FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Técnicas de simulação computacional para o cálculo de probabilidades de sorteios desportivos

Rodrigo Afonso Pita Silva

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Mestrado em Engenharia Informática e Computação

Supervisor: Rosaldo Rossetti

August 6, 2024

Técnicas de simulação computacional para o cálculo de probabilidades de sorteios desportivos

Rodrigo Afonso Pita Silva

Mestrado em Engenharia Informática e Computação

August 6, 2024

Resumo

Os modelos de sorteio dos desportos modernos enfrentam frequentemente várias restrições, o que torna difícil ou impraticável calcular probabilidades realistas de resultados de sorteio utilizando métodos matemáticos tradicionais. Estas restrições podem incluir as posições das equipas no ranking, a sua divisão em "potes", a proibição de confrontos entre equipas do mesmo país, evitar emparelhamentos entre países/clubes com tensões ou conflitos políticos, evitar confrontos repetidos com adversários recentes, restrições de calendário que impedem as equipas da mesma cidade de jogar no mesmo dia e outras numerosas restrições ocasionais.

A presente dissertação propõe um novo sistema de simulação de sorteios, passo a passo, para determinar as probabilidades reais de confronto em sorteios desportivos, centrando-se especificamente em sorteios de torneios a eliminar. O simulador, refletindo com precisão as configurações do sorteio, será capaz de incorporar várias fases do sorteio e aplicar restrições de forma eficaz. Gerará valores de probabilidade realistas e analisará o impacto de cada restrição nas probabilidades, podendo fornecer informações valiosas aos organizadores da competição.

Este sistema de simulação implica vários desafios. Implicará a criação de sorteios customizáveis através de um conjunto de parâmetros, restrições e configurações de equipas. Um processo de simulação passo a passo otimizado para simulações em larga escala. Técnicas de estimativa de probabilidades serão incorporadas nos algoritmos de simulação desenvolvidos para verificar a legitimidade de cada passo do sorteio e realizar estimativas de emparelhamentos de equipas.

Prevê-se que esta dissertação produza um sistema de simulação robusto e eficiente que possa replicar com confiança procecssos de sorteio bem definidos. Fornecerá uma análise exaustiva do impacto de várias restrições nas probabilidades efectivas dos resultados dos sorteios. Além disso, contribuirá significativamente para o domínio da análise das probabilidades de sorteios no desporto.

Palavras-chave: simulação discreta de eventos, sorteios desportivos, probabilidades, simulação, Monte Carlo, restrição

Abstract

Modern sports' draw models often face various restrictions, making it challenging or impractical to calculate realistic probabilities of draw outcomes using traditional mathematical methods. These restrictions can include the ranking positions of the teams, their division into "pots", prohibitions on matchups between teams from the same country, avoidance of pairings between countries/clubs with political tensions or conflicts, avoiding repeat matchups with recent opponents, scheduling constraints preventing teams from the same city from playing on the same day, and other numerous occasional restrictions.

This dissertation proposes a novel step-by-step draw simulation system to determine real matchup probabilities in sports draws, focusing specifically on knockout tournament draws. The simulator, accurately reflecting the draw configurations, will be capable of incorporating various draw stages and applying restrictions effectively. It will generate realistic probability values and analyze the impact of each restriction on the probabilities, possibly providing valuable insights for competition organizers.

This simulation system will involve several challenges. It will involve the creation of customized draws through a set of parameters, restrictions, and teams configurations. A step-by-step simulation process optimized for large-scale simulations. Probability estimation techniques will be incorporated in the developed simulation algorithms to verify the legitimacy of each draw step and perform teams matchups estimations.

This dissertation is anticipated to produce a robust and efficient simulation system that can confidently replicate well defined draw processes. It will provide a comprehensive analysis of the impact of various restrictions on the actual probabilities of draw outcomes. Moreover, it will contribute significantly to the field of sports draw probability analysis.

Keywords: discrete-event simulation, sports' draws, probabilities, simulation, Monte Carlo, restriction

Acknowledgements

To Professor Rosaldo Rossetti, for the orientation. His guidance and support were crucial for the fulfilment of this project.

To my family, my father, my mother, my sister and my grandparents, who have always been there for me through this journey, providing the support I needed. They are truly my fuel tank.

To my friends, who made the process much smoother. Thank you for all the laughs and moments.

To my girlfriend, for her daily unconditional support, that made everything easier.

Rodrigo Silva

*"Excellence is not a destination.
It is a continuous journey that never ends."*

Brian Tracy

Contents

List of Figures

List of Tables

Abbreviations, Acronyms and Symbols

ACM	Associazione Calcio Milan
BAY	Fußball-Club Bayern München
BEN	Sport Lisboa e Benfica
BRU	Club Brugge Koninklijke Voetbalvereniging
CHE	Chelsea Football Club
CIT	Manchester City Football Club
CP	Constraint programming
CSP	Constraint Satisfaction Problem
CSPs	Constraint Satisfaction Problems
DOR	Borussia Dortmund
FEUP	Faculdade de Engenharia da Universidade do Porto
FIFA	Fédération Internationale de Football Association
FRK	Eintracht Frankfurt
INT	Football Club Internazionale Milano
LIV	Liverpool Football Club
LTP	Law of Total Probability
MC	Monte Carlo
NAP	Società Sportiva Calcio Napoli
POR	Futebol Clube do Porto
PSG	Paris Saint-Germain Football Club
RA	Restriction Algorithm
RBL	RasenBallsport Leipzig
RMA	Real Madrid Club de Fútbol
TOT	Tottenham Hotspur Football Club
UCL	UEFA Champions League
UEFA	Union of European Football Associations
ZZ	zerozero

Chapter 1

Introduction

1.1 Context

Modern sports draw models have sometimes complex formats that directly influence their results. These models can include rules and restrictions that generate unbalanced probabilities of encounters between participants of these draws. As such, it is in the interest of participants and competition organizers to have concrete or estimated knowledge of these probability distributions, which is sometimes impractical using traditional mathematical methods due to the complexity of the rules and restrictions. These restrictions can include the ranking positions of the teams, their division into "pots", prohibitions on match-ups between teams from the same country, avoidance of pairings between countries/clubs with political tensions or conflicts, avoiding repeat match-ups with recent opponents, scheduling constraints preventing teams from the same city from playing on the same day, and other numerous occasional restrictions. This dissertation proposes a novel step-by-step draw simulation system to determine realistic match-up probabilities in sports draws, focusing specifically on knockout tournament draws.

Figure 1.1: UCL 2023 draw ceremony. Source: (1)

1.2 Problem and Motivation

The primary problem addressed in this dissertation is the aforementioned difficulty of correctly estimating the probabilities of matches in sports draws. Traditional methods mathematical methods struggle to incorporate these complex rules and often fail to provide realistic probabilities. The motivation for this dissertation is to explore new approaches, using computer simulation techniques to realize a realistic estimation of these probabilities. With this, we want to understand until what extent, techniques such as Monte Carlo (MC), Constraint Programming (CP) and Markov-Chain Analysis, are able to respond to the needs and be a viable alternative to estimate realistic probabilities in knockout tournament draws.

1.3 Aim and Goals

The main objective is to develop a simulation tool capable of creating and simulating knockout tournament draws and estimating realistic draw match-up probabilities at each step, including and respecting restrictions. This main aim, can be divided into specific goals:

- Design and implement a modular simulation framework.
- Implement a draw creation process capable of generating draws with different specifications, within the knockout tournaments scope.
- Ensure all rules and restrictions are satisfied during the simulation process.
- Explore different technologies in the area of event sequence simulation and use them to estimate the match-up probabilities in each step of the draw.
- Evaluate the performance of the different probability estimation methods.

1.4 Research Questions and Hypotheses

To address the problem presented and the objectives described, the following research questions and hypotheses were developed :

- **RQ1:** What are the limitations of mathematical methods in obtaining real match-up probabilities for sports draws with seeding rules and restrictions?
- **RQ2:** Can parametrization processes in simulation-based methods be used to generate a wide variety of different draws?
- **RQ3:** Can simulation-based and heuristic methods, such as Constraint Programming techniques, Monte Carlo Simulation, and Markov Chain be utilized to estimate realistic probabilities for sports draws with seeding rules and restrictions?

The hypotheses are:

- **Hypothesis 1:** Simulation-based methods are alternative methods that can reliably help infer match-up probabilities for sports draws.
- **Hypothesis 2:** It is possible to design a simulator that can accurately reproduce the draw process for various types of knockout tournaments, adhering to specific rules and restrictions.

1.5 Expected Contributions

This dissertation is expected to contribute significantly to the field informatics and computing engineering applied to the context of sports draws, through the exploration of probabilistic simulation techniques. The contributions can be categorized into three domains: scientific, applicational and technological.

One of the expected scientific contributions passes for developing a systematic literature review on the existing research on probabilistic methods and their application in sports draw simulation and identify the gaps in the current literature around this topic. The other scientific contribution is the exploration of probabilistic methods and simulation techniques and provide a comprehensive analysis on their performance and behavior under the context described.

The expected applicational contributions consist on the development of a system that can be used by competition organizers to automate draw processes and at the same time avoid manual errors, ensuring the integrity of the competitions. The system can also be used by fans as an entertainment tool to follow real draws and promote public interest in competitions.

Finally, the expected technological contribution passes by the development of a customizable draw simulation system, capable of creating and replicating different kinds of draws for knockout tournaments, while providing comprehensive probabilistic data on the simulation outcome through the incorporation of techniques as Monte Carlo simulation, Constraint Programming, and Markov Chain analysis.

These are important contributions to the field of informatics engineering by providing innovative solutions and methodologies for simulating and analyzing sports draws.

1.6 Structure

Besides the Introduction, this dissertation is composed of the following chapters: Chapter 2 of the State of the Art provides the theoretical contextualization necessary for a better understanding of the topics covered, as well as a literature review on the related work in the field. Chapter 3 of the Methodological Approach details the research methodology, including the problem formalization, methodological pipeline, and the simulator architecture. This chapter also details the different approaches used for estimating the draws match-up probabilities. Chapter 4 of the Empirical Studies, presents a comprehensive analysis of the results obtained with the different probability estimation approaches and discusses the main findings in the context of this dissertation research questions and hypothesis. Chapter 5 of the Conclusion, summarizes the main findings, outlines the key contribution and suggests directions for future research.

Chapter 2

State of the Art

Sports draws play a crucial role in determining the match-ups and group formations in various sports competitions. Ideally, the draws aim to ensure fairness and competitive balance among all participants, while also adding an element of unpredictability and excitement to the competition.

These draws appear in various types of sports competitions and tournaments, each with its own structure and dynamics. In Table 2.1 are some of the most common types of competition formats:

Table 2.1: Most common sport's competition formats (2)

Competition Type	**Description**	**Examples**
Single-Elimination or Knockout Competition	Opponents are paired against each other and the loser is eliminated from the competition. This process repeats itself until only one participant remains, which is declared the winner.	Tennis Grand Slams
Round-Robin	All participants in a group or a league play against each other at least once, obtaining a final classification based on points and tiebreak criteria.	English Premier League, Portuguese "Primeira Liga", Brazilian "Serie A", etc
Combination Format	This format results from the combination of single-elimination and round-robin formats. Some competitions include a group stage with a round-robin format that is followed by a knockout stage with a single-elimination format based on the teams' previous classification. Or vice-versa, a knockout stage is used to classify teams for a following round-robin stage	UEFA Champions League, UEFA Europa League, FIFA World Cup, etc

Sports draws serve as the initial step in determining the match-ups or group formations in various competition formats, existing different approaches tailored to the specific format and desired outcomes.

For common league competitions, random draws are commonly employed to determine the order of matches. This type of draw consists of the process of randomly assigning participants to matches without any specific criteria, which ensures that no team has an unfair advantage.

However, for the majority of other competitions, the draw processes increase in complexity, for being guided by competition rules. These competition rules can involve seeding or other specific rules of the tournaments, often used to promote fairness and competitive balance. Seeding is the process of putting competitors in a certain position ensuring that the best teams/players do not meet each other too early in the competition and have a high impact on the competition outcome (3). They are normally based on rankings or the competitors' performance so far in the competition (4).

When forming a tournament's group stage, a stratified draw process is often used. This process involves separating teams into pots based on their rankings, previous performance, or other relevant factors. Teams are then randomly drawn from each pot, forming groups with teams from different pots. This process is made to ensure that all the groups are even and to promote clashes between strong competitors further in the competition.

For single-elimination tournaments, both seeding and stratified draw processes are commonly applied. For example, in most tennis Grand Slams, the highest-ranked players only enter in latest stages of the competition graph (seeding). On the other hand, the UEFA Champions League (UCL) knockout phase games setup is obtained through a stratified draw process, where the teams are divided into 2 pots based on their previous group classification and teams are drawn randomly, forming matches with one team from each pot.

Some other rules, specific to each competition, can be applied to the previously mentioned draw processes. These types of rules are usually introduced to avoid unwanted matches. Picking up on the UCL knockout phase setup example, the draw process has some additional constraints such as that teams from the same country can´t be drawn against each other (5). This leads to the need to make small adjustments during the draw process to ensure all teams end up with a valid opponent.

We can now understand that these draws have an important role in shaping the structure and dynamics of sports competitions, having a direct impact on their outcome. Random draws, stratified draw processes, and seeding are all used strategically to ensure fair match-ups, competitive balance, and overall excitement for the events.

2.1 Background

2.1.1 Event Simulation

Simulation is generally defined as the process of replication of the real world based on a set of assumptions and conceived models of reality (6). Moreover is a computing technique that involves modeling and analyzing complex systems. Event simulation models represent a sequence of events, each representing a change in the system's state. These models can fit into different categories, such as discrete or continuous and even deterministic or stochastic.

Discrete event simulation: In discrete event simulation, events occur at specific points in time, and the system state changes only at those points. This is often used to model systems where the state of the system is represented by a set of discrete variables, such as the number of customers in a queue or the number of jobs in a system (7; 8).

Continuous event simulation: In continuous event simulation, events can occur continuously over time, and the system state changes continuously in response to these events. This is often used to model systems where the state of the system is represented by a set of continuous variables, such as the temperature of a fluid in a pipe or the population of a species (7).

Stochastic event simulation: Stochastic event simulation allows uncertainty to be incorporated into the model as it involves random variables and probabilities to generate the timing and outcome of events. This type of event simulation applies to model systems where the behavior is inherently unpredictable, such as most queuing and inventory systems (9).

Deterministic event simulation: This type of event simulation assumes that the system's behavior is entirely predictable as it has no probabilistic components, which means the simulation will always produce the same results if it runs with the same initial conditions. Deterministic event simulation models systems with well-defined rules, such as manufacturing processes or worst-case analysis systems (9).

2.1.2 Monte Carlo Simulation

The Monte Carlo method is a mathematical technique that is used for modeling and simulating complex systems that involves generating random numbers within a predefined range and can be used to solve mathematical problems and make predictions (10; 11). These random numbers represent the input variables of the system being modeled, and by repeatedly simulating the system with different sets of random inputs, one can effectively explore the entire range of possible outcomes.

Monte Carlo simulation is particularly well-suited for estimating probabilities of events that are difficult or impossible to predict analytically. This is because the repeated generation of random samples allows for direct computation of the frequency of occurrence of each event. For instance, Monte Carlo techniques can be used for environmental simulation to estimate the release strength and reconstruct the dispersion of hazardous substances in marine environments (12), or in sports tournament modeling, to predict outcomes such as the teams' winning probabilities (11; 13; 14; 15; 16).

Monte Carlo simulation offers several advantages over traditional analytical methods, particularly when dealing with complex systems with multiple interacting variables and uncertainty (17; 18). It has high adaptability as it can be applied to a wide range of problems, from simple deterministic systems to highly nonlinear and stochastic models. Monte Carlo methods are also not sensitive to small changes in model parameters, making them less prone to errors compared to analytical approaches. It also takes care of input covariances or dependencies automatically (19; 20).

Although Monte Carlo simulation offers several advantages for probability estimation it can also raise some concerns. As the accuracy of the estimated probabilities depends on the number of random samples generated this process can be computationally intensive, especially for large systems with numerous random variables (21; 18).

Another important topic to address about this concern is the validation of the results of these type of simulations to ensure their accuracy and reliability. Several validation methods and techniques can be used to assess the validity of Monte Carlo simulations.

Convergence analysis is a critical method for evaluating the accuracy of Monte Carlo simulations. It involves assessing how the estimated probabilities converge to a stable value as the number of random samples increases. If the estimated probabilities converge as the number of samples increases, it indicates that the simulation is accurate and the results are reliable.

There are several different convergence analysis methods, including graphical convergence analysis, statistical convergence analysis, and probabilistic convergence analysis. Graphical convergence analysis involves plotting the estimated probabilities against the number of samples to visualize the convergence pattern (22). Statistical convergence analysis utilizes statistical tests to assess the significance of the convergence. Probabilistic convergence analysis involves defining a convergence criterion and determining whether the estimated probabilities meet this criterion (23).

Sensitivity analysis evaluates how the results of a Monte Carlo simulation vary in response to changes in the input parameters. By performing sensitivity analysis, researchers can identify the key input parameters that have the most significant impact on the simulation outcomes (16). This information can be used to refine the model and improve its accuracy.

Sensitivity analysis methods can be classified into two main categories: local sensitivity analysis and global sensitivity analysis. Local sensitivity analysis focuses on the impact of small changes in individual input parameters, while global sensitivity analysis examines the overall sensitivity of the model to changes in all input parameters simultaneously.

Another method for Monte Carlo validation could be the comparison with historical data. When available, comparing the results of a Monte Carlo simulation with historical data can provide valuable insights into the validity of the simulation. This comparison can help to identify potential biases or inaccuracies in the model and suggest areas for improvement.

Historical data can be used to calibrate the model parameters, ensuring that the simulation accurately reflects the real-world behavior of the system under study. Additionally, comparing the simulated outcomes with historical data can provide a measure of the model's predictive accuracy.

In addition to these general validation methods, there are also specific techniques that can be applied to validate Monte Carlo simulations for different types of applications. For example, in environmental modeling, validation methods may involve comparing the simulated dispersion of pollutants with measured data from field experiments (24). In sports tournament modeling, validation methods may involve comparing the simulated team rankings with actual tournament results.

The choice of validation methods depends on the specific application and the available data. However, by employing a combination of these methods, researchers can gain a comprehensive understanding of the validity of their Monte Carlo simulations and ensure that they are producing reliable results.

2.1.3 Constraint Programming

Constraint programming (CP) is a paradigm in the computer science field used in different areas such as artificial intelligence and operations research for solving combinatorial problems. It involves defining a set of constraints that describe the properties and relationships between variables, and then finding values for these variables that satisfy all the constraints. CP has proven to be highly effective in a variety of domains, including scheduling, planning, resource allocation, and sports scheduling, among others (25).

At the core of CP there are three main elements: variables, domains and constraints. These elements are the basis to represent constraint satisfaction problems (CSPs). Variables are the elemental structures, the unknowns to be determined. Domains are the values that these variables can have. Each variable has its own domain of possible values. Constraints are the logical relations between the variables' values, that must be satisfied. The goal in CP is to find an assignment of values to the variables that satisfies all the problem constraints.

The definition of a CSP is called modeling and it basically consists in defining variables, their domains, and constraints. For example, in a sports scheduling problem, variables could represent the teams, match times, and venues, while constraints could enforce that no team plays more than once at the same time or that certain teams must not be matched against each other due to scheduling conflicts or political tensions.

Solving CSPs typically involves a combination of search and inference. Search strategies systematically explore possible assignments for the variables. Common search methods include backtracking, where the algorithm incrementally builds candidates to the solution and abandons a candidate as soon as it determines that the candidate cannot possibly be completed to a valid solution. Constraint propagation involves deducing variable domains based on constraints to reduce the search space. Techniques such as arc consistency, path consistency, and k-consistency are used in constraint propagation to iteratively narrow down the domains of the variables. Heuristic-based search strategies, such as the use of minimum remaining values or least constraining value heuristics, guide the search process to explore more promising areas of the search space first. CP libraries implement algorithms to solve CSPs using these methods. Examples of such libraries IBM ILOG CPLEX CP Optimizer (26), Gecode (27), Google OR-Tools (28), and MiniZinc (29).

CP has proven to have a positive impact in different areas and has a vast number of applications. CP optimizes resources and solves allocation problems for scheduling tasks, such as employee shifts, class timetables, sports schedules, etc. In manufacturing and logistics, CP helps in the allocation of resources efficiently, while taking in consideration aspects like production capacities and delivery times. CP is also used in product configuration, where products are customized based on customer requirements and constraints on component compatibility.

CP is a paradigm that provides flexibility in modeling various types of constraints, robustness in finding solutions, and the ability to handle relatively large complex problems. However, the main challenges faced in this area include the complexity of certain problems, which makes them not trivial to model, requiring the need of expert knowledge, and the computational intensity of solving large-scale CSPs (30).

2.1.4 Probability Estimation

Probability estimation is a fundamental concept in statistics and is used in a wide range of applications, including scientific research, engineering, and business (31). It consists of the process of determining the likelihood of an event or outcome. There are two main approaches to probability estimation: frequency-based methods and subjective methods.

Frequency-based methods estimate probabilities based on the observed frequency of an event (32). These methods are often based on the law of large numbers, which states that the relative frequency of an event will converge to its true probability as the number of trials increases (33).

One common frequency-based method is sampling. In sampling, a small subset of data is randomly selected from a larger population, and the frequency of the event of interest is estimated from the sample data (34). For example, a sports analyst might sample a team's past performance against different opponents to estimate the team's win-loss record against those opponents.

Another common frequency-based method is statistical modeling. In statistical modeling, mathematical models are used to relate the probability of an event to other variables (35; 36). For example, a sports analyst might develop a statistical model that predicts the probability of a team winning a game based on the team's current form, the opponent's current form, and other relevant factors (37).

Subjective methods estimate probabilities based on an individual's judgment or belief about the likelihood of an event (38). These methods are often used when there is no or limited historical data available to estimate probabilities.

One common subjective method is expert judgment. In expert judgment, experts in a particular field are asked to provide their subjective probabilities for an event. For example, a sports analyst might ask a panel of experts to provide their probabilities for the outcome of a particular game.

Another common subjective method is Bayesian inference. Bayesian inference is a statistical method that allows for updating probabilities based on new information (39).

In scientific research, various specific techniques are employed for probability estimation. These methods help to refine statistical models and improve the accuracy of predictions (36):

- **Maximum likelihood estimation:** This method estimates the parameters of a statistical model by maximizing the likelihood of the observed data (40).
- **Bootstrapping:** This method involves repeatedly resampling the data and re-estimating the parameters of the statistical model. The resulting distribution of estimates can be used to assess the uncertainty in the parameter estimates (39).
- **Cross-validation:** This method involves dividing the data into a training set and a test set. The parameters of the statistical model are estimated on the training set, and the model is

then tested on the test set. The performance of the model on the test set can be used to assess its generalizability to new data (41).

- **Sensitivity analysis:** This method involves varying the parameters of the statistical model and examining how the model's predictions change. This can help to identify the key parameters that affect the model's predictions (42).
- **Model validation:** This is used to assess the performance of the model by comparing the predictions of the statistical model to the observed data (34; 37).

2.1.5 Similarity of Probability Matrices

In the context of comparing probability matrices, two prominent methods are the Euclidean distance (Frobenius norm) and cosine similarity. These methods provide different perspectives on the similarity and dissimilarity between matrices, each with its own advantages and applications.

2.1.5.1 Euclidean Distance

The Euclidean distance between two matrices is a measure of their dissimilarity. It is often referred to as the Frobenius norm when applied to matrices. For an $M \times N$ matrix A, the Frobenius norm is defined as:

$$\|A\|_F = \sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}|a_{ij}|^2} \tag{2.1}$$

Geometrically, the set of vectors Ax (where x has a length of 1) forms an ellipsoid in R^N, and the lengths of its semi-axes correspond to the singular values of A. The Frobenius norm measures how long the vector Ax is. This norm can be used to compute the distance between two matrices A and B as follows:

$$Euclidean\_distance(A,B) = \|A-B\|_F = \sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}|a_{ij}-b_{ij}|^2} \tag{2.2}$$

This distance is useful in various applications, including data analysis and machine learning, as it provides a straightforward way to quantify the dissimilarity between matrices ((43), (44)).

2.1.5.2 Cosine Similarity

Cosine similarity measures the cosine of the angle between two vectors, which can be extended to matrices by vectorizing them. The cosine similarity between two matrices A and B is defined as:

$$cosine\_similarity(A,B) = \frac{\langle vec(A), vec(B)\rangle}{\|vec(A)\|\|vec(B)\|} \tag{2.3}$$

Where $vec(A)$ and $vec(B)$ denote the vectorized forms of matrices A and B, respectively. This similarity ranges from -1 to 1, where 1 indicates that the matrices are identical in direction, -1 indicates that they are opposite, and 0 indicates orthogonality. Cosine similarity is particularly useful in applications where the magnitude of the matrices is less important than their direction, such as in text analysis and clustering (45).

2.1.5.3 Other Relevant Methods

Apart from Euclidean distance and cosine similarity, here are other methods for comparing probability matrices, based on (46) and (47):

- **Manhattan Distance (L1 Norm)**: This distance is the sum of the absolute differences between corresponding elements of the matrices. It is defined as:

$$Manhattan\_distance(A,B) = \sum_{i=1}^{M} \sum_{j=1}^{N} |a_{ij} - b_{ij}| \tag{2.4}$$

- **Jaccard Similarity**: This similarity measures the size of the intersection divided by the size of the union of two sets, and can be adapted for matrices. For binary matrices, it is defined as:

$$Jaccard\_similarity(A,B) = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} \min(a_{ij}, b_{ij})}{\sum_{i=1}^{M} \sum_{j=1}^{N} \max(a_{ij}, b_{ij})} \tag{2.5}$$

- **Pearson Correlation Coefficient**: This coefficient measures the linear correlation between two sets of data. For matrices, it can be computed as the correlation between their vectorized forms.

- **Kullback-Leibler Divergence**: This measure is used to compare probability distributions. For two matrices representing probability distributions P and Q, the Kullback-Leibler divergence is given by:

$$KL(P\|Q) = \sum_{i=1}^{M} \sum_{j=1}^{N} p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right) \tag{2.6}$$

These methods can all be used for comparing probability matrices, each offering unique insights depending on the nature of the data and the specific application requirements.

2.1.6 Bayesian Networks

Bayesian networks are a type of probabilistic graphical model that is used to represent and reason about uncertainty in a variety of applications, such as medical diagnosis, fraud detection, robotics, risk assessment, etc (48). They are particularly indicated for modeling problems with complex causal relationships between variables. In a Bayesian network, each variable is represented by a node, and the relationships between variables are represented by edges. The edges are associated with conditional probabilities, which represent the probability of one variable given the values of other variables (49; 50; 51; 52; 53; 54). Bayesian networks can be used to estimate probabilities in two ways:

One way is Forward propagation. This method involves starting with the observed values of some variables and then propagating the probabilities through the network to estimate the probabilities of other variables. For example, a Bayesian network could be used to diagnose a patient's illness by starting with the observed symptoms and then propagating the probabilities through the network to estimate the probability of different diagnoses (55).

The other way is called Backward propagation. This method involves starting with the query variable and then propagating the probabilities back through the network to estimate the probabilities of the parent variables. For example, a Bayesian network could be used to predict the weather by starting with the current temperature and pressure and then propagating the probabilities back through the network to estimate the probability of different weather conditions (56).

In scientific research, various methods are employed for Bayesian network inference:

- **Exact Inference:** Employs mathematical algorithms to calculate exact probabilities for all network variables, often computationally expensive and practical for small networks (57).
- **Approximate Inference:** Utilizes heuristic algorithms to approximate probabilities efficiently, making it more practical for larger networks than exact inference (52).
- **Monte Carlo Sampling:** Generates numerous samples from the posterior distribution to approximate the probability of variable configurations, commonly used in large networks for approximate inference (51).
- **Markov Chain Monte Carlo (MCMC) Methods:** Specifically designed for sampling from complex probability distributions, MCMC methods are employed for approximate inference in Bayesian networks (17; 58).

2.2 Related Work

Several studies have investigated the use of computing simulation techniques to predict competition's outcomes and analyze its formats.

(3) compared and evaluated four different tournament designs for the World Men's Handball Championships using Monte Carlo simulations with a probabilistic model. The study found that the round-robin format is the most effective at selecting the strongest teams as winners, but it is not practical due to the large number of games required. G66 was found to be the best design in terms of selecting the strongest teams as winners among the designs considered by the study.

(59) investigated the use of official ATP computer tennis rankings for simulating major tennis tournaments. The focus was on predicting a player's chance of winning based on their ATP

rankings. The study found that a logistic regression model fitted to ATP rankings successfully estimates the probability of winning a set based on the rating difference. This provides a predictive element to the official ATP rankings, allowing for simulations of tournament outcomes.

(13) presented and evaluated a simulation/probability model designed for predicting outcomes in major football tournaments, with a focus on the World Cup and the European Cup championships. The study developed a simulation/probability model that incorporates scoring intensities, estimated as a weighted average of goals scored. The model was used to compute the probability of each team winning the tournament, taking into account the Poisson distribution for the number of goals scored.

(14) developed a simulation model for the Brazilian National Football Championship, focusing on estimating scores required for specific positions in the final classification ranking. The simulation model utilized Monte Carlo sampling to randomly generate the number of points obtained by each team in each match. The model considered the rules of the championship, the number of competitors, and the probability of a match ending in a draw. It provided estimators for the necessary scores to achieve specific positions in the final classification ranking.

(60) evaluated the impact of different point systems and formats on the quality of the FIFA World Cup group stage matches to predict match-fixing. The study explored a new format to understand the impact in competitiveness. A classification method was developed to categorize games in relation to their competitiveness. Monte Carlo simulations were conducted for different tournament formats. Analysis of historical data was used to validate the simulation results.

(15) evaluated the probability of performance of a team in the Spanish 1st division (maintenance, playing European tournaments or winning the league). The objective was to obtain the amount of points a team should aim to stay in the first division. The study employed Monte Carlo simulation based on historical data using UEFA draw procedure. It used normal approximation and Monte Carlo simulation to estimate the likelihood of a team achieving different performance outcomes.

(61) investigated the impact of match scheduling on less competitive games in UCL. The study provided insights to tournament organizers to optimize scheduling for more competitive games. The study developed a simulation model (four-parameter Poisson) based on pot allocation to simulate group matches. The model was used to analyze different schedules for the last 2 match days, based on UEFA regulations and historical data. The study calculated the probabilities of stake-less games for each schedule with 1 million simulation runs. The study recommended that the strongest teams play against mid-tier teams in the last round to avoid stake-less games. It also found that certain schedules significantly reduce the probability of stake-less games.

So far we have seen that several studies used Monte Carlo simulations to replicate sports competitions with different types of formats and with that, obtain several statistical indicators about the implications that the various factors that regulate these competitions influence their outcome and the way they develop. On the other hand, other studies were found, that focused more on the sport's draw component and evaluating how the draw regulations and seeding policies impact the competition's dynamics. These studies suggest that draw restrictions and policies can have direct implications on the performance of certain teams or groups of teams in many different ways.

(62) developed a mathematical model to evaluate the impact of draw restrictions on same-nation match-ups in the UEFA Europa League. The study explored how these restrictions influence the probability of a team from a specific association winning the tournament and how different draw methods can promote diversity and engagement in the competition. Monte Carlo simulations were conducted based on historical data using UEFA's draw procedure to simulate the round of 32 and round of 16. The results revealed that draw constraints in the round of 32 increased the probability of same-nation match-ups by 2-3% and in the round of 16 by 6-12%. Interestingly, the association constraint in the round of 32 slightly favored Spanish clubs while reducing the winning probabilities for teams outside England and Spain by up to 4%. Expanding the association constraint to the round of 16 was found to significantly reduce the likelihood of same-nation match-ups, with a cumulative probability reduction reaching up to 60% and an average gain of about 20%. The author concluded by recommending further investigation into the rationale behind UEFA's draw rules, highlighting the potential for uneven distribution and unfair outcomes arising from these restrictions.

(63) also conducted a study to quantify incentive incompatibility in the context of sports. They evaluated the probability of tanking in the original tournament design and analyzed the impact of adding draw constraints to mitigate this issue. The study assessed the effectiveness of the proposed constrained format and examined the implications of alternative designs on the FIFA World Cup qualifying process. Their findings indicated that the initial probability of effective tanking was approximately 1.4%, and the proposed draw constraints significantly reduced this risk to less than 0.013%. The study presented a practical solution to mitigate incentive incompatibility by introducing draw constraints, demonstrating its effectiveness in improving fairness and integrity.

(4) evaluated the impact of a new seeding system in the UEFA Champions League and Europa League. The study evaluated the changes in competitiveness and analyzed the implications of removing the association's factor by conducting simulations across fluctuation amplitudes ranging from 0.03 to 10 and generating 100 random setups for each amplitude. The study produced empirical distribution functions for winner rating difference, competitive balance, and final quality under different seeding procedures. The findings revealed that the new UEFA Champions League seeding policy led to a decline in tournament quality. In contrast, the seeding system in the UEFA Europa League improved the quality (expected rating of the winner and final quality but with less competitive balance). Robustness analysis confirmed the stability of the metrics. The study provided valuable insights into the immediate sporting effects of changes in the UEFA Champions League and Europa League seeding systems. Additionally, the study recommended evaluating other UEFA tournament format changes.

(5) explored the UEFA Champions League Round of 16 draw procedure's impact on the probabilities of different match-ups and the financial consequences for participating clubs. Using specific software, the authors analyzed the draw procedure and calculated the number of feasible outcomes as well as their corresponding probabilities using an algorithm for probabilities of UEFA

draw. The algorithm for calculating probabilities considers all possible order combinations for the runners-up and their corresponding probabilities. For each order, the algorithm identifies the available opponents for each runner-up and calculates the probability of each possible outcome. By iterating through all possible orders and summing the probabilities, the algorithm determines the overall probability of each draw result. They further compared the probabilities generated by their software with the ideal probabilities for equal outcomes. The study found that the UEFA procedure does not produce equal probabilities for all outcomes, leading to significant deviations from the ideal scenario. The study also showed that these deviations, while relatively small in both absolute and relative terms, can still have a material impact on the financial fortunes of participating clubs.

Finally, it was discovered a system that simulates the draw procedures of the UEFA Champions League, Europa League, and Europa Conference League knockout stages, providing probabilities for each possible match-up during the draw process. This system employs a recursive function with the law of total probabilities (LTP) to efficiently calculate the probabilities, even for a large number of teams (64).

$$LTP : P(A) = \sum_n P(A \cap B_n) \tag{2.7}$$

The system employs a bipartite graph data structure to represent the draw process, with each node representing a team and edges connecting teams that are allowed to be matched based on the tournament regulations. The edges in the graph are weighted with values between 0 and 1, representing the probability of each potential match-up.

2.3 Gap analysis and discussion

The Table 2.2 consists of a systematic mapping applied to the articles described in the subsection 2.2. This analysis aimed to understand which techniques were most used by the works collected, the methods they used in common, their objectives and approaches, as well as the areas that remained to be explored or improved. With regard to simulation techniques, there was unanimity in the use of the Monte Carlo simulation technique to infer probabilities in the different domains. Analyzing the scope and breadth of the work, it was found that only 1 article ((4)) considered applying its work to more than one sporting competition, while the vast majority of the others chose to focus their efforts on just one competition. Of the papers collected, a significant number implemented a system to simulate tournament processes, and of these, only a smaller subset also considered simulating the draw processes of the respective tournaments, leaving a total of 5 scientific papers that also considered simulating draws. However, in 2 of these 5 papers, there is no mention of implementing a detailed simulation of the draw processes, which guarantees compliance with the sequence of events taking into account the rules. Only (5), (62) and (63) explain in detail the process of simulating the draw to ensure compliance with the rules. In the scientific

Table 2.2: Systematic Mapping

Features	**Articles**
Uses Monte Carlo simulations	All articles
Focus only in one competition	(3), (59), (13), (**?**), (5), (62)
Considers several competitions	(4)
Simulation of a tournament process	(3), (62), (63), (4), (60), (15), (5)
Simulation of regulated draw processes as a whole	(4), (62), (60), (5), (63)
Simulation of the regulated draw processes step by step	LTP system
Evaluation of the impact of different point systems or formats in competition outcomes	(3), (59), (60), (4)
Evaluation of the impact of different draw rules in competition outcomes	(5), (63)
Evaluation of the impact of different draw rules in the competitor's match-ups probabilities	(62), (5)
Calculation of the probability matrices of the competitor's match-ups a priori of the draw	(5)
Calculation of the probability matrices of the competitor's match-ups a priori and during the draw	LTP system
Calculation of draw probabilities through probabilistic formulas	(5), LTP system
Calculation of draw probabilities through Monte Carlo simulation	(62)

literature found, all the studies considered the draw simulation process as a whole, largely due to the fact that their only objective was to obtain the final results of that process. However, the practical system found ("LTP system"), breaks this paradigm, by having considered the implementation of a "step-by-step" simulation of the draw process in which the sequence of the draw events is defined manually by the user, always complying with the regulations. Within the set of articles that implemented tournament simulations, it was possible to observe a smaller group (3; 4; 59; 60) that aimed to evaluate the impact of different formats and scoring systems on the way these tournaments end up developing. Of the works that focused on reliably simulating the draw processes, it was possible to observe that (5) not only evaluated the effects that the rules applied in the draws have on the course of the tournament events but also sought to obtain the probabilities of matches between the teams that are part of the draw. On the other hand, the work carried out in (62) was more focused on the impact of the draw rules on the outcome of the tournament, while in the (63) article the focus was on how the rules influence the probabilities of matches between teams with certain characteristics. (5) work, however, goes further in the chapter of the probabilities of competitor's match-ups, being the only scientific article that reported calculating a complete probabilities matrix for the draw's result. It is important to note that the LTP system implements an even more complete approach, calculating the probability matrices during the entire manual simulation process, offering a tool that allows you to observe the evolution of the probabilities during the draw process. In both (5) and the LTP system, probabilities are calculated using probabilistic formulas, but in (62) the draw probabilities are obtained through Monte Carlo simulations of the draw process. With this in mind, we recognize the potential to develop a work validated by scientific methods that comprehensively addresses the calculation of probability matrices throughout the draw process using the Monte Carlo simulation method, breaking away from the approaches used in (5) and the LTP system and propelling the exploration of this method in an environment with a huge variety of knockout draws. In this Dissertation we also aim to explore other techniques such as Constraint Programming and Markov Chain, that were not found in the literature.

Chapter 3

Methodological Approach

This chapter provides a comprehensive overview of the methodological framework employed in this project and it is structured into four main sections. Section 3.1 identifies and formalizes the problem, based on the research questions and hypothesis. Section 3.2 outlines the key phases of the project and the tasks performed in each phase, guiding the development process. Section 3.3 delves into the technical foundation and structural components of the simulator, detailing the design and functionality of its main modules. Finally, in Section 3.4 discusses the various methods used to estimate match-up probabilities.

3.1 Problem Formalization

In order to ensure that the objectives of this research are clearly defined and systematically approached, we will identify and formalize the problem that we intend to address with the help of some of the research questions and hypotheses that guide this study.

3.1.1 Research Questions and Hypotesis

The primary research questions guiding this project are as follows:

- **RQ1**: What are the limitations of mathematical methods in obtaining real match-up probabilities for sports draws with seeding rules and restrictions?
- **RQ2**: Can parametrization processes in simulation-based methods be used to generate a wide variety of different draws?
- **RQ3**: Can simulation-based and heuristics methods, such as constraint programming techniques, Monte Carlo Simulation and Markov chain be utilized to estimate realistic probabilities for sports draws with seeding rules and restrictions?

As we try to answer the aforementioned questions, it is possible to suggest the following hypothesis:

- **Hypothesis 1**: Simulation-based methods are alternative methods that can reliably help infer match-up probabilities for sports draws.
- **Hypothesis 2**: It is possible to design a simulator that can accurately reproduce the draw process for various types of knockout tournaments, adhering to specific rules and restrictions.

3.1.2 Problem Definition and Objectives

The problem of this research can be defined as developing a comprehensive simulation framework for sports draws that integrates automation, probability estimation, and constraint handling. The framework needs to accurately simulate the draw process, ensuring all procedural steps are followed consistently. The framework must handle various constraints, such as restrictions on certain match-ups, ensuring that the draw results are valid according to predefined rules. Additionally, it should incorporate methods to estimate the probabilities of various match-ups, providing insights into the likelihood of different outcomes. Finally, through the framework results it should be performed a comparison of different probability estimation methods to identify their strengths and weaknesses in various scenarios.

To address the defined problem, the following specific objectives have been established:

1. **Design and implement a modular simulation framework**: Develop a system that can handle the different stages of the draw process, from team creation to draw simulation and probability estimation.
2. **Develop and apply constraints**: Create mechanisms to create constraints during the draw process, ensuring compliance during the draw process and results.
3. **Integrate multiple probability estimation methods**: Implement Monte Carlo simulations, constraint programming, and Markov chain analysis within the framework to estimate match-up probabilities.
4. **Evaluate the performance of estimation methods**: Conduct a comparative analysis of the different probability estimation methods, assessing their accuracy and computational efficiency.

By addressing these objectives, this research aims to provide a comprehensive solution to the problem of simulating sports draws and estimating match-up probabilities, thereby contributing to the field of computational sports analytics.

3.2 Methodological Pipeline

In order to try to address the problem and objectives described in an efficient manner, a methodological pipeline was defined to guide the work. The development of this pipeline was intended to identify the main phases of the project and predict the tasks for each one of them.

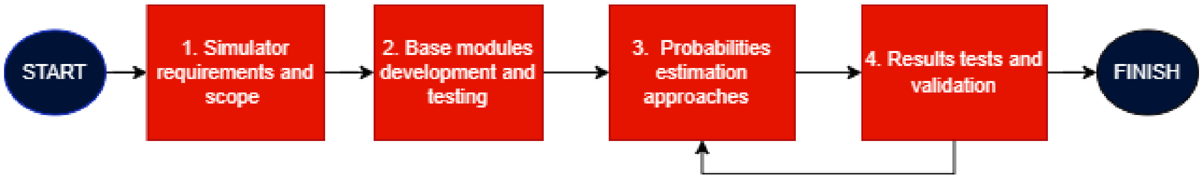

Figure 3.1: Project main phases.

Figure 4.1 shows the 4 main phases identified for the project. In an initial phase, all the requirements for the simulator were aggregated and a scope was established to delimit the possible draws that the simulator would be able to reproduce. This was followed by a software development phase, which involved the building of the basic modules for the web platform, followed by a phase of implementation of the techniques studied for estimating probabilities. Finally, a phase was carried out for testing and validating the proofs of concept for the different techniques implemented, which provoked corrections and adaptations to the respective methods.

3.2.1 Simulator's requirements and scope

The initial phase of the methodological pipeline played an important role in defining the specific area of operation for the sports draw simulator. While the overall focus was on draws for knockout tournaments, this phase delved deeper to establish the specific characteristics and functionalities the simulator would possess. The key tasks undertaken during this phase were outlined as follows:

- **Legislation Analysis**: Analyzing the regulations governing elimination tournament draws to comprehend the various characteristics and potential variations across different competitions.
- **Functional Requirements Gathering**: Working with zerozero collaborators to understand their specific needs and functionalities they expect from the simulator.
- **Draw creation parameters**: Defining the set of parameters required by the simulator to generate customized draws.

By completing these crucial tasks within the requirements and scope definition phase, the project established a clear road-map for the development of the simulator that catered to the specific needs of the possible users while complying with regulations governing knockout tournament draws.

3.2.2 Base modules development and testing

Following the initial phase of requirements and scope definition, the project transitioned to the software development phase. This phase focused on building the core functionalities of the web-based simulator. A breakdown of the key tasks undertaken is detailed below:

- **Technology selection**: Evaluating and identifying suitable web development frameworks, prioritizing efficiency and scalability.
- **Web Development**: Working on back-end and front-end development, by designing user interfaces and building the core application logic.
- **Tests**: Performing unit tests to each module individually and some integration tests between modules to ensure the application's robustness.

By completing these tasks within the base modules' development and testing phase, a solid foundation was established for the following phases of the pipeline.

3.2.3 Probabilities and tests

With the probability estimation and testing phase we were able to delve deeper on the conceptual approaches proposed for estimating match-up probabilities at each step of the draw. This phase involved the development and implementation of proof-of-concept models, followed by testing and comparison to ensure the reliability of the results. The key tasks undertaken during this phase were outlined as follows:

- **Modeling and Implementation**: Developing and implementing proof-of-concept models for each proposed approach (Monte Carlo, Constraint Programming and Markov Chain).
- **Test Set Setup**: Establishing a comprehensive set of draw scenarios to evaluate the robustness of each method. This included creating various draw configurations to test the methods under different conditions.
- **Comparative Testing**: Conducting tests to compare the results of the different probabilistic estimation methods.
  - *Benchmarking*: Comparing the results from the developed methods to evaluate performance.
  - *Similarity Testing*: Assessing the similarity of the probability distributions generated by each method to understand how closely they align.

By completing these tasks within the probability estimation and testing phase, we were able to validate the usefulness of the proposed probabilistic methods and ensure their robustness in estimating match-up probabilities for sports draws.

3.3 Simulator Architecture

This section delves into the technical foundation and structural components of the simulator, highlighting the choice of technologies, the core architecture, and the main modules that drive the system. The architecture integrates a comprehensive data management strategy with a well-defined presentation layer to facilitate seamless interaction and efficient processing. Key entities were created to manage essential information, such as teams, restrictions, and draws, enabling the simulator to operate effectively within the defined parameters. Additionally, this section provides a detailed explanation of the main modules, their functionalities, and the implementation of various probability estimation approaches, including Monte Carlo simulation, Constraint Programming, and Markov Chain.

3.3.1 Technology

Python was selected for the simulator algorithms development due to its readability and clear syntax, enhancing code comprehension and long-term maintenance. Python's scientific libraries offer powerful tools for numerical computations and data manipulation necessary for simulating complex scenarios. The language's extensive community and resources ensure support and learning materials during development.

For web development, Django was the chosen framework. As a Python web framework, Django capitalizes on the previous choice of the simulator's algorithms coding language. Django enforces a well-defined Model-View-Template (MVT) architecture, separating data logic from the presentation layer. This promotes clean and maintainable code, crucial for complex projects. Django also streamlines development by providing pre-built functionalities for common tasks like user authentication and form handling, saving time and effort compared to building a solution from scratch.

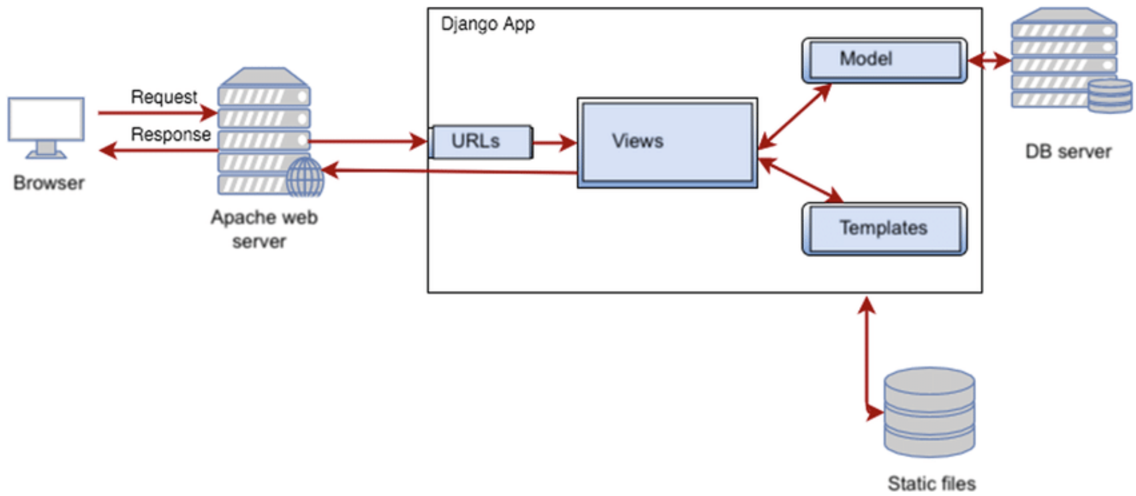

Figure 3.2: Django Project Architecture. Source: (65)

Figure 3.2 provides an overview of the Django application architecture, highlighting the flow of data and interactions between the browser, the web server, and the Django application components. Requests from the browser are processed by the Apache web server, which forwards them to the Django application. Within the Django app, URLs route the requests to the appropriate views, which then interact with the model (data layer) and templates (presentation layer) to generate a response. This architecture ensures a clean separation of concerns, promoting maintainable and scalable code.

3.3.2 Simulator's modeling

Building on the robust technological foundation established with Python and Django, the platform architecture of the simulator was designed leveraging the incorporated SQLite database, facilitating clear data management and interaction between various components.

The platform's data storage system reflects on the key classes of the simulator, each managing specific aspects of the simulation process. These classes and their interactions form the backbone of the simulator, allowing the simulation of the draws.

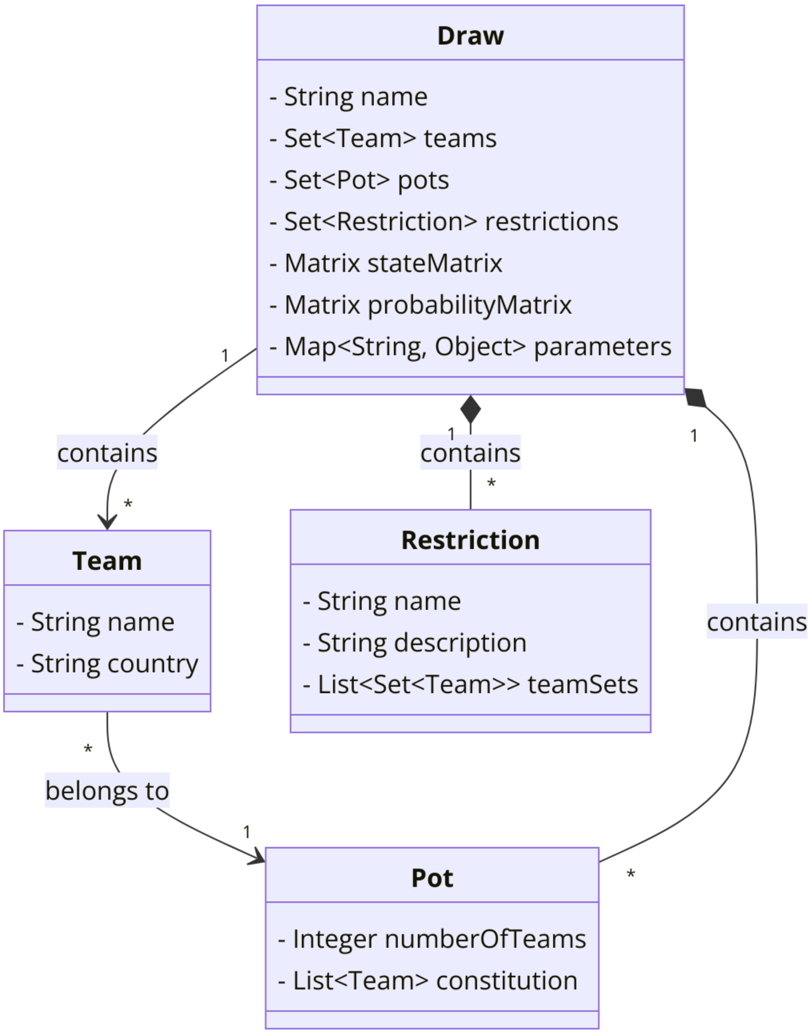

Figure 3.3: UML diagram of the simulator.

In Figure 3.3 we can observe the UML diagram of the simulator. Team is the most elementary entity of the system, representing the basic unit of competition, with each team having a name and a country. Pots are groups of teams used during the draw process, each with a specified number of teams and a constitution listing the teams it contains. Pots are associated with specific draws. Restrictions are rules that prevent certain match-ups between teams. Each restriction has a name, a description explaining its logic, and a list of sets of teams to which the restriction applies, ensuring no games occur between teams in the same set within a draw. The central class in the simulator is the Draw, representing the entire draw event. A draw includes a name, a list of teams, a set of pots, a list of restrictions, a state matrix representing the current state of the draw simulation, a probability matrix storing pairing probabilities, and parameters that dictate the simulation logic.

As previously mentioned, the parameters for creating the draws were defined by taking into account various existing draw models for knockout tournaments and considering recommendations from zerozero collaborators. The set of parameters includes the number of pots, the number of teams per pot, the initial distribution of the teams in the pots, the starting pot, and the method for removing teams from the pots. This set of parameters illustrates the simulator's flexibility, allowing different types of draws to be created and simulated. It is possible to define a number of pots for a draw, as well as an individual number of teams for each pot, allowing draws with various numbers of pots of different sizes. It is also possible to explicitly define the teams for each pot, or to choose the option of randomly initializing the pots, where the user only needs to choose the total set of teams for the draw and they are initially randomly distributed among the pots every time that draw is performed. The user can also define which pot the draw starts in and whether the teams are drawn alternately or sequentially. In an alternating draw, as the name suggests, teams are drawn alternately from the pots, starting with the initial pot and going through all the pots until all the teams have been drawn. In a sequential draw, teams are drawn pot by pot, starting with the initial pot and only changing pots when the previous one no longer has any teams remaining.

3.3.3 Main modules

The simulator's platform has 4 main functional modules, that allow the user to generate customized draws and perform simulations with match-up probabilities estimations.

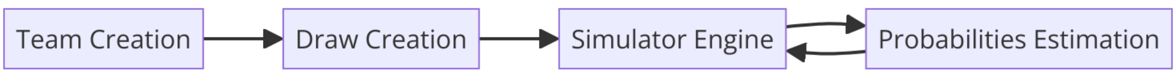

Figure 3.4: Modules flow diagram.

Figure 3.4 illustrates the flow of the application, showcasing how the different modules interact. The process begins with the Team Creation module, where users can input and manage team information. Next, the Draw Creation module allows users to set up a draw by defining its parameters, associating it with teams and implementing restrictions. The Simulator Engine module then takes over, simulating the draw step-by-step. This engine ensures that the draw adheres to the defined parameters and updates the state matrix accordingly. Finally, the Probabilities Estimation module calculates and updates the probability matrix, providing real-time match-up probabilities for the current state of the simulation. This modular design of the simulator allows a clear separation of concerns, where each module is essential for the overall functionality of the simulator.

3.3.3.1 Team Creation Module

The Team Creation module provides a straightforward interface for users to create and manage team information. This module consists of a simple form that allows users to input the name, country, and logo of a team. Upon submission, the team data is permanently saved.

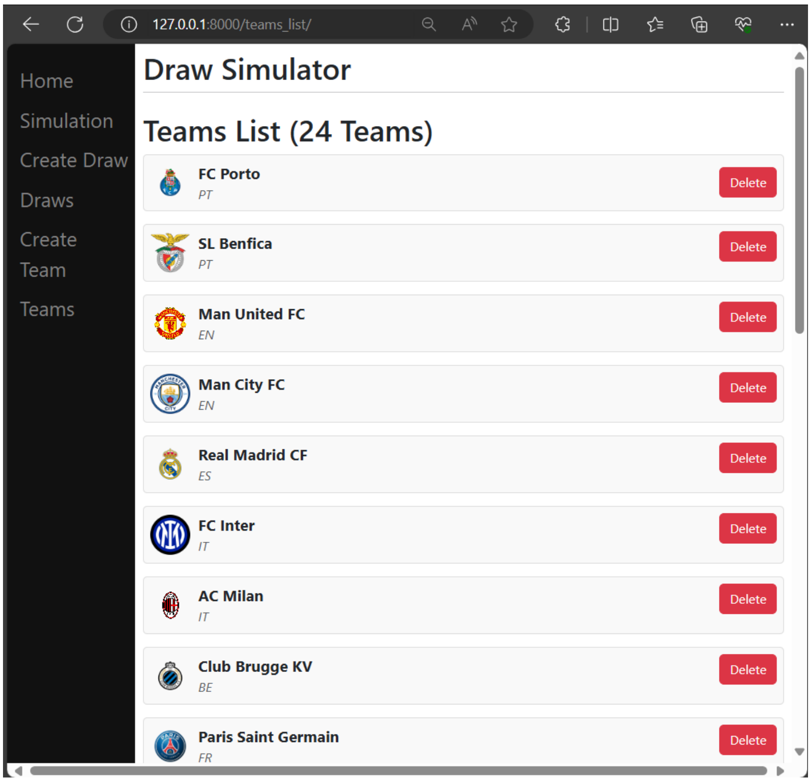

Figure 3.5: Teams' Management Interface.

As shown in Figure 3.5, in addition to the creation form, this module includes an interface to manage the created teams. Users can view, edit, and delete existing teams through this interface, facilitating easy maintenance of team data.

3.3.3.2 Draw Creation Module

The Draw Creation module takes users through a step-by-step process to generate a new draw. This process is designed to ensure flexibility and customization in order to create draws for different competitions with different characteristics.

This process starts with an initial form in which the user provides the name of the draw and the number of pots it will have. Allowing to quickly establish the basic framework of the draw.

In the second step, the user defines the draw parameters. These parameters include the number of teams per pot, the initial pot, the pot initialization process (static or random), and the mode for drawing the teams from the pots (alternated or sequential). This step allows users to adapt the logic of the draw process, allowing for different combinations of parameters.

The third step varies based on the selected pot initialization process. If the static option is chosen, the user manually specifies the teams for each pot. On the other hand, if the random option is selected, the user defines the total set of teams for the draw, and the system randomly distributes the teams among the pots in each draw instance.

The final step is a confirmation page where the user reviews all the draw details. This page provides a comprehensive overview of the draw configuration, allowing the user to confirm the draw creation or to add restrictions. If the user wants to add restrictions, they can do so by pressing the "Add Restrictions" button, which initiates the restriction creation process.

The restriction creation process begins with an initial form where the user defines the name and description of the restriction. This is followed by a final step where the user adds sets of teams of the draw to the restriction, preventing games between teams of the same set.

This modular and systematic approach to draw creation ensures that the user can efficiently set up and customize draws while maintaining a high degree of control over the draw parameters and restrictions.

3.3.3.3 Simulator Engine Module

The Simulator Engine module is the core component that sets up a draw based on its parameters and allows the user to perform and visualize the draw process. This module provides an interactive interface where users can observe and manipulate the draw as it unfolds and interacts with the Probability Estimation module, sending information about the draw state at each step of the simulation.

The interface of the draw engine is divided into several key sections:

In Figure 3.6, the yellow box highlights the part of the interface that shows the pots' composition, listing the teams in each pot. This section allows the user to see the initial distribution of teams and how they are grouped for the draw. The blue box highlights the table that displays the match-up probabilities between teams at each step of the draw. This table provides real-time updates on the likelihood of different match-ups occurring, helping users to understand the dynamics of the draw as it progresses.

In Figure 3.7, the orange box highlights the match fixtures table, showing the teams that have already been drawn and their match-ups. This section allows users to track the progress of the draw and see which teams have been paired. The green box highlights the options menu buttons, which provide the main functionalities of the simulator that the user can manipulate:

- **Direct Selection**: The user can directly choose the team that is drawn in each step of the draw. This feature is helpful for certain cases of analysis or specific scenarios the user wants to test.

- **Random Button**: The user can choose to randomly pick a team from the available teams for each step by pressing this button.

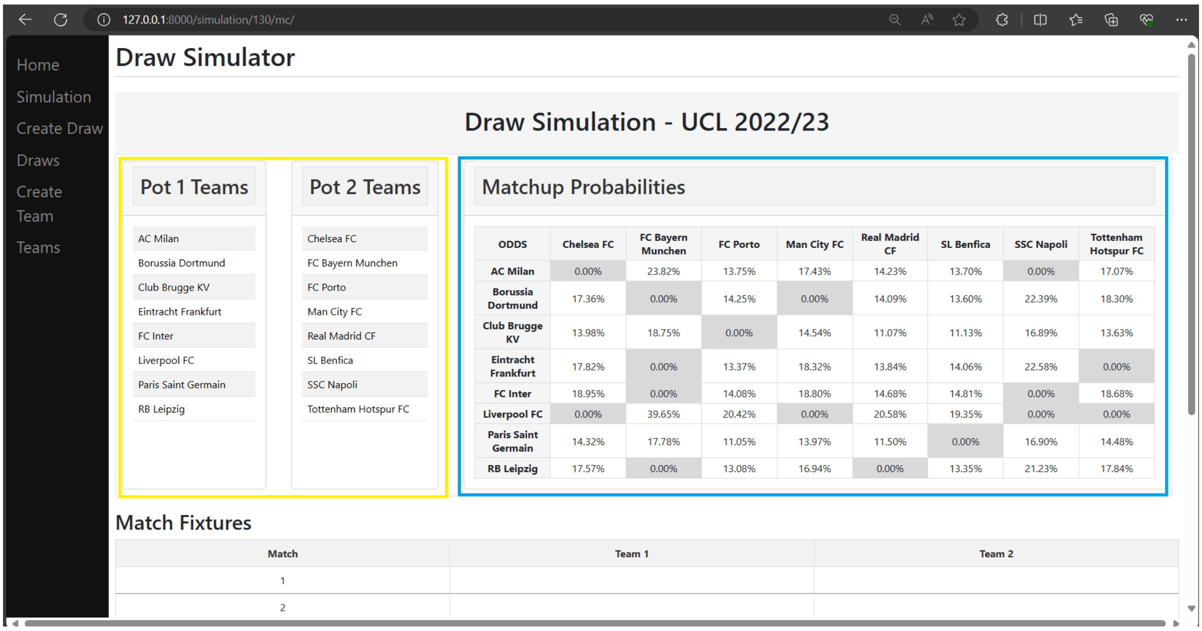

Figure 3.6: Simulator engine Interface

- **Play Button**: When the user presses the "Play" button, the simulator generates a step-by-step random draw with a timeout between each drawn team. This automated process allows the user to observe a fully randomized draw sequence.
- **Clear Button**: The clear button resets the draw process, taking it back to the initial step and clearing all progress. This allows the user to start over and create a new draw if needed.

This interactive approach ensures that users have full control over the draw simulation process, enabling them to both observe and influence the outcomes in real-time.

3.3.3.4 Probabilities Estimation Module

The Probabilities Estimation Module is a crucial component of the simulator architecture, responsible for updating the match-up odds table based on the current state of the draw. Unlike other modules, it operates without a user interface, working behind the scenes to perform complex calculations and provide essential data for the simulation process.

This module is structured to receive two primary inputs:

- **State Matrix**: The matrix of match fixtures with the teams already drawn.
- **Draw Information**: Details about the draw, including pot composition and the parameters that govern the draw process.

Using these inputs, the Probabilities Estimation Module outputs two pieces of information:

- **match-up Odds Matrix**: A matrix that represents the probabilities of pairing teams for the next steps of the draw.

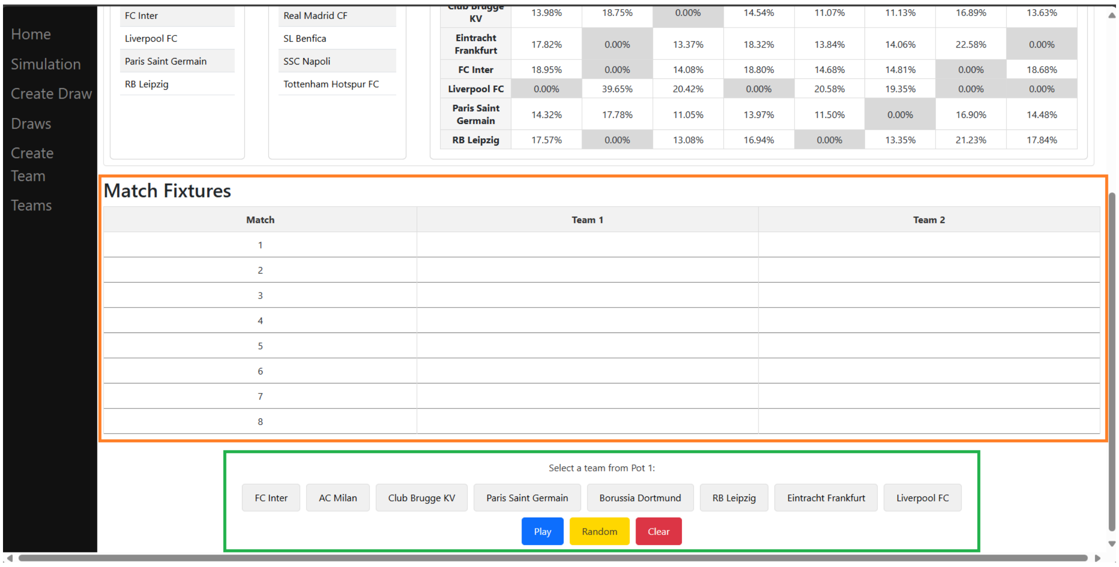

Figure 3.7: Simulator engine Interface

- **Selectable Teams**: A list of teams eligible for selection in the next step of the draw, considering the current state and draw rules.

The module is integrated with the Draw Simulator Engine, ensuring seamless communication and data exchange. Its versatile format, with designed endpoints, allows easy integration of the various probability estimation approaches, including Monte Carlo simulation, constraint programming, and Markov chain analysis.

3.4 Probability estimation approaches

The different techniques proposed for estimating probabilities have been implemented in the probability estimation module. As mentioned, this module receives as input the information about a draw and its current status, more specifically the state matrix with the teams that have already been drawn and the draw parameters as well as the composition of the pots. With this information, we have tried to estimate the probabilities of matches between the teams in the draw using different methods, described in more detail in the next sections.

3.4.1 Monte Carlo

The implementation of a Monte Carlo simulation approach to estimating draw probabilities involves generating a certain number of samples of draw results and calculating the frequency of matches between teams and, consequently, the probabilities.

3.4.1.1 Algorithm

The algorithm to generate each sample in the Monte Carlo simulation reproduces the draw process by randomly picking a team in each step. This means that if, for a certain step, there are N teams that can be picked, a team is randomly drawn with a probability of $\frac{1}{N}$. This method, however, can result in invalid draws because, in each step of the MC draw simulation process, we are not calculating dead-ends and simply making direct restriction checks. This means that when the second team of a match is to be drawn, we are considering the teams of the respective pot that do not have a restriction with the team waiting for an opponent, and making further checks.

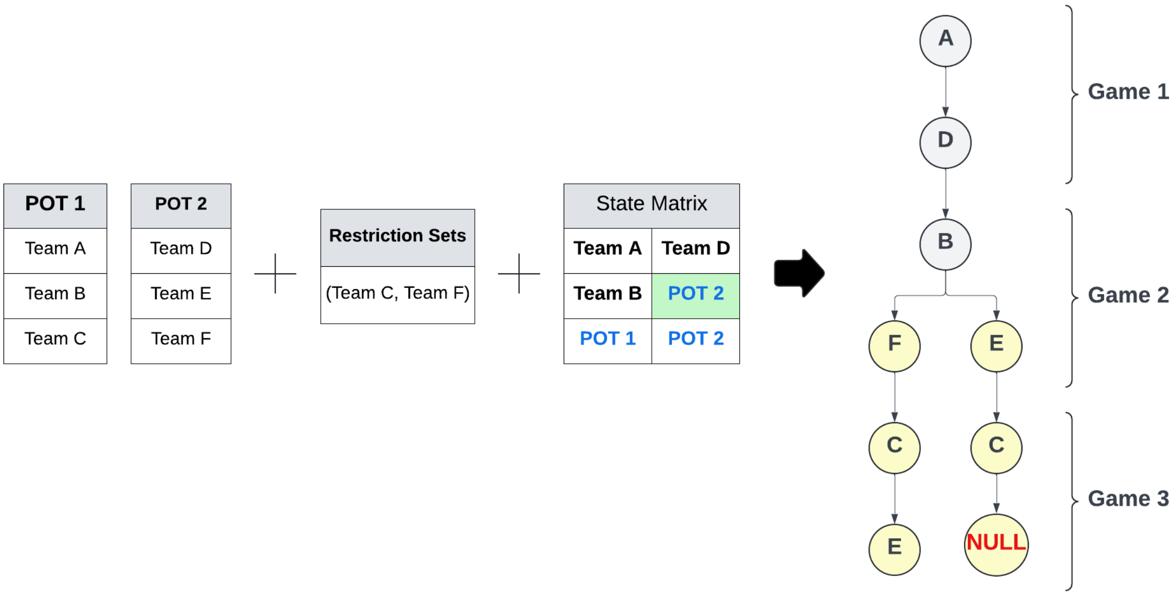

Figure 3.8: Draw dead-end case analysis.

Figure 3.8 shows the analysis of a concrete dead-end scenario for a certain draw. In the draw represented in the figure, is known that Team C and Team F are not allowed to play and Team B is waiting for an opponent that either can be Team E or Team F. Team B does not have any restriction with either Team E or Team F, so the algorithm will consider both options with a 50% probability. But if Team E is drawn against Team B, it will not be any valid opponent remaining for Team C, resulting in a dead-end. To ensure that we work only with valid draws, the samples that resulted on a dead-end are not considered for the probability estimation.

3.4.1.2 Match-up Probabilities

The probability estimation is performed by taking all the valid MC simulations and analyzing the frequency of matches between teams. The selectable teams for the next step of the draw are all the different teams that feature in the first unfilled slot of the samples generated through the Monte Carlo simulations.

Initial observations immediately highlighted one imprecision of this method for calculating probabilities through the frequency of matches in the MC simulations, which was the unbalanced set of simulation results. Figure 3.9 illustrates this problem and the mechanism used to correct this issue. In the provided example there are 8 MC generated samples, for a certain draw. Note

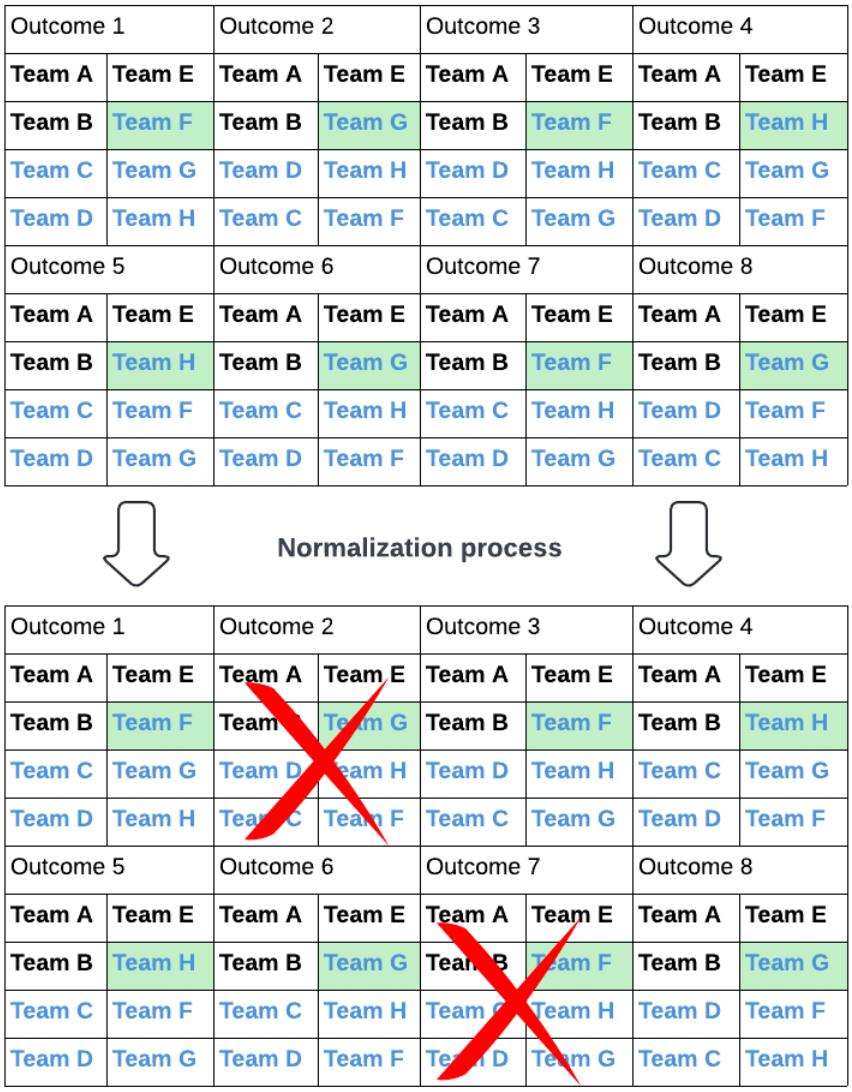

Figure 3.9: Monte Carlo sample's normalization.

that for this draw three teams have already been drawn ('Team A', 'Team B' and 'Team C'), being the teams in blue the ones obtained through simulation. The slot in green represents the next slot to be drawn, and it is possible to see that there are 3 different options of teams for that slot: 'Team F', 'Team G' and 'Team H'. Assuming that those 3 teams are all the possible options for the next drawing team, it is known that the probability of each one of them being picked is $\frac{1}{3}$ at that point. However, the number of samples generated by MC simulations that start with each different team is not exactly the same, which cause skewed odds. In this particular case, we get $\frac{3}{8}$ of probability for Team G and Team F and $\frac{2}{8}$ for Team H. Through the normalization process we randomly eliminate the samples from the exceeding cases to ensure that there are exactly the same number of simulations starting with each option, securing mathematical logic at the first level of the draw simulation and the correct probability of $\frac{1}{3}$ for each option. Initially, the number of samples starting with each team may vary slightly, but after normalization, we ensure that the number of simulations starting with each team is balanced, leading to more accurate and reasonable probability estimation.

3.4.2 Constraint Programming Approach

The approach for estimating probabilities in sports draws using constraint programming was implemented through the *python-constraint* library. This library provides several key functions that enable the creation and solving of constraint satisfaction problems:

- "*Problem()*": is the function used to instantiate a CSP.
- "*addVariable(variable, values)*": is the function that allows the creation of variables and definition of their values' domain. It is associated to a CSP instance.
- "*addConstraint(constraint_function, variables)*": is the function that allows the creation of constraints and associates them with the problem.
- "*getSolutions()*": is the function that calls the solver of this constraint library and provides all the solutions for the defined CSP.

3.4.2.1 Modeling the Draw as a Constraint Programming Problem

Modeling the draw process as a constraint programming problem involves several steps. The first stage involves defining the problem variables. In this case, our variables are the slots in the result grid. Each slot is identified by a letter and a number, where the letter identifies the match and the number (1 or 2) the team's position for that match. The next step is to assign possible values to the variables. These values are the possible teams to be drawn for each slot. Using the information about the parameterization and constitution of the pots of a draw, it is possible to calculate in advance which pot will provide the teams for each of the slots in the results matrix.

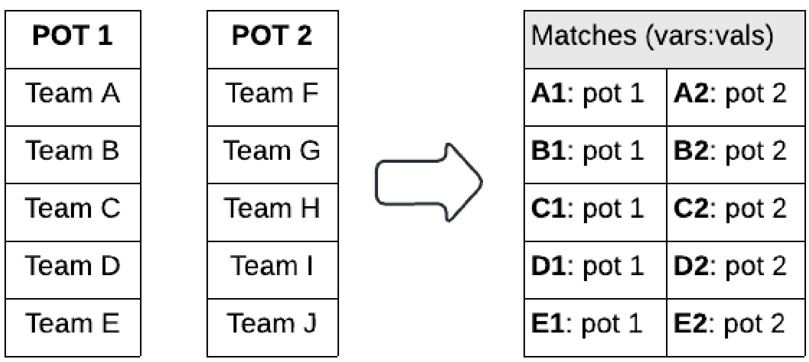

Figure 3.10: Variables initialization example.

Figure 3.10, shows an example of an alternated draw with 2 pots, each one of them with 5 teams and its pots-slot correspondence. For this particular example, the possible values for the variables "A1", "B1", "C1", "D1" and "E1" are the teams from pot 1 ("Team A", "Team B", "Team C", "Team D" and "Team E"), while the values for the slots "A2", "B2", "C2", "D2" and "E2" are the teams from pot 2 ("Team F", "Team G", "Team H", "Team I" and "Team J")

The next step of the modeling process is to introduce the constraints. For our problem, 2 different constraints were introduced.

The *AllDifferentConstraint()*, is a built-in constraint in the library that makes sure that for every solution of the problem, all variables have different values. For our context, this constraint ensures that each team is assigned to exactly one slot of the result grid.

The second constraint is a customized one created from scratch to ensure that no matches occur between teams that are restricted from playing against each other. This constraint discards solutions where teams that belong to the same restricted set are paired in a match.

After defining the constraints, the final step is to solve the problem using *getSolutions()*. This function finds all possible solutions according to the defined variables and constraints. The estimation of the match-up probabilities is based on the results of this function. By analyzing the frequency of matches between teams across all valid solutions, we can estimate the probability of each match-up occurring in the draw. This method ensures that the estimated probabilities are based on all possible valid configurations of the draw, considering the constraints imposed by the draw parameters and restrictions.

3.4.3 Markov Chain

In this section, we discuss our approach using Markov Chains to estimate probabilities for sports draws. Markov Chains provide a framework for modeling stochastic processes where the next state depends only on the current state. This characteristic can be leveraged to simulate draws while considering constraints.

To implement this approach, we designed a Markov Chain model that simulates the draw process with restrictions. The key idea is to represent the slots in the result grid as states in the Markov Chain. Transitions between states are determined by the teams available in the respective pots and the constraints applied. For instance, moving from state S_i to S_{i+1} involves selecting a team for slot S_{i+1} from the corresponding pot, ensuring no restrictions are violated.

The process of this method can be split in 4 main steps:

1. **Initialization**: Define the slots in the draw as states in the Markov Chain. Initialize the pots with teams and define the sequence of the pots.

2. **State Transitions**: For each slot, determine the possible teams that can be assigned based on the current state of the draw. This involves checking the teams available in the pot and applying the constraints to avoid restricted match-ups.

3. **Simulation**: Simulate the draw process by traversing through the states, assigning teams to slots while ensuring the constraints are respected. If a state transition leads to a dead-end (no valid teams available), the simulation is considered invalid and discarded.

4. **Probability Estimation**: Run multiple simulations to generate a sufficient number of valid draw configurations. The probabilities of match-ups are estimated by analyzing the frequency of each match-up across all valid simulations.

While the Markov Chain model successfully simulates the draw process, it failed in determining the match-up probabilities by itself. The implemented approach to estimating probabilities involves running multiple simulations, similar to the Monte Carlo method. The structured nature of the Markov Chain provides a clear representation of the draw process, but without a specific mechanism for probability estimation, it essentially replicates the Monte Carlo approach. The draw process can be modeled using Markov Chain, but as probability estimation approach it becomes redundant.

Chapter 4

Empirical Studies

This chapter will analyze and discuss the results obtained from implementing the methods described in Chapter 3 for probabilities estimation. The goal is to compare the draw probabilities and performance of the different methods and existing tools in specific relevant scenarios.

4.1 Performance

To evaluate performance, we created draws with 2 Pots, with the same number of teams each, in which the teams are drawn alternately, without any restrictions.

Table 4.1: Performance measures.

Teams per Pot	MC1000 (sec)	MC10000 (sec)	MC50000 (sec)	MC100000 (sec)	RA (sec)
2	<1	<1	< 1	1.53	< 1
3	<1	< 1	1.12	2.11	<1
4	<1	<1	1.86	2.89	<1
5	<1	<1	2.04	3.1	1.65
6	<1	<1	2.37	5.9	24.77
7	<1	<1	3.68	6.25	2425.80
8	<1	1.06	3.96	7.36	-
9	<1	1.34	4.53	7.86	-
10	<1	1.54	5.32	11.25	-

Table 4.1 presents the performance of two algorithms, MC (Monte Carlo) and RA (Restriction Algorithm), in calculating initial odds for draws with varying numbers of teams. MC simulations were run with 1,000, 10,000, 50,000, and 100,000 iterations. It can be observed that for draws with up to 5 teams, both algorithms achieved very fast computation times. However, the execution time for the RA algorithm increased significantly for draws with more teams. For instance, calculating odds for a 6-team draw with RA took over 4 times longer than 100,000 MC simulations. This trend continued for draws with more teams, with the RA taking approximately 40 minutes to run a 7-team draw, which is over 100 times longer than the previous 6-team draw. This abrupt decrease in performance of the RA made it expendable to measure the execution time for draws with more

than 7 teams. On the other hand, MC computation times remained manageable for all scenarios. It is easy to understand why the RA approach becomes nonviable for draws with more teams if we know how the algorithm works and look at how the number of draw outcomes increases with the addition of teams.

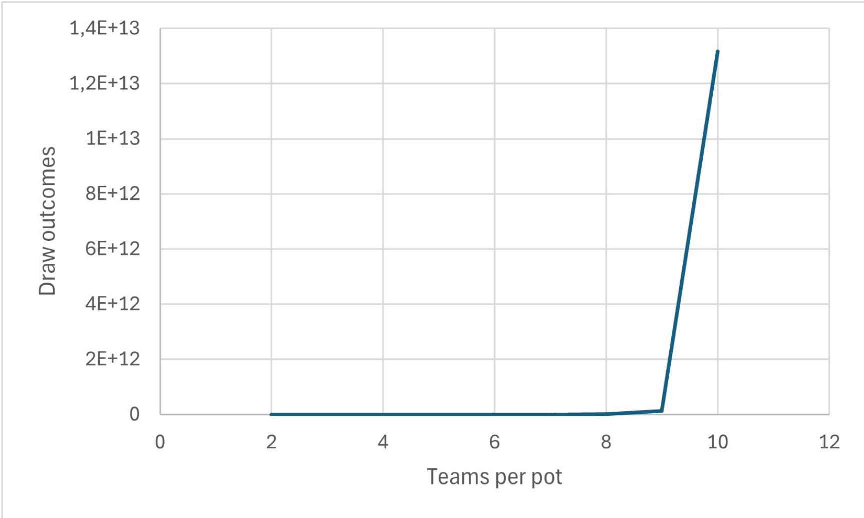

Figure 4.1: Draw outcomes per number of teams.

Unlike MC which simulates a fixed number of random possibilities, RA finds all possible outcomes for the draw. As the number of teams increases, the number of possible outcomes grows exponentially, as shown in Figure 4.1. The total number of different outcomes for the type of draw created with N teams per Pot is given by $(N!)^2$. For instance, a 7-team draw has 254,016,000 possible outcomes, but for an 8-team draw, there are already 1,625,702,400 possibilities, which is 64 times larger. This exponential growth in possibilities overwhelms the RA algorithm, making it slow and impractical for larger draws.

4.2 Comparative Testing

In order to analyze the results obtained through the different methods implemented, a series of tests were carried out to understand their behaviour in greater detail. We began by constructing a set of draws, so that we could apply the implemented approaches and then calculate similarity metrics between the probability matrices'. The data set of draws was formed by the past 10 editions of the UCL R16 draw, so that we can include the LTP system discussed in the Literature Review. This type of draw is a 2-Pot draw, with the same number of teams per Pot, in which they are drawn alternately, with restrictions between teams from the same country and those that came from the same group in the previous round. Due to the inefficiency of the RA algorithm, mentioned above, the number of teams in each Pot was reduced from 8 to 6.

Tables 4.2 and 4.3 present similarity metrics between the three different systems (LTP, MC and RA) used to estimate probabilities of match-ups for the draws' set. The similarity metrics used are Euclidean Distance and Cosine Similarity, applied to the match-up probabilities matrices. The

Table 4.2: Euclidean distances between different methods.

Draw	Eucl. Dist. (LTP x MC)	Eucl. Dist. (LTP x RA)	Eucl. Dist (MC x RA)
UCL 24	1.28708974	0.977803661	0.749333037
UCL 23	2.016085316	2.058421725	0.909749366
UCL 22	0.556147462	0.577061522	0.574891294
UCL 21	3.225368196	3.816595341	1.059716943
UCL 20	2.32282156	3.161755841	1.268227109
UCL 19	1.835919388	2.327509398	0.912195155
UCL 18	1.383645969	1.065987561	0.789112856
UCL 17	0.813557944	0.891156274	0.795642881
UCL 16	2.774165228	3.356482254	1.156899945
UCL 15	1.700564221	1.742331186	0.847141633
Average	1.7915365024	1.99750979581	1.106042001481

Euclidean Distance between MC and RA is the lowest among all comparisons, with an average of 1.106. This indicates a strong similarity between the MC and RA methods. But if we compare both methods with the LTP, it is possible to observe that the Euclidean Distance between LTP and MC is consistently lower compared to the distance between LTP and RA across all editions, except for UCL 24 where the distance was higher between MC and LTP(1.287 for LTP x MC vs 0.978 for LTP x RA). Suggesting that the probability distributions generated by LTP and MC are more similar than those generated by LTP and RA.

Table 4.3: Cosine similarity between different methods.

Draw	Cos. Sim. (LTP x MC)	Cos. Sim. (LTP x RA)	Cos. Sim. (MC x RA)
UCL 24	0.999934432	0.999962158	0.999977776
UCL 23	0.999871313	0.99986585	0.99997378
UCL 22	0.999988842	0.999987988	0.999988078
UCL 21	0.999653457	0.999514875	0.999962597
UCL 20	0.999836307	0.999693847	0.999951276
UCL 19	0.999881117	0.999808966	0.999970663
UCL 18	0.999920494	0.999959863	0.999977424
UCL 17	0.999978706	0.999964117	0.999979586
UCL 16	0.999787399	0.999627748	0.999960615
UCL 15	0.999900372	0.999852755	0.999969885
Average	0.999875244	0.999823817	0.999971168

The Cosine Similarity values are extremely high for all pairs, indicating that the probability distributions are nearly identical in terms of direction, though their magnitudes may differ. The highest cosine similarity is consistently between MC and RA, with values approaching 1 (e.g., 0.999978 for UCL 24). This reaffirms the close resemblance between these two methods in terms of the direction of their probability distributions.

The high similarity between MC and RA, as indicated by both Euclidean Distance and Cosine Similarity, suggests that these two methods produce highly comparable results. This could imply that both approaches are effectively capturing the constraints and dynamics of the knockout tournament draws in a similar manner. However, it may be safe to say that for practical applications, the MC method might overtake the RA, as it has shown very similar probabilistic results with much higher efficiency.

The closer resemblance of LTP to MC (compared to RA) in terms of Euclidean Distance (more significantly) and Cosine Similarity (less significant) could mean that while LTP's approach is somewhat more aligned with the probabilistic nature of the Monte Carlo method, diverging more significantly from the constraint-focused RA.

Although this similarity analysis showed some patterns of the probabilistic distributions of all the methods, we tried to break down what those quantitative values may suggest by analyzing 2 specific cases of draws simulations.

4.2.1 Case Study 1

For the first case, we used the 2023 UCL R-16 draw, whose rules have already been described above. Let's start by analysing the probabilities of the LTP and MC systems for the entire draw, excluding the RA system due to its efficiency problems for larger draws, as already mentioned.

Table 4.4: LTP Probability Matrix for UCL 23 Draw.

Teams	CHE	BAY	POR	CIT	RMA	BEN	NAP	TOT
ACM	0.00	23.95%	13.40%	17.80%	13.65%	13.40%	0.00%	17.80%
DOR	17.95%	0.00%	13.75%	0.00%	14.20%	13.75%	22.22%	18.14%
BRU	13.90%	19.46%	0.00%	13.88%	10.98%	10.79%	17.11%	13.88%
FRK	17.95%	0.00%	13.75%	18.14%	14.20%	13.75%	22.22%	0.00%
INT	19.08%	0.00%	14.22%	18.89%	14.70%	14.22%	0.00%	18.89%
LIV	0.00%	37.12%	20.80%	0.00%	21.28%	20.80%	0.00%	0.00%
PSG	13.90%	19.46%	10.79%	13.88%	10.98%	0.00%	17.11%	13.88%
RBL	17.23%	0.00%	13.30%	17.41%	0.00%	13.30%	21.34%	17.41%

Table 4.5: MC Probability Matrix for UCL 23 Draw.

Teams	CHE	BAY	POR	CIT	RMA	BEN	NAP	TOT
ACM	0.00%	23.89%	13.57%	17.36%	14.04%	13.83%	0.00%	17.31%
DOR	18.04%	0.00%	13.85%	0.00%	14.11%	13.79%	22.18%	18.04%
BRU	13.88%	18.21%	0.00%	13.98%	11.21%	11.20%	17.30%	14.22%
FRK	18.01%	0.00%	13.83%	18.35%	14.07%	13.63%	22.11%	0.00%
INT	18.80%	0.00%	14.29%	19.00%	14.70%	14.30%	0.00%	18.91%
LIV	0.00%	39.19%	19.95%	0.00%	20.73%	20.14%	0.00%	0.00%
PSG	13.81%	18.71%	11.19%	14.14%	11.14%	0.00%	16.86%	14.15%
RBL	17.46%	0.00%	13.33%	17.17%	0.00%	13.11%	21.55%	17.37%

Examining the LTP probability matrix (Table 4.4), we observe that certain match-ups have a probability of 0%. For instance, ACM cannot face CHE, NAP, or CIT. These zero probabilities reflect the restrictions of the draw, such as teams from the same country or those who have already played against each other in the group stage. The highest probability for LIV is to face BAY (37.12%), indicating a limited number of viable opponents due to these restrictions. Similarly, DOR has a high chance of facing NAP (22.22%).

The Monte Carlo probability matrix (Table 4.5) shows a similar pattern, with zero probabilities for restricted match-ups. The highest probabilities are consistent with the LTP matrix, such as "LIV vs BAY" (39.19%) and "DOR vs NAP" (22.18%). The Monte Carlo method also predicts a notable probability for ACM facing BAY (23.89%), similar to LTP's 23.95%.

To further analyze the differences between the LTP and MC methods, we calculate the differences in their probability estimations.

Table 4.6: Difference Matrix (LTP - Monte Carlo) for UCL 23 Draw.

Teams	CHE	BAY	POR	CIT	RMA	BEN	NAP	TOT
ACM	0.00%	0.06%	-0.17%	0.44%	-0.39%	-0.43%	0.00%	0.49%
DOR	-0.09%	0.00%	-0.10%	0.00%	0.09%	-0.04%	0.04%	0.10%
BRU	0.02%	1.25%	0.00%	-0.10%	-0.23%	-0.41%	-0.19%	-0.34%
FRK	-0.06%	0.00%	-0.08%	-0.21%	0.13%	0.12%	0.11%	0.00%
INT	0.28%	0.00%	-0.07%	-0.11%	0.00%	-0.08%	0.00%	-0.02%
LIV	0.00%	-2.07%	0.85%	0.00%	0.55%	0.66%	0.00%	0.00%
PSG	0.09%	0.75%	-0.40%	-0.26%	-0.16%	0.00%	0.25%	-0.27%
RBL	-0.23%	0.00%	-0.03%	0.24%	0.00%	0.19%	-0.21%	0.04%

Table 4.6 highlights the discrepancies between the LTP and Monte Carlo methods in their probability estimations. The differences span between -2.07% in the match of LIV and BAY, and 1.25% in the match involving BAY and BRU. These differences can be attributed to the inherent variations in the probabilistic algorithms and how each method handles the restrictions imposed by the draw. Despite these discrepancies, the overall probability estimation behavior of both methods is quite similar. Most differences are minor and do not significantly affect the overall predictions, suggesting that both methods provide consistent and reliable predictions for this draw.

For further analysis of this draw, we manually picked some teams to produce a scenario with less amount of draw outcomes, where possibly the restrictions impact more on having higher and more defined probabilities. The matches defined manually were: "LIV vs POR", "BRU vs NAP", "ACM vs CIT" and "INT vs RMA". With this reduction in the number of teams to be drawn we could now include the RA algorithm results for this scenario.

Upon examining the RA and LTP probability matrix for the UCL 23 Draw scenario (Table 4.7), we see that both methods respected the restrictions. For instance, there is a 100% probability of BAY facing PSG, due to the restriction involving German teams, and a 0% probability for FRK and TOT to face each other, as they were in the same group during the group stage.

The RA and LTP methods show identical probabilities because both are deterministic algorithms, which incorporate the same constraints and draw logic. On the other hand, MC method

Table 4.7: RA and LTP probability matrix for UCL 23 Draw scenario.

Teams	CHE	BAY	BEN	TOT
DOR	25.00%	0.00%	25.00%	50.00%
FRK	50.00%	0.00%	50.00%	0.00%
PSG	0.00%	100.00%	0.00%	0.00%
RBL	25.00%	0.00%	25.00%	50.00%

while stochastic by nature, also respects the same restrictions, as shown in Table 4.8. As expected, MC shows minor differences due to the inherent randomness of its approach. For example, DOR's probabilities against CHE and BEN are 25.01% and 25.13% in the MC matrix compared to the exact 25.00% in the RA and LTP matrices.

These results show that despite the differences in methodology, all three approaches converge on similar probability distributions in a smaller scenario where the restrictions are clearly defined and the number of outcomes is significantly reduced. The small variations observed in the MC method are expected and acceptable given its probabilistic nature.

4.2.2 Case Study 2

For the second test case, a draw was created with different parameters than the previous draws, in order to analyze the robustness and flexibility of the simulator and the RA and MC methods for estimating the match-up probabilities in a simulation with different configurations. The draw was created with 3 pots. Pots 1 and 2 have 3 teams each and Pot 3 has 6 teams. The draw starts with Pot 1 and the teams are drawn alternately from the Pots.

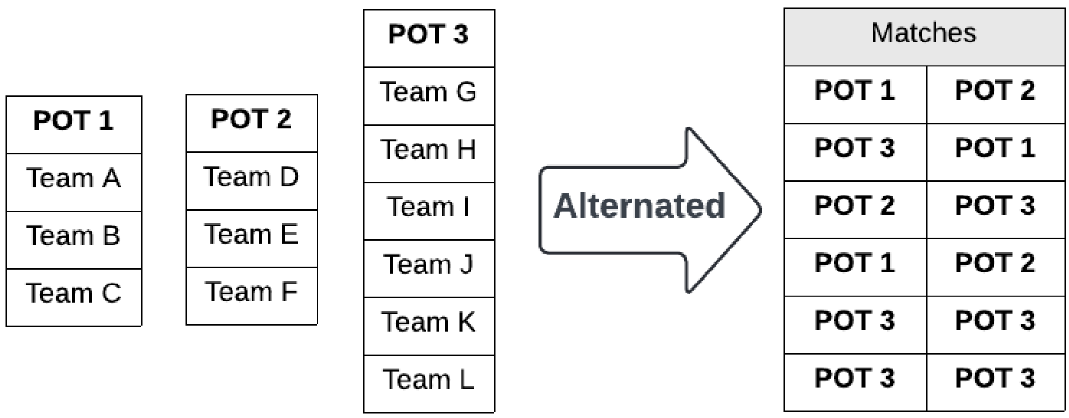

Figure 4.2: Case 2 draw with alternated picks.

Figure 4.2 shows the configuration of the pots and their correspondence in the results matrix. In the previous case, the games were always made up of 2 teams from different pots. In this case, there is greater diversity in the make-up of the games, with 2 games between teams from Pot 1 and

Teams	CHE	BAY	BEN	TOT
DOR	25.01%	0.00%	25.13%	49.86%
FRK	50.46%	0.00%	49.54%	0.00%
PSG	0.00%	100.00%	0.00%	0.00%
RBL	24.53%	0.00%	25.33%	50.14%

Table 4.8: MC probability matrix for UCL 23 Draw scenario.

Pot 2, 1 game between teams from Pot 3 and Pot 1, 1 game between teams from Pot 2 and Pot 3, and 2 games between teams from Pot 3. Additionally, the following games were restricted: "Team A vs Team L", "Team B vs Team F", "Team H vs Team I", "Team D vs Team J""Team D vs Team K" and "Team J vs Team K".

Table 4.9: RA Probability Matrix for case 2 alternated draw.

Teams	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**
A	0.00%	0.00%	0.00%	24.88%	21.95%	27.32%	4.39%	5.12%	5.12%	5.61%	5.61%	0.00%
B	0.00%	0.00%	0.00%	30.73%	25.37%	0.00%	6.34%	7.32%	7.32%	8.29%	8.29%	6.34%
C	0.00%	0.00%	0.00%	23.66%	20.73%	25.37%	4.39%	5.12%	5.12%	5.61%	5.61%	4.39%
D	24.88%	30.73%	23.66%	0.00%	0.00%	0.00%	4.63%	5.61%	5.61%	0.00%	0.00%	4.88%
E	21.95%	25.37%	20.73%	0.00%	0.00%	0.00%	4.63%	5.61%	5.61%	5.61%	5.61%	4.88%
F	27.32%	0.00%	25.37%	0.00%	0.00%	0.00%	6.83%	8.29%	8.29%	8.29%	8.29%	7.32%
G	4.39%	6.34%	4.39%	4.63%	4.63%	6.83%	0.00%	13.41%	13.41%	15.12%	15.12%	11.71%
H	5.12%	7.32%	5.12%	5.61%	5.61%	8.29%	13.41%	0.00%	0.00%	17.68%	17.68%	14.15%
I	5.12%	7.32%	5.12%	5.61%	5.61%	8.29%	13.41%	0.00%	0.00%	17.68%	17.68%	14.15%
J	5.61%	8.29%	5.61%	0.00%	5.61%	8.29%	15.12%	17.68%	17.68%	0.00%	0.00%	16.10%
K	5.61%	8.29%	5.61%	0.00%	5.61%	8.29%	15.12%	17.68%	17.68%	0.00%	0.00%	16.10%
L	0.00%	6.34%	4.39%	4.88%	4.88%	7.32%	11.71%	14.15%	14.15%	16.10%	16.10%	0.00%

Table 4.9 shows the RA probability matrix for Case 2 draw. We can see that all the restricted games were respected by the algorithm, for example, Team D cannot face Team J or Team K, as indicated with a 0.00% probability. The probability 5.61% is the most common, being found in matches between teams in Pot 1 and Pot 3 and between teams in Pot 2 and Pot 3, which are the least frequent matches. The highest probabilities are found in matches that happen the most, namely matches between teams from Pot 1 and Pot 2, with the probability of the match between Team D and Team B reaching 30.73%, and in the matches between teams from pot 3, with a probability of 17.68% between teams such as I, J and K. Although the results matrix shows the same number of matches between teams from Pot 1 and Pot 2 and between teams from Pot 3, the probabilities of matches between teams in Pot 3 are naturally lower, as pot 3 has more teams, over which the probabilities are distributed.

Table 4.10 shows the MC probability matrix for Case 2 draw. Firstly it is clear that, like the RA algorithm, the Monte Carlo method respected all the restrictions initially defined for this draw.

Table 4.10: MC Probability matrix for case 2 alternated draw.

Teams	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**
A	0.00%	0.00%	0.00%	22.54%	22.51%	30.38%	4.74%	4.83%	4.88%	5.05%	5.07%	0.00%
B	0.00%	0.00%	0.00%	28.74%	28.47%	0.00%	6.53%	6.69%	6.61%	6.70%	6.72%	9.54%
C	0.00%	0.00%	0.00%	20.46%	20.49%	26.40%	4.82%	4.89%	4.94%	5.03%	5.17%	7.79%
D	22.54%	28.74%	20.46%	0.00%	0.00%	0.00%	7.13%	7.12%	7.16%	0.00%	0.00%	6.84%
E	22.51%	28.47%	20.49%	0.00%	0.00%	0.00%	4.73%	4.93%	4.77%	4.72%	4.83%	4.55%
F	30.38%	0.00%	26.40%	0.00%	0.00%	0.00%	7.05%	7.17%	7.30%	7.23%	7.23%	7.23%
G	4.74%	6.53%	4.82%	7.13%	4.73%	7.05%	0.00%	12.34%	12.21%	16.00%	15.67%	8.78%
H	4.83%	6.69%	4.89%	7.12%	4.93%	7.17%	12.34%	0.00%	0.00%	19.88%	20.10%	12.06%
I	4.88%	6.61%	4.94%	7.16%	4.77%	7.30%	12.21%	0.00%	0.00%	20.04%	19.73%	12.35%
J	5.05%	6.70%	5.03%	0.00%	4.72%	7.23%	16.00%	19.88%	20.04%	0.00%	0.00%	15.37%
K	5.07%	6.72%	5.17%	0.00%	4.83%	7.23%	15.67%	20.10%	19.73%	0.00%	0.00%	15.48%
L	0.00%	9.54%	7.79%	6.84%	4.55%	7.23%	8.78%	12.06%	12.35%	15.37%	15.48%	0.00%

Looking at the table, we can also say that the MC method was in line with RA, in terms of assigning higher probabilities to more frequent games and lower probabilities to less frequent games. The less frequent matches (between teams from Pot 1 and Pot 3 and between teams from Pot 2 and Pot 3) have probabilities ranging from 4.55% to 9.54%. The games between teams from Pot 3 have values mostly between 15% and 21% and the games between teams from Pot 1 and Pot 2 have probabilities above 20% and up to 30%, following the same logic as the RA estimated probability distribution. Once again, the overall pattern of MC is aligned with the RA pattern, having slight variations. For example, the MC method assigned a probability of 30.38% to the match between team A and team F, the highest in its probability matrix, unlike the RA method which estimated the match between team B and team D as the highest probability, with a probability of 30.73%. For further analysis on the differences between the RA and MC methods on this case, we calculate the differences on their probability estimations.

Table 4.11 shows the difference matrix for Case 2 draw that quantifies the variations between the RA and MC probability estimations. The differences span between -3.40% and 3.20%. For example, the probability of Team A facing Team D shows a slight increase in the RA method (24.88%) compared to the MC method (22.54%), resulting in a difference of 2.34%. Similarly, the RA method slightly underestimates the probability of Team E facing Team K (5.61%) compared to the MC method (4.83%), with a difference of 0.78%. It is not possible to associate the biggest discrepancies with a particular group of games, as those values are widely distributed throughout the matrix. Both in the most frequent games (Pot 1 vs Pot 2 and Pot 3 vs Pot 3) and in the least frequent games (Pot 1 vs Pot 3 and Pot 2 vs Pot 3), there are values with differences of around |3%| and values with differences smaller than |0.5%|. In this case, the difference matrix between the RA and MC method did not show any values that would suggest that either method is making a significant error in processing the draw or estimating probabilities that are very far

Table 4.11: Difference Matrix (RA-MC) for case 2 alternated draw.

Teams	A	B	C	D	E	F	G	H	I	J	K	L
A	0.00%	0.00%	0.00%	2.34%	-0.56%	-3.06%	-0.35%	0.29%	0.24%	0.56%	0.54%	0.00%
B	0.00%	0.00%	0.00%	1.99%	-3.10%	0.00%	-0.19%	0.63%	0.71%	1.59%	1.57%	-3.20%
C	0.00%	0.00%	0.00%	3.20%	0.24%	-1.03%	-0.43%	0.23%	0.18%	0.58%	0.44%	-3.40%
D	2.34%	1.99%	3.20%	0.00%	0.00%	0.00%	-2.50%	-1.51%	-1.55%	0.00%	0.00%	-1.96%
E	-0.56%	-3.10%	0.24%	0.00%	0.00%	0.00%	-0.10%	0.68%	0.84%	0.89%	0.78%	0.33%
F	-3.06%	0.00%	-1.03%	0.00%	0.00%	0.00%	-0.22%	1.12%	0.99%	1.06%	1.06%	0.09%
G	-0.35%	-0.19%	-0.43%	-2.50%	-0.10%	-0.22%	0.00%	1.07%	1.20%	-0.88%	-0.55%	2.93%
H	0.29%	0.63%	0.23%	-1.51%	0.68%	1.12%	1.07%	0.00%	0.00%	-2.20%	-2.42%	2.09%
I	0.24%	0.71%	0.18%	-1.55%	0.84%	0.99%	1.20%	0.00%	0.00%	-2.36%	-2.05%	1.80%
J	0.56%	1.59%	0.58%	0.00%	0.89%	1.06%	-0.88%	-2.20%	-2.36%	0.00%	0.00%	0.73%
K	0.54%	1.57%	0.44%	0.00%	0.78%	1.06%	-0.55%	-2.42%	-2.05%	0.00%	0.00%	0.62%
L	0.00%	-3.20%	-3.40%	-1.96%	0.33%	0.09%	2.93%	2.09%	1.80%	0.73%	0.62%	0.00%

from what the real probability would be for any of the games. However, although in general terms the two methods are aligned in estimating the probabilities for this draw, the values of the discrepancies between the probability matrices have increased compared to the previous case, and it is not possible to say with certainty that this difference is associated exclusively with the stochastic nature of the MC method.

4.3 Result Analysis

The results presented through this chapter provide a comprehensive analysis of the performance and behavior of the RA and MC methods in estimating match-up probabilities for knockout tournament draws. One of the key findings is the significant difference in performance between the two methods implemented. The RA method exhibits substantially longer computation times as the number of outcomes increases, making it impractical for larger draws. RA's performance showed a significant drop when faced with alternated draws of 2 pots with more than 6 teams in each pot, toking approximately 40 minutes to complete a 7-team draw, whereas the MC method remained efficient, handling 100,000 simulations in just over 6 seconds. This discrepancy in performance, evidenced the scalability issue of the RA method, attributed to the exponential growth in possible outcomes with the addition of more teams. This happens because the RA method calculates all of the draw solutions, having no practical way around this problem using constraint programming.

The comparative testing using Euclidean Distance and Cosine Similarity metrics revealed a strong similarity between the MC and RA methods. The MC method consistently showed lower Euclidean Distance and higher Cosine Similarity values compared to RA when measured against the LTP method. This indicates that the probability distributions generated by MC and RA are closely aligned. The LTP method, diverged more significantly from RA, rather than MC, suggesting that LTP and MC share a closer probabilistic framework, probably more capable of capturing subtle details inherent to the draw process.

With further analysis through specific case studies, we intended to practically interpret what those similarity metrics meant in concrete draw scenarios probabilities. In the first case, for the UCL 22/23 R-16 draw initial state, the LTP and MC probability matrices exhibited a similar pattern, with minor discrepancies due to MC stochastic nature. Despite these differences, both methods provided consistent and reliable predictions, interpreted as the same probability distribution. For a manually provoked scenario within the same draw, where the restrictive environment significantly reduced the outcomes of the draw, the 3 methods converged on similar probability distributions, with the RA and LTP providing the exact same probability matrix, further suggesting their accuracy in restricted environments.

The second case was a more complex scenario to test the robustness and flexibility of the RA and MC methods. The RA and MC methods both respected the predefined restrictions and provided consistent probability distributions. However, the difference matrix (Table 4.11) revealed slightly larger discrepancies between the RA and MC methods compared to Case 1. These differences, while still low, highlighted the impact of increased complexity and diversity in draw configurations on the probabilistic estimations, emphasizing the differences between the two algorithms.

It is known that the MC method's stochastic nature introduces variations, but this bigger increase in differences might suggest a bit more than that. We have seen that for larger restricted draws, such as the UCL R-16 draw, the MC method tended to converge more to the specific system designed exclusively for that draw (LTP system), than the RA method. If the LTP system is considered as the baseline for that draw scenario, the MC method seemed to be more accurate than the RA in terms of probability estimation in the initial stages of the draw. This is probably happening as the RA method considers all the possible outcomes with the same weight when calculating the probabilities, by dividing each match-up by the total number of different outcomes. In reality, some outcomes happen more times than others and possibly the MC captures better those nuances in the probabilities by generating each sample through a draw simulation.

However, it is not possible to say that any method is wrong, as the overall results followed the same patterns, in terms of probability distribution and remained within acceptable limits, providing confidence for both methods.

On the other hand, these findings suggest that the MC method, with its superior computational efficiency and close alignment with both LTP and RA in probability estimations, is well-suited for practical applications in sports draw simulations. Its ability to handle multiple types of draws, differentiates it from the LTP system and the capability to perform a large number of simulations quickly while providing very similar results compared to RA, indicates its potential to provide very satisfactory results for real-time applications in sports tournaments or even other scenarios involving complex draw processes. These results not only showed that the proposed methods could be implemented for estimating match-up probabilities in knockout draws, but revealed the potential for further exploration of probabilistic methods in other types of sports draws and other areas that involve complex probabilistic processes.

Chapter 5

Conclusion

This section synthesizes the research findings in Section 5.1, highlights the contributions made by this study in Section 5.2, and discusses future work possibilities in Section 5.3.

5.1 Research Questions and Hypotheses demonstration

The primary objective of this thesis was to address the following research questions and hypotheses:

- **RQ1:** What are the limitations of mathematical methods in obtaining real match-up probabilities for sports draws with seeding rules and restrictions?
- **RQ2:** Can parameterization processes in simulation-based methods be used to generate a wide variety of different draws?
- **RQ3:** Can simulation-based and heuristics methods, such as constraint programming techniques, Monte Carlo Simulation, and Markov chain, be utilized to estimate realistic probabilities for sports draws with seeding rules and restrictions?

Correspondingly, the hypotheses were:

- **H1:** Simulation-based methods are alternative methods that can reliably help infer match-up probabilities for sports draws.
- **H2:** It is possible to design a simulator that can accurately reproduce the draw process for various types of knockout tournaments, adhering to specific rules and restrictions.

Throughout the course of this dissertation, we have attempted to answer the research questions initially proposed and to verify the hypotheses raised. With regard to RQ1, it was noted in Chapter 2 (State of the Art) that several articles have shown that it is not trivial to calculate probabilities in various types of sports draws, and have even emphasized the ineffectiveness of traditional mathematical methods when introducing restrictions into these processes. In regard to RQ2, Chapter 3

(Methodological Approach) detailed the approach adopted for creating draws for knockout tournaments, based on a series of configurations and parameters. This parameterization process allowed the generation of several types of draws, with different combinations of parameters.

Regarding RQ3, Chapter 4 (Empirical Studies) evaluated the possibility and demonstrated that Monte Carlo simulations and other heuristic methods are indeed effective in estimating realistic match-up probabilities for various types of knockout tournaments with different rules and restrictions. With this, we also confirm both hypotheses, as simulation-based methods have shown to be alternative methods that can estimate reliable probabilities for sports draws and perform decently in scenarios with higher complexity of seeding rules and restrictions, providing a robust framework for sports draws simulations.

5.2 Main Contributions

The main contributions of this thesis can be categorized into scientific, applicational, and technological domains:

5.2.1 Scientific Contributions

- A systematic literature review identified gaps in existing research and validated the use of Monte Carlo methods for probability estimation and analysis in sports draw processes.
- This research explored and evaluated various probabilistic methods and simulation techniques, such as Monte Carlo sampling, Constraint Programming and Markov Chain, applied to sports draws simulation environment, providing a detailed analysis on their performance and behavior under relevant scenarios.

5.2.2 Applicational Contributions

- A system that can be utilized by competition organizers to avoid manual errors and promote fairness in their competitions. By automating the draw process and ensuring compliance with seeding rules and restrictions, the tool helps in maintaining the integrity of competitions and preventing potential cheating.

5.2.3 Technological Contributions

- The development of a system capable of creating and replicating customizable draws and estimating reliable match-up probabilities, for knockout tournaments.

5.3 Future Work

There are several paths for future research and development based on the work developed and findings of this dissertation:

- **Probability estimation methods:** Exploring new ways to perform the probability estimation of this type of draws or even combining different probabilistic methods to enhance performance and accuracy. For instance, hybrid models that integrate Monte Carlo simulations with machine learning techniques could provide even more precise estimations.

- **Expansion to Different Types of Draws:** The current system is only focused on knockout tournament draws. Future work could expand this to include group-stage draws or round-robin tournaments, and implement a solution with the explored methods for estimating probabilities of other draw formats.

- **Adaptation to New Formats:** With sports organizations constantly evolving their tournament structures, the system should be adaptable to new formats, by providing the possibility to create different draw's logics and formats from scratch.

In conclusion, this dissertation has laid a solid foundation for the application of probabilistic methods and techniques based on computer simulation in sports draws. This work evaluated and confirmed the viability of methods such as Monte Carlo simulation or Constraint Programming for a step-by-step simulation of knockout tournament draws, with estimation of match-up probabilities. The MC approach showed to be the most promising, combining efficiency with a reliable behavior when estimating match-up probabilities. This work addressed current challenges but leaves a clear door open for new possibilities and future improvements and research.

References

[1] Jack Flintham. Champions League 2023 group stage draw details and live stream, August 2023. Section: Football News.

[2] Philip Scarf, Muhammad Mat Yusof, and Mark Bilbao. A numerical study of designs for sporting contests. *European Journal of Operational Research*, 198(1):190–198, October 2009.

[3] László Csató. A simulation comparison of tournament designs for the World Men's Handball Championships. *International Transactions in Operational Research*, 28(5):2377–2401, September 2021. Publisher: John Wiley & Sons, Ltd.

[4] Dmitry Dagaev and Vladimir Yu. Rudyak. Seeding the UEFA Champions League participants: evaluation of the reforms. 15(2):129–140, 2019.

[5] Stefan Klößner and Martin Becker. Odd odds: The UEFA Champions League Round of 16 draw. *Journal of Quantitative Analysis in Sports*, 9(3):249–270, September 2013. Publisher: De Gruyter.

[6] G. Nicolis. *Introduction to Nonlinear Science*. Cambridge University Press, June 1995. Google-Books-ID: PFZWrlU33KMC.

[7] Jesús Isaac Vázquez-Serrano, Rodrigo E. Peimbert-García, and Leopoldo Eduardo Cárdenas-Barrón. Discrete-event simulation modeling in healthcare: A comprehensive review. *International Journal of Environmental Research and Public Health*, 18(22), 2021.

[8] Xiange Zhang. Application of discrete event simulation in health care: a systematic review. *BMC health services research*, 18(1):687, September 2018.

[9] Reinaldo Padilha França, Ana Carolina Borges Monteiro, Vania V. Estrela, and Navid Razmjooy. Using Metaheuristics in Discrete-Event Simulation. In Navid Razmjooy, Mohsen Ashourian, and Zahra Foroozandeh, editors, *Metaheuristics and Optimization in Computer and Electrical Engineering*, pages 275–292. Springer International Publishing, Cham, 2021.

[10] Scopus - Document details - Monte Carlo methods.

[11] J. Krčadinac, E.M. Maruševec, L. Jerković, I. Kovač, J. Zloić, A. šarčević, and M. Vranić. Modeling Tennis Matches Using Monte Carlo Simulations Incorporating Dynamic Parameters. In *2023 46th MIPRO ICT and Electronics Convention (MIPRO)*, pages 1281–1286, May 2023. ISSN: 2623-8764.

[12] Y. Liu, C.M. Harvey, F.E. Hamlyn, and C. Liu. Bayesian estimation and reconstruction of marine surface contaminant dispersion. *Science of the Total Environment*, 907, 2024.

[13] R.H. Koning, M. Koolhaas, G. Renes, and G. Ridder. A simulation model for football championships. *European Journal of Operational Research*, 148(2):268–276, 2003.

[14] C. F. Silva, E. S. Garcia, and E. Saliby. Soccer championship analysis using Monte Carlo simulation. In *Proceedings of the Winter Simulation Conference*, volume 2, pages 2011–2016 vol.2, December 2002. Journal Abbreviation: Proceedings of the Winter Simulation Conference.

[15] I. Díaz-Emparanza and V. Núñez-Antón. On the use of simulation methods to compute probabilities: Application to the first division Spanish soccer league. *SORT*, 34(2):181–200, 2010.

[16] Tushar Aggarwal. Mastering Monte Carlo Simulation for Data cience, April 2024.

[17] W. R. Gilks, S. Richardson, and David Spiegelhalter. *Markov Chain Monte Carlo in Practice*. CRC Press, December 1995. Google-Books-ID: T2G1DwAAQBAJ.

[18] Yunyu Zhang. The value of Monte Carlo model-based variance reduction technology in the pricing of financial derivatives. *PLoS ONE*, 15(2):e0229737, February 2020.

[19] H.W. Coleman and W.G. Steele. Engineering application of experimental uncertainty analysis. *AIAA Journal*, 33(10):1888–1896, 1995.

[20] Ali Naseri, Mehdi Jamei, Iman Ahmadianfar, and Mohammad Behbahani. Nanofluids thermal conductivity prediction applying a novel hybrid data-driven model validated using monte carlo-based sensitivity analysis. *Eng. with Comput.*, 38(Suppl 1):815–839, apr 2022.

[21] John Geweke and Michael Keane. Chapter 56 - Computationally Intensive Methods for Integration in Econometrics**The authors gratefully acknowledge financial support from National Science Foundation grants SBR-9511280, SBR-9731037, SES-9814342, and SES-9819444. In James J. Heckman and Edward Leamer, editors, *Handbook of Econometrics*, volume 5, pages 3463–3568. Elsevier, January 2001.

[22] The Use of the Monte Carlo Method in Sensitivity Analysis and Its Advantages, November 2020.

[23] Averill M. Law and W. David Kelton. *Simulation modeling and analysis*. McGraw-Hill series in industrial engineering and management science. McGraw-Hill, Inc, New York St. Louis San Francisco Auckland, second editon edition, 1991.

[24] Quantitative validation of Monte Carlo SPECT simulation: application to a Mediso AnyScan GATE simulation. 10.

[25] Pierre Schaus. *Integration of Constraint Programming, Artificial Intelligence, and Operations Research: 19th International Conference, CPAIOR 2022, Los Angeles, CA, USA, June 20-23, 2022, Proceedings*. Springer Nature, June 2022. Google-Books-ID: 34J0EAAAQBAJ.

[26] ILOG CPLEX Optimization Studio 12.10.0.

[27] GECODE - An open, free, efficient constraint solving toolkit.

[28] OR-Tools.

[29] The MiniZinc Handbook — The MiniZinc Handbook 2.6.4.

[30] Francesca Rossi, Peter van Beek, and Toby Walsh. Chapter 4 constraint programming. In Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter, editors, *Handbook of Knowledge Representation*, volume 3 of *Foundations of Artificial Intelligence*, pages 181–211. Elsevier, 2008.

[31] Dayue Qin, Yanzhu Chen, and Ying Li. Error statistics and scalability of quantum error mitigation formulas. *npj Quantum Information*, 9(1):1–14, April 2023. Number: 1 Publisher: Nature Publishing Group.

[32] Bradley Efron. Prediction, Estimation, and Attribution. *Journal of the American Statistical Association*, 115(530):636–655, April 2020.

[33] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman & Hall/CRC, 2015.

[34] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, and Jonathan Taylor. Statistical Learning. In Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, and Jonathan Taylor, editors, *An Introduction to Statistical Learning: with Applications in Python*, pages 15–67. Springer International Publishing, Cham, 2023.

[35] Weibin Lan and Jingying Chang. Statistical Model of College Students' Mental Health Based on the Law of Large Numbers. *Applied Mathematics and Nonlinear Sciences*, 8(1):2975–2984, January 2023.

[36] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[37] Robert Tibshirani. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, January 1996.

[38] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, New York, NY, 2009.

[39] Bradley Efron and Trevor Hastie. *Computer Age Statistical Inference, Student Edition: Algorithms, Evidence, and Data Science*. Institute of Mathematical Statistics Monographs. Cambridge University Press, 2021.

[40] Shijie Wang, Minsuk Shin, and Ray Bai. Fast bootstrapping nonparametric maximum likelihood for latent mixture models, 2024.

[41] Bryan Cai, Fabio Pellegrini, Menglan Pang, Carl Moor, Changyu Shen, Vivek Charu, and Lu Tian. Bootstrapping the cross-validation estimate. 07 2023.

[42] Minh Thu Bui, Cornelis J. Potgieter, and Akihito Kamata. Penalized likelihood methods for modeling count data. *Journal of Applied Statistics*, 50(15):3157–3176, 2023.

[43] Mathematics Stack Exchange. Geometric interpretation of euclidean distance between matrices. Accessed: 2024-06-17.

[44] Scikit-Learn. Pairwise euclidean distances. Accessed: 2024-06-17.

[45] Arjun Prakash. Understanding Cosine Similarity: A key concept in data science, September 2023.

[46] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. *Noise reduction in speech processing*, pages 1–4, 2009.

[47] Fernando Perez-Cruz. Kullback-leibler divergence estimation of continuous distributions. In *2008 IEEE International Symposium on Information Theory*, pages 1666–1670, 2008.

[48] Richard Neapolitan. *Learning Bayesian Networks*. January 2003.

[49] Kevin P. Murphy. *Machine learning: a probabilistic perspective*. Adaptive computation and machine learning series. MIT Press, Cambridge, MA, 2012.

[50] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, July 2009.

[51] Pavel Loskot. Bayesian methods and monte carlo simulations. In Ali Soofastaei, editor, *Numerical Simulation*, chapter 5. IntechOpen, Rijeka, 2022.

[52] Amani A. Alahmadi, Jennifer A. Flegg, Davis G. Cochrane, Christopher C. Drovandi, and Jonathan M. Keith. A comparison of approximate versus exact techniques for Bayesian parameter inference in nonlinear ordinary differential equation models. *Royal Society Open Science*, 7(3):191315, March 2020.

[53] Shawn X. Ma, Ali H. Dhanaliwala, Jeffrey D. Rudie, Andreas M. Rauschecker, Douglas Roberts-Wolfe, Peter Haddawy, and Charles E. Kahn. Bayesian networks in radiology. *Radiology: Artificial Intelligence*, 5(6):e210187, 2023.

[54] Train Bayesian Neural Network - MATLAB & Simulink.

[55] Lin Wei, Hanyue Yu, and Bin Li. Energy financial risk early warning model based on bayesian network. *Energy Reports*, 9:2300–2309, 2023.

[56] Snehal Prabhudesai, Jeremiah Hauth, Dingkun Guo, Arvind Rao, Nikola Banovic, and Xun Huan. Lowering the computational barrier: Partially Bayesian neural networks for transparency in medical imaging AI. *Frontiers in Computer Science*, 5, February 2023. Publisher: Frontiers.

[57] Aidan C. Daly, Jonathan Cooper, David J. Gavaghan, and Chris Holmes. Comparing two sequential Monte Carlo samplers for exact and approximate Bayesian inference on biological models. *Journal of The Royal Society Interface*, 14(134):20170340, September 2017. Publisher: Royal Society.

[58] Joshua J. Bon, Adam Bretherton, Katie Buchhorn, Susanna Cramb, Christopher Drovandi, Conor Hassan, Adrianne L. Jenner, Helen J. Mayfield, James M. McGree, Kerrie Mengersen, Aiden Price, Robert Salomone, Edgar Santos-Fernandez, Julie Vercelloni, and Xiaoyu Wang. Being Bayesian in the 2020s: opportunities and challenges in the practice of modern applied Bayesian statistics. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 381(2247):20220156.

[59] S.R. Clarke and D. Dyte. Using official ratings to simulate major tennis tournaments. *International Transactions in Operational Research*, 7(6):585–594, November 2000. Publisher: John Wiley & Sons, Ltd.

[60] M. Chater, L. Arrondel, J.-P. Gayant, and J.-F. Laslier. Fixing match-fixing: Optimal schedules to promote competitiveness. *European Journal of Operational Research*, 294(2):673–683, 2021.

[61] L. Csató, R. Molontay, and J. Pintér. Tournament schedules and incentives in a double round-robin tournament with four teams. *International Transactions in Operational Research*, 2023.

[62] László Csató. The effects of draw restrictions on knockout tournaments. 18(4):227–239, 2022.

[63] László Csató. Quantifying incentive (in)compatibility: A case study from sports. *European Journal of Operational Research*, 302(2):717–726, October 2022.

[64] CL R16 Draw Probabilities.

[65] Joao Bioco and Álvaro Rocha. Web application for management of scientific conferences. pages 765–774, 04 2019.

Appendix A

Simulator Interfaces

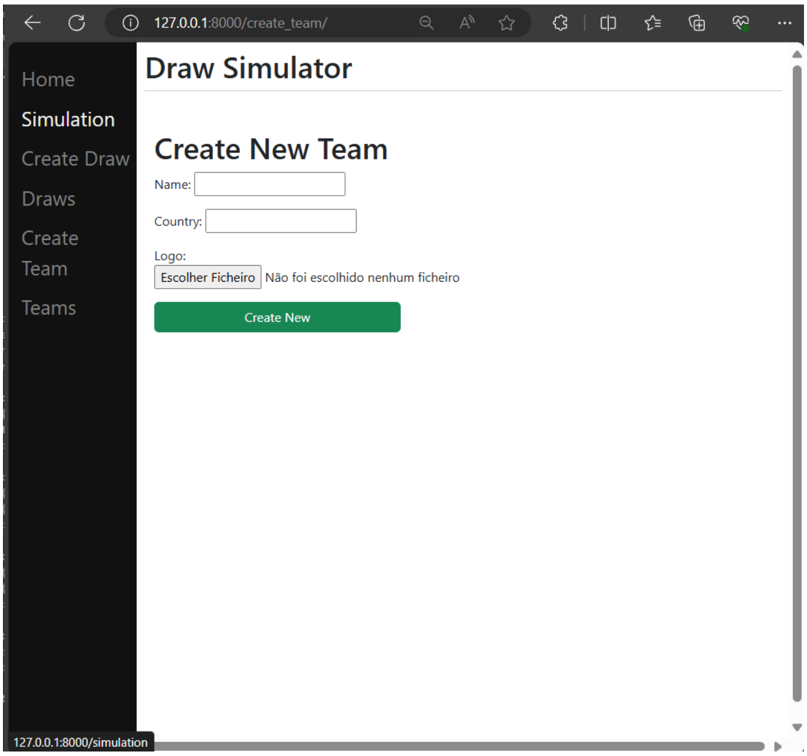

Figure A.1: Create team form.

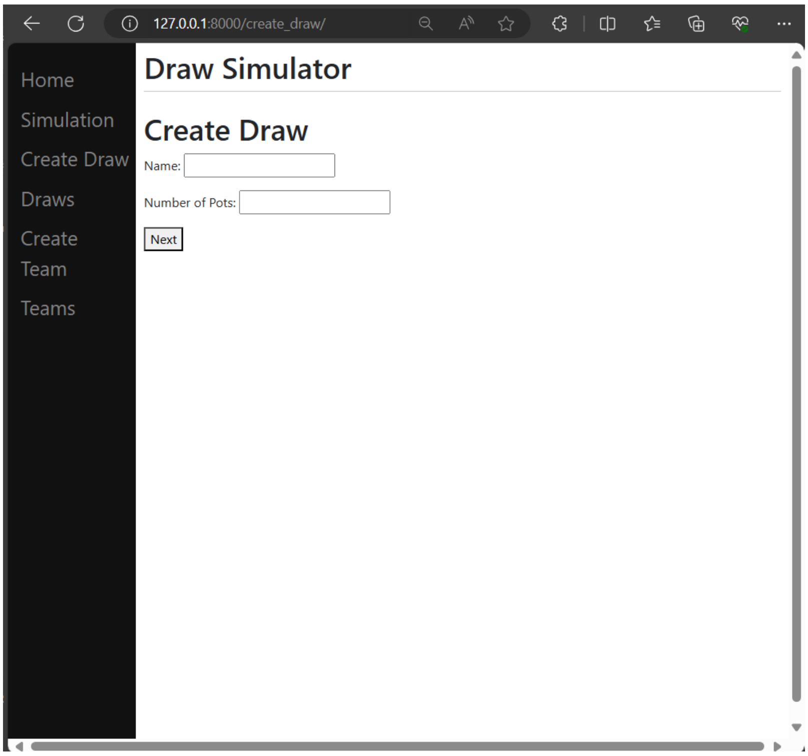

Figure A.2: Create draw first step.

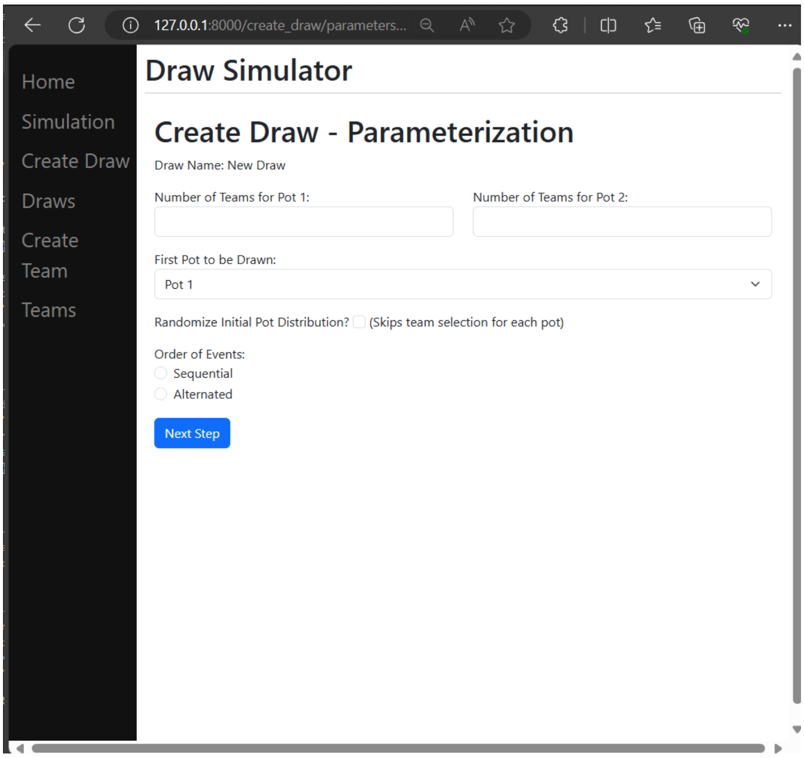

Figure A.3: Draw second step.

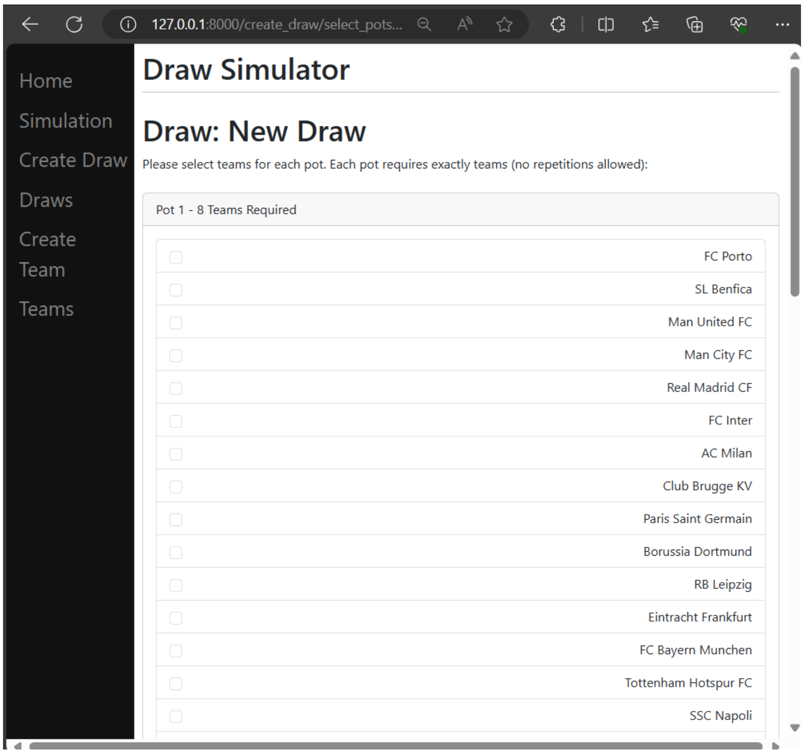

Figure A.4: Create draw third step.

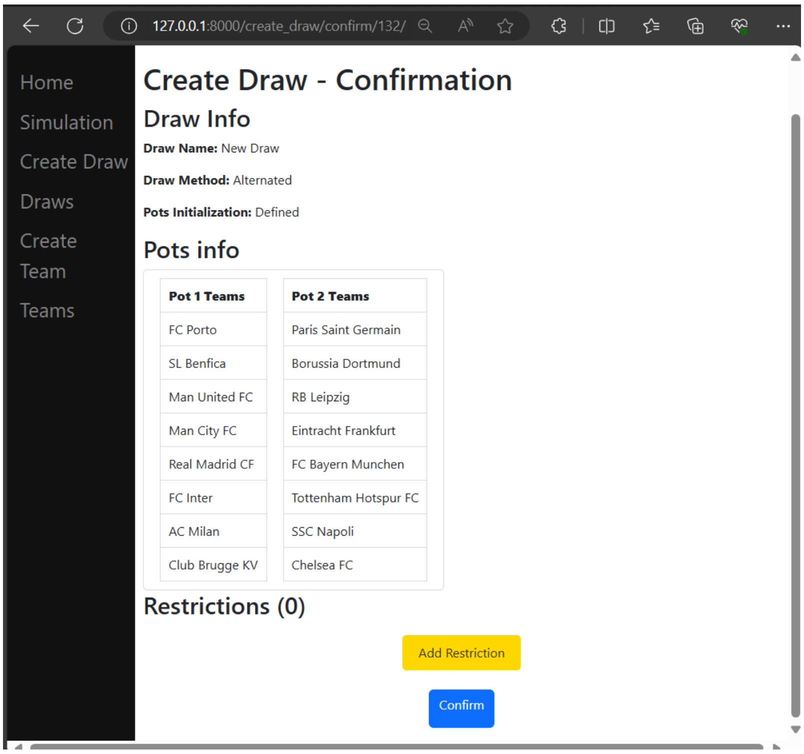

Figure A.5: Draw Confirmation step.

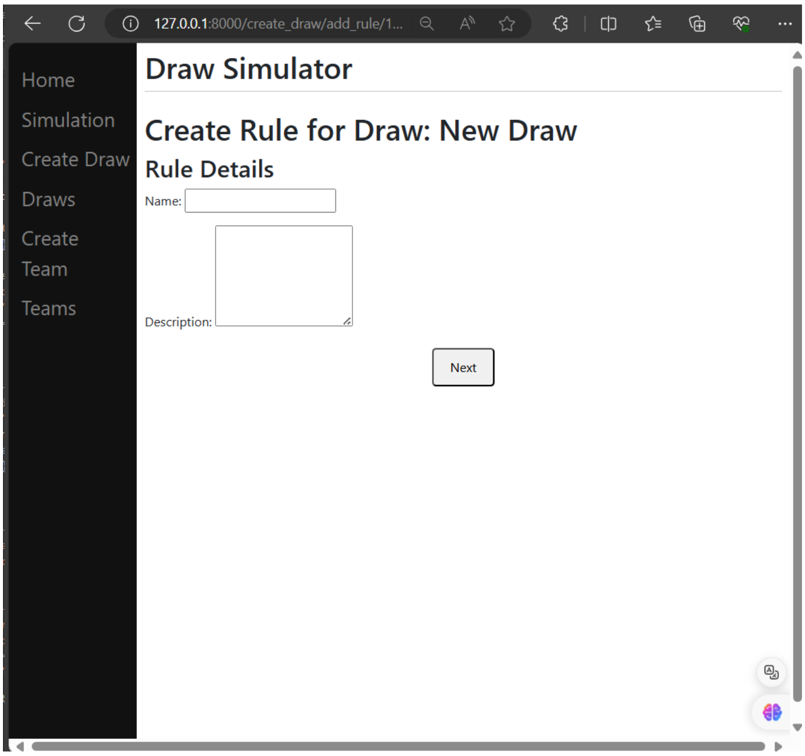

Figure A.6: Create Restriction first step.

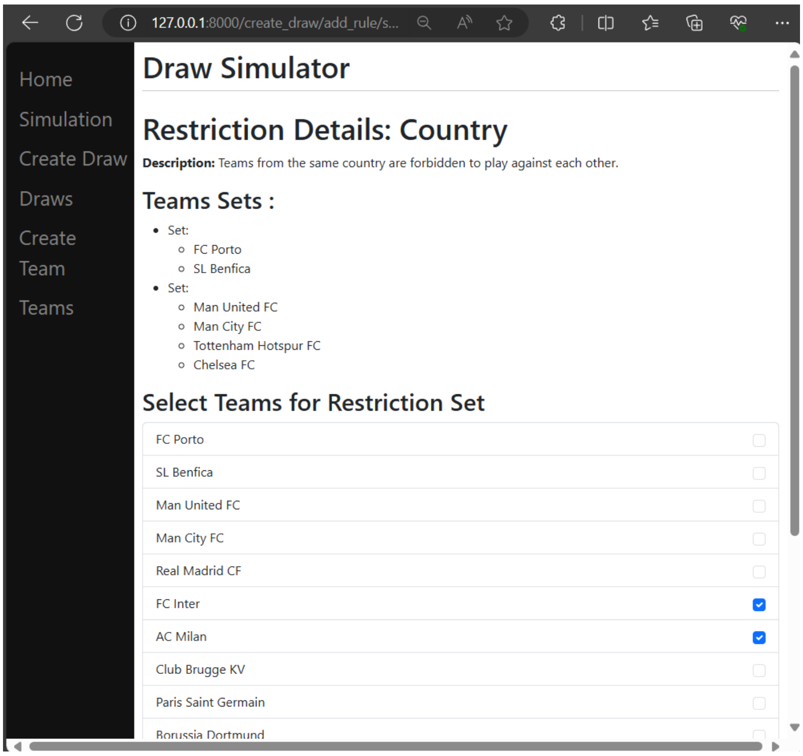

Figure A.7: Adding sets of teams to a restriction.

[table]xcolor