Critical Analysis of the FPL Team Picker Algorithm

A Technical Review of the Expected Points Model,
Transfer Optimizer, Starting XI Selection, and Captain Strategy

Algorithm Analysis Report
FPL Team Picker 2025–26

December 2025

Abstract

This report provides a critical technical analysis of the Fantasy Premier League (FPL) Team Picker algorithm, consisting of four main components: (1) the Machine Learning Expected Points (xP) model with 122-feature engineering, (2) the Linear Programming / Simulated Annealing transfer optimizer, (3) the Starting XI selector with formation enumeration, and (4) the situation-aware captain selection system. We examine the mathematical foundations, implementation choices, and identify both strengths and weaknesses with recommendations for improvement.

Contents

1 System Architecture Overview

The FPL Team Picker implements a layered optimization architecture with four distinct layers:

1. **Prediction Layer**: ML Expected Points Service (122 features)
2. **Tactical Layer**: Transfer & Squad Optimization (LP/SA)
3. **Selection Layer**: Starting XI & Captain Selection
4. **Risk Layer**: Uncertainty Quantification & Template Protection

The core decision pipeline is:

$$\text{Raw Data} \xrightarrow{\text{Feature Engineering}} \text{xP Predictions} \xrightarrow{\text{LP/SA}} \text{Optimal Squad} \xrightarrow{\text{Selection}} \text{Starting XI + Captain}$$

2 Expected Points Model Training

2.1 Mathematical Formulation

The ML Expected Points model predicts the total points y_i for player i in gameweek $t+1$ using:

$$\hat{y}_{i,t+1} = f_\theta(\mathbf{x}_{i,t}) \tag{1}$$

where f_θ is a trained ensemble model (RandomForest, XGBoost, LightGBM, or GradientBoosting) and $\mathbf{x}_{i,t} \in \mathbb{R}^{122}$ is the feature vector constructed from historical data up to gameweek t.

2.2 Feature Engineering Architecture (122 Features)

The `FPLFeatureEngineer` transformer generates features across 12 categories:

Category	**Description**	**Count**
Static Features	Price, position encoding, games played	4
Cumulative Season Stats	Goals, assists, clean sheets, etc.	11
Cumulative Per-90 Rates	Goals/90, points/90, xG/90	7
Rolling 5GW Form	Recent performance windows	13
Rolling 5GW Per-90	Efficiency over recent form	3
Defensive Metrics	Goals conceded, saves, xGC	4
Consistency & Volatility	Std dev of points/minutes, form trend	3
Team Features	Team-level rolling stats	13
Fixture Features	Home/away, opponent strength	6
Enhanced Ownership	Transfer momentum, bandwagon score	5
Enhanced Value	Points per pound, price volatility	4
Penalty/Set-Piece	Penalty, corner, FK taker flags	4
Betting Odds	Win probability, expected goals	15
Injury/Rotation Risk	Injury risk, rotation likelihood	5
Venue-Specific Strength	Home/away attack/defense ratings	6
Player Rankings	Form rank, ICT rank, tackles	7
Data Quality Indicators	Flags for available data	5
Elite Interactions	Elite player × fixture difficulty	4
Total		**122**

Table 1: Feature categories in the ML pipeline

2.3 Temporal Leak Prevention

A critical design principle is **leak-free features**. All temporal features use strict lookback:

$$\text{rolling\_5gw\_points}_{i,t} = \frac{1}{5}\sum_{k=1}^{5} y_{i,t-k} \tag{2}$$

The implementation uses `shift(1)` operations to ensure no future data leakage.

Exception: Betting odds features are *forward-looking* by nature (available before match kickoff) and require no temporal shift.

2.4 Uncertainty Quantification

The system extracts prediction uncertainty from ensemble disagreement:

$$\sigma_i = \text{std}\left(\{f_k(\mathbf{x}_i)\}_{k=1}^{K}\right) \tag{3}$$

where f_k represents individual trees in the ensemble. For boosting algorithms (XGBoost, LightGBM), tree contributions are scaled by learning rate:

$$\sigma_i^{\text{XGB}} = \frac{\text{std(tree contributions)}}{\eta} \tag{4}$$

where η is the learning rate.

2.5 Strengths of the XP Model

- **Comprehensive feature set**: 122 features capture diverse signal sources (form, team strength, betting markets, ownership dynamics)
- **Leak-free design**: Strict temporal ordering prevents training-test contamination
- **Uncertainty quantification**: Tree-level variance enables risk-aware decisions
- **Domain-aware imputation**: Position-specific defaults instead of generic fillna(0)
- **Per-gameweek team strength**: Avoids using future information for fixture difficulty
- **Multi-horizon prediction**: Cascading 3GW/5GW predictions use synthetic data

2.6 Weaknesses of the XP Model

- **Feature redundancy**: Several highly correlated features exist (e.g., `net_transfers_gw` and `ownership_velocity`). Feature selection via RFE helps but adds complexity.
- **Limited temporal dynamics**: Rolling 5GW window is fixed; adaptive windows based on recent performance volatility could improve signal extraction.
- **Position-agnostic model**: A single model predicts all positions, yet scoring mechanisms differ significantly (GKP: saves, DEF: clean sheets, FWD: goals). Position-specific models could improve accuracy.
- **No explicit fixture sequence modeling**: Features capture opponent strength but not fixture *patterns* (e.g., tough-easy-tough sequences that affect rotation).
- **Betting odds availability**: Forward-looking betting features assume pre-match odds exist; for future gameweeks, these must be estimated or set to neutral defaults.
- **Cold-start problem**: New players lack historical features. Current solution: fallback to position-based defaults, but this loses player-specific signal.

3 Transfer Optimization

3.1 Problem Formulation

The transfer optimization is formulated as an Integer Linear Program (ILP):

$$\begin{aligned}
\max_{\mathbf{x}} \quad & \sum_{i=1}^{N} \mathrm{xP}_i \cdot x_i - c \cdot \max(0, T - F) && (5)\\
\text{s.t.} \quad & \sum_{i=1}^{N} p_i \cdot x_i \leq B \quad \text{(budget)} && (6)\\
& \sum_{i \in \mathcal{P}_j} x_i = n_j \quad \forall j \in \{\text{GKP, DEF, MID, FWD}\} \quad \text{(positions)} && (7)\\
& \sum_{i \in \mathcal{T}_k} x_i \leq 3 \quad \forall k \in \text{Teams} \quad \text{(team limit)} && (8)\\
& \sum_{i \in \mathcal{S}} x_i \geq 15 - M \quad \text{(transfer limit)} && (9)\\
& x_i \in \{0, 1\} \quad \forall i && (10)
\end{aligned}$$

where:

- $x_i = 1$ if player i is in the squad
- xP_i is expected points for player i
- p_i is player price, B is total budget
- $c = 4$ is the transfer cost, F is free transfers, T is total transfers
- $\mathcal{S}$ is the current squad, M is max transfers allowed

3.2 LP vs SA Comparison

Aspect	Linear Programming	Simulated Annealing
Optimality	Guaranteed optimal	Approximate (95-99%)
Speed (transfers)	1-2 seconds	10-15 seconds
Speed (wildcard)	1-2 seconds	30-60 seconds
Determinism	Yes	No (seed required)
Constraint handling	Native	Penalty functions
Non-linear objectives	No	Yes

Table 2: Comparison of optimization methods

3.3 Starting XI Optimization

For 1-gameweek optimization, the LP additionally optimizes the starting lineup:

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{N} \mathrm{xP}_i \cdot s_i - \beta \sum_{i=1}^{N} p_i (x_i - s_i) && (11) \\
\text{s.t.} \quad & s_i \le x_i \quad \forall i \quad \text{(can only start if in squad)} && (12) \\
& \sum_{i=1}^{N} s_i = 11 \quad \text{(exactly 11 starters)} && (13) \\
& \text{Formation constraints: } 1 \text{ GKP}, 3\text{–}5 \text{ DEF}, 2\text{–}5 \text{ MID}, 1\text{–}3 \text{ FWD} && (14)
\end{aligned}
$$

The term $-\beta \sum p_i(x_i - s_i)$ penalizes expensive bench players, encouraging budget allocation to starters.

3.4 Strengths of Transfer Optimization

- **Guaranteed optimality**: LP provides provably optimal solutions for the defined objective
- **10-50x speedup**: CBC solver finds solutions in 1-2 seconds vs. 30+ seconds for SA
- **Joint squad-and-XI optimization**: For 1GW, simultaneously optimizes who to buy and who to start
- **Proper budget handling**: Uses selling price for current players, market price for new
- **Safety check**: If LP solution is worse than keeping current squad, reverts to "no transfer"
- **Forced transfer handling**: Correctly handles unavailable players (injured/suspended)
- **Free Hit support**: Recognizes Free Hit chip and optimizes for 1GW only

3.5 Weaknesses of Transfer Optimization

- **Myopic objective**: LP maximizes xP for current horizon but doesn't consider:
  - Future fixture swings (selling Salah before DGW)
  - Price rises/falls and team value growth
  - Upcoming BGW/DGW chip strategies
- **Linear objectives only**: Cannot directly optimize non-linear objectives like Sharpe ratio or variance minimization.
- **No exploration**: LP always returns the same optimal point. For wildcards, users might want multiple near-optimal alternatives.
- **Price volatility ignored**: A player about to rise £0.2m might be more valuable than xP alone suggests.
- **Rolling transfer banking not modeled**: Cannot model "bank this week to have 2 FT next week".

Formation	GKP	DEF	MID	FWD
3-4-3	1	3	4	3
3-5-2	1	3	5	2
4-3-3	1	4	3	3
4-4-2	1	4	4	2
4-5-1	1	4	5	1
5-3-2	1	5	3	2
5-4-1	1	5	4	1

Table 3: Valid FPL formations

4 Starting XI Selection

4.1 Formation Enumeration

The system enumerates all valid FPL formations and selects the one maximizing starting XI xP:

Algorithm:

1. Sort players by xP within each position
2. For each valid formation (d, m, w):
   (a) Select top 1 GKP, top d DEF, top m MID, top w FWD
   (b) Calculate total xP for this formation
3. Return formation with maximum xP

4.2 Availability Filtering

Players with status $\in \{i, s, u\}$ (injured, suspended, unavailable) are excluded from starting XI consideration *before* formation enumeration.

4.3 Strengths of Starting XI Selection

- **Exhaustive search**: All 7 valid formations evaluated, guaranteeing optimal selection
- **xP-adaptive**: Naturally adapts formation based on where xP is concentrated
- **Availability-aware**: Automatically excludes unavailable players
- **Multi-horizon support**: Can optimize for 1GW, 3GW, or 5GW xP

4.4 Weaknesses of Starting XI Selection

- **No auto-sub modeling**: FPL auto-substitutes bench players if starters don't play. Optimal bench order should consider auto-sub probabilities:
$$\mathbb{E}[\text{Points}] = \sum_{i \in \text{XI}} \text{xP}_i \cdot P(\text{plays}_i) + \sum_{j \in \text{bench}} \text{xP}_j \cdot P(\text{autosub}_j)$$
- **Binary availability**: Players are either available or excluded. Doesn't model partial availability (e.g., 75% chance of playing).
- **Formation fixation**: Once selected, formation is fixed. No consideration of differential formations.
- **No doubtful player strategy**: A doubtful premium might be better benched for a certain budget player to avoid 1-point cameos.

5 Captain Selection

5.1 Upside-Seeking Captain Score

The captain selection uses a **ceiling-seeking** approach rather than expected value:

$$\text{Captain Score}_i = 2 \cdot \text{xP}_i^{(90)} \cdot (1 + \alpha_{\text{own}} + \alpha_{\text{match}}) \tag{15}$$

where the 90th percentile upside is:

$$\text{xP}_i^{(90)} = \text{xP}_i + 1.28 \cdot \sigma_i \tag{16}$$

5.2 Template Protection

For high-ownership players ($> 50\%$ ownership), a template protection bonus applies:

$$\alpha_{\text{own}} = \min\left(\frac{\text{ownership} - 50}{20}, 1\right) \cdot \text{fixture\_quality} \cdot 0.40 \tag{17}$$

This ensures that highly-owned players in favorable fixtures receive up to 40% scoring boost, preventing rank-damaging differential picks against template.

5.3 Situation-Aware Strategy

The intelligent captain recommendation adapts strategy based on manager context:

Rank Category	**Season Phase**	**Recommended Strategy**
Elite ($< 100k$)	Late	Template Lock
Elite ($< 100k$)	Early/Mid	Protect Rank
Comfortable ($< 500k$)	Any	Balanced
Chasing ($< 2M$)	Declining momentum	Chase Rank
Trailing ($> 2M$)	Any	Maximum Upside
Any	Triple Captain active	Maximum Upside
Any	Free Hit active	Chase Rank

Table 4: Captain strategy auto-detection logic

5.4 Haul Probability Matrix

The system calculates outcome probabilities assuming normal distribution:

$$P(\text{blank}) = \Phi\left(\frac{\tau_{\text{blank}} - \text{xP}}{\sigma}\right) \tag{18}$$

$$P(\text{return}) = \Phi\left(\frac{\tau_{\text{return}} - \text{xP}}{\sigma}\right) - P(\text{blank}) \tag{19}$$

$$P(\text{haul}) = 1 - \Phi\left(\frac{\tau_{\text{return}} - \text{xP}}{\sigma}\right) \tag{20}$$

where $\tau_{\text{blank}} = 2$ and $\tau_{\text{return}} = 8$ are configurable thresholds.

5.5 Strengths of Captain Selection

- **Ceiling-seeking philosophy**: Using 90th percentile prioritizes haul potential—correct for doubling points
- **Template protection**: Avoids rank-destroying differential picks when template captain is clearly optimal
- **Situation awareness**: Auto-detects appropriate strategy based on rank and season phase
- **Betting odds integration**: Uses team win probability and expected goals for matchup quality
- **Haul probability transparency**: Shows blank/return/haul probabilities to inform decision
- **Rank impact estimation**: Models differential potential based on ownership

5.6 Weaknesses of Captain Selection

- **Gaussian assumption**: Real FPL point distributions are heavily skewed (many 2s, occasional 15+). Normal distribution underestimates blank probability.
- **No historical captain accuracy tracking**: System doesn't learn from past captain decisions. A feedback loop could improve calibration.
- **Vice-captain underutilized**: Simply second-highest scorer. Should consider correlation with captain and ownership differential.
- **Simplistic rank impact model**: Uses rough heuristics. Could be improved with historical rank movement data.
- **No mini-league awareness**: Captain strategy should differ in head-to-head mini-leagues (target opponent's captain).
- **Double gameweek captain**: No special handling for players with two fixtures—should boost their ceiling.

6 End-to-End System Analysis

6.1 Data Pipeline

The data flow through the system:

1. **Data Loading**: 12 data sources including historical performance, fixtures, betting features
2. **Feature Engineering**: Raw data $\rightarrow$ 122 leak-free features
3. **Model Training**: Walk-forward CV with temporal splits
4. **Inference**: Load trained model, generate xP predictions with uncertainty
5. **Optimization**: LP/SA for transfer decisions
6. **Selection**: Starting XI and bench order
7. **Captain**: Situation-aware captain pick

6.2 Cross-Validation Strategy

Walk-forward validation ensures temporal integrity. For training on GW 1–n, test on GW $n+1$. The first trainable GW is 6 due to 5GW rolling feature requirements.

6.3 Model Comparison Framework

The system supports systematic model comparison:

$$\text{Composite Score} = 0.4 \cdot \text{MAE} + 0.3 \cdot \text{Spearman} + 0.2 \cdot \text{RMSE} + 0.1 \cdot \text{Captain Accuracy} \quad (21)$$

7 Identified Gaps and Recommendations

7.1 Strategic Gaps

1. **BGW/DGW Planning**: System lacks proactive handling of blank and double gameweeks. *Recommendation*: Integrate FPL calendar with lookahead optimization.
2. **Chip Strategy**: No systematic chip timing optimization. *Recommendation*: Add chip assessment with simulation.
3. **Team Value Management**: Price change prediction not integrated. *Recommendation*: Add expected value growth as secondary objective.
4. **Transfer Banking**: LP cannot model "save transfer" strategy. *Recommendation*: Multi-stage stochastic programming or heuristic rules.

7.2 Model Improvements

1. **Position-Specific Models**: Train separate models for each position to capture scoring differences.
2. **Non-Gaussian Uncertainty**: Replace normal assumption with empirical distribution for more accurate haul/blank probabilities.
3. **Adaptive Rolling Windows**: Use player-specific window sizes based on form stability.
4. **Auto-Sub Modeling**: Incorporate playing time uncertainty into starting XI decisions.

7.3 Optimization Improvements

1. **Multi-Week LP**: Extend LP to 3-5 gameweek horizon with transfer continuity constraints.
2. **Robust Optimization**: Optimize for worst-case or CVaR under prediction uncertainty.
3. **Exploration Mode**: SA generates diverse near-optimal squads for wildcard exploration.

8 Conclusion

The FPL Team Picker implements a sophisticated multi-stage optimization system with notable strengths:

- **Comprehensive ML pipeline**: 122 leak-free features with uncertainty quantification
- **Provably optimal transfers**: LP guarantees optimal solution for defined objective

- **Situation-aware captaincy**: Adapts strategy to rank, season phase, and chip status
- **Clean architecture**: Well-separated domain services with clear responsibilities

Key areas for improvement:

- **Strategic planning**: BGW/DGW awareness, chip optimization, team value growth
- **Distributional modeling**: Non-Gaussian uncertainty for more accurate captain analysis
- **Multi-week horizon**: Extend LP to consider future fixture swings
- **Position-specific models**: Capture scoring mechanism differences across positions

The system provides a strong foundation for competitive FPL management, with the LP optimizer representing a significant improvement over pure heuristic approaches. The main limitation is the myopic nature of single-gameweek optimization—extending to multi-week planning would provide substantial strategic benefit.

A Mathematical Notation Summary

Symbol	Description
y_i	Actual points for player i
$\hat{y}_i$	Predicted expected points (xP)
σ_i	Prediction uncertainty (std dev)
x_i	Binary decision variable (1 if in squad)
s_i	Binary starter variable (1 if in starting XI)
p_i	Player price
B	Total budget
F	Free transfers
c	Transfer cost (4 points)
$\mathcal{P}_j$	Set of players in position j
$\mathcal{T}_k$	Set of players from team k
$\mathcal{S}$	Current squad player set

B Configuration Parameters

Key configurable parameters in the system:

```
# Feature Engineering
rolling_window: 5  # GW lookback for form features

# LP Optimization
transfer_cost: 4  # Points penalty per transfer
max_transfers: 5  # Maximum transfers allowed

# Captain Selection
elite_rank_threshold: 100000
comfortable_rank_threshold: 500000
chasing_rank_threshold: 2000000
```

```
blank_threshold: 2  # Points below this = blank
return_threshold: 8  # Points above this = haul
template_ownership_threshold: 50  # % for template
```