Optimizing Fantasy Football Lineups

Fraser Walker (001219429)

ECE 710 Final Project

McMaster University

2018
May

1 Introduction

Daily Fantasy Sports (DFS) is an emerging meta-game played by millions of people, particularly in North America. Recently with the popularity of websites such as DraftKings or FanDuel, has come the monetization of fantasy sports: a shift from a hobby where one competes against his friends, to a legitimate form of gambling. As such, a strong performance from ones fantasy team can now have serious financial implications, so gaining an edge over competing players is more important than ever.

In fantasy football for example, the idea is simple: draft a team of real life NFL players with the goal being to maximize the number of fantasy points they will score based on their performance in their respective NFL games. Each player has a salary with premium players (like Tom Brady) costing more than bottom of the barrel players (say, Brock Osweiller). Your team has a budget, which limits how expensive your players can be, ie: you cannot build a team of superstars at every position.

There are two important phases in team building. The first is projecting how a player will do that week. This is extremely challenging, and often projecting fantasy points can be arbitrary and will inevitably be riddled with poor predictions. However, assuming you have confidence in your projections, another challenging problem is how to select your team. This is an optimization problem where the objective is to maximize (projected) points while staying under the salary cap. There are other tertiary constraints with player positions: a certain number of players at each position must be present on your team. For example there must be one (1) quarterback on the team, and only one. Quarterbacks score more points on average than other positions, but one cannot load up on quarterbacks for this reason.

This is a challenging optimization problem since, for an average main slate of NFL games in a given week there can be almost 600 players present in the pool. In this report we look at a numerical example where we optimize a lineup over a player pool of 578 players. This can lead to many, many unique lineups: so simply enumerating the possible lineups is out of the question. Though the constraints are all linear, they only take integer values (a lineup can only have three or four players at the runningback position; 3.65 players is not possible). This problem is then an integer program (IP), decidedly more challenging then linear programs since there aren't any computationally effective general purpose optimization algorithms for integer problems that guarantee arriving at an optimal solution[1].

2 Formulating the problem

Let $\mathcal{P}$ be the set of all players (the player pool) available to choose from. We define subsets of of $\mathcal{P}$ according to position: let $\mathcal{Q}$ be the set of quarterbacks (QB), $\mathcal{R}$ the set of runningbacks (RB), $\mathcal{W}$ the set of wide receivers (WR), $\mathcal{T}$ the set of tight ends (TE), and $\mathcal{D}$ the set of defenses (DST). Note that each of these subsets are mutually exclusive: the intersection of any two is empty, for example:

$$\mathcal{Q} \cap \mathcal{R} = \emptyset \qquad \mathcal{Q}, \mathcal{R} \subset \mathcal{P}$$

It is worth noting that NFL DFS doesn't allow for 'cross-position' ellibility: a player is one **and only** one of the five listed positions. In NBA DFS for example, lineups are more flexible, and players can be listed at two positions, creating several possible placements in a lineup. For example, Ben Simmons of the Philadelphia 76ers has been listed at two positions in the past, and as a result could be placed in one of five different spots in an NBA lineup. In the NFL, a player is elligible for at most two spots in a lineup (more on that later).

The idea is to have every available player be a binary variable, that is:

$$\forall x \in \mathcal{P} : \quad x(x-1) = 0$$

That is, $x = 0$ (player x is not selected) or $x = 1$ (player x is selected). Each player x_i has a projected points attribute p_i and a salary s_i. The goal is to draft the team with the most projected points, making this a maximization problem with the objective function f_0:

[1]`http://people.brunel.ac.uk/ mastjjb/jeb/or/ip.html#capbud`

$$f_0 = \sum_{x_i \in \mathcal{P}} p_i x_i$$

The first obvious constraint is the salary cap. DraftKings allows for a $\$50,000$ budget for your lineup, but in general let c be the available budget, then:

$$\sum_{x_i \in \mathcal{P}} s_i x_i \leq c$$
$$\Longrightarrow f_1 = \sum_{x_i \in \mathcal{P}} s_i x_i - c \leq 0$$

The rest of the constraints pertain to team composition according to a player's position. For quarterbacks and defenses, this is straight forward: a lineup must have **exactly** one quarterback and defense. However when it comes to RB, WR, and TE there is a little flexibility: there is one spot on the lineup designated the 'flex' which can be one RB, WR, or TE. A lineup can then have 2-3 RBs, 3-4 WRs, or 1-2 TEs:

$$\sum_{x_i \in \mathcal{Q}} x_i = 1$$
$$2 \leq \sum_{x_i \in \mathcal{R}} x_i \leq 3$$
$$3 \leq \sum_{x_i \in \mathcal{W}} x_i \leq 4$$
$$1 \leq \sum_{x_i \in \mathcal{T}} x_i \leq 2$$
$$\sum_{x_i \in \mathcal{D}} x_i = 1$$
$$\sum_{x_i \in \mathcal{P}} x_i = 9$$

The last equality says the size of the team must be nine players. So we have ten equality/inequality constraints in total (note how three of the constraints are double sided, these amount to two separate ineqaulities each). This is the full optimization problem, however an insight into how NFL DFS is played will allow us to restrict the problem further and simplify it in the process. Although one is allowed to roster two tight ends, this is generally not advised since the tight end position is far more volatile then either the runningback or the wide receiver positions. Tight ends also consistently score less than comparably priced receivers or runningbacks. Our strategy then is to only roster one tight end:

$$\sum_{x_i \in \mathcal{R}} x_i \leq 3$$
$$\sum_{x_i \in \mathcal{W}} x_i \leq 4$$
$$\sum_{x_i \in \mathcal{T}} x_i = 1$$
$$\sum_{x_i \in \, (\mathcal{R} \cup \mathcal{W})} x_i = 6$$

This strategy has gotten rid of four inequalities and replaced them with one equality constraint. The flex spot on the roster is captured in the last equality, and by turning the tight end position into an equality constraint, we have implicit lower bounds on both the runningback and wide receiver positions. Based on this formulation, a feasible lineup cannot roster one RB for example because the last equality constraint implies there would be five WRs, which is infeasible. The complete integer program is thus:

$$
\begin{aligned}
min: \quad & -f_0 = -\sum_{x_i \in \mathcal{P}} p_i x_i \\
subject\ to: \quad & f_1 = \sum_{x_i \in \mathcal{P}} s_i x_i - c \le 0 \\
& f_2 = \sum_{x_i \in \mathcal{Q}} x_i - 1 = 0 \\
& f_3 = \sum_{x_i \in \mathcal{R}} x_i - 3 \le 0 \\
& f_4 = \sum_{x_i \in \mathcal{W}} x_i - 4 \le 0 \\
& f_5 = \sum_{x_i \in \mathcal{T}} x_i - 1 = 0 \\
& f_6 = \sum_{x_i \in\, (\mathcal{R} \cup \mathcal{W})} x_i - 6 = 0 \\
& f_7 = \sum_{x_i \in \mathcal{D}} x_i - 1 = 0 \\
& \forall x \in \mathcal{P}: \quad x(x-1) = 0
\end{aligned}
$$

3 Solution Strategy

The primary solution method will be the *Branch and Bound* algorithm for integer and mixed integer programs (MIP)[2][3]. Due to the size of the player pool (anywhere from 550 - 600 players), simply enumerating all the lineups, though finite, is not an effective strategy. The branch and bound algorithm relaxes the integer restrictions, turning the original problem P_0 into a Linear Program (LP) which can be solved effectively with the simplex algorithm or with interior point methods. The optimal value of the relaxation gives an upper bound on the optimal value of the original (thinking as a maximization problem), and the optimal point will most likely not be feasible for the original IP. From here, the algorithm chooses a variable that has

[2] `http://www.gurobi.com/resources/getting-started/mip-basics`

[3] `http://people.brunel.ac.uk/ mastjjb/jeb/or/ip.html#capbud`

a rational value and *branches* the problem in the following way. Suppose for the variable x_1 the optimal point is $x_1 = 0.6$, the algorithm branches the problem by creating two restricted problems P_1 and P_2 where in $P_1 : x_1 = 0$ and in $P_2 : x_1 = 1$. The algorithm then proceeds in the same way, solving the relaxed versions of P_1 and P_2 and continues the branching process. If p_1^* and p_2^* represent the optimal values of the IPs P_1 and P_2 respectively, then $max\{p_1^*, p_2^*\} = p_0^*$, the optimal value of the original problem P_0.

This process can be represented as a tree diagram where the original problem is the *root node* and each sub-problem is another node in the tree. A node is *fathomed* if it is 'complete': for one reason or another there is no need to continue branching at this node, making it a permanent node (a leaf in the tree diagram). The best integer solution found at any point in the branching is known as the *incumbent*, and the incumbent increases as better integer solutions are found. The branch and bound algorithm doesn't enumerate all possible integer solutions because if it determines at a node P_j that the optimal value of the relaxed LP is smaller than the incumbent, then it doesn't need to continue branching at that node, and thus the node is fathomed. The algorithm, like enumeration, does still depend on powers of 2 (ie: the branching that occurs at each node) and thus will not scale well for large problems, however it does guarantee finding the optimal solution vs. a heuristic algorithm which only approximates it.

4 Numerical Example

The salaries for Week 15 of the 2017 NFL season (2017/12/14 - 2017/12/18) were downloaded from DraftKings[4] as a .csv. The data contains the player's name, their salary for the upcoming week, position, and average number of fantasy points scored per game that season. In the absence of any projections from the time, the average points per game (PPG) will be used as the 'projected points' over which we wish

[4] `https://www.draftkings.com/`

	L. Bell	**A. Brown**	**A. Kamara**	**J. Jones**
Projected Points	23.646	25.377	20.054	16.738
Salary	9300	9100	8600	8500
QB	0	0	0	0
RB	1	0	1	0
WR	0	1	0	1
TE	0	0	0	0
FLEX	1	1	1	1
DST	0	0	0	0

Table 1: The head of the 8×578 matrix representing the data

to optimize our lineup. How points are projected for the upcoming week may not be in the scope of this report, PPG does have some inherent problems (an injured player may have a high PPG but his salary will be low to reflect the fact that he won't play - making him an obvious value play for the optimizer) but in general it does a decent job: salary and ppg have a high correlation. The player pool contains 578 players, each with eight attributes, and this information can be represented as a 8×578 matrix (Table 1).

Gurobi[5], a commercial software for LP, quadratic programs (QP), and in particular MIPs is used to optimize the lineup. This report uses a Gurobi API available as a package for the *R* programming language. DraftKings uses a $\$50,000$ team budget so in the formulation from earlier we have $c = 50000$. In this example, Gurobi found the optimal solution extremely quickly, show in Table 2. When the original unrestricted problem (with two TE feasible lineups) is run through the optimizer, the same lineup is returned as when we only allow one tight end. In addition to the branch and bound algorithm, Gurobi implements additional mechanisms to make optimization simpler. For example, *Presolving* refers to a process by which variables and/or constraints are

[5] `http://www.gurobi.com/index`

	Projected Points	**Actual Points**	**Salary**
QB T.J. Yates	18.100	7.12	4300
RB T. Gurley	23.285	48.0	8300
RB E. Elliot	22.913	0	6800
WR A. Brown	25.377	4.4	9100
WR D. Hopkins	22.408	18.0	7700
WR N. Agholar	12.715	18.9	4400
TE C. Clay	8.790	11.8	3000
FLEX W. Fuller	14.443	9.4	3500
DST Rams	11.154	17.0	2800
TOTAL	159.185	134.62	49,900

Table 2: The optimal lineup, with comparison between projected vs. actual points scored that week

removed before the branch and bound algorithm according to some internal logic. For example, in the player pool there are several players that have zero or negative projected points, and the algorithm can presolve to remove these players from the pool before attempting the branch and bound algorithm. Gurobi was able to find the optimal solution in roughly 0.12 seconds and only explored one node and found one suboptimal incumbent solution before finding the optimal integer solution. We know this is the optimal solution because the incumbent and the *best bound* are equal. The best bound is the maximum of the optimal values reached at each of the current 'leaf nodes'.

In Table 2 we see how the optimal solution would have done that week[6]. In general, 134.62 is a good score in NFL DFS, however this is number is inflated a bit by Todd Gurley's 48 points - one of the best performances of the year, so the average score in head-to-head games that week would have been larger than usual as a result. One

[6]Zeke Elliot was suspended for this game and Antonio Brown was injured during his game

additional consideration for future work would be to project ownership percentage: as in, estimate how many people will roster player x this week. Ownership percentage is important in DFS since in large tournaments, having highly owned players (known as the chalk) doesn't give an individual lineup much of a competitive advantage when that player is on a large percentage of teams[7]. This is less of an issue in head-to-head games where two lineups square off against each other. Ownership percentage could be seen as a secondary objective over which to minimize, not as important as maximizing score, but something to consider. This could also add additional constraints (example: do not roster any player projected to have > 35% ownership). Another consideration would be to project player points as an interval instead of a single point;

$$p_i = (a_i, b_i) \quad a_i < b_i$$

This can be thought of as a 95% confidence interval. Boom-or-bust players (less consistent week to week) would have comparitively wider ranges than Antonio Brown or Le'Veon Bell - blue chip players. Now there are additional considerations such as, how large should the 'tolerance' for risky players be? What is an acceptable 'worst case scenario' (lower bound) on the projected points total for an entire lineup? This would lead to additional constraints and/or a hierarchy of objective functions, but the solutions could be more robust to Black Swans (rare, unforseen events that carry a significant impact, can be both good or bad).

[7] http://cs229.stanford.edu/proj2015/104_report.pdf