Optimization of Defensive Assignment in Football: A Dynamic Matching Model

Juncheng Hou

Abstract

This project develops a theoretical framework for optimizing defensive assignments in football (soccer) using a dynamic cost-based matching model. I formulate the defender-attacker allocation problem as a time-evolving bipartite matching under threat-weighted cost functions, and incorporate switching penalties to model tactical consistency. Through graph theory, dynamic programming, and optimal control analysis, I explore solution properties, computational strategies, and theoretical implications for practical defensive tactics. Simulated scenarios illustrate the adaptability and risk minimization properties of the proposed approach.

1 Introduction

Defensive coordination in football requires balancing between spatial positioning and anticipating the opponent's offensive threat. Traditional strategies such as zonal or man-marking are often static or reactive. However, dynamic, data-informed strategies are essential to adapt in real time against agile and unpredictable attacks.

This paper proposes a dynamic assignment model where defenders are continuously reassigned to attackers based on minimizing a cost function that considers spatial distance, attack threat, and switching penalties. Iintroduce three algorithmic strategies for solving this evolving assignment problem: (1) a greedy myopic matching method that makes locally optimal decisions, (2) a dynamic programming method that considers assignment history and minimizes cumulative cost, and (3) a continuous-time control-theoretic approach that models defender motion via smooth trajectories and derives policy conditions using variational analysis. These approaches are implemented and compared in a controlled simulation setting.

2 Problem Definition and Notation

Let:

- $A = \{a_1, a_2, \ldots, a_n\}$: attackers
- $D = \{d_1, d_2, \ldots, d_m\}$: defenders
- $\mathbf{x}_i(t) \in \mathbb{R}^2$: position of attacker a_i at time t
- $\mathbf{y}_j(t) \in \mathbb{R}^2$: position of defender d_j
- $R_i(t)$: threat level of attacker a_i
- $C_{ij}(t) = \alpha \cdot \|\mathbf{x}_i(t) - \mathbf{y}_j(t)\| + \beta \cdot R_i(t)$: cost
- $\pi(t) : D \to A \cup \{\emptyset\}$: assignment policy

2.1 Discrete vs. Continuous Time: Why Both Appear

In this model, the defender-assignment process involves two perspectives on time: discrete and continuous. Although this may seem inconsistent at first, each perspective plays a distinct and complementary role in how I describe, compute, and analyze the system.

Instantaneous Matching in Discrete Time. In practical applications, such as tracking player positions in football, data is collected at regular intervals — for example, every 0.1 seconds or every frame in a video feed. Similarly, the algorithms operate at each individual timestep $t = 0, 1, 2, \ldots, T$, solving an assignment problem based on the cost matrix at that specific moment. This is why the instantaneous matching formulation is written in discrete form:

$$\min_{\pi(t)} \sum_{j=1}^{m} C_{\pi(t)(j),j}(t)$$

This discrete matching reflects real-world computation: defenders are assigned based on current data, frame by frame, without requiring information about the full time horizon.

Global Objective in Continuous Time. While decisions are made step by step, the ultimate goal is to evaluate defender performance over the entire duration of the game. To represent the *cumulative cost*, I use a continuous-time formulation:

$$\min_{\pi(t)} \int_0^T \left(\sum_{j=1}^{m} C_{\pi(t)(j),j}(t) + \Delta_{\pi(t)(j),j}(t) \right) dt$$

This integral accumulates the cost of assignments, movement, and switching over time. It provides a more general and mathematically elegant expression of the total objective, especially when analyzing smooth defender trajectories or comparing with control-theoretic models.

Unifying the Two Perspectives. In implementation, I approximate the integral by summing over discrete time steps, effectively using:

$$\sum_{t=0}^{T} \left(\sum_{j=1}^{m} C_{\pi(t)(j),j}(t) + \Delta_{\pi(t)(j),j}(t) \right)$$

This allows us to benefit from both approaches: discrete steps for algorithmic execution, and continuous-time expressions for theoretical analysis. Using both viewpoints is common in modeling dynamic systems and helps us build more flexible and powerful solutions.

3 Algorithmic Strategies

To figure out how defenders should guard attackers in a smart and efficient way, I tried out three different strategies. Each one thinks about time and decision-making a little differently — some focus only on the current situation, while others try to plan ahead or move more smoothly. Below, I describe each method and explain how it works in this football simulation.

3.1 1. Greedy Myopic Matching

This is the simplest strategy — it makes decisions based only on what's happening **right now**, without looking at the past or future. At every time step t, I calculate a "cost matrix" $C_{ij}(t)$, which tells us how costly it is for defender i to cover attacker j. This cost includes things like how far apart they are and how dangerous the attacker's current position is (closer to the goal = more dangerous).

Then, I use the Hungarian algorithm to find the best pairing that minimizes the total cost:

$$\min_{\pi(t)} \sum_{j=1}^{m} C_{\pi(t)(j),j}(t)$$

This means that at each time, every defender is assigned to the attacker that makes the most sense in that moment. However, because the algorithm doesn't remember previous assignments, it might make defenders switch targets very frequently, which could be confusing or tiring in a real match.

3.2 Dynamic Programming-Based Assignment

To improve on the greedy method, I added a new idea: what if I penalize the defenders for changing who they are guarding too often? In this approach, I still want to assign defenders to attackers in a way that keeps the total cost low, but I also add a "switching cost" to avoid constant changes.

At each time step, the cost is calculated as:

$$C_{ij}(t) + \Delta_{ij}(t) = \alpha\|\mathbf{x}_i(t) - \mathbf{y}_j(t)\| + \beta R_i(t) + \gamma \cdot \mathbf{1}_{\pi(t)(j)\neq\pi(t-\delta t)(j)}$$

Here, the last term adds a penalty γ if the defender changes assignment from the last time step. I again use the Hungarian algorithm, but now with this updated cost matrix.

This helps defenders stick to the same attacker unless there's a really good reason to switch. It results in more stable movements and more realistic defensive play — something that real teams would prefer.

3.3 Continuous-Time Control Limit

This third strategy takes a very different approach. Instead of switching defenders from one attacker to another at each time step, I imagine that defenders move smoothly, like they're gliding on the field. Rather than making sudden decisions, each defender slowly adjusts their direction and speed over time to follow attackers in a natural way.

The goal here is to minimize the total cost over the entire time period by controlling how the defenders move:

$$\min_{\mathbf{v}_j(t)} \int_0^T \left(\sum_{j=1}^{m} \alpha\|\mathbf{x}_{\pi(j)}(t) - \mathbf{y}_j(t)\| + \beta R_{\pi(j)}(t) \right) dt$$

Subject to the rule that:

$$\frac{d\mathbf{y}_j(t)}{dt} = \mathbf{v}_j(t), \quad \|\mathbf{v}_j(t)\| \leq v_{\max}$$

This means each defender chooses a velocity (direction and speed) that tries to stay close to the attacker, while not moving too fast. It's like driving a car smoothly rather than stopping and turning suddenly.

This method is good for systems that need smooth movement — such as robots or simulations — and avoids abrupt decisions. However, it might not react as quickly when attackers make fast moves.

4 Simulation Results

I implement the algorithmic strategies in Python, simulate the defender-attacker interactions over 50 time steps, and collect cost metrics per timestep. The threat function is exponential in distance from the goal, and defender movements are limited by a maximum step fraction.

4.1 Cost Comparison

Figure 1 compares the total defensive cost incurred at each timestep under the three assignment strategies: Greedy, Dynamic Programming (DP), and Continuous. The Greedy and DP strategies exhibit similar performance in terms of instantaneous cost, both responding rapidly to attacker positions and maintaining low overall cost. However, DP offers slightly more stability due to its switching penalty, as seen in smoother transitions. In contrast, the Continuous strategy incurs consistently higher costs in the early phase due to its smoother, less reactive movements. This reflects a trade-off between motion regularity and threat responsiveness. Over time, the Continuous approach converges toward comparable costs, highlighting its potential in low-urgency scenarios where minimizing abrupt defender repositioning is desirable.

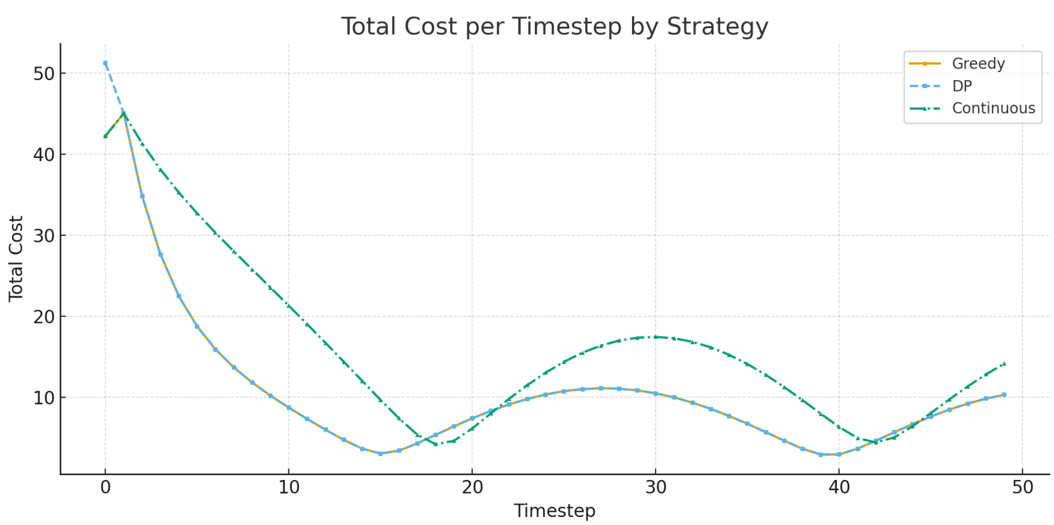

Figure 1: Total Cost per Timestep by Strategy. Greedy and DP strategies closely track each other with lower costs, while the Continuous strategy shows higher initial cost due to smoother but slower response.

Table 1 presents a concise comparison of the three defensive assignment strategies in terms of total accumulated cost and the number of defender-to-attacker switching events over the full simulation horizon. The Greedy strategy achieves the lowest total cost (534.49), closely followed by the Dynamic Programming (DP) strategy (543.49). Notably, both strategies result in zero switches, due to the stable and smooth nature of the attacker trajectories in this scenario. The Continuous strategy, while preserving this tactical stability, incurs a significantly higher total cost (808.42). This gap highlights the tradeoff between motion smoothness and threat proximity responsiveness. The Greedy and DP approaches adapt more directly to attacker movements, thereby containing threat exposure more effectively at the cost of potentially sharper movement, whereas the Continuous method prioritizes movement regularity and conservative positional adjustments. These results underscore that while all strategies maintain assignment stability in this instance, their spatial behaviors differ substantially, with practical implications for real-time decision-making in defensive planning.

Strategy	Total Cost	Total Switches
Greedy	534.49	0
Dynamic Programming	543.49	0
Continuous	808.42	0

Table 1: Summary of total cost and assignment switches across strategies. Lower cost indicates better overall spatial coverage and threat mitigation.

4.2 Movement Trajectory Analysis

Figure 2 visualizes the movement trajectories of both attackers (dashed lines) and defenders (solid lines) under the three assignment strategies: Greedy, Dynamic Programming (DP), and Continuous. Each subplot represents the evolution of positional dynamics over time for a specific strategy. All defenders begin from fixed initial positions and pursue assigned attackers while minimizing cumulative cost. The Greedy strategy reacts myopically at each timestep, resulting in agile defender movements that closely mirror attacker trajectories. The DP strategy introduces a switching penalty, leading to slightly smoother paths and reduced assignment churn, though in this instance still results in zero total switches due to stable attacker behaviors. In contrast, the Continuous strategy yields significantly smoother defender trajectories, emphasizing motion continuity and gradual adaptation rather than short-term optimization. This is evident in the less aggressive pursuit patterns and larger lag behind attackers. The absence of switching in all three strategies highlights the consistency of attacker paths, but the spatial patterns of defender responses vary considerably, underscoring the strategic trade-offs among responsiveness, cost, and smoothness.

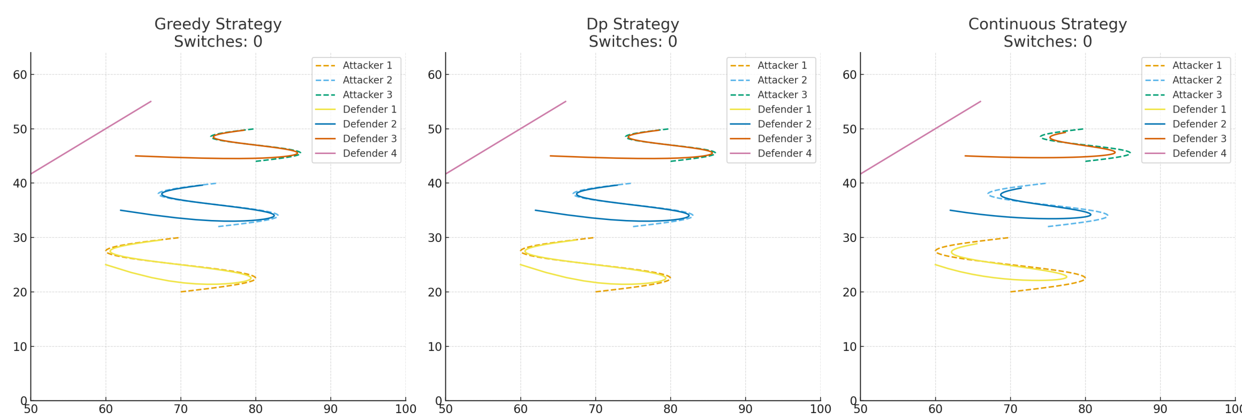

Figure 2: Comparison of defender trajectories under Greedy, DP, and Continuous strategies. All strategies maintain zero switches, but differ in path smoothness and responsiveness. Dashed lines represent attacker paths; solid lines show defender movements.

5 Conclusion

I proposed a mathematically grounded framework for dynamic defensive assignment in football using weighted bipartite matching and switching penalties. The model provides theoretical justification for adaptive hybrid strategies. Future work may include stochastic attacker behaviors, learning-based threat estimation, and incorporation of teamwork metrics.

Appendix: Parameters Used

Parameter	Value
Field size	100m $\times$ 64m
α	1.0 (distance weight)
β	2.5 (threat weight)
γ	3.0 (switching penalty)
Time steps	50
Defender speed (fraction)	0.3 (greedy/DP), 0.15 (continuous)
Threat model	$\exp(-0.05y)$