x2_flat,X_min]});
 nonlinfc= @(in1)deal(G(in1),H(in1));
```

```
%% New scenarios
%  x0 = [zeros([20,4]);zeros([20,4]);  zeros( [20,4])];
%  opts = optimset('Algorithm','sqp','Display','off');
%
%  runfmincon(F,x0, nonlinfc, opts);
%
%   x0 = [zeros([20,4]); zeros([20,4]); zeros( [20,4])];
%  opts = optimset('Algorithm','active-set','Display','off');
%
%
%  runfmincon(F,x0, nonlinfc, opts);
%
%
%
%   x0 = [zeros([20,4]); zeros([20,4]); zeros( [20,4])];
%  opts = optimset('Algorithm','interior-point','Display','off');
%
%  runfmincon(F,x0, nonlinfc, opts);
%
%   x0 = [ones([20,4]);ones([20,4]); ones( [20,4])];
%  opts = optimset('Algorithm','sqp','Display','off');
%
%  runfmincon(F,x0, nonlinfc, opts);
%
%   x0 = [ones([20,4]);ones([20,4]); ones( [20,4])];
%  opts = optimset('Algorithm','active-set','Display','off');
%
%
%  runfmincon(F,x0, nonlinfc, opts);
%
%
%
%   x0 = [ones([20,4]);ones([20,4]); ones( [20,4])];
%  opts = optimset('Algorithm','interior-point','Display','off');
%  x0_flat = reshape(x0,1,[]);
%
%  runfmincon(F,x0_flat, nonlinfc, opts);





x0 = [ones([20,4]); ones([20,4]); ones( [20,4])];
opts = optimset('Algorithm','sqp','Display','iter','MaxFunEvals',inf,'MaxIter',inf);
 [x fval x1_out x2_out x_min_out output lambda grad hessian] = runfmincon(F,x0, nonlinfc, opts);
 lambda;
 output;
 grad;
 hessian;
 Eigen_values = eig(hessian);

%  x0_flat = reshape(x0,1,[]);
%  [x fval] = fmincon(F,x0_flat,[],[],[],[],zeros(1,length(x0_flat)),[], nonlinfc);
```

```
% options = optimoptions('ga','Display','iter','PopulationSize',2400, 'InitialPopulationMatrix', 2*ones
% nvars = 240
% [x Fval exitflag] = ga(F,nvars,[],[],[],[],zeros(1,length(x0_flat)),[],nonlinfc,options)
% x1_out = x(1:80);
% x2_out = x(81:160);
% x1_out=reshape(x1_out,20,[]);
% x1_out(x1_out<1E-5) = 0;
% table(x1_out)
%
% x2_out=reshape(x2_out,20,[]);
% x2_out(x2_out<1E-5) = 0;
% table(x2_out)
%
%
% if length(x)>160
%     x_min_out = x(161:240);
%   x_min_out=reshape(x_min_out,20,[]);
% x_min_out(x_min_out<1E-5) = 0;
% table(x_min_out)
%  end


%  x1_out=reshape(x1_out,1,[]);
%  x2_out=reshape(x2_out,1,[]);
%  x_min_out=reshape(x_min_out,1,[]);

%  x_min_out = (x_min_out-100)./(200 - 100)

%  f = -sum(x_min_out.*W_flat.*x1_out,'all')- sum((1-((sum(x_min_out.*x1_out,'all'))/11))*x_min_out.*W_

%  x0 = [x1_out; x2_out; x_min_out];
%  opts = optimset('Algorithm','sqp','Display','off','MaxFunEvals',inf,'MaxIter',inf);
%  [x fval x1_out x2_out x_min_out] = runfmincon(F,x0, nonlinfc, opts);
%   x0 = [x1_out; x2_out; x_min_out];
%  opts = optimset('Algorithm','sqp','Display','off','MaxFunEvals',inf,'MaxIter',inf);
%  [x fval x1_out x2_out x_min_out] = runfmincon(F,x0, nonlinfc, opts);
%   x0 = [x1_out; x2_out; x_min_out];
%  opts = optimset('Algorithm','sqp','Display','off','MaxFunEvals',inf,'MaxIter',inf);
%  [x fval x1_out x2_out x_min_out] = runfmincon(F,x0, nonlinfc, opts);
%   x0 = [x1_out; x2_out; x_min_out];
%  opts = optimset('Algorithm','sqp','Display','off','MaxFunEvals',inf,'MaxIter',inf);
%  [x fval x1_out x2_out x_min_out] = runfmincon(F,x0, nonlinfc, opts);
writematrix(x1_out,'sqpx1.xls')
writematrix(x2_out,'sqpx2.xls')
writematrix(x_min_out,'sqpxmin.xls')
%   x0 = [rand([20,4]); rand([20,4]); rand( [20,4])];
%  opts = optimset('Algorithm','active-set','Display','off','MaxFunEvals',inf,'MaxIter',inf);
%  [x fval x1_out x2_out x_min_out] = runfmincon(F,x0, nonlinfc, opts);
%  writematrix(x1_out,'asx1.xls')
% writematrix(x2_out,'asx2.xls')
% writematrix(x_min_out,'asxmin.xls')
%    x0 = [zeros([20,4]); zeros([20,4]); zeros( [20,4])];
%  opts = optimset('Algorithm','interior-point','Display','off','MaxFunEvals',inf, 'MaxIter',inf);
```

```
%  runfmincon(F,x0, nonlinfc, opts);
```

```
 %% Runfmincon function
 function [x fval x1_out x2_out x_min_out output lambda grad hessian] = runfmincon(F,x0, nonlinfc, opts
  x0_flat = reshape(x0,1,[]);
[x fval exitflag output lambda grad hessian] = fmincon(F,x0_flat,[],[],[],[],...
    zeros(1,length(x0_flat)),[],...
    nonlinfc,opts);
disp('---------------------------------------------------');
disp([ 'Algorithm:', opts.Algorithm]);
disp([ 'Exitflag:', num2str(exitflag)]);
disp('x0');

table(x0)
disp (['Fval = ' ,num2str(fval)]);
disp(lambda)


x1_out = x(1:80);
x2_out = x(81:160);
x1_out=reshape(x1_out,20,[]);
x1_out(x1_out<1E-5) = 0;
table(x1_out)

x2_out=reshape(x2_out,20,[]);
x2_out(x2_out<1E-5) = 0;
table(x2_out)


if length(x)>160
    x_min_out = x(161:240);
  x_min_out=reshape(x_min_out,20,[]);
x_min_out(x_min_out<1E-5) = 0;
s = 1-((sum(x_min_out.*x1_out,'all'))/11)
su = sum(x_min_out,'all')
table(x_min_out)
 end

 end
% f_new = double(subs(f,{x1_flat,reshape(x1_out,1,[])}))
```

Appendix 2: Future Work

In the future, we hope to add multi-objective optimization to the problem at hand. In this case our primary objective will be to maximize the Expected Value scored by our squad and the secondary objective will be to minimize the total amount of money spent on procuring the squad.

The goal behind this is to check if there are multiple Pareto points that offer similar results with respect to the primary objective function while differing in the secondary objective, it would be beneficial from the squad management perspective to choose the cheaper squad.

We may also consider a multi-week approach to this problem. At the moment we are only predicting performance for the current week, but this model can be expanded on to include other mechanics from the Fantasy Premier League. Possible mechanics include variation in player costs, different game modes, and multi-week planning.

Further, we could convert the club based representation into player based representation using multiple methodologies such as probabilistic interpretations, or stochastic integer methods to create physical representations of the optimal solution. Naturally this will lead to more than one possible team and we could test all of them out to see which is the most optimal solution.