The Ultimate Team Optimization Problem: A Multi-Objective MILP Formulation

Giovanni Lo Bianco

December 7, 2025

Abstract

This document presents a formal mathematical model for the Ultimate Team Optimization Problem, a challenge faced by players of sports video games like EA FC. The goal is to assemble the best possible team of players by simultaneously optimizing two conflicting objectives: maximizing the team's overall rating and maximizing its chemistry. We begin with a detailed, non-technical description of the problem and then present the formal Multi-Objective Mixed-Integer Linear Program (MOMILP). We also describe a solution methodology based on an iterative search with dominance-breaking constraints to generate the Pareto frontier of non-dominated solutions.

1 Problem Description

In the popular Ultimate Team game mode of the EA FC video game series, players are tasked with building their dream football squad from a collection of player items. Success in this mode is not just about assembling a team of superstars; it's a delicate balancing act between two key metrics: **Team Rating** and **Team Chemistry**.

Team Rating is a straightforward average of the individual "Overall" ratings of the 11 players in the starting lineup. A higher rating generally signifies a more talented team on paper.

Team Chemistry , however, is a more complex system that rewards players for creating links between teammates. Players receive chemistry points if they share the same club, league, or nationality with others in the team. A team with high chemistry receives in-game performance boosts, making the players more effective than their base statistics would suggest.

The core challenge, which we call the **Ultimate Team Optimization Problem**, arises from the conflict between these two objectives. Often, the highest-rated players do not share the same club or league, making it difficult to achieve high chemistry. Conversely, building a team with perfect chemistry might require using lower-rated players, thus sacrificing overall team talent. More specifically, the collection of players available is limited by the resources (in-game currency, player items) of the user. Thus, this problem is even more constrained as the user has limited resources.

The Role of Formations An additional layer of complexity is the choice of formation (e.g., 4-4-2, 4-3-3, 5-2-1-2). Each formation dictates the specific positions that must be filled, which in turn affects which players can be used and what chemistry links are possible. The model presented here is designed to solve the optimization problem for a *fixed* formation. To find the absolute best teams, the entire optimization process must be repeated for each desired formation. As we will see, solutions found for one formation can be used to inform and improve the search for subsequent formations.

1.1 How the Optimization Model Works

Instead of manually testing combinations, we define a set of goals and rules for an optimization solver. The solver's job is to find the best possible team that follows all the rules we've given it.

The Two Primary Goals The model is designed as a multi-objective problem. This means it tries to do as well as possible on two competing goals at the same time:

1. **Maximize Team Rating:** Find a combination of 11 players whose average rating is as high as possible.

2. **Maximize Team Chemistry:** Find a combination of 11 players whose total chemistry score (the sum of each player's individual chemistry, up to a maximum of 33) is as high as possible.

The Core Decisions The model's fundamental task is to make a series of yes/no decisions for every available player and every position in a chosen formation. For a player like Kylian Mbappé and the "ST" (Striker) position, the model asks: "Should I place this player in this position? Yes or No?" By making one "yes" decision for each of the 11 positions, it builds a complete team.

1.2 The Rules of the Game (Constraints)

To ensure the solver builds a valid team, we enforce a series of strict rules, or "constraints."

Rule 1: Player Assignment Rules These are the basic rules for filling out a team sheet.

- **Fill Every Position:** Each of the 11 positions in the formation must be filled by exactly one player. No more, no less.

- **One Player, One Spot:** A single player cannot be used in more than one position at the same time.

- **Play Your Position:** A player can only be placed in a position if they are eligible for it (i.e., it's listed as their primary or alternate position).

Rule 2: Chemistry Calculation Rules This is the most complex part of the model, designed to perfectly mimic the in-game chemistry system. It works in three main steps: unlocking chemistry tiers, calculating each player's potential chemistry, and then finalizing the team's total chemistry.

- **Step 1: Unlocking Chemistry Tiers.** The model first counts how many players from the same nation, league, and club are in the starting 11. Based on these counts, it determines which "tier" of chemistry points has been unlocked for that specific group.

  - **Nation Chemistry:** Tier 1 (1 point) for 2 players, Tier 2 (2 points) for 5 players, Tier 3 (3 points) for 8 players. *Bonus: ”Icon” players count as two players for their nation.*
  - **League Chemistry:** Tier 1 (1 point) for 3 players, Tier 2 (2 points) for 5 players, Tier 3 (3 points) for 8 players. *Bonus: ”Hero” players count as two players for their league.*
  - **Club Chemistry:** Tier 1 (1 point) for 2 players, Tier 2 (2 points) for 4 players, Tier 3 (3 points) for 7 players.

- **Step 2: Calculating Individual Chemistry.** Once the tiers are determined, the model calculates the potential chemistry for each individual player. A player's score is the sum of the points from the tiers they belong to, capped at a maximum of 3 points.

- **Step 3: Finalizing Team Chemistry.** The model calculates the total team chemistry by summing the scores from Step 2 for all 11 players, with one last adjustment: if the player is an ”Icon” or a ”Hero,” their chemistry is automatically set to the maximum of 3.

2 Multi-Objective Problem Formulation

To translate this complex problem into a solvable format, we now define its components using the formal language of mathematical optimization. This involves specifying the sets, parameters, variables, objectives, and constraints that constitute the model.

2.1 Sets and Indices

First, we define the fundamental building blocks of our model. These sets and their corresponding indices provide a way to reference all the players, positions, and attributes involved.

- I: The set of available players, indexed by i.

- $I_{special} \subseteq I$: The subset of players who are either Icons or Heroes.

- K: The set of 11 positions for a given formation, indexed by k.

- $J_{nat}, J_{club}, J_{league}$: Sets of unique nationalities, clubs, and leagues, each indexed by j.

- $M = \{0, 1, 2, 3\}$: The set of chemistry point tiers, indexed by m.

2.2 Parameters

Next, we define the parameters of the model. These are the known, fixed values that are derived directly from the game's data, such as a player's rating or nationality.

- R_i: The overall rating of player i.

- P_{ik}: Binary, 1 if player i can play in position k.

- $nat_i, club_i, league_i$: The specific nation, club, and league of player i.

- $Icon_i, Hero_i$: Binary, indicating if player i is an Icon or Hero.

- T_m^{cat}: Integer thresholds required to achieve chemistry tier m for a category ‘cat‘.

2.3 Decision Variables

At the heart of the model are the decision variables. These are the unknown values that the solver will determine. They represent the choices we need to make to build the optimal team.

- $x_{ik} \in \{0, 1\}$: 1 if player i is assigned to position k.
- $\gamma_{jm}^{cat} \in \{0, 1\}$: 1 if category instance j achieves chemistry tier m.
- $ch_i \in \{0, 1, 2, 3\}$: Potential chemistry points for player i.
- $ch_k^{pos} \in \{0, 1, 2, 3\}$: Chemistry of the player in position k.
- $ch_k^{final} \in \{0, 1, 2, 3\}$: Final chemistry contribution from position k.

2.4 Objective Functions

The goals of our optimization are captured in the objective functions. Since we have two competing goals, we define two distinct functions to maximize.

$$\text{Maximize} \quad Z_{Rating} = \sum_{i \in I} \sum_{k \in K} R_i \cdot x_{ik} \tag{1}$$

$$\text{Maximize} \quad Z_{Chemistry} = \sum_{k \in K} ch_k^{final} \tag{2}$$

2.5 Constraints

To ensure that the solutions generated by the model are valid according to the rules of the game, we must impose a series of constraints. These mathematical statements enforce all the team-building and chemistry rules.

2.5.1 Player Assignment Constraints

Each position must be filled by exactly one player.

$$\sum_{i \in I} x_{ik} = 1 \quad \forall k \in K \tag{3}$$

Each player can be assigned to at most one position.

$$\sum_{k \in K} x_{ik} \leq 1 \quad \forall i \in I \tag{4}$$

A player must be eligible for an assigned position.

$$x_{ik} \cdot P_{ik} = x_{ik} \quad \forall i \in I, \forall k \in K \tag{5}$$

2.5.2 Chemistry Calculation Constraints

Chemistry tiers are unlocked based on player counts.

$$\sum_{m\in M} T_m^{nat} \cdot \gamma_{j,m}^{nat} \leq \sum_{i\in I, nat_i=j} \sum_{k\in K} (1 + Icon_i) \cdot x_{ik} \quad \forall j \in J_{nat} \tag{6}$$

$$\sum_{m\in M} T_m^{league} \cdot \gamma_{j,m}^{league} \leq \sum_{i\in I, league_i=j} \sum_{k\in K} (1 + Hero_i) \cdot x_{ik} \quad \forall j \in J_{league} \tag{7}$$

$$\sum_{m\in M} T_m^{club} \cdot \gamma_{j,m}^{club} \leq \sum_{i\in I, club_i=j} \sum_{k\in K} x_{ik} \quad \forall j \in J_{club} \tag{8}$$

Only one chemistry tier can be active per category.

$$\sum_{m\in M} \gamma_{jm}^{cat} \leq 1 \quad \forall j \in J_{cat}, \forall cat \in \{\text{nat, league, club}\} \tag{9}$$

Individual chemistry is the sum of points from active tiers.

$$ch_i \leq \sum_{m\in M} m \cdot \gamma_{nat_i,m}^{nat} + \sum_{m\in M} m \cdot \gamma_{league_i,m}^{league} + \sum_{m\in M} m \cdot \gamma_{club_i,m}^{club} \quad \forall i \in I \tag{10}$$

Individual chemistry is capped at 3 points.

$$ch_i \leq 3 \quad \forall i \in I \tag{11}$$

2.5.3 Team Chemistry and Linearization Constraints

The chemistry of a position is linked to the assigned player.

$$ch_k^{pos} \leq (1 - x_{ik}) \cdot 3 + ch_i \quad \forall i \in I, \forall k \in K \tag{12}$$

Final chemistry accounts for Icon and Hero bonuses.

$$ch_k^{final} \leq ch_k^{pos} + 3 \cdot \sum_{i'\in I_{special}} x_{i'k} \quad \forall k \in K \tag{13}$$

3 Solution Methodology: Generating the Pareto Frontier

Having defined the problem in its pure, multi-objective form, we now turn to the practical methodology for finding the optimal solutions. Because this is a multi-objective problem, there is no single solution that is optimal for all objectives. Instead, we seek the **Pareto frontier**, a set of non-dominated solutions where improving one objective is only possible by degrading another.

Standard MILP solvers, such as Google OR-Tools, are designed to handle single-objective problems. Therefore, we must employ a strategy to explore the solution space and construct the Pareto frontier ourselves. The chosen approach is an iterative search with dominance-breaking constraints.

3.1 Scalarized Objective Function

To utilize standard MILP solvers, we convert the multi-objective problem into a single-objective one by scalarizing the objectives. We use a simple weighted sum with a fixed weight $\alpha \in [0, 1]$ (typically 0.5):

$$\text{Maximize} \quad (1 - \alpha) \left(\frac{Z_{Rating}}{Z_{Rating}^{max}} \right) + \alpha \left(\frac{Z_{Chemistry}}{Z_{Chemistry}^{max}} \right) \tag{14}$$

Where Z^{max} terms are normalization constants. While traditionally one might vary α to trace the Pareto frontier, our approach relies on a more robust iterative search mechanism.

3.2 Iterative Search and Pareto Constraints

The core of our methodology is an iterative search process that guarantees the exploration of the Pareto frontier. Since the solver is constrained to never return a solution that is dominated by or identical to a previously found solution, it is forced to find a new point on the Pareto front in each iteration. This renders the variation of the weight α largely redundant; even with a fixed weight, the constraints drive the solver to discover the entire frontier.

Let $\mathcal{P}$ be the set of all Pareto-optimal teams found so far. In each new iteration, we add constraints to ensure the next solution found is not dominated by any solution $p \in \mathcal{P}$.

Banning Existing Solutions: This constraint prevents the solver from returning the exact same set of 11 players.

$$\sum_{i \in I_p} \sum_{k \in K} x_{ik} \leq 10 \tag{15}$$

Dominance-Breaking Constraints: These constraints are the core of the iterative search. For each team $p \in \mathcal{P}$, which has a known normalized rating R_p and chemistry C_p, we tell the solver that the new team it finds must not be dominated by team p. This means the new team must be either strictly better on at least one objective, or have the same objective values but with a different set of players.

- $\delta_p^{rating}, \delta_p^{chem} \in \{0, 1\}$: Binary variables. $\delta_p^{rating} = 1$ if the new team's rating is strictly better than team p's rating.
- $\rho_p^{rating}, \rho_p^{chem} \geq 0$: Continuous variables that measure the ratio of the new objective value to the old one.

The following constraints work together for each existing Pareto solution p:

$$\delta_p^{rating} + \delta_p^{chem} + 0.5 \cdot \rho_p^{rating} + 0.5 \cdot \rho_p^{chem} \geq 1 \tag{16}$$

This is the main logic constraint. It forces the solver into one of three situations:

1. The new solution is strictly better in rating ($\delta_p^{rating} = 1$).
2. The new solution is strictly better in chemistry ($\delta_p^{chem} = 1$).
3. The new solution has identical objective values to team p. In this case, both δ variables must be 0. The constraint can then only be satisfied if both ρ variables are forced to be 1.

The next constraints define when the δ and ρ variables can be activated. The subtlety of the model lies in how the ρ variables allow for the third case.

$$\frac{Z_{Rating}}{Z_{Rating}^{max}} \geq R_p + (\delta_p^{rating} - 1) + \epsilon \tag{17}$$

$$\frac{Z_{Chemistry}}{Z_{Chemistry}^{max}} \geq C_p + (\delta_p^{chem} - 1) + \epsilon \tag{18}$$

These ensure that if δ_p^{rating} is 1, the new rating must be greater than the old rating R_p by at least a small amount ϵ. If δ_p^{rating} is 0, the constraint simply becomes $\frac{Z_{Rating}}{Z_{Rating}^{max}} \geq R_p - 1$, which is always true.

$$R_p \cdot \rho_p^{rating} \leq \frac{Z_{Rating}}{Z_{Rating}^{max}} \tag{19}$$

$$C_p \cdot \rho_p^{chem} \leq \frac{Z_{Chemistry}}{Z_{Chemistry}^{max}} \tag{20}$$

These constraints define the ratio variables. Consider the rating ratio ρ_p^{rating}. If the new solution's rating is greater than R_p, then ρ_p^{rating} can be greater than 1. If the new rating is less than R_p, ρ_p^{rating} will be less than 1. Crucially, if the new rating is exactly equal to R_p, the constraint becomes $R_p \cdot \rho_p^{rating} \leq R_p$, which implies $\rho_p^{rating} \leq 1$. Since the main logic constraint (25) is trying to maximize its left-hand side, the solver will push ρ_p^{rating} to its maximum possible value, which is exactly 1 in this case. This is the mechanism that allows the model to find alternative teams with the same objective scores, satisfying constraint (25) via the ρ variables when the δ variables are both zero.

3.3 Algorithm for Iterative Search

The entire process for finding the complete Pareto frontier across all formations can be summarized in the following algorithm.

The function `UpdateParetoFrontier` takes a list of teams and filters out any team that is dominated by another in the list, ensuring that $\mathcal{P}$ always contains only non-dominated solutions. This overall approach allows for a systematic exploration of the trade-offs between rating and chemistry.

Algorithm 1 Iterative Pareto Frontier Generation

```
**Input:** Player set $I$, list of Formations $F$
**Output:** The final Pareto frontier $\mathcal{P}$
Initialize $\mathcal{P} \leftarrow \emptyset$
**for** each formation $f$ in $F$ **do**
  keep_searching $\leftarrow$ true
  **while** keep_searching **do**
    Build MILP model for formation $f$
    Add Dominance-Breaking constraints for each team $p \in \mathcal{P}$
    Solve the MILP model
    **if** a new team $T$ was found **then**
      $\mathcal{P} \leftarrow \mathcal{P} \cup \{T\}$
      $\mathcal{P} \leftarrow$ UpdateParetoFrontier($\mathcal{P}$) {Remove any newly dominated teams}
    **else**
      keep_searching $\leftarrow$ false {No more solutions for this formation}
**return** $\mathcal{P}$
```