# From Projections to Picks: A Decision Analytics Approach to Fantasy Football Draft Strategy

Trevor Squires

July 14, 2025

## 1    Introduction

Fantasy football is a way for fans to engage with the sport. Through various formats and platforms, fantasy pits fans against each other by allowing them to compile their own teams of professional players to see whose team can perform the best. It presents opportunities for fans to function as team managers, often requiring the managers to make a wide variety of decisions. Despite the fact that:

1. I'm currently 29 years old and have been a football fan for more than 25 of those,

2. In 2014, I went to college to become an engineer, and after a bit of back and forth, in 2022 I graduated with my PhD in mathematics,

3. I absolutely love environments which require analyzing strategic decision making,

it is now 2025 and I haven't really dedicated any time applying or studying the analytics of fantasy football. This document details my first efforts to optimize a bit of my fantasy football gameplay. The goal is not to completely solve fantasy football - if I could do that, I would certainly not be sharing that with you all. No, the goal is to find parts of the game we can optimize, and show clear, practical improvements.

In this way, fantasy football analytics mirror that of everyday business problems. Aside from the occasional luddite, people all tend to agree that some form of decision science can be used to make better decisions. However, in many places the field struggles with adoption. When decision science goes wrong, it's usually for one of two reasons: either it overreaches and tries to do too much, or it spits out something complicated that nobody understands or trusts even if it is technically sound. In both cases, the science fails not because the math is wrong, but because the decision-maker can't (or won't) use it. The real challenge is finding that balance between technical rigor and practical usability.

## 2    Predictive vs. Prescriptive Analytics

Most people who play fantasy football - at least semi-seriously - use some form of analytics, even if they wouldn't call it that. Whether it's looking up rankings, browsing average draft positions (ADP), or checking a player's projected points, they're using predictive models to guide their decisions. These projections are statistical estimates of how many points each player will score, typically over the course of a season. They're built from historical performance, expected workload, team context, injury risk, and other factors.

### A Quick Primer on Fantasy Football Drafts

If you're new to fantasy, here's the basic setup: each manager in a league takes turns selecting (drafting) real-life NFL players to build their fantasy team. Once a player is taken, no one else can draft them. Most leagues use a "snake draft" format, where the pick order reverses each round (e.g., picks 1–12, then 12–1, and so on). Teams must fill out a lineup, which typically includes one quarterback (QB), multiple running backs (RBs) and wide receivers (WRs), a tight end, and a flex spot. Later on, we will discuss the full dynamics of the league type that was analyzed. After the draft, you score points each week based on how

your drafted players perform in real life. So the goal during the draft isn't just to take good players, it's to build a complete, high-scoring roster under positional constraints and with limited information about what other managers will do.

## An Example: When Greedy Drafting Goes Wrong

I claim that predictive analytics is not sufficient to make fully optimal decisions, let's walk through a simplified example that showcases exactly that. Suppose we have two managers, Alice and Bob, and they're drafting from a pool of just 6 players. Each team must end the draft with exactly one QB, one RB, and one WR. The draft will go in a snake format: Alice picks first, then Bob picks twice, Alice finishes her team with the next two picks, and Bob is left with the last player. Here are the available players and their projected season-long fantasy points:

| Player | Position | Projected Points |
| --- | --- | --- |
| Mahomes | QB | 330 |
| Allen | QB | 320 |
| CMC | RB | 280 |
| Bijan | RB | 250 |
| Jefferson | WR | 300 |
| Chase | WR | 250 |

Alice is picking first. If she drafts greedily - just taking the top projected player - she grabs Mahomes (330 points). Bob then takes CMC (280) and Jefferson (300), locking in high-value RB and WR picks. Alice now looks at the remaining pool. She still needs an RB and WR. The best available are Bijan (250) and Chase (250). Her final team totals: 330 (QB) + 250 (RB) + 250 (WR) = **830 points**. Bob, on the other hand, still needs a QB. He picks up Allen (320) with his last pick. His team: 280 (RB) + 300 (WR) + 320 (QB) = **900 points**. Alice had the first overall pick, took the best player, and ended up with the worse team.

The issue wasn't with the projections. The problem was that her drafting strategy didn't account for how her early choices limited her future options. Drafting Mahomes meant giving up the chance to grab a higher-value RB or WR before Bob could. A simple greedy strategy that always picks the highest-projected player doesn't look ahead or consider opportunity cost. Sure, this toy example is contrived and simple, but this happens in real fantasy drafts, just at a larger scale with less easy to visualize mistakes. Roster composition, positional scarcity, and future pick dynamics all matter. Projections tell you how good a player is, but not when you should draft them, hence a need for prescriptive analytics.

# 3 Methodology

## 3.1 Data Collection

As with any data-related project, I always underestimate how much time I'll need to spend on the data. To counteract this slightly, I try to require as little raw, clean data as possible and make use of whatever is most easily available - though I do enjoy challenging myself from time to time by learning a new tool.

For this project, I started by making the easiest possible league assumptions. By focusing on a 12-team PPR league, I maximized my chances of finding usable data since this is by far the most common format. To get started, I needed two basic ingredients: projected point totals for each player and their average draft position (ADP). These two datasets form the foundation of most fantasy football analytics. Projections

estimate how valuable a player is expected to be. ADP tells us where other people are actually drafting them. When we put the two together, we get a clearer sense of opportunity, who might be undervalued, who's being over-drafted, and what kind of players are likely to be available at each pick.

Finding ADP data for this format was easy-Fantasy Football Calculator offers a downloadable CSV with all the major positions covered. Unfortunately, finding usable projections was a bit harder. Since we're still a ways out from the season, I assume many outlets are holding off on full projections due to uncertainty around injuries and depth charts. The one place that did have projections, ESPN, didn't offer an easy download. So instead of manually copying and pasting data across dozens of pages, I wrote a short Selenium script to scrape their sortable projections table. I haven't really touched webscraping in the past so I was a bit out of my element, but at the end of the day the script handled the job. Through a series of simulated actions: click the right tab, scroll through pages, extract player names, teams, positions, and projected point totals, I was able to get the projections I needed. I cleaned up some quirks (DST formatting, empty rows, missing data) and ended up with a dataset of roughly 300 fantasy-relevant players.

## 3.2  Value Over Replacement (VOR) Motivation

Let's return to the Alice and Bob example from earlier. Recall that Alice, picking first, grabbed the top projected player overall - a quarterback, Mahomes. Bob responded by taking a running back and a wide receiver, leaving Alice to pick less valuable players for those positions later. Despite having the first pick, Alice ended up with the worse team.

In the example, Mahomes scored 330 projected points, which sounds great on its own. But the second-best quarterback, Allen, was projected for 320 points. That difference, 10 points, tells us that picking Mahomes in that position is not going to generate much of an advantage over your opponent. On the other hand, the top wide receiver, Jefferson, was projected at 300 points, while the next best WR was only 250. That differential is much greater and suggests that a Jefferson first pick may have been preferred.

So let's assume that Alice opened with Jefferson. Bob can then take the best players of the other two positions: Mahomes and CMC. The draft would finish with Alice taking the remaining players at her position: Bijan and Allen while Bob gets Chase at last pick. Alice finishes with 320 (QB) + 250 (RB) + 300 (WR) = **870** points while Bob gets 330 (QB) + 280 (RB) + 250 (WR) = **860** points. In this alternate universe, Alice comes out ahead of the draft by shifting away from a greedy strategy.

Why did this happen? The answer lies in the concept of Value Over Replacement, or VOR. Simply put, VOR measures how much better a player is compared to a "replacement-level" player at the same position. The replacement-level player represents roughly the best freely available option after the top players have been taken. VOR helps capture this nuance by comparing each player's projected points to the baseline of replacement-level players at their position. The replacement level is usually defined as the projected points of the player at the draft slot where starting rosters run out for that position. For our analysis, we define the replacement as the best player at their position that is still available on our **next** pick. Formally, for a player $i$ at draft pick $p$ given next draft pick $p'$:

$$\text{VOR}_i = \text{ProjPoints}_i - \text{ProjPoints}_{\text{replacement}(i,p')}$$

Essentially, the VOR is the difference between taking a particular player now vs taking the same position next round instead.

In practice, we can estimate what the replacement value is using ADP. For any given draft pick, we'll know when our next draft pick is going to occur. Using that information along with some ADP estimates, we can predict what players will be remaining as replacements for our next pick. To add a bit of robustness to our calculations, rather than assuming the next round will follow ADP exactly, we can assume our replacement value is actually the average of the best available player at: our next pick, our next pick + 6 more picks, our next pick - 6 more picks. This hedges slightly against the assumption that the draft follows the ADP exactly. In general, assuming that ADP holds over a small sample size (like the difference between two consecutive player picks) is dangerous. However, assuming that ADP holds on a larger scale (like the first 5 rounds of a draft) is much more reliable.

Using VOR instead of raw projections guides smarter drafting. It prioritizes players who provide the most incremental value over what you could get later at the same position. Additionally, VOR acknowledges that fantasy drafting is a multi-player game and your goal is not just to get a high scoring team, but to maximize

the difference between your team's performance and everyone else's. This helps manage the opportunity cost of each pick, ensuring that your team builds around scarce, high-impact positions before filling in depth.

## 3.3  Full-Draft Mixed Integer Programming (MIP)

In many ways, VOR can be considered a one-step lookahead. We are estimating the opportunity cost of selecting a particular position at a particular point in time vs picking it one round later. But you know what's better than a one-step lookahead? A 12-step, full draft lookahead. Indeed, what if instead of just estimating how good of shape our team will be in the next round, we look at how our entire draft will play out if we select a particular position in a particular round? This framework requires a slightly different technical toolkit for analysis. Immediately, I see dynamic (DP) or mixed integer programming (MIP) as frameworks to help to answer this question. For this work and the analysis to follow, I chose to use the latter.

By formulating the draft as an optimization problem, we can maximize the projected points of the final team while respecting all roster constraints and anticipating player availability across all picks. The primary goal of the model is simple: maximize the sum of projected points for all players selected throughout the draft.

$$\max \sum_{\text{pick } i} \sum_{\text{position } p} x_{i,p} \times \text{ProjPoints}_{i,p}$$

Here, $x_{i,p}$ is our binary decision variable indicating whether the player chosen at pick $i$ fills position $p$. The model must also respect team-building rules typical in fantasy football. For example, a standard 12-team PPR league would require:

- Exactly one player chosen per pick,

- Position limits such as a fixed number of quarterbacks, running backs, wide receivers, tight ends, kickers, and team defenses

- A FLEX role which can be filled by anyone one of a few positions (usually WR/RB/TE)

These constraints are encoded as linear inequalities involving the $x_{i,p}$ variables and auxiliary variables that track FLEX usage. This ensures that the final roster meets all league requirements and that positional flexibility is handled properly. The full model is written out in (MIP) and explained in slightly more detail in Section A

One key input to the model is a projected points matrix, $\text{ProjPoints}_{i,p}$, which estimates how many points you can expect if you draft a player at pick $i$ for position $p$. This matrix is computed using ADP we forecast which players will still be on the board when it's our turn, which in turn informs which players can reasonably be chosen at each pick. Again, we pull the same ADP averaging trick as in the VOR section to make our estimates a bit more robust. By combining projections, roster constraints, and ADP-informed availability, this MIP optimizes draft decisions across all picks, rather than just one at a time. This full draft lookahead captures the complex tradeoffs and opportunity costs better than simple heuristics like VOR alone. Again, more details are discussed in the appendix, but the main takeaway is that our MIP model can be solved to tell us what the optimal position to target is for any player/round combination in a 12 man standard PPR league.

One last note about the solution itself. It isn't enough to just model the problem, we also have to be able to optimize the model. In our case, this model was built with Pyomo which allows the use of different solvers. I chose Gurobi as a solver for it's ease of integration with Python/Pyomo, it's speed as a solver, and due to the fact the model was small enough to be solved under the free license. Though, for the same reason, any MIP solver supported by Pyomo would have easily solved this problem as well.

# 4  Results

With the methodology spelled out, it's now time to review some of the results. To be extremely specific about what type of league we're optimizing, we assume the following:

- There are 12 players in the draft

- They drafted in snake order

- Each team must consist of 1 QB, 2 WR, 2 RB, 1 TE, 1 team defense (DEF), and 1 place kicker (PK)

- Each team is allowed 1 FLEX position that may be filled by a WR, RB, or TE

- Since there are only 9 players on the team, only the first 9 rounds are optimized

- The projections and ADP are based on PPR scoring rules

As discussed previously, we will split these out by difficulty of implementation to present different levels of adoption.

## 4.1 Position-Level Playbook

One of the simplest outputs from the full-draft MIP is a position-level playbook: a table that tells you, for each pick in a 12-team league, what position you should be targeting. This assumes the draft follows ADP reasonably closely, and that players go roughly where we expect them to. In other words, this playbook is the default plan-something you can print out and bring with you to your draft to help guide your decisions in real time.

| Round | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 | P11 | P12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | WR | RB | WR | RB | WR | WR | WR | RB | RB | RB | RB | RB |
| 2 | QB | QB | TE | TE | TE | TE | TE | RB | TE | TE | RB | RB |
| 3 | TE | TE | QB | QB | QB | QB | QB | TE | QB | QB | QB | QB |
| 4 | RB | WR | RB | RB | RB | RB | RB | RB | RB | RB | TE | TE |
| 5 | RB | RB | RB | WR | RB | RB | RB | WR | WR | WR | WR | WR |
| 6 | WR | WR | WR | WR | WR | WR | WR | WR | WR | WR | WR | WR |
| 7 | WR | WR | WR | WR | WR | WR | WR | QB | WR | WR | WR | DEF |
| 8 | DEF | DEF | DEF | DEF | DEF | DEF | DEF | DEF | DEF | DEF | DEF | WR |
| 9 | PK | PK | PK | PK | PK | PK | PK | PK | PK | PK | PK | PK |

Note that this table should not be read more than 1 column at a time. For example, player 3 is recommended to take a TE in round 2, but so are players 4-7. If players 4,5,6, and 7 all take TEs before player 3 in the second round (which they can because of the snake draft ordering), then player 3 is unlikely to get a strong tight end. However, player 3 is recommended to take a tight end **because** there are usually good ones there in an average draft. If instead of an average draft, you assume that everyone is drafting optimally, this board may look different. But in practice, assuming average draft strategies is much more accurate which is what drove this assumption in the first place.

That being said, there are a few clear takeaways:

- **Wide receiver has depth.** The WR position is recommended to be filled out last for every player. If you don't land one of the very top wide receivers in round 1 (say, the top 5 or 6), the model often prefers to wait and load up on WRs in the mid-to-late rounds. There's just a lot of WR value available throughout the draft.

- **Running back dries up quickly.** RBs dominate the early rounds, and then mostly disappear after round 5. If you don't grab your core RBs early, the model has a hard time finding replacements later. If you can't get one of the premium wide receivers early, load up on RBs. There just aren't that many RBs projected for high workloads, so this is one of the few positions where early aggression pays off.

- **Quarterback value peaks in round 3.** Most of the top QB value is concentrated in the third round. But if you have a shot at the very top QB overall-say, one projected far above the rest-it's worth grabbing them in round 2. After that, unless you're in a league that starts more than one QB, it's best to wait.

5

- **Tight end is top-heavy.** TE is always tricky. There are often one or two standouts, but fall off quickly - usually not even enough to fill 12 teams. The model recommends taking the top tight end above its ADP for almost every player save players 11 and 12 who are too late to expect to grab a good TE early.

- **Wait on defense and kicker.** No surprise here: both DEF and PK are almost always drafted in the last two rounds. They don't offer much upside compared to skill positions, and they're easy to stream during the season. This playbook confirms what most experienced players already do instinctively.

- **Wide recievers are the go to FLEX**. Again, this isn't revolutionary, but the model agrees that using your FLEX role on a WR is the preferred approach. This is the commonly accepted strategy in PPR leagues - the league type we are analyzing.

This position-level strategy is useful for a few reasons. It's easy to use, robust to small deviations from ADP, and requires very little real-time computation. The downside, of course, is that it's static. If your draft goes off-script, say, a surprise position run starts, or your opponents reach heavily, it might steer you into suboptimal choices. Still, for most standard drafts, this kind of playbook offers a clean, precomputed baseline strategy. It's a perfect example of prescriptive analytics applied at a practical level: it doesn't try to do everything, but it gives clear guidance you can actually use.

## 4.2  Player-Level Targets

Beyond general positional trends, one of the more interesting outputs from the full-draft optimizer is identifying specific players who consistently show up in the optimal draft strategy across many different teams and draft slots. Because the MIP is simulating the entire draft for each drafter-from pick 1 through pick 12-it selects players based on their marginal value in context. So if a player is being selected over and over again, that usually means they are providing strong value relative to others available at the same position and pick. Note that this doesn't mean these players are the top targets, just that they are high value players who are consistently available without having to reach. Here are a few players who appeared in optimal drafts for at least 4 out of the 12 player positions:

- **Trey McBride (TE)** - Appears in **10** optimal drafts, usually in **round 3**. This is a clear signal that McBride offers one of the strongest values at the tight end position, especially if you're not spending a second-round pick on the top options.

- **Omarion Hampton (RB)** - Also appears in **10** optimal drafts, most often in **round 4**. Hampton seems to represent a turning point in the RB pool. After the top 10–12 running backs are gone, there's a steep dropoff, so Hampton showing up this frequently suggests he's one of the last RBs with strong projected value. A small caveat is that he's a rookie so his projections are extremely volatile, but that's not a consideration in our mathematical model.

- **Rome Odunze (WR)** - Chosen in **9** drafts around **round 6**. This is a sweet spot in the WR depth chart. Odunze seems to be a strong target after loading up on RBs early, and helps fill out the mid-round WR depth while offering upside.

- **Calvin Ridley (WR)** - Selected in **8** drafts, typically in **round 6**. Similar to Odunze, Ridley offers solid value in the middle rounds and may be underpriced relative to his projected usage.

- **Jayden Daniels (QB)** - Appears in **7** optimal builds, usually around **round 3**. He seems to represent the inflection point in QB value-if you miss out on the elite options in round 2, Daniels is often the next-best bet and offers strong upside. Interestingly enough, due to other upside, top QBs are never recommended for any player with Jalen Hurts being the highest recommended selection. Daniels appears to be the best mix of value and position.

- **Zay Flowers (WR)** - Found in **6** optimal builds, most commonly in **round 5**. Again, this reinforces the idea that the WR depth is strong in the mid rounds, and Flowers provides a solid value in that tier.

- **Jonathan Taylor (RB)** - Chosen in **5** drafts, usually in **round 2**. Taylor seems to fall into a slightly undervalued zone. If top-tier RBs are off the board, the model frequently recommends Taylor as a strong fallback option with workhorse potential.

- **Jaylen Waddle (WR)** - Shows up in **4** drafts, most commonly around **round 6**. This supports the same broader pattern: WRs in the 5–7 range are often the best options available once the RB pool starts to thin out.

These kinds of player-level insights can help sharpen your draft board. Rather than relying purely on consensus rankings, you can look for players who are repeatedly selected in optimal draft strategies and prioritize them accordingly. That said, there's a tradeoff here. This insight assumes that ADP holds reasonably true and that you'll be able to get these players around their projected rounds. In more chaotic drafts, players may be reached for or fall unexpectedly, and rigidly sticking to these targets could backfire. Still, having a shortlist of high-VOR names gives you a leg up when it comes to making informed choices in the mid rounds-where most drafts are won or lost.

## 4.3 Real Time Integration

If the position playbook was minimal integration, the opposite end of that spectrum is certainly the real time integration. All of the insights discussed so far rely on ADP as a prediction for what will be available at a particular point in time. As soon as that assumption breaks down, the previous results struggle to provide value. The solution? Reoptimize. By connecting directly with Yahoo's drafting interface, I was able to reuse the MIP framework by rebuilding and solving a new optimization problem using the actual draft board rather than relying on a static one. Yes, it still requires the ADP assumption from this point forward as a prediction for what will happen in the future, but the real time integration allows for a resolve at each round rather than relying on the round 1 prediction for that point in time.

As far as insights go, the real time engine doesn't provide much takeaways. However, it offers a fully synced approach for those who want to lean all in to the model.

# 5 Conclusion

This project explored two different approaches for improving fantasy football draft strategy: a Value Over Replacement (VOR) method and a full Mixed Integer Programming (MIP) solve. Both approaches are valid, but they serve different needs.

The VOR method is simple, interpretable, and easy to communicate. You can usually explain to someone why a particular player has high VOR in a sentence or two. The MIP, on the other hand, is more comprehensive. It considers the entire draft in one shot, accounting for future opportunity costs, positional depth, and roster constraints all at once. But the tradeoff is that the decisions it produces are harder to explain. It'll tell you what to do, but not always why. Depending on what you need - explainability vs optimality - you might lean toward one over the other.

Interestingly, in this case, the two approaches ended up agreeing more than I expected. That makes the VOR-based methods especially appealing in most cases, since they're easier to reason through while still offering solid performance. Though users that are more than comfortable with the MIP, like myself, will still rely on the MIP.

Each technique also produced insights at different levels of generality. The first level was the position-level playbook: simple, useful, and applicable across most drafts. If you don't know anything else, that's a good place to start. The second level, the player targets, offered a bit more precision - highlighting players who repeatedly showed up in optimal strategies. These insights are great when they line up with your draft position, but they're more sensitive to shifts in ADP or early-round surprises. The third level - full integration - is the most technically powerful. It gives you optimal decisions in real time, tailored to the exact draft unfolding in front of you. But that power comes at a cost. There's no useful takeaway to bring into the draft beforehand. You're either running the optimizer on the fly or you're not using it at all.

So which approach is best? That entirely depends on the person using it. If you want something you can print out, skim between picks, and not overthink, the MIP-derived position playbook is a great balance of

rigor and usability. If you're drafting on the fly, value explainability, or need to justify your picks to others, the VOR logic may be a better fit. And if you're all-in on optimization and don't mind complexity, live integration with the full MIP can give you the edge in tight leagues.

# 6 Future Work

Now, to be fair, there's still a lot of ways to improve this. The most obvious next step is to incorporate some measure of risk. Right now, both the projections and ADP are treated as fixed point estimates. But in practice, player performance is uncertain. Some players are high-floor, low-ceiling; others are the complete opposite. I would say the majority of amateur fantasy decisions are made based off of "I think this person will outperform his projections". Adding variance or confidence intervals to projections could lead to more robust strategies, especially in early rounds where a bust can derail an entire draft. Similarly, ADP isn't perfect - some players have a wide range of draft positions, and that uncertainty could be explicitly modeled to better simulate real drafts. In fact, by accounting for yearly or even weekly variance in point output, one could extend this model to optimize a full 15 rounds with the latter rounds being used to mitigate risk in your starters. Instead of assuming a fixed amount of points per week for your two starting RBs, you could assume that collect the two largest point totals from your roster of 3+ RBs. Though it's trivial in the deterministic case, adding weekly variance is a great way to make the model more realistic (at the cost of additional data requirements).

Another improvement would be to shift the optimization objective itself. Instead of maximizing your own projected points, you could try to maximize your projected advantage over your most likely opponent. After all, fantasy is zero-sum: you don't just want a good team, you want a better team than the person you're facing. This adds some complexity to the modeling, since your opponents' picks now matter too, but it's definitely doable.

Bye weeks are another detail currently ignored. Most fantasy leagues require you to start a full lineup each week, and if your top RBs are all on bye in Week 10, that's a problem. Adding constraints to stagger bye weeks could help ensure more consistent performance across the season. There are probably dozens of other tweaks and ideas - better priors on team construction, injury modeling, schedule strength, stacking considerations for best ball - but the goal here wasn't to solve every nuance of the draft. Even still, I'm happy with how this has turned out with such simple data assumptions.

# A MIP Details

Let's say you want to draft the best possible team. Not just make the best next pick, but the best set of picks across your entire draft slot. That's the core goal of this model: find the combination of positions that will lead to the highest total projected points across a full draft. We can think of this as a constrained optimization problem. At each pick, we must choose a position to fill. Across the draft, we must respect roster rules - e.g., we can only take a limited number of quarterbacks, running backs, etc. - and at the end of the draft, we want to maximize our total projected points. To model this, we use a Mixed Integer Program (MIP), where the binary variables represent which position we choose to draft at each pick. The model considers the value of each decision, the positional requirements, and even flexible lineup options. Here's how it works in detail. In addition to the obvious, we will receive the following information

- A list of position constraints. Each element in the list contains a set of positions and the maximum amount of players of all those positions that we can take

- A current roster (which can be completely empty)

- A list of picks we get to make

- A database of players, their position, average draft position, and yearly projections

- A flex limit

## A.1 Notation

Let:

- $R = \{0, 1, \ldots, N-1\}$: the set of rounds or pick indices the user will make.

- $P$: the set of positions (e.g., QB, RB, WR, TE, DEF, etc.).

- $J \subseteq P$: the set of positions which can count as a FLEX

- $\ell$: the total FLEX limit

- $\text{pick}_r$: the overall pick number for round $r \in R$.

- $\text{proj}_{r,p}$: the projected points for the best available player at position $p \in P$ at pick number $\text{pick}_r$.

- $\text{limit}_j$: maximum number of players allowed for group constraint $j$.

- $\text{positions}_j \subseteq P$: the subset of positions constrained in group $j$ (e.g., WR+RB+TE for FLEX).

- $\text{roster}_p$: number of players already drafted at position $p \in P$.

With these definitions, we can write out the full model

$$
\max \sum_{r \in R} \sum_{p \in P} x_{r,p} \cdot \text{proj}_{r,p}
$$

$$
\text{s.t.} \sum_{p \in P} x_{r,p} = 1 \quad \forall r \in R
$$

$$
\sum_{r \in R} \sum_{p \in \text{positions}_j} x_{r,p} + \sum_{p \in \text{positions}_j} \text{roster}_p \le \text{limit}_j + f_j \quad \forall j \in P \qquad \text{(MIP)}
$$

$$
f_j = 0 \quad j \notin J
$$

$$
\sum_j f_j \le \ell
$$

## A.2 Model Components

### Decision Variables

The most critical piece of the model are the decision variables.

- $x_{r,p} \in \{0, 1\}$: binary variable indicating whether we draft position $p$ at pick $r$.

- $f_j \in \mathbb{Z}_{\ge 0}$: integer variable indicating how many of the drafted players from group $j$ count as FLEX.

The $x$ binary variables are a standard in MIP modeling. The $f_j$ are necessary to simultaneously ensure that each position limit is satisfied with exceptions only in the case of the FLEX positions.

### Objective

Our goal is to maximize projected points across the draft:

$$
\max \sum_{r \in R} \sum_{p \in P} x_{r,p} \cdot \text{proj}_{r,p}
$$

Each $x_{r,p}$ indicates whether we selected position $p$ at pick $r$, and $\text{proj}_{r,p}$ gives the expected points for that choice. This ensures the optimizer always selects the most valuable position combinations over the full draft.

## Constraints

### Pick constraint: One position per pick

At each pick, exactly one position must be chosen:

$$\sum_{p \in P} x_{r,p} = 1 \quad \forall r \in R$$

### Roster limit constraints by position group

Each position group $j$ has an associated cap, which accounts for players already on the roster and those drafted during the optimization:

$$\sum_{r \in R} \sum_{p \in \text{positions}_j} x_{r,p} + \sum_{p \in \text{positions}_j} \text{roster}_p \leq \text{limit}_j + f_j \quad \forall j \in P$$

This makes sure we don't draft more players than we can use. For example, if you can start up to 3 WRs, and you already have one, you can take up to 2 more, unless you're using a FLEX spot.

### FLEX permissions

Only allow FLEX assignments for groups where $\text{flex}_j = 1$. If FLEX isn't allowed for a group, then:

$$f_j = 0 \quad j \notin J$$

This forces $f_j$ to only be positive where applicable (e.g., RB/WR/TE combinations).

### Overall FLEX cap

You only get a limited number of FLEX slots overall (usually just one):

$$\sum_j f_j \leq \ell$$

This keeps things realistic. Even if 5 players "fit" into a FLEX-capable group, we're only allowed to start one of them in that slot.