17th September 2025

MIGe Departement
University of Studies of Trieste

Fantasy Football Draft using Genetic Algorithms with Monte Carlo sampling

Exam presentation for:
Optimization for AI

Student:
Gabriel Masella [SM3800048]

Table of Contents:

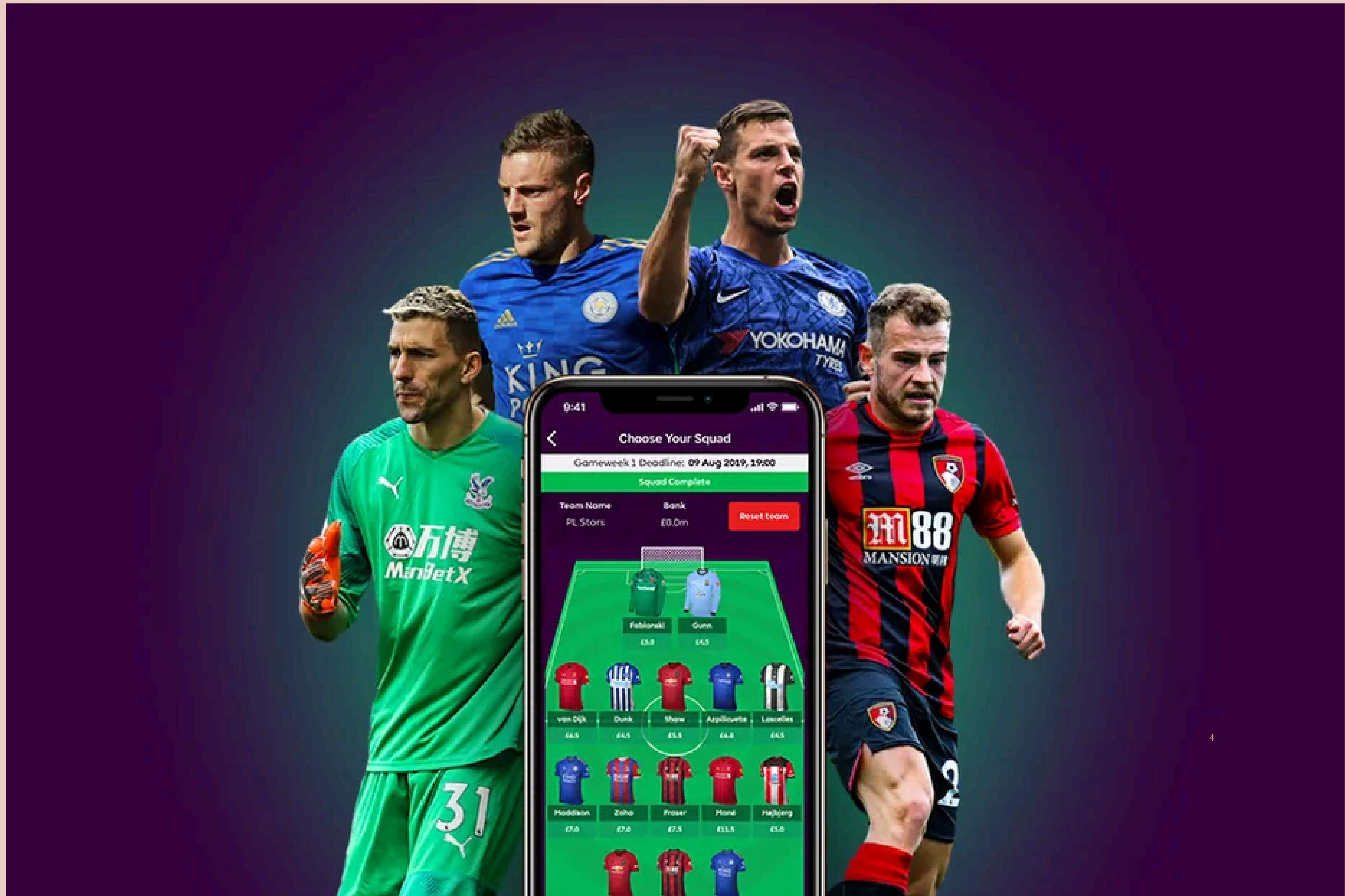
1.

Introduction

An overview of what a Fantasy Football League is, including its statistical variant, and outlines the purpose of this project within the context of the "Optimization for AI" course.

What is a Fantasy Football League?

- **Goal:** Build your own dream team of real players at the beginning of the season
- **Starting Point:** Everyone gets the same virtual budget (e.g., €250M)
- **Player Prices:** Set based on real-world skill/performances
- **Draft Rule:**
  - Stay within budget. same number of players for everyone;
  - same player can be chosen by multiple managers
  - Usually 11 startes + 7 substitues (2 GK, 6 DF, 6 MF, 4 FW)
- **Scoring:** Points earned from players' real match performances (goals, assists, saves, etc.)
- **Competition:** Compare total points with other managers over the season



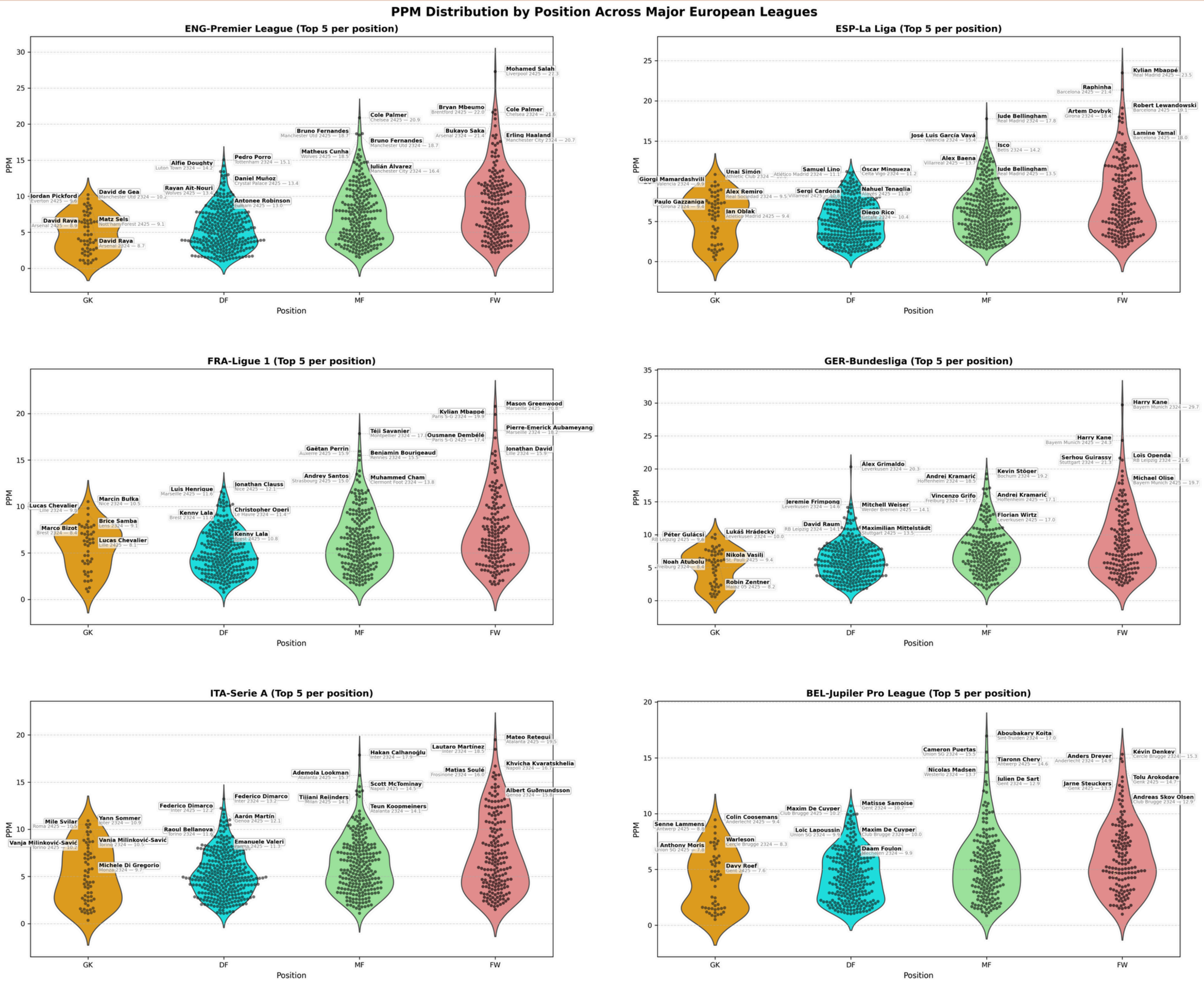
1. Introduction

PPM (Points per Match) and Market Price

Six datasets were created from various football leagues:

- Premier League (453/493 eligible players)
- Serie A (470/533 eligible players)
- Bundesliga (383/441 eligible players)
- La Liga (463/546 eligible players)
- Ligue 1 (392/482 eligible players)
- Jupiler Pro League (334/425 eligible players)

Sufficient data from their second divisions was available for the evaluation of player performance over a two-year period.
For each season, the **PPM (Points per Match)** of all players was calculated and used, together with **age**, **team strength**, **position**, and **player strength** compared to other players in his role and his teammates, to calculate the **market value**.



Aim of the project

The objective of this project is to automatically choose the ideal roster of players at the start of each season:

- Implementing an Optimization Algorithm, specifically a Genetic Algorithm (GA) for our needs,
- Adhering to common constraints found in the fantasy leagues,
- Utilizing player performances from the previous season while taking uncertainty into consideration.

2.

Genetic Algorithms

The theoretical foundation of the project, from the mathematical interpretation of this real-world scenario to the Genetic Algorithm (GA) cycle that has been implemented.

Mathematical Formulation of the problem

The Fantasy Football Draft problem can be elegantly formulated as a variant of the **binary knapsack problem with additional constraints.** This is a classic combinatorial optimization problem where we must select items (in this case, football players) to maximize value while respecting multiple constraints.

For each player i in our set of available players $N = \{1, 2, \ldots, n\}$, we define a binary decision variable:

$$x_i = \begin{cases} 1 & \text{if player } i \text{ is selected} \\ 0 & \text{if player } i \text{ is not selected} \end{cases}$$

Our goal is to maximize the total **expected Points Per Match (PPM)** of the selected players:

$$\max \sum_{i=1}^{n} \mathrm{PPM}_i \cdot x_i$$

Note: In the implementation, this is converted to a minimization problem by negating the objective

Mathematical Formulation of the problem

To the previous considerations, we must add the following constraints:

1. **Budget Constraint (Inequality)**
   a. Total player cost must not exceed budget
2. **Position Constraints (Equality):**
   a. Exact number of players required for each position
3. **Team Diversity Constraint (Inequality):**
   a. At most M players from any single team

Combing all this we obtain the complete model on the right, that we must optimize.

$$\text{maximize} \sum_{i=1}^{n} \text{PPM}_i \cdot x_i$$

subject to:

$$\sum_{i=1}^{n} \text{price}_i \cdot x_i \leq B$$

$$\sum_{i \in P_{\text{GK}}} x_i = N_{\text{GK}}$$

$$\sum_{i \in P_{\text{DF}}} x_i = N_{\text{DF}}$$

$$\sum_{i \in P_{\text{MF}}} x_i = N_{\text{MF}}$$

$$\sum_{i \in P_{\text{FW}}} x_i = N_{\text{FW}}$$

$$\max_{t \in T} \left(\sum_{i \in T_t} x_i \right) \leq M$$

$$x_i \in \{0, 1\} \quad \forall i \in N$$

Mathematical Formulation of the problem

The model was extended to incorporate uncertainty in player performance.

Instead of using fixed PPM values, we can sample from a **gaussian distribution:**

$$\mathrm{PPM}_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$$

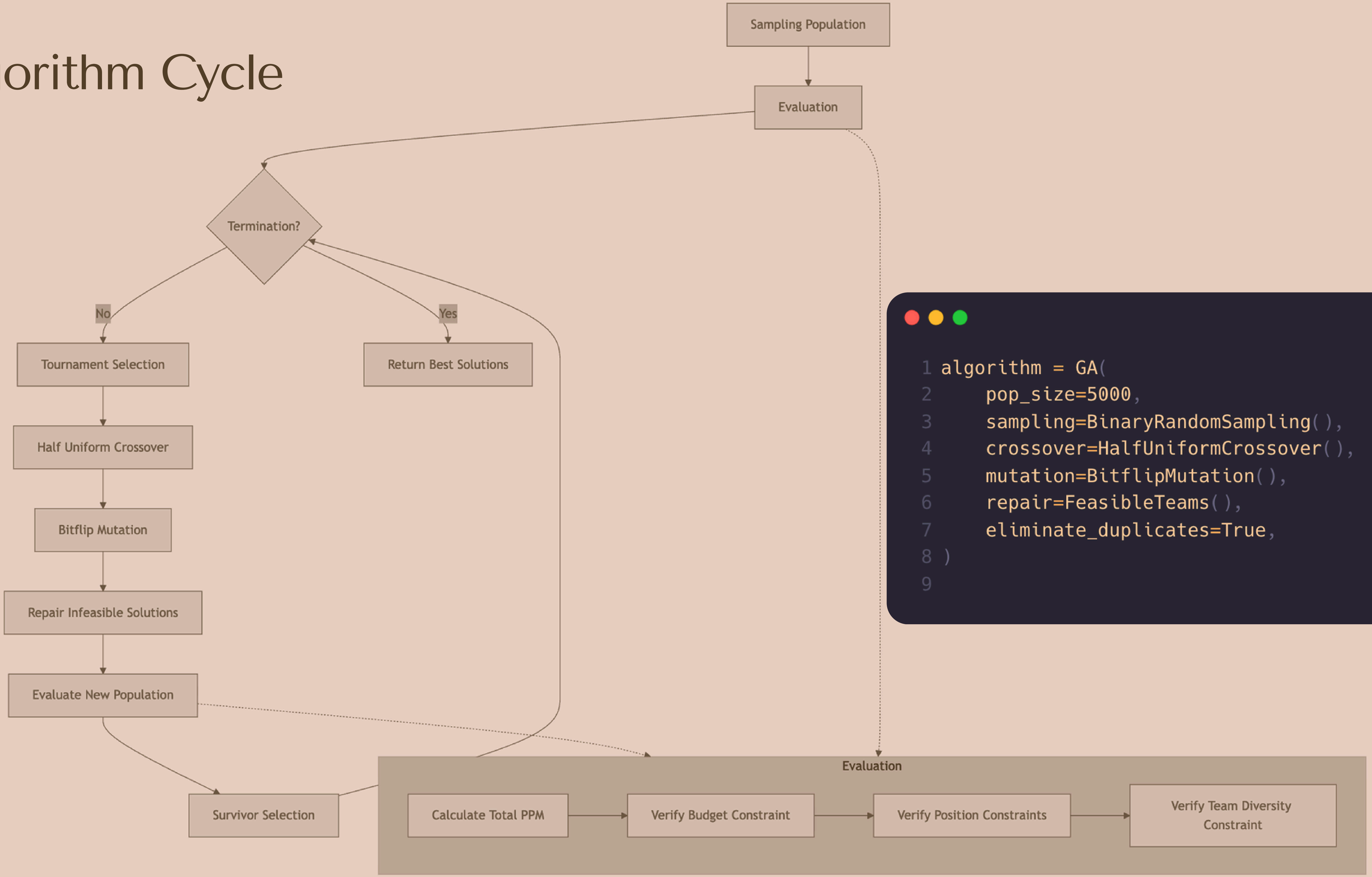
The objective function then becomes:

$$\max \mathbb{E}\left[\sum_{i=1}^{n} \mathrm{PPM}_i \cdot x_i\right]$$

This stochastic objective is approximated through **Monte Carlo simulations** in the implementation, providing a more robust solution that accounts for performance variability.

2. Genetic Algorithm

Genetic Algorithm Cycle




```python
algorithm = GA(
    pop_size=5000,
    sampling=BinaryRandomSampling(),
    crossover=HalfUniformCrossover(),
    mutation=BitflipMutation(),
    repair=FeasibleTeams(),
    eliminate_duplicates=True,
)
```

3.

Experimental Results

Let's examine the outcomes derived from the implemented algorithm, comparing the "Stochastic" and "Deterministic" approaches, along with their performance against the "perfect" solution provided by Linear Programming (LP).

3. **Experimental Results**

Convergences Comparison

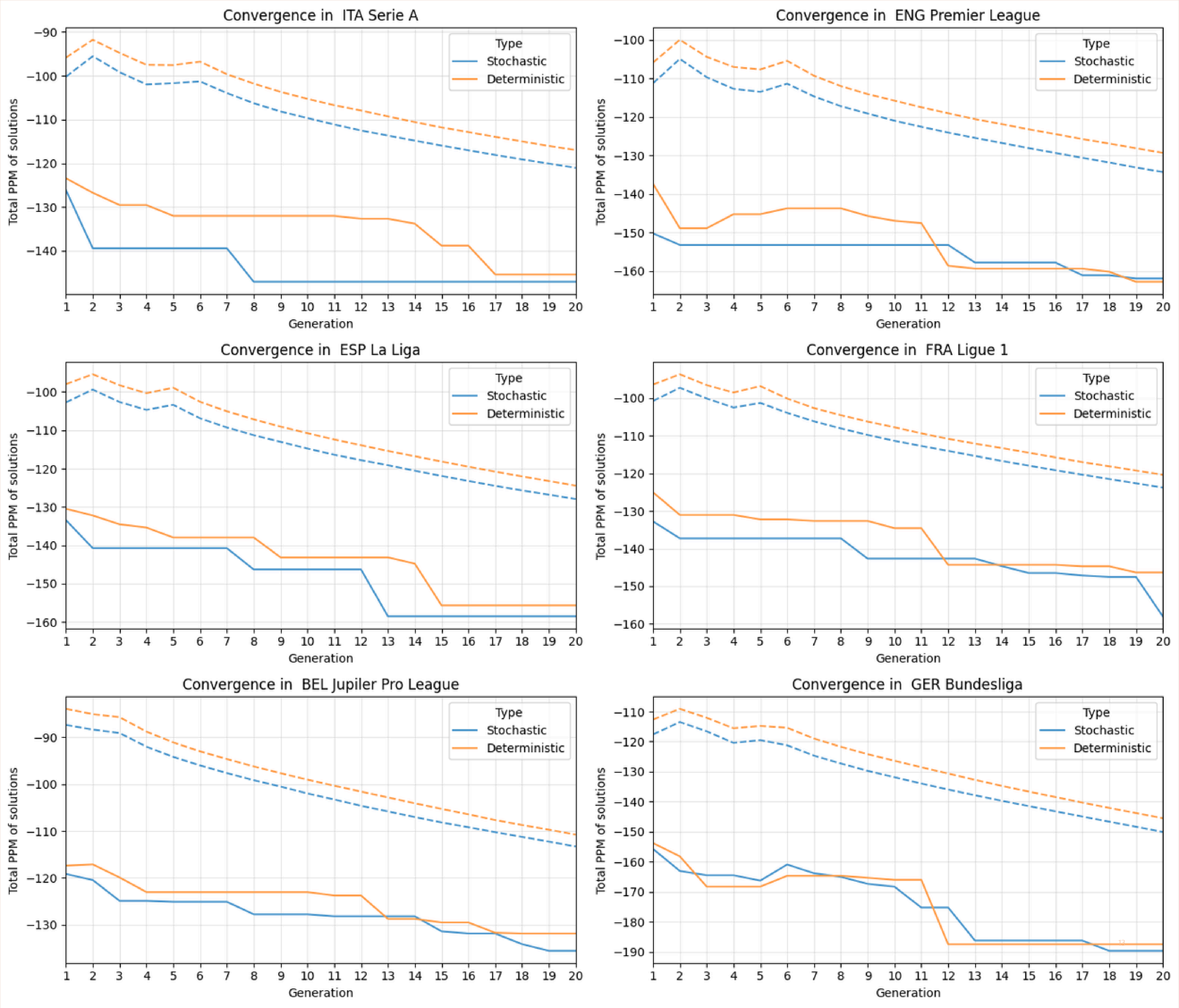
The experiment was run for the six different leagues with two different approaches:

- **"Stochastic"** means when the Monte Carlo sampling was used in evaluation phase
- **"Deterministic"** with the regular GA.

The scenario under analysis is the following:

```
problem = FantasyTeamProblem(fantasy_datasets[league],
                budget=250,# Total Money available
                max_team=4, # maximum 4 players from the same club
                num_gk=2, num_df=6, num_mf=6, num_fw=4,
                num_simulations=20) # Number of sampling,
```

where the price of the players are clipped between £4M and £25M



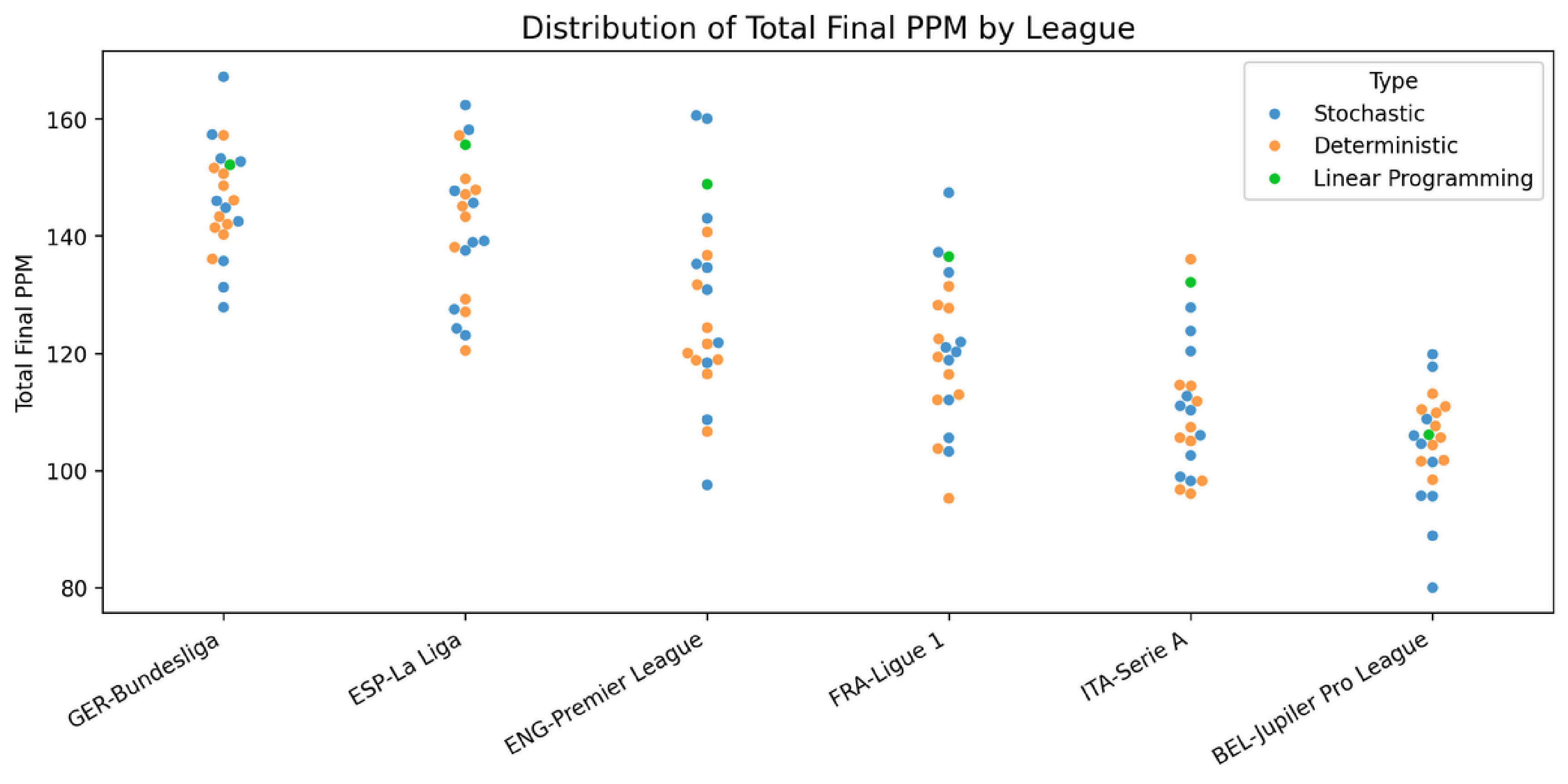
3. **Experimental Results**

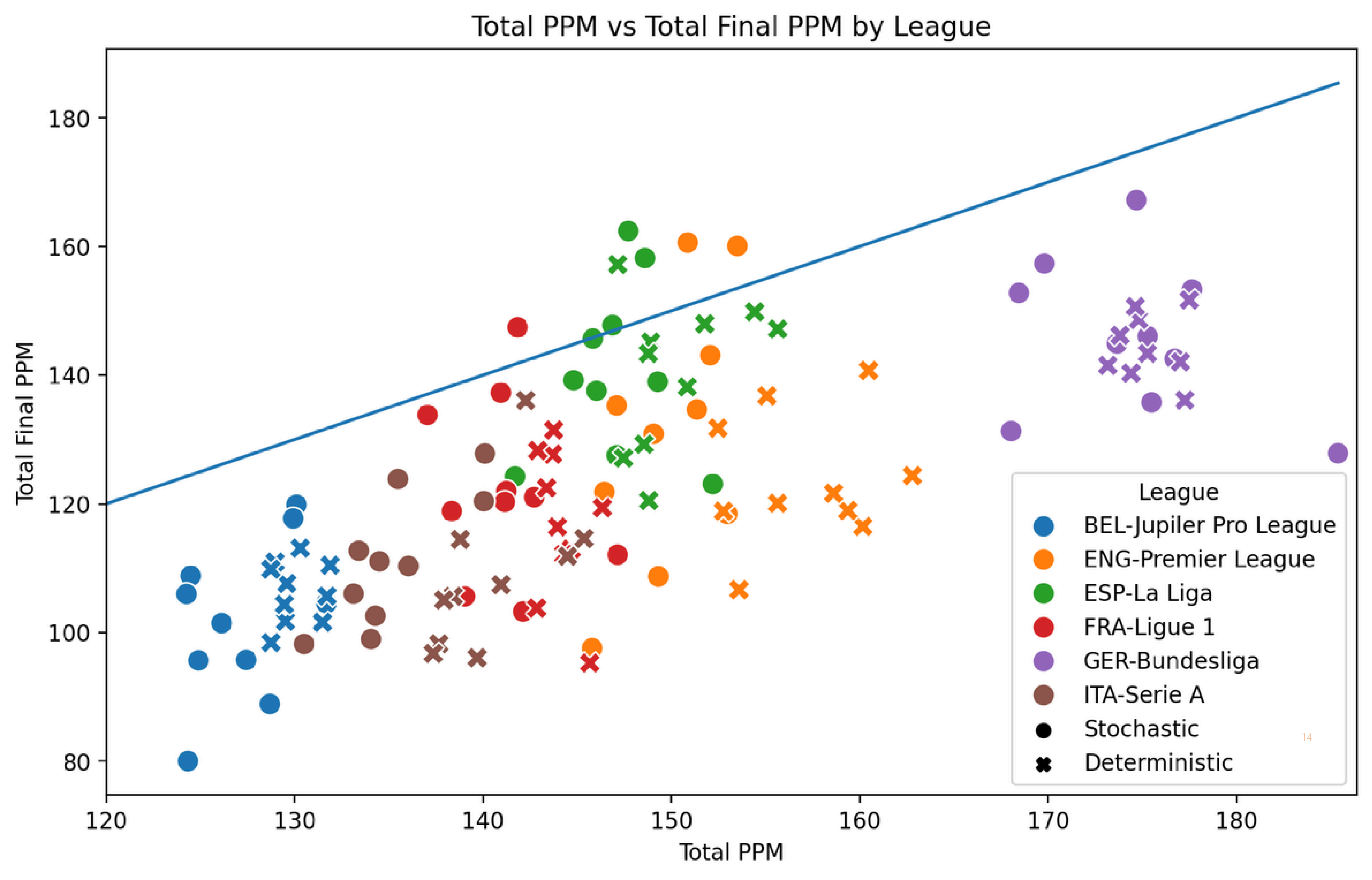
Comparison of Results Between Different Approaches and LP solution

- **Stochastic GA**: This approach leads to broader distributions and heavier upper tails across many leagues. It explores riskier portfolios that can achieve greater realized outcomes (higher upside), but it also incurs a greater spread (increased downside risk).

- **Deterministic GA**: This method results in narrower distributions (less variance) and often focuses on high expected PPM players. However, it can overfit point estimates, as many solutions fall below the line y=x in scatter plots, indicating an optimistic bias.

As reference we use also an LP algorithm to find the "best" solution maximizing purely PPM, with the same constraints.

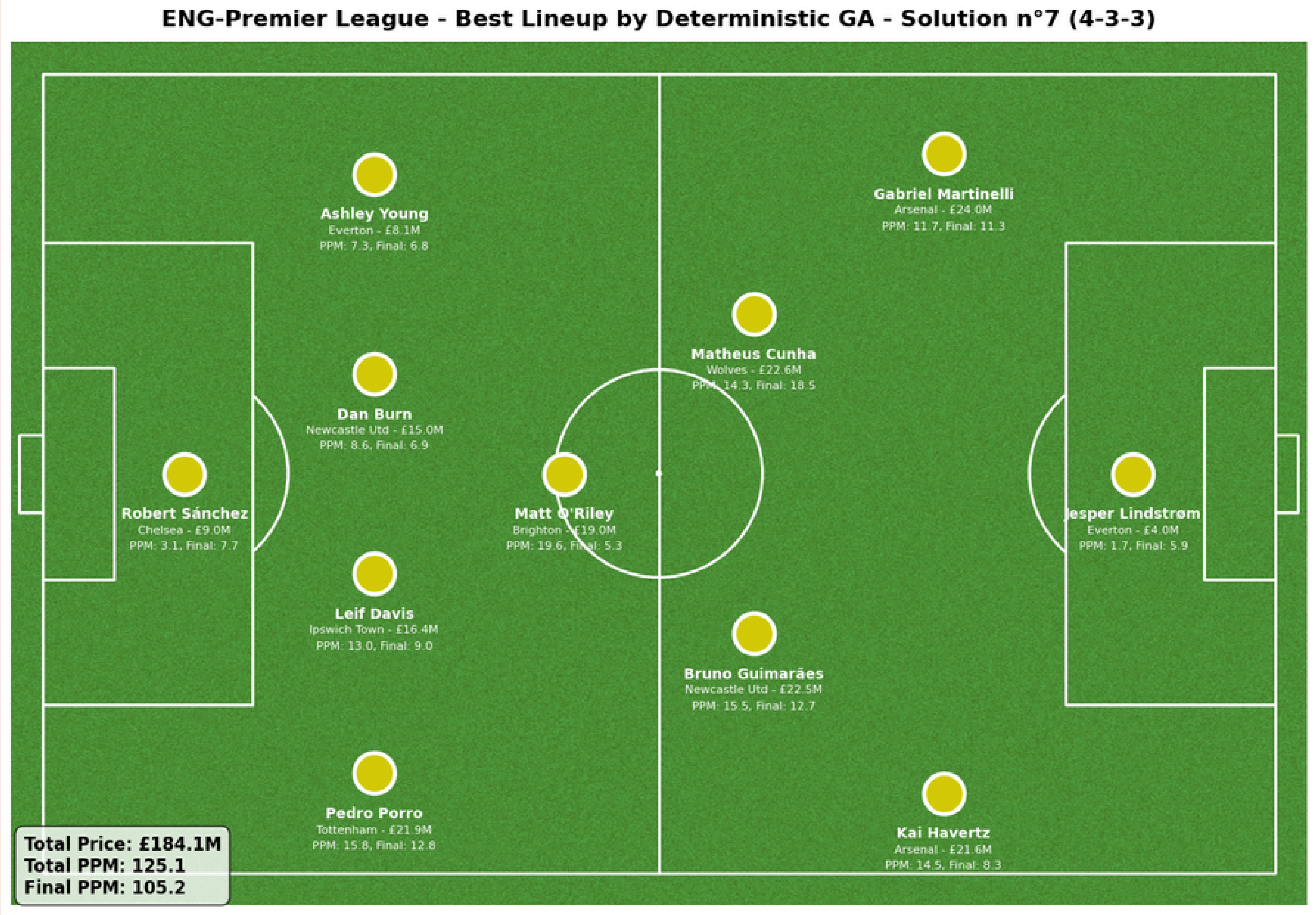


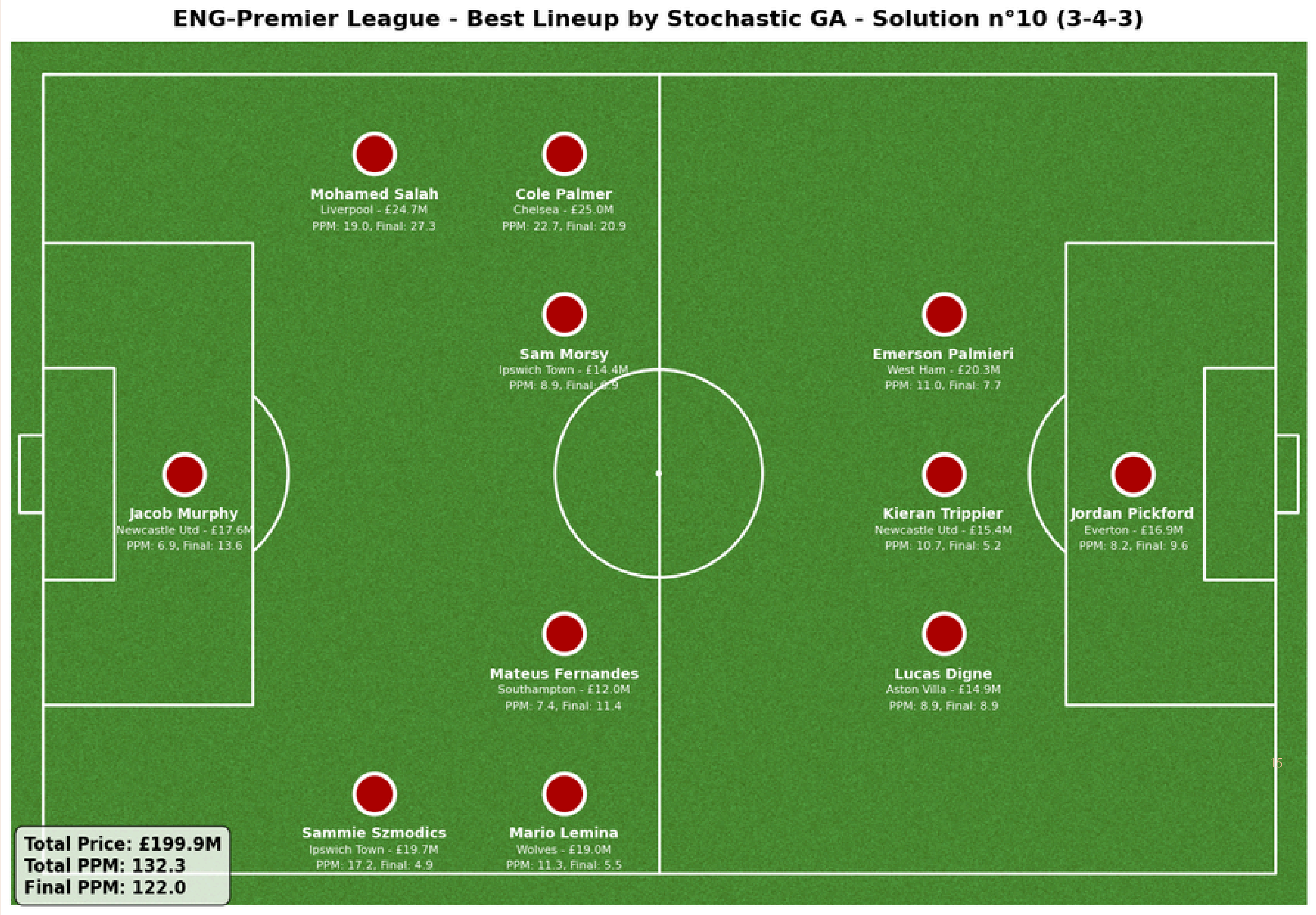


3. Experimental Results

Comparison of Results by Lineups

To assess the performance of the two algorithm versions, we determine the optimal lineup for each roster. We then calculate the Points Per Match (PPM) and Final PPM for comparison, using the best possible roster generated by the LP Algorithm as reference.



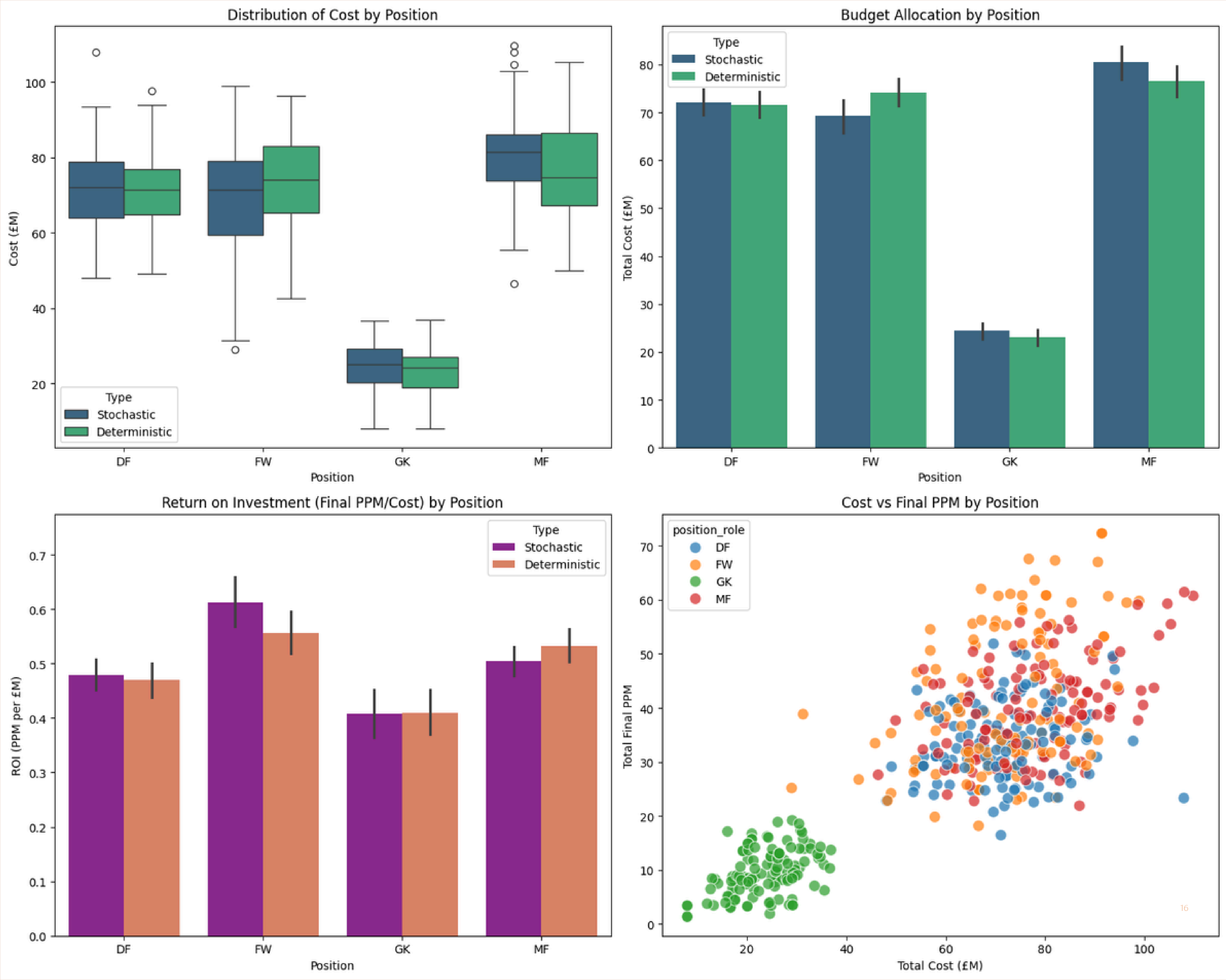


3. **Experimental Results**

Budget Allocation Analysis

Another interesting aspect to analyze for a better decision-making process it's to study how the algorithms decides to spend their tight budget

Position Role	Average Cost	Cost Percentage (%)	Average Final PPM	Avg %ROI (PPM per £M)
DF	71,78	29,21	33,53	48,00
FW	71,77	29,20	41,31	58,00
GK	23,73	9,66	9,54	41,00
MF	78,46	31,93	40,17	52,00



4. FURTHER IMPROVEMENTS

This project represent a good point of starts, but of course, we can always do better and here's how we can improve results:

- **Improved Data Quality and Advanced Prediction Models**
  - Machine Learning techniques to forecast player performances.
  - Transition from a seasonal overview to gameweek optimization.
- **Multi-Objective Optimization using NSGA-II:**
  - Analyze PPM in relation to Budget Efficiency, or strenght ratio between lineups/sub, or other
- **Parallel Optimization:**
  - Better results through distributed training, leveraging diverse populations (ensuring variety with over 300 players in the league) while employing Star Topology to gather the best outcomes.

MIGe Departement
University of Studies of Trieste

Thanks for your attention!

Feel free to ask any questions

1. **Introduction**

Classical Fantasy League vs. Statistical Format

Feature	Classical Fantasy Leagues	Statistical Fantasy League
Scoring Basis	Primarily subjective ratings from sports journalists + a few key bonuses (e.g., goals, assists, clean sheets)	Purely statistical: every measurable action (passes, tackles, shots, dribbles, etc.) counts
Bonuses/Penalties	Few but impactful (goal = big boost, red card = big drop)	Many small contributions from a wide range of actions; cumulative effect
Transparency	Dependent on human evaluations, which can be debated	Fully objective — data from official providers like Opta
Strategy Focus	Pick players likely to get good ratings and score/assist	Balance star players with high-activity players who rack up stats across the board

1. **Introduction**

Statistical Rules adopted for this project

The rules for scoring are modeled after Kickest's Fantasy League, using available league data from FBRef, our free data source provider. Points are assigned as follows:

- **5 points** for each **start**, **2 points** for **substitute** appearances
- **Goals: 14 points** (forwards)**, 16** (midfielders)**, 18** (defenders)**, 50** (goalkeepers)
- **10 points** for **penalty goals**, **-10 for missed penalties**
- **7 points** for **assists**
- **-2 points** for **yellow cards**, **-5** for **red cards**
- **2 points** for **shots on target**, **1 point** for **shots**
- **-1 point** for **fouls**, **0.5** for **fouls won**
- **-0.5** for **offsides**, **0.5** for **crosses**
- **1 point** for **interceptions**, **2 points** for **tackles won**
- **10 points** for **clean sheets** (goalkeepers only)
- **-2 points** for **goals conceded** (goalkeepers only)
- **1 point** for **saves** (goalkeepers only), **15 points** for **penalty saves** (goalkeepers only)

Genetic Algorithm Cycle

The implementation utilizes the **PyMOO optimization framework** for a genetic algorithm (GA) with the following components:

1. **Selection Mechanism:**
   - Tournament selection drives the population towards optimal solutions.
2. **Crossover Operator:**
   - Half Uniform Crossover (HUX) exchanges half of differing bits.
3. **Mutation Operator:**
   - Bitflip Mutation introduces random variation by flipping bits.
4. **Repair Mechanism:**
   - A custom FeasibleTeams operator addresses constraint violations.
   - Ensures solutions adhere to position, budget, and diversity constraints.
5. **Evaluation Strategy:**
   - Supports both deterministic and stochastic evaluations using Monte Carlo sampling.
   - Employs multi-threading for efficiency in evaluations.
6. **Termination Criteria:**
   - The algorithm stops after a predetermined number of generations (20).

```
algorithm = GA(
    pop_size=5000,
    sampling=BinaryRandomSampling(),
    crossover=HalfUniformCrossover(),
    mutation=BitflipMutation(),
    repair=FeasibleTeams(),
    eliminate_duplicates=True,
)

```

3. Experimental Results

Comparison of Results by Lineups

To assess the performance of the two algorithm versions, we determine the optimal lineup for each roster. We then calculate the Points Per Match (PPM) and Final PPM for comparison, using the best possible roster generated by the LP Algorithm as reference.