FPL picker – An Optimization approach with Goal Programming

# Introduction to Goal Programming

Goal programming is a branch of multi-objective optimization that can be used to solve problems with **multiple objectives** that are generally incommensurable and often **conflict with each other** in a decision-making horizon. It can be thought of as an extension or generalization of linear programming to handle multiple, normally conflicting objective measures.

Hence, this technique is suited to pick an FPL squad that could be based on stats/metrics (or we will them ”goal”) from each players. More than often those stats are quite conflict with each other. For examples, we wants to pick 2 goalkeepers with good shot saving ratio and clean sheets per game, but at the same time we also want to pick strikers who will have a lot of shots/shots on targets. Those stats are conflicting as goalkeepers will have 0 shots/shots on targets; strikers do not need stats/metrics of shot saving or clean sheet. To harmonize those ”goals”, goal programming seems a balanced and suitable approach.

In this case we will use Minimax GP technique. Minimax GP is a variant of goal programming that involves minimizing the maximum deviation from the goals. In minimax GP, each goal has a priority level and a target value. The optimization algorithm attempts to minimize the maximum deviation from the target values while satisfying all of the goals.

Text

Description automatically generated with low confidence

\*Remember: Minimax GP is based on the concept of deviation from the target. Hence, we will often see variable such as d-  or d+  which means negative and postive deviation. In this case of FPL squad picking, negative and postive deviation do ot differ much from each other.

# Problem formulation

Now let try to set a formulation to pick a FPL squad for the first week of FPL.

For GP formulation, always identify the goals first. In this case, they are FPL players stats (I used stats from 2021/2022 seasons). Also, I only have stats from 17 teams (no stats for 3 teams of promotion for week 1 yet, as it is hard to find those data from the lower league)

In this example, we can set 7 goals(stats): Shots, Key pass, clean sheets (GK), clean sheets (defender), Saves (GK), Accurate crosses (defender) and Minutes played. We can call them from s1 to s7 respectively.

Hence, the objective for the problem will be:

Min Z = s1+ + s2+ + s3+ + s4+ + s5+ + s6+ + s7+

(Important note: In this example, I consider the importance of all the stats will be equal, hence, I do not set the weight. It is always better to set specific weight values to each goal as some goal/stats are more important than the other. In this case, it requires a bit of your insight/view on football and FPL about which stats is crucial and more valuable than others. I also believe that 7 stats chosen for this example are probably not enough, we might need more for a more robust preditction.)

Now moving on to the variables of the problem. First we will identify our main variable in this case as **xi** where **xi** is binary. **xi = 1** means we will pick player **i** into our squad, **xi = 0** means player **i** not selected.

We also set **c1i** to **c7i** as stats metric equivalent to our 7 stats/goals.

For example, **c1i** means the Shots stat of player **i**.

Moving on to the **target value.** This is very important step and a brief investigation into the data will always helpful to set a attainable and robust target.

My approach in this case is balancing so I will set the target value slightly above the average stats value of 300+ players. For example, the (slightly above) average number of total shots last season will be 43. So the target value for s1 will be 43\*15 (multiply by 15 because I want to pick 15 players who shoots more often)(for other cases such as goalkeeper, you only need to multiply by 2 as we will only pick 2 goalkeeper). All in all, picking target value also requires a bit of investigation and it is very case-specific. You can obcivously try to raise the target value for a more robust model but remember that those values must be attainable.

After that we can formulate our constraints for the 7 goals.

All in all we will have a general model formulation as below:

Min Z = s1+ + s2+ + s3+ + s4+ + s5+ + s6+ + s7+

Subject to:

**+ s1+ - s1- = 43\*15**

**+ s2+ - s2- = 16\*15**

**+ s3+ - s3- = 5\*2**

**+ s4+ - s4- = 4\*5**

**+ s5+ - s5-  = 45\*2**

**+ s6+ - s6- = 16\*5**

**+ s7+ - s7- = 1300\*15**

**Xi = 0 or 1**

**s1+ ,s1-** ,**s2+ ,s2-** ,**s3+ ,s3-** ,**s4+ , s4-** ,**s5+ ,s5-** ,**s6+ ,s6-** ,**s7+ ,s7- >= 0**

Also, remember to add constraints for FPL rules (cost restriction, positon restrictions, maximum 3 players from the same club, etc.) Those constraints are easily coded (you can check from the main code)

Remember to update your data every week so model can perform every gameweek.

# Packages

For implementation, the problem was coded with Python. Packages used: numpy, pandas, pyomo (<http://www.pyomo.org/> ), gurobi (<https://www.gurobi.com/>).