University of Padova

Department of Mathematics

Master Thesis in Data Science

Optimal Solution Bounds for the Random Assignment problem

Supervisor
Professor Marco Formentin
University of Padova

Master Candidate
Sofia Bazzan

Academic Year
2023-2024

Abstract

The Random Assignment Problem is a significant challenge in optimization theory that involves allocating a set of n jobs to an equal number of machines to minimize total cost. The costs are assigned to each machine to complete each job and in this formulation are considered as random variables with an uniform distribution between 0 and 1. Despite its apparent simplicity, this problem has captivated mathematicians and researchers for decades. A key difficulty lies in estimating the expected value of the cost associated with the optimal solution, denoted as $\mathbb{E}[A_n]$. Early attempts to bound this expected value resulted in orders of $\log n$ until Walkup's result in 1979, where he demonstrated that $\mathbb{E}[A_n]$ is bounded independently of n, establishing an unexpected constant bound of $3 + o(1)$. This result, that was grounded in graph theory and linear programming, laid the foundation for further advancements in understanding the problem and subsequent research led to the discovery of bounds for $\mathbb{E}[A_n]$ using diverse mathematical techniques. However, the most significant breakthrough occurred in 1987 when Parisi and Mézard applied concepts from statistical physics to derive a remarkable estimation for $\mathbb{E}[A_n]$. Their work, based on the replica method and spin glass theory, suggested that as n tends to infinity, $\mathbb{E}[A_n]$ converges to $\frac{\pi^2}{6}$, providing a surprising result that also came from a different field of research. The only aspect left to address was to obtain a rigorous mathematical proof instead of an estimate and the first formal demonstration of this result was presented years later by Aldous in his article "The $\zeta(2)$ Limit in the Random Assignment Problem" [1], which will be the main focus of this thesis. After the introductory chapter that aims to synthesize the problem's evolution, Chapter 2 will outline the main steps of Aldous's proof for a comprehensive understanding, and Chapter 3 will examine the details necessary to complete the demonstration. To emphasize the practical relevance of the random assignment problem, Chapter 4 will explore one of its possible applications. Specifically, it will focus on the article: "Privacy preserving wireless communication using bipartite matching in social big data" [2] which explain how to employ a bipartite matching method to address a privacy protection issue showing the practical impact of theoretical advancements in combinatorial optimization.

Contents

Listing of figures

1
Introduction

1.1 The Random Assignment Problem

The problem of interest of this thesis is the assignment problem that can be explained as follows [1]. Let's consider the need to allocate n jobs to n machines, each numbered from 1 to n. Suppose each job has a specific cost depending on the machine that performs it, denoted by $c(i, j)$, where i represents the job index and j denotes the machine index. The objective is to assign exactly one job to each machine with the goal of minimizing the total cost. Consequently, the basic input can be depicted by an $n \times n$ matrix:

$$\begin{bmatrix} c(1,1) & \dots & c(1,n) \\ \vdots & \ddots & \vdots \\ c(n,1) & \dots & c(n,n) \end{bmatrix}$$

Furthermore, this cost matrix can help to better visualize the assignment problem. In this formulation, each row represents a job or task, and each column represents a machine or resource. The objective is to select one machine for each job, ensuring that each column (machine) is chosen exactly once for each row (job), with the aim of minimizing the total sum of the costs associated with all job-machine assignments, where each cost is represented by an entry in the matrix.

Because there are n jobs and n machines both numbered from 1 to n the solution to this problem lies in finding a permutation π that minimizes the total cost [3]:

$$A_n = \min_{\pi} \sum_{i=1}^{n} c\left(i, \pi(i)\right).$$

In particular, this work will concentrate on solving the Random Assignment Problem where the costs associated with assigning jobs to machines are random variables. In this scenario, the simplest stochastic model assumes that the costs $c(i,j)$ are independent and identically distributed according to a uniform distribution between 0 and 1. Therefore, we have $c(i,j) \sim \mathit{Unif}\left([0,1]\right), \forall i, j = 1, ..., n$.

1.1.1 Graph formulation

Another formulation of the problem is based on graph theory, specifically utilizing the concepts of bipartite graphs and matchings [4]:

Definition 1. *A graph is bipartite if its set of vertices can be partitioned into two sets A and B such that every edge of the graph has one endpoint in A and the other in B.*

Definition 2. *A set of edges in a graph is called a matching if no two edges share a common vertex. With respect to a given matching, a vertex is said to be matched or covered if there is an edge in the matching incident to it. Conversely, if a vertex is not covered, it is said to be unmatched or uncovered.*

These concepts are relevant to the random assignment problem. By considering a complete bipartite, weighted graph with $2N$ vertices, where N belong to one partition and N to the other (representing jobs and machines, respectively), the random assignment problem can be reformulated as finding a matching in the graph that covers all vertices with minimum total cost.

1.2 History of the Matching Problem

1.2.1 Advancements in Graph Theory Fundamentals for the Assignment Problem

One of the earliest instances of the assignment problem [5] can be traced back to 1784 in Monge work [6]. Monge introduced a problem similar to what is now known as the assignment problem, but within a distinct context and that initially appears continuous in nature: the transportation problem. Motivated by the challenge of transporting earth, which he analogized

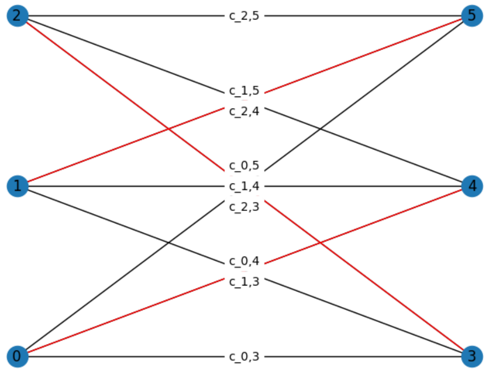


Figure 1.1: An example with $N = 3$. The random assignment problem asks, given that $c_{i,j}$ is a random cost between 0 and 1 for each pair of vertices *i,j* belonging to two different partitions V_1 and V_2, to find the subset of edges with minimum total cost that connects each node of V_1 to exactly one of V_2. In this picture V_1 and V_2 are the nodes on the left and on the right respectively and a possible matching is highlighted in red.

to the combinatorial task of moving molecules, Monge's scenario involved two equal-acreage regions: one filled with earth and the other empty. The primary objective was to efficiently relocate the earth from the first area to the second while minimizing the total transportation distance, computed as the cumulative distance each molecule needed to traverse. This variation of the assignment problem later became formally recognized as the transportation problem, a fundamental concept in optimization and logistics where the objective is to determine the most efficient way to transport goods or resources from suppliers to consumers.

This early formulation was important not just by itself but because it presented a specific case, setting the stage for later developments in graph theory. Notably, significant progress in this field emerged between 1912 and 1931 through the works of Frobenius and König. The former, focusing on matrix theory, and the latter, emphasizing graph theory, provided foundational contributions to matching theory from distinct perspectives.

The pivotal advancements in graph theory began with Frobenius's work in 1912, where he introduced a theorem concerning the determinant of a matrix [7]. This theorem laid essential groundwork for subsequent developments in matching theory and marked a significant milestone in the evolution of mathematical concepts related to matrices and determinants.

Theorem 1. *Let the elements of a determinant of degree n be n^2 independent variables. One sets some of them equal to zero, but such that the determinant does not vanish identically. Then it remains an irreducible function, except when for some value $m < n$ all elements vanish that have m rows in common with $n \times m$ columns.*

In 1915, König [8] recognized that Frobenius's findings could be reformulated in terms of

bipartite graphs by introducing the concept of representing a graph with its adjacency matrix. König proceeded to provide a proof of Frobenius's result, demonstrating a significant connection between matrix theory and graph theory.
During the *Congrés de Philosophie mathématique* in Paris in 1914, König presented the following:

Theorem 2. *Each regular bipartite graph has a perfect matching.*

Where a perfect matching is defined [4] as:

Definition 3. *A matching is perfect if it covers all the vertex of the graph.*

König also derived the following consequence:

Corollary 1. *The edge set of any regular bipartite graph can be decomposed into perfect matchings.*

In order to give an elementary proof of his previous result described above, Frobenius in 1917 proved the following which now is a fundamental theorem in graph theory:

Theorem 3. *If in a determinant of the n^{th} degree all elements vanish that $p(\leq n)$ rows have in common with $n-p+1$ columns, then all members of the expanded determinant vanish.
If all members of a determinant of degree n vanish, then all elements vanish that p rows have in common with $n-p+1$ columns for $p=1$ or $2, \ldots$ or n.*

That is saying that if $A=(a_{i,j})$ is an $n \times n$ matrix, and for each permutation π of $\{1, ..., n\}$ one has $\prod_{i=1}^{n} a_{i,j} = 0$ then for some p there exist p rows and $n-p+1$ columns of A such that their intersection is all-zero.

Frobenius's theorem identifies which bipartite graphs possess a perfect matching but a characterization of the maximum size of a matching in bipartite graph was given by König's more general theorem in 1931 [9].

Theorem 4. *In an even circuit graph, the minimal number of vertices that exhaust the edges agrees with the maximal number of edges that pairwise do not contain any common end point.*

This theorem was stating that the maximum size of a matching in a bipartite graph is equal to the minimum number of vertices needed to cover all edges.

In 1931, Egerváry, a Hungarian mathematician, made a significant advancement building upon König's work by characterizing the maximum weight of a matching in a bipartite graph. This concept, applicable to the assignment problem, led to the formulation of the following theorem.

Theorem 5. *If the elements of the matrix $(c_{i,j})$ of order n are given nonnegative integers , then under the assumption*

$$\lambda_i + \mu_j \geq c_{i,j} \, \textit{for} \, i, j = 1, \dots n, (\lambda_i, \mu_j \; \textit{non negative integers}),$$

we have:

$$\sum_{k=1}^{k=n} (\lambda_k + \mu_k) = \max \left(a_{1,\pi_1} + a_{2,\pi_2} \dots a_{n,\pi_n} \right).$$

where $\pi_1 \dots \pi_n$ *run over all possible permutation of the numbers* $1, \dots, n$.

1.2.2 First modern formulation of the Matching Problem

Egerváry's theorem and proof method laid a crucial foundation for Kuhn to develop the first comprehensive formulation and solution of the assignment problem in his work *"The Hungarian Method for the Assignment Problem"* in 1955 [10].

This work is widely acknowledged as the starting point for the practical development of solution methods and variations for the classic assignment problem. Kuhn's Hungarian method, named in honor of Egerváry's contribution, stands out as a combinatorial optimization algorithm that remains of note today for its efficiency in solving the assignment problem within polynomial time.

Kuhn's analysis of König's work on graph theory led to the realization that the matching problem for a bipartite graph on 2 sets of n vertices was exactly the same as a n by n assignment problem with all $a_{ij} = 0$ or 1. Notably König had given a combinatorial algorithm based on augmenting paths that produces optimal solutions to the matching problem. While this algorithm seemed promising it had the problem of how to reduce the general assignment problem to the $0 - 1$ special case. Reading König's works Kuhn discovered that the solution to his problem might be found in Egerváry's paper and after studying his results was able to identify a method that allow the reduction of a general assignment problem to a finite number of $0 - 1$ assignment problems.

Kuhn developed a method that enhanced Egerváry's approach for solving the problem by finding augmenting paths to achieve a perfect matching or the required sets. His technique focused on maximizing the improvement of λ and μ in Egerváry's formulation not just by one, but by the largest value possible.
The method was summarized by Flood [11] as the following sequence of steps.

1. Subtract the smallest element of the adjacency matrix A from each element of A obtaining a matrix A_1 with nonnegative elements and at least one zero.

2. Find a minimal set S_1 of lines, n_1 in number, which contain all the zero of A_1. If $n_1 = n$ there is a set of n independent zeros (where in this context independent means that they are not on the same horizontal or vertical line) and the elements of A in this n positions constitutes the required solution.

3. If $n_1 < n$, let h_1 denote the smallest element of A_1 which is not in any line of S_1. Then $h_1 > 0$. For each line in S_1 add h_1 to every element of that line; then subtract h_1 for every element of A_1. Call the new matrix A_2.

4. Repeat steps 2 and 3, using A_2 in place of A_1.

Kuhn noted that the sum of the elements of the matrix is decreased by $n(n-n_k)h_k$ in each application of step 3, so the process must terminate after a finite number of steps. Later Munkres [12] observed that the method runs in strongly polinomial time $(\mathcal{O}(n^4))$.

1.2.3 Development of the Randomized Version of the Assignment Problem and applications

Kuhn's work gave origin to the research of different methods to solve the AP and other (both linear and non-linear) application and variation of the problem in different contexts. From the numerous modifications there is also the focal point of this thesis: the random assignment problem which can be explained as a variation of the general problem where the costs are random with an uniform distribution between 0 and 1.

The origins of studying this problem [3] can be traced back to 1962 when Kurtzberg [13] initiated investigations by analyzing the execution time of heuristic methods for the assignment problem. Kurtzberg's innovative approach of introducing random costs uniformly distributed between 0 and 1 introduced a method for understanding the average time required by algorithms to solve this problem. Employing techniques like greedy matching, he derived upper bounds on $\mathbb{E}A_n$ for uniformly distributed costs inspiring further investigations into bounds for the expected value of A_n by subsequent researchers.

This work has also highlighted the significance of the random assignment problem, emphasizing its critical role in analyzing random instances of optimization problems. This analytical approach is essential for understanding the characteristics of solutions, feasible regions, and optimal values, particularly in large-scale scenarios. Operating within a probabilistic framework, where problem data conforms to a probability distribution, provides a foundation for maintaining consistency across problem instances of varying sizes. This methodology enables the evaluation of optimal values and solutions as functions of problem size, thereby advancing our comprehension of the random assignment problem, which is crucial for every application within the matching problem domain.

Furthermore, deriving upper bounds on expected optimal values for specific problems has yielded significant advancements in understanding the computational complexities involved.

Today, the analysis of random instances of the assignment problem remains highly relevant, especially given the increasing number of potential applications across various disciplines in science and engineering. These applications span fields such as chemistry, biology, physics, archaeology, electrical engineering, sports, and beyond.

To illustrate the diverse range of applications of this problem, here are some instances that can be formulated as bipartite graph matching:

- **Privacy-preserving wireless communications**: In the article [2], the authors discuss how enhanced wireless data transmissions have significantly improved service deployment, especially in social networks and big data applications. They highlight multi channel wireless communication as a key approach for disseminating information in dynamic and heterogeneous wireless networking environments. Channel Scheduling Controllers (CSCs) play a crucial role in data transmissions by facilitating real-time task scheduling. However, fixed communication scheduling struggles to meet the demands of higher-level privacy protections due to conflicts between performance and security requirements. To address this challenge, the authors propose an optimal task scheduling technique that allocates tasks on heterogeneous Nodes to enhance system resilience. They address an NP-complete problem representing the privacy protection issue of CSC systems and solve it using a bipartite matching method.

- **DNA sequencing**: In this work [14], was analized the computational complexity of a combinatorial problem arising in DNA sequencing by hybridization. This problem involves an input comprising an integer l and a set S of words of length k formed from the symbols A, C, G, T. The core challenge is to determine the existence of a word of length l that includes each word in S at least once as a subword while excluding any other subword of length k. The computational complexity of this problem has long been a subject of inquiry and remains unresolved. The authors establish that this problem is polynomial time equivalent to the exact perfect matching problem in bipartite graphs.

- **Image feature matching**: In [15] the authors address the challenging task of determining 2D image feature correspondences across a set of images. They introduce two affinity measures for image points and lines from different images, which are utilized to create unweighted and weighted bipartite graphs. Therefore the research demonstrates that the problem of image feature matching can be reformulated as a general maximum-weight bipartite matching problem.

To show how the bipartite matching can be applied to a practical problem, Chapter 4 will focus completely on explaining the first example provided: the application to privacy-preserving wireless communication as detailed in [2].

1.3 Early Solutions to the Random Assignment Problem

This seemingly simple model has been extensively studied for decades without a definitive solution being found, employing various approaches. The first results are briefly reported in the following paragraph [1].

1.3.1 Rigorous Bounds via Linear Programming

The first bounds obtained by Kurtzberg for $\mathbb{E}A_n$ were of order $\log n$ and these were not improved until 1979 when Walkup [16] showed that $\mathbb{E}A_n$ is bounded independently of n, a surprising result at the time. Specifically, he established that:

$$\mathbb{E}[A_n] \leq 3 + o(1).$$

The approach he utilized to derive this inequality is intriguing, and here is a summary of the key steps. Initially, he considered the natural strategy of focusing on assigning edges with small costs. The idea involved constructing a bipartite graph where an edge (i, j) belongs to the edge set E if and only if $c_{ij} \leq \alpha/n$. The hope was that if such a graph had a perfect matching, the cost of the assignment problem would be bounded by α. However, this method alone fell short of yielding the desired constant bound. One notable challenge was the presence of isolated vertices, with the expected number scaling as $2ne^{-\alpha}$.

In addition to that, investigation revealed that achieving a perfect matching with all edges less than α/n required setting $\alpha = \mathcal{O}(\log n)$, resulting in a bound for the expected value no better than that obtained through greedy methods.

The key idea to solve this problem involved the introduction of k-out multigraphs, which are graphs that allow connections with the same end nodes, and where each vertex has exactly k outgoing edges. Walkup developed a method for generating a random two-out bipartite multigraph by partitioning the vertex set V_i, V_j into two subsets with $|V_i| = |V_j| = n$. For each vertex v in V_i, two random elements w and w' from V_j were selected, forming the edges (v, w) and (v, w'). Similarly, for each vertex v in V_j, two random elements of w and w' from V_i were chosen, resulting in the corresponding edges.

Although this process could lead to some edges being added twice, resulting in a multigraph rather than a graph, the distinction proved inconsequential. Crucially, this approach ensured that with high probability, a perfect matching existed, addressing the challenges posed by isolated vertices. This critical insight was formalized in Walkup's theorem:

Theorem 6. *Let G be a random two-out bipartite multigraph with bipartition* V_1, V_2*, where* $|V_1| = |V_2| = n$*. The probability that G fails to contain a perfect matching is bounded by* $5/n$*.*

To establish the connection between the assignment problem and the theory of random two-out bipartite multigraphs, Walkup introduced edge weights $c_{i,j}^{(1)}$ and $c_{i,j}^{(2)}$ for each edge cost c_{ij}. These edge weights, representing outgoing and ingoing edges, were independently and identically distributed in $[0, 1]$, resulting in a total of $4n$ independent random variables. Walkup then selected the two smallest outgoing costs for each vertex in V_i and the two smallest ingoing costs for each vertex in V_j. He demonstrated that the smallest value had an expectation of

$$\mathbb{E}[c_i^{(1)}(1)] = \mathbb{E}[c_j^{(2)}(1)] = \frac{2}{n},$$

and similarly, the second smallest value had

$$\mathbb{E}[c_i^{(1)}(2)] = \mathbb{E}[c_j^{(2)}(2)] = \frac{4}{n}.$$

Furthermore, considering sets of out-edges for vertices, that for each i was denoted with O_i, Walkup showed that the expectations of the edge weights conditioned on the presence of these edges followed specific patterns, explaining the constant 3 in the final inequality. The values obtained where

$$\mathbb{E}\left(c_{ij}^{(1)}|(i,j) \in O_i\right) = \frac{1}{2}\mathbb{E}\left(c_{ij}^{(1)}|c_{ij}^{(1)} = c_i^{(1)}(1)\right) + \frac{1}{2}\mathbb{E}\left(c_{ij}^{(1)}|c_{ij}^{(1)} = c_i^{(2)}(1)\right) = \frac{3}{n},$$

and just in the same way:

$$\mathbb{E}\left(c_{ij}^{(2)}|(i,j) \in O_i\right) = \frac{3}{n}.$$

In the concluding steps, Walkup utilized these observation to define a partition of the bipartite multigraph G that combined with the results from the theorem and derived inequalities to obtain the desired estimation for the assignment problem. For further details on the proof one can look at [17].

Eight years later, Karp [18] introduced a new approach to the estimation of $\mathbb{E}A_n$ that was based on linear programming, obtaining the upper bound:

$$\mathbb{E}[A_n] \leq 2.$$

Specifically Kurp's proof made use of results obtained for the transportation problem, knowing that the assignment problem could be represented as it's particular instance, which assured that both the primal and dual problems of the transportation problem can be expressed in terms of spanning trees. His key observation was that, conditional on certain events related to the spanning tree of a complete bipartite graph, the distribution of edge costs followed a uniform distribution. Building upon this observation, Karp conducted a thorough computation to derive the inequality.

Later on, in 1993, Lazarus [19] was also able to provide a lower bound:

$$\mathbb{E}A_n \geq 1 + \frac{1}{e}.$$

Inspired in part by Karp's result, Dyer, Frieze, and McDiarmid [20] developed an innovative approach to deriving general bounds for the objective function of linear programming problems with random costs. Their method exploits the optimality criterion of the simplex method and the lack-of-memory property of certain random distributions. The technique involves conditioning on random costs based on particular events, such as surpassing a predetermined threshold. By utilizing these conditional probabilities and the properties of the random distributions, they derived bounds on the moment-generating function of the optimal solution to the linear programming problem. So they were able to obtain the same bound proved by Karp, ($\mathbb{E}A_n \leq 2$) but without the need for specialized bases.

Subsequent research has been dedicated to enhancing these bounds, resulting in refinements such as:

$$\mathbb{E}[A_n] \leq 1.94,$$
$$\mathbb{E}[A_n] \geq 1.51.$$

Obtained respectively by Olin in 1992 [21] and Coppersmith - Sorkin in 1999 [22].

1.3.2 Parisi and Mézard conjecture through the replica method

The most significant advancement in estimating the actual solution was provided by Parisi and Mézard, who employed a completely different method based on the apparently unrelated concept of spin glasses. In their seminal work [23], they applied results from the statistical physics of spin glasses to demonstrate that, under the assumption of replica symmetry, the expected value of A_n converges to $\frac{\pi^2}{6} = \zeta(2)$ as n approaches infinity.

The formula they derived from the graph representation of the problem revealed that the true value of the optimal solution is $\frac{\pi^2}{6}$, with an additional term of the order of $\frac{1}{n}$. Specifically, they stated:

$$\mathbb{E}[A_n] = \frac{\pi^2}{6} - \frac{1}{n}\left(\frac{\pi^2}{12} + 2\zeta(3)\right) + \mathcal{O}\left(\frac{1}{n}\right), \tag{1.1}$$

where $\zeta(3)$ refers to the Riemann zeta function at argument 3.

In order to validate their estimation, was conducted an experiment in [24] where the authors analyzed several samples and plotted both the results obtained from a series of simulations and the prediction (1.1) as a function of $\frac{1}{2N}$. They were able to study a range of points between 100 and 800 and considered different sample sizes for varying values of $2N$. Specifically, they analyzed approximately 20000 samples for $2N = 100$, 10000 samples for $2N = 200$, 5000 for $2N = 400$ and 1600 for $2N = 800$.

In this work was replicated their experiment in order to better visualize the accuracy of the prediction (1.1). In particular were created different number of possible graphs for each value of $2N$, namely $100, 150, 200, 400, 800$. Then for each graph, costs were randomly assigned from an uniform distribution between 0 and 1, and the optimal solution was computed using the Hungarian algorithm. Subsequently, for each value of N, were calculated the mean and the relative distance from the prediction as:

$$\Delta L = \frac{|L_{simulated} - L_{theorical}|}{L_{theorical}}. \tag{1.2}$$

where $L_{simulated}$ is the average value of the length of the optimal matching, founded in the simulations, while $L_{theorical}$ is the one computed using (1.1) the optimal solutions across the different samples, providing the results reported in the table 1.1 These values are plotted together

$2N$	Mean	ΔL	Samples
100	1.5784923226880974	0.0012086335130640858	20000
150	1.5995813068263864	0.001455561053957953	10000
200	1.6116590883243114	0.0006257766414101237	10000
400	1.6271405713344287	0.0010195175158027854	5000
800	1.6341784006457363	0.0016429026262728137	1600

Table 1.1: Table of means and relative distances obtained from the simulation. For each value of N were created a number of graphs reported in the sample column with $2N$ nodes and computed the optimal solution of the matching problem where the table report the average among the different graph and the relative distances, computed as in (1.2).

with the line of prediction (1.1) in the plot 1.2 and here can be observed that the results obtained from the various simulations closely resemble those conjectured using the replica theory, showing the practical validity of the estimation.

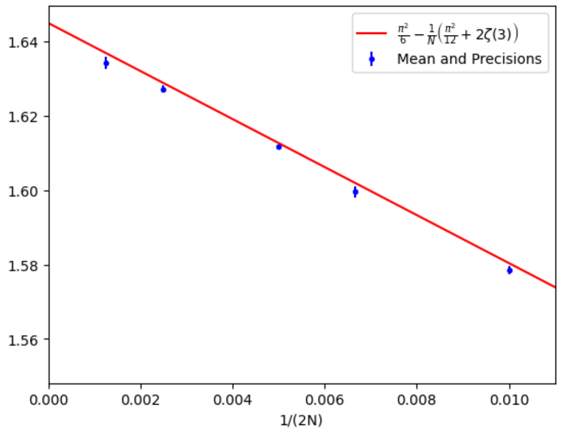


Figure 1.2: Plot of both the line of the theoretical prediction obtained using Parisi and Mézard equation (1.1) and mean and relative distance, as reported in the table 1.1, of the optimal solution of the matching problem computed with the hungarian algorithm.

The $\zeta(2)$ bound on the expected value of A_n, as obtained by Parisi and Mézard, gave significant insight into the solution of the random assignment problem. However, the only aspect left to address was providing a rigorous mathematical proof instead of an estimate. The first proper demonstration of this result emerged later in the work of Aldous [1]. In his article, the problem was revisited using linear programming techniques, successfully proving Parisi's conjecture. Furthermore, he extended the understanding of the problem by deriving additional results that elucidate the obtained expectation value. His contributions culminate in a series of key theorems that will be provided in the next chapter, along with an outline of the proof, while the remaining details will be covered in Chapter 3.

2
Fundamental Steps of Aldous's Proof

2.1 Presentation of the results

As mentioned in the previous chapter, the focus of this part is to explain the main steps of Aldous's proof of Parisi and Mézard estimation. Here, will be provided an overview of its key components, rather than explaining all its exhaustive details, which will be presented later in Chapter 3.

A key starting point of Aldous's proof derives from his prior research [25], where he demonstrated that the distribution of $c(i, j)$ influences the limit of $\mathbb{E}[A_n]$ only through its density function at 0. Consequently, for the sake of simplicity and without loss of generality, will be adopted the Exponential(1) distribution for $c(i, j)$ instead of the uniform distribution. This adjustment eliminates the need for successive normalizations.

Then, it becomes necessary to reframe the problem in a more convenient manner, for which the following formulation was considered:

$$A_n = \min_{\pi} \frac{1}{n} \sum_{i=1}^{n} c\,(i, \pi(i))\,.$$

where the $(c(i, j))$ are independent with exponential distribution with mean n.

The main results obtained by Aldous confirm Parisi and Mézard's estimation while also providing additional insights. These results are summarized in the following theorems, which will be the central focus of this and the subsequent chapter. In all the following results, the permutation obtaining the minimum will be indicated as π_n.

Theorem 7. $\lim_n \mathbb{E}A_n = \frac{\pi^2}{6}$.

Theorem 8. $c(1, \pi_n(1))$ *converges in distribution; the limit distribution has density:*

$$h(x) = \frac{e^{-x}(e^{-x} - 1 + x)}{(1 - e^{-x})^2}, 0 \leq x < \infty. \tag{2.1}$$

Theorem 9. *For each* $k \geq 1$ *define:*

$$q_n(k) = P\left(c(1, \pi_n(1)) \textit{ is the k'th smallest of } \{c(1,1), c(1,2), \ldots, c(1,n)\}\right).$$

Then $\lim_n q_n(k) = 2^{-k}$.

Theorem 10. *For each* $0 < \delta < 1$ *there exists* $\epsilon(\delta) > 0$ *such that, if* μ_n *are permutations (depending on* $(c(i,j))$*) such that* $\mathbb{E}n^{-1}\#\{i : \mu_n(i) \neq \pi_n(i)\} \geq \delta$*, then*

$$\liminf_n \mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n} c\left(i, \mu_n(i)\right)\right) \geq \frac{\pi^2}{6} + \epsilon(\delta).$$

The first theorem is the most important for this thesis because it proves the conjecture obtained through replica theory, so it's the only one that will be completely proved, while the other three are able to add information about the distribution of the optimal solution and a characterization of the costs.

The key steps of the proof of Theorem 7 can be summarized as follows:

1. Construction of the Poisson Weighted Infinite Tree (PWIT) and proof of the equivalence of the random assignment problem with the optimal matching problem on the PWIT.

2. Finding the optimal matching $\mathcal{M}_{\text{opt}}$ on the PWIT and analysis of its main properties.

3. Deduction of the results presented in Aldous's article from the properties of $\mathcal{M}_{\text{opt}}$, by a sequence of calculations.

In this chapter, the essential steps will be outlined, demonstrating how it is possible to prove the desired result, without explicitly presenting all the proofs in this section. These proofs will be carried out in the next chapter, while the aim of this one is to summarize them.

2.2 The Poisson Weighted Infinite Tree

The first part of Aldous's proof of the Parisi-Mézard conjecture involves establishing the equivalence of the random assignment problem with the matching problem on the Poisson Weighted Infinite Tree (PWIT). So in order to provide crucial results for the matching problem, this section will first explain the structure of the PWIT and it's connection with the problem of interest of this thesis.

The Poisson Weighted Infinite Tree can be described by considering a set V comprising finite words, denoted as $v = v_1v_2 \ldots v_d$, where each v_i is a natural number, subject to the constraint $0 \leq d < \infty$. This set includes the empty word, denoted as ϕ. Within this framework, there exists a natural tree, denoted as T, with a vertex-set V and an edge-set E. An edge, $e \in E$, takes the form $e = (v, v_j)$, where j is greater than or equal to 1. Here, for a given $v = v_1v_2 \ldots v_d$, vj is defined as $v_1v_2 \ldots v_dj$, representing the j'th child of v, with v serving as the parent of vj.

To introduce randomness into this structure the edge-weights are assigned such that for each $v \in V$, the weights $(W(v, vj), j \geq 1)$ on the edges $((v, vj), j \geq 1)$ are determined by the points of a Poisson point process with a rate of 1, independently as v varies.

A Poisson point process [26] is a fundamental concept in probability theory and stochastic processes. It represents a scenario where points are distributed randomly in a mathematical space. What distinguishes a Poisson point process is the independence between these points, in fact they occur randomly and without influence from each other and that when counting the number of points within any specified region of the space, the distribution of these counts follows what's known as a Poisson distribution, so if N is denoted as the number of points in the process the probability that N is equal to a certain number n is given by:

$$Pr\{N = n\} = \frac{\lambda^n}{n!}e^{-\lambda}.$$

where λ is the expected value of the distribution, 1 in this case.

This whole structure is referred to as the Poisson Weighted Infinite Tree (PWIT). In addition is possible to denote the probability distribution of the entire configuration $(W(e))$ of edge-weights as λ. Consequently, λ represents a probability measure on the space $W = (0, \infty)^E$, containing all possible configurations $w = (w(e), e \in E)$ of edge-weights.

A matching on the Poisson Weighted Infinite Tree (PWIT) can be defined similarly to how it was presented for other graphs in the introductory chapter. So a matching on the PWIT is a set of edges from T, such that each vertex is incident to exactly one edge in the set. The set that contains all the possible matching for the PWIT is denoted with $\mathrm{M} \subseteq \{0, 1\}^E$ and

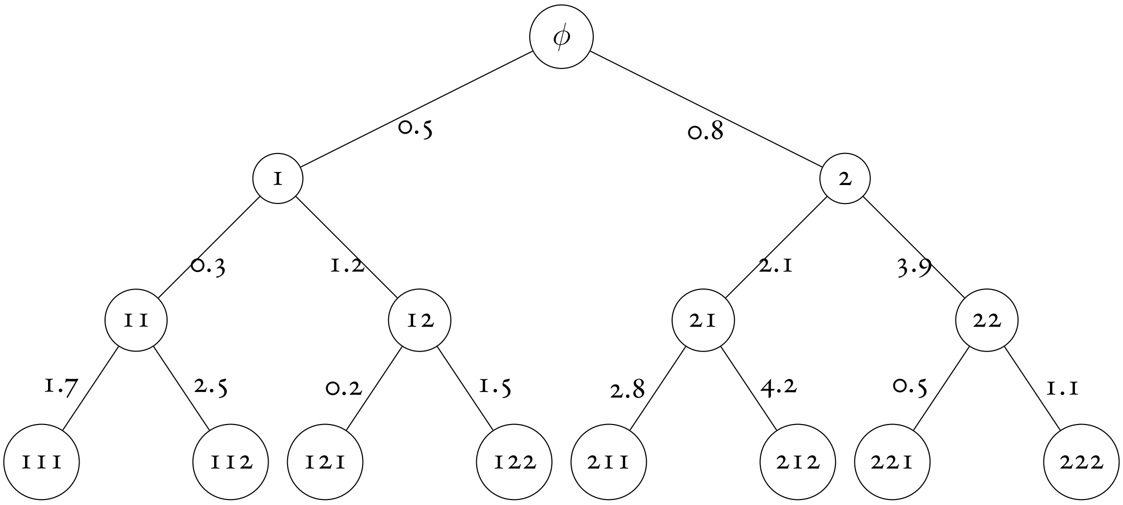


Figure 2.1: One possible realization of a Poisson Weighted Infinite Tree (PWIT) representing nodes with associated weights along the edges.

m is considered a matching, so $m = m(e) \in \mathrm{M}$, if and only if $\sum_{e:v\in e} m(e) = 1 \; \forall \; v \in V$. In addition, given that the weights of the edges for the PWIT are random, it is important to acknowledge that each random matching will depend on the edge weights. Therefore, it is necessary to consider a joint distribution between edge weights and indicators of edges in the matching, denoted by μ on $W \times \mathrm{M}$, where W represents the set of edge weights and M represents the set of matchings. The marginal distribution of this joint distribution on W is denoted by λ.

The relationship between the random matching M in the Poisson Weighted Infinite Tree (PWIT) and the matching π_n in the $n \times n$ random assignment problem is established by the following theorem, which provides a link between the two different questions, implying that can be studied only the optimal solution of the matching on the PWIT in order to prove the result on the random assignment problem.

Theorem 11.

$$\lim_n \mathbb{E}\left[A_n\right] = \inf \mathbb{E}\left[W(\phi, \overrightarrow{M}(\phi))\right],$$

where the infimum is taken over all spatially invariant random matchings $\mathcal{M}$ on the PWIT.

In this section, the precise meaning of the notation $\overrightarrow{M}$ for spatially invariant matching will not be explicitly defined but it will be clarified in the subsequent chapter. At this stage, it is important to understand that not all possible matchings can be considered in the theorem. Indeed, including all possible matchings would render the bound to be proved incorrect due to the existence of a greedy matching, denoted as M_{greedy}. This greedy matching can be constructed by considering all edges as $(\phi, 1), (2, 21), (3, 31), \ldots, (11, 111), (12, 121), \ldots$, and so forth. The average cost of this matching is 1, as each selected edge's average cost is $\frac{1}{\lambda}$, and in our case, the rate λ is equal to 1, resulting in a much smaller cost than the desired limit of $\mathbb{E}[A_n]$, which is $\frac{\pi^2}{6}$.

However, imposing the constraint of spatially invariant matching complicates the result, as the decision regarding whether an edge e belongs to the matching must be spatially invariant, meaning it should not depend on which vertex is selected as the root of T. This addition will exclude from the statement of Theorem 11 some matching, including the greedy matching that was presented, guaranteeing that is not possible to find better bounds than $\frac{\pi^2}{6}$. To define this notion, it is essential to introduce the concepts of local convergence and the unfolding map, which will be addressed in subsequent chapters. For the purposes of this chapter's summary, these theories will be postponed.

For now, it's important to understand the key message of Theorem 11, which is that the expected value of the optimal solution of the random assignment problem, $\mathbb{E}[A_n]$, converges to the average cost per edge in a minimum cost matching on the PWIT as n tends to infinity.

2.3 The Optimal Matching on the PWIT

2.3.1 Fundamental idea

There is a fundamental idea behind the theoretical construction of the optimal matching on the PWIT, which will be rigorously developed in the next sections. This idea is the following: consider a realization of the PWIT and define

$$X_\phi = \text{cost of optimal matching on } T - \text{cost of optimal matching on } T - \{\phi\}. \qquad (2.2)$$

The equation represents how the cost of the matching increases with the addition of the root. However, due to the involvement of total cost, which essentially deals with infinity minus infinity, the equation does not hold up as a precise definition. Nevertheless, statistical physics commonly employs such renormalization arguments. Let's assume, for the sake of the argument, that it works as intended because as at this point, only the idea behind all the demonstrations that will be done in the next chapter is being considered.

If each vertex v in the set V is considered, X_v which represents the cost difference caused by the addition of v, can be defined in a similar manner by focusing on the subtree T_v rooted at v

that consists of v and all its descendant vertices.

$$X_v = \text{cost of optimal matching on } T^v - \text{cost of optimal matching on } T^v - \{\phi\}.$$

And so is possible to obtain the recursion:

$$X_v = \min_{1 \le j < \infty} \left(W(v, v_j) - X_{v_j} \right). \tag{2.3}$$

The recursion arises from the fact that in this case, the left side represents the cost difference between using or not using the vertex v in a matching on T^v and to use an edge (v, v_j) is necessary to pay the cost of the edge and the difference between the cost of not using or using v_j, which is the right side. Next, it will be demonstrated that the argument could be made rigorous by first constructing random variables satisfying the recursion (2.3), and then defining a matching using these random variables. However, since (2.2) is not rigorously defined, it cannot be claimed with certainty that the matching obtained in this way is optimal. Nevertheless, it will be shown (in Proposition 2) that weak optimality is easy to prove.

2.3.2 The Logistic Distribution

The solution introduced earlier will be developed in the next subsection. But first, to better understand the statement and provide a proof, it is necessary to recall the definition of a logistic distribution and establish some results about it. This probability distribution will play a crucial role in proving the Parisi and Mézard conjecture, so additional attention will be dedicated to it here. The logistic distribution is the symmetric probability density:

$$f_X(x) = \left(e^{x/2} + e^{-x/2}\right)^{-2}, \infty < x < \infty.$$

It was proved in [27] that the corresponding distribution function and variance are:

$$F_X(x) := \int_{-\infty}^{x} f_X(y) dy = \left(1 + e^{-x}\right)^{-1}, \, -\infty < x < \infty,$$

$$\text{var}X := \int_{-\infty}^{\infty} x^2 f_X(x) dx = \frac{\pi^2}{3}. \tag{2.4}$$

and that this particular distribution has the following property that characterize it:

$$f(x) = F(x)(1 - F(x)), \, -\infty < x < \infty. \tag{2.5}$$

Now the first result that will be necessary for the construction of the optimal solution is the following, whose demonstration rely rather than on "brute force calculus" more on the symmetry and the structure of the logistic distribution.

Lemma 1. *Let* $0 < \xi_1 < \xi_2 < \dots$ *be the points of a rate* 1 *Poisson process. Then let* $(X; X_i, i \geq 1)$ *be independent random variables with common distribution* μ*. Then:*

$$\min_{1 \leq i < \infty} (\xi_i - X_i) \overset{d}{=} X, \tag{2.6}$$

if and only if μ *is the logistic distribution.*

Proof. Consider a set of points (ξ_i, X_i) forming a Poisson point process $\mathcal{P}$ on $(0, \infty) \times (-\infty, \infty)$ with mean intensity $\rho(z, x)dzdx = dz\mu(x)$. The distribution function F of μ satisfies:

$$1 - F(y) = P\left(\min_{1 \leq i < \infty} (\xi_i - X_i) \geq y\right),$$

by property (2.6). Then, since stating that the minimum is $\geq y$ is equivalent to stating that all of them are $\geq y$, which happens if and only if none of them is $\leq y$, is possible to write:

$$P\left(\min_{1 \leq i < \infty} (\xi_i - X_i) \geq y\right) = P\left(\text{no points of } \mathcal{P} \text{ in } (z, x) : z - x \leq y\right).$$

Now, $\mathcal{P}$ is a Poisson process, so the number of points will follow a Poisson distribution with probability of being equal to n: $\frac{\lambda^n}{n!}e^{-\lambda}$, which needs to be computed over the considered region. In this case, the mean intensity is $\lambda = \int\int_{z-x\leq y} \rho(z, x)dzdx$, and $n = 0$. Thus, the last term in the series of equations can be rewritten as:

$$\exp\left(-\int_0^\infty \bar{F}(z - y)dz\right),$$

where $\bar{F}(y) = 1 - F(y) = \exp\left(-\int_{-y}^\infty \bar{F}(u)du\right)$, as obtained through a change of variables.

Differentiating $1 - F(y) = \exp\left(-\int_{-y}^\infty \bar{F}(u)du\right)$ yields

$$F'(y) = \bar{F}(-y)\bar{F}(y). \tag{2.7}$$

This implies that the density $F'(\cdot)$ is symmetric and so (2.7) is equivalent to the condition (2.5) characterizing the logistic distribution. □

Another property of this distribution that will be essential for the construction of the optimal matching is the following:

Lemma 2. *Let X_1 and X_2 be independent random variables with the logistic distribution. Then:*

$$h(x) := P\left(X_1 + X_2 > x\right), 0 \leq x \leq \infty,$$

is the density of a probability distribution on $[0, \infty)$ with mean $\frac{\pi^2}{6}$.

Proof. To demonstrate that it is a probability density, it is necessary to verify that the integral across its domain equals 1. In this case this integral is:

$$\begin{aligned}\int_0^\infty h(x)dx &= \int_0^\infty P\left(X_1 + X_2 \geq x\right) dx \\ &= \int_0^\infty P\left((X_1 + X_2)^+ \geq x\right) dx \text{ (because } x \text{ is non-negative)} \\ &= \mathbb{E}(X_1 + X_2)^+.\end{aligned}$$

This is due to the fact that for non-negative random variables holds the integral identity:

$$\mathbb{E}(X) = \int_0^\infty P(x > t)dt.$$

Then is also possible to see that:

$$\mathbb{E}(X_1 + X_2)^+ = \mathbb{E}(X_1 - X_2)^+ \text{ (by symmetry of the logistic distribution)}$$

$$= \int_\infty^\infty P(X_1 \geq y \geq X_2).$$

Where the last equality is justified by the following general identity (for arbitrary random variables V, W)

$$\mathbb{E}(V - W)^+ = \int_{-\infty}^\infty P(V > x > W)dx.$$

Then:

$$\begin{aligned}\int_{-\infty}^\infty P(V > x > W)dx &= \int_{-\infty}^\infty (1 - F(y))F(y)dy \\ &= \int_{-\infty}^\infty f(y)dy \text{ because of the property (2.5)} \\ &= 1.\end{aligned}$$

And the mean is:
$$\begin{aligned}\int_0^\infty xh(x)dx &= \int_0^\infty xP\left(X_1+X_2\geq x\right)dx\\ &= \frac{1}{2}\mathbb{E}\left((X_1+X_2)^+\right)^2\\ &= \frac{1}{4}\mathbb{E}(X_1+X_2)^2 \text{ by symmetry}\\ &= \frac{1}{2}\mathbb{E}X_1^2\\ &= \frac{\pi^2}{6} \text{ using (2.4).}\end{aligned}$$

□

2.3.3 Construction of the optimal matching

In the process of constructing the previously outlined solution heuristically, each edge e within the set E of T corresponds to two directed edges $\overrightarrow{e}$ and $\overleftarrow{e}$, collectively referred to as $\overrightarrow{E}$. These directed edges represent familial relations, where each edge $\overrightarrow{e}=(v',v)$ generates infinitely many children in the form (v,y), where $y\neq v'$. For instance, the directed edge $(273,27)$ gives rise to children such as $(27,2)$, $(27,271)$, $(27,272)$, $(27,274)$, and so forth. The Poisson Weighted Infinite Tree (PWIT) assigns weights $W(e)$ to undirected edges, and the weight of the corresponding directed edges $\overrightarrow{e}$ and $\overleftarrow{e}$ is denoted as $W(\overrightarrow{e})=W(\overleftarrow{e})=W(e)$.

With this established, is now possible to proceed to articulate the first lemma for the construction of the optimal matching.

Lemma 3. *Jointly with the edge-weights* $(W(e), e\in E)$ *of the PWIT is possible to construct* $\{X(\overrightarrow{e}), \overrightarrow{e}\in\overrightarrow{E}\}$ *such that:*

1. *Each* $X(\overrightarrow{e})$ *has the logistic distribution.*
2. *For each* $\overrightarrow{e}$ *with children* $\overrightarrow{e_1}, \overrightarrow{e_2}, \ldots$ *say,*
$$X(\overrightarrow{e})=\min_{1\leq j<\infty}\left(W(\overrightarrow{e_j}-X(\overrightarrow{e_j})\right). \tag{2.8}$$

Proof. For a vertex $v=i_1i_2\ldots i_h$ write $|v|=h$. For $h\geq 1$ write:
$$\overrightarrow{E_h}=\{\overrightarrow{e}=(v,v_j):|v|=h, j\geq 1\},$$
$$\overrightarrow{E}_{\leq h}=\{\overrightarrow{e}=(v,y):|v|\leq h, |y|\leq h\}.$$

Consider a set of independent logistic random variables $\{X(\overrightarrow{e}) : \overrightarrow{e} \in \overrightarrow{E}_h\}$, which are independent of the family $(W(e))$. Then recursively define $X(\overrightarrow{e})$ for each $\overrightarrow{e} \in \overrightarrow{E}_{\leq h}$ using equation(2.8). By Lemma 1 because the $X(\overrightarrow{e_j})$ follow a logistic distribution and $W(\overrightarrow{e_j})$ belongs to a Poisson point process then also $X(\overrightarrow{e})$ has a logistic distribution. This construction yields a joint distribution for $\{W(e), e \in E; X(\overrightarrow{e}), \overrightarrow{e} \in \overrightarrow{E}_h \cup \overrightarrow{E}_{\leq h}\}$. The proof is then completed using the Kolmogorov consistency theorem. In fact the theorem states that:

Theorem 12. *Given an arbitrary family of marginal probability distribution P_n on a compact metric space X, there exists at least one probability distribution on X, such that for each n, the marginal distribution of this distribution on every compact subset of X coincides with P_n.*

Thus, Kolmogorov's theorem guarantees the existence of the joint distribution $\{W(e), e \in E; X(\overrightarrow{e}), \overrightarrow{e} \in \overrightarrow{E}_h \cup \overrightarrow{E}_{\leq h}\}$ and ensures that the members $W(e)$ and the $X(\overrightarrow{e})$of the joint distribution still have the marginal distributions defined initially, where all the weights of the edges $W(e)$ constitute a Poisson point process and $X(\overrightarrow{e})$ has the logistic distribution. □

For $v, v' \in V$ from now will be written $v \sim v'$ if (v, v') is an undirected edge. Once is fixed a realization of the weights $(W(e), e \in E; X(\overrightarrow{e}), \overrightarrow{e} \in \overrightarrow{E})$. Then for each $v \in V$ define $v*$ as the vertex that cost less (for the specific actualization of costs W) to add to the matching. Formally is defined as:

$$v* = \arg\min_{v' \sim v}(W(v, v') - X(v, v')). \tag{2.9}$$

Then the following lemma states that by adding step by step vertices like $v*$, that represent the ones that can be added with less cost, is guarantee to obtain a matching on the PWIT.

Lemma 4. *The set of undirected edges $\{(v, v*) : v \in V\}$ is a matching on the PWIT.*

Proof. To prove this result, it suffices to demonstrate that if $v*$ represents the vertex of minimum cost to be added to the matching given vertex v, then the converse is also true, i.e., $(v*)* = v$. This implies that for all vertices v, selecting the unordered edge $(v, v*)$ results in a valid matching. Therefore, the set of undirected edges $(v, v*) : v \in V$ forms a matching on the PWIT.

Now fix v, by definition of $v*$ follows that:

$$W(v, v*) - X(v, v*) < \min_{y \sim v, y \neq v*} (W(v, y) - X(v, y)),$$

and:

$$(W(v, y) - X(v, y)) = X(v*, v) \text{ by (2.8)}.$$

In other words:

$$X(v, v*) + X(v*, v) > W(v*, v). \tag{2.10}$$

Now suppose, for contradiction, that $(v*)* = z \neq v$; then:

$$W(v*, z) - X(v*, z) < W(v*, v) - X(v*, v).$$

In this case (2.8) would imply that:

$$X(v, v*) \leq W(v*, z) - X(v*, z).$$

Combining these last two inequalities yields the result that $X(v, v*) \leq W(v*, v) - X(v*, v)$ which contradicts the equation (2.10) and proves the lemma. □

Formulate $\mathcal{M}_{opt}$ for the stochastic matching as outlined in Lemma 4. It's noteworthy that the reasoning leading to equation (2.10) can be reversed, offering a more symmetrical criterion for determining whether an edge belongs to $\mathcal{M}_{opt}$:

An edge, e, is part of $\mathcal{M}_{opt}$ if and only if:

$$W(e) < X(\overrightarrow{e}) + X(\overleftarrow{e}),$$

where $\overrightarrow{e}$ and $\overleftarrow{e}$ denote the directed edges corresponding to e. It appears intuitive that $\mathcal{M}_{opt}$ should be invariant across space; This will be proved in the next sections.

2.4 Analysis of the optimal matching and deduction of the estimation

This subsection aims to present the fundamental result that implies the Parisi and Mézard conjecture, along with the other theorems stated at the beginning of the chapter. While the primary focus is on the former, all mentioned theorems can be proved using the results that will be provided. The central object of study in this subsection is the random cost $W(\phi, \overrightarrow{\mathcal{M}}_{opt}(\phi))$ of the edge $(\phi, \overrightarrow{\mathcal{M}}_{opt}(\phi))$ of $\mathcal{M}_{opt}$ which contains the root. The most crucial property of this random cost is as follows:

Proposition 1. *The random variable $W(\phi, P\overrightarrow{\mathcal{M}}_{opt}(\phi))$ has the probability density function $h(\cdot)$ described in lemma 2, and so $\mathbb{E}W(\phi, \mathcal{M}_{opt}(\phi)) = \frac{\pi^2}{6}$.*

Proof. The weights associated with edges $(W(\phi, i), i \geq 1)$ and the X-values $(X(\phi, i), i \geq 1)$ follow a Poisson process (ξ_i) and are independently and identically distributed as logistic random variables (X_i) as stated in Lemma 1. Utilizing the notation introduced earlier and the definition 2.9 of $\mathcal{M}_{opt}$ is possible to write:

$$W(\phi, \overrightarrow{\mathcal{M}}_{opt}(\phi)) = \xi_I, \text{where } I = \arg\min_{i \geq 1}(\xi_i - X_i).$$

To derive this distribution in a refined manner first fix $0 < y < \infty$ and condition on the event $A_y := \{\exists J : \xi_J = y\}$. Conditionally, the remaining points $(\xi_i, i \neq J)$ and corresponding X-values $(X_i, i \neq j)$ adhere to a Poisson process $(\xi'_j, j \geq 1)$ and are i.i.d. logistic variables $(X'_j, j \geq 1)$, respectively. Notably, these are independent of X_J, with its conditional distribution retaining a logistic form. Thus,

$$P(I = J|A_y) = P(y - X_J < X') \text{ where } X' = \min_{j \geq 1}(\xi'_j - X'_j).$$

But according to Lemma 1 , X follows a logistic distribution. Given its independence from X_J, is possible to deduce from the definition of $h(\cdot)$ that $P(I = J|A_y) = h(y)$. Consequently, it establish:

$$P\left(\xi_I \in [y, y + dy]\right) = P\left(I = J|A_y\right) P\left(\text{ some } \xi_i \text{ in } [y, y + dy]\right) = h(y)dy,$$

thus confirming the statement. □

The last result that is needed to prove that A_n converges to $\frac{\pi^2}{6} = \zeta(2)$ is the following proposition.

Proposition 2. *Let $\mathcal{M}$ be a spatially invariant matching of the PWIT such that $P(\overrightarrow{\mathcal{M}}(\phi) \neq \overrightarrow{\mathcal{M}}_{opt}(\phi)) > 0$. Then $\mathbb{E}W(\phi, \overrightarrow{\mathcal{M}}(\phi)) > \mathbb{E}W(\overrightarrow{\mathcal{M}}_{opt}(\phi))$.*

Before diving into the proof of this proposition in the upcoming chapter, it's crucial to elucidate how it connects to the original statement. Propositions 1 and 2 lay the groundwork for deriving all four theorems (7-10). The central theorem of focus, Theorem 7, can be inferred through the following reasoning: Propositions 1 and 2 collectively imply that:

$$\inf\left\{\mathbb{E}W(\phi, \overrightarrow{\mathcal{M}}(\phi)) : \overrightarrow{\mathcal{M}} \text{ is spatially invariant matching on the PWIT}\right\} = \frac{\pi^2}{6}.$$

This arises from the fact that Proposition 2 implies that every spatially invariant matching on the PWIT has an expected value of weights greater than that of $\mathcal{M}_{opt}$, while Proposition 1 indicates that the expected value of $\mathcal{M}_{opt}$ is indeed $\frac{\pi^2}{6}$. When combined with Theorem 11, it follows that:

$$\lim_n \mathbb{E}[A_n] = \inf\left\{\mathbb{E}W(\phi, \overrightarrow{\mathcal{M}}(\phi)) : \overrightarrow{\mathcal{M}} \text{ is spatially invariant matching on the PWIT}\right\} = \frac{\pi^2}{6}$$

thus affirming Theorem 7 and giving the desired result.

3
Complete proof

In this chapter will be completed the proof explained in Chapter 2 providing all the missing parts.

3.1 Equivalence with the matching on the PWIT

3.1.1 Spatially invariant matching

In the previous chapter, Theorem 11 stated the equivalence of the limit of the expected value of A_n for the random assignment problem with the infimum over the expected values of the weights for spatially invariant matching on the PWIT. However, the precise definition of spatially invariant matching was not provided. This section will cover this part by constructing and defining spatially invariant matching.

First of all, for each $w \in W$ and each $i \geq 1$, define a bijection θ_i^w from V, the set of vertices, to itself, which induces a bijection on the edge set. These bijections are denoted by $\theta_i^w(\cdot)$ and can be defined as follows:

$$w(i, i(k-1)) < w(\phi, i) < w(i, ik).$$

Setting the left hand side of the equation as 0 for $k = 1$. Then define the bijection in a way that set the vertex i as the root and relabel all the vertex to preserve order structure, so by defining:

- $\theta_i^w(i) = \phi$,

- $\theta_i^w(\phi) = k$,
- θ_i^w takes vertices $(i1, i2, \dots, i(k-1); ik, i(k+1), \dots)$ to vertices $(1, 2, \dots, k-1; k+1, k+2, \dots)$,
- for $v = ij_2 \dots j_l$ and $l \geq 3$ let $\theta_i^w(v) = (\theta_i^w(ij_2))\, j_3 \dots j_l$,
- for $v = j_1 j_2 \dots j_l$ where $j_1 \neq i$ let $\theta_i^w(v) = kj_1 j_2 \dots j_l$.

Before, the space $W \times M$ was described as the set of possible combinations (w, m) of weights w and matchings m. Now, by considering the choice of the root vertex, the space expands to $Z = W \times M \times \{1, 2, \dots\}$, where each element is denoted as (w, m, k), with k representing the distinguished vertex. The maps θ_i^w induce a mapping from this space Z to itself. This mapping can be interpreted as relabeling the distinguished vertex as the root, preserving the order structure of the other vertices, and transforming the previous root into the distinguished vertex. In particular, $\theta(w, m, i) = (\hat{w}, \hat{m}, k)$ where:

- $w(e) = \hat{w}(\theta_i^w(e))$,
- $m(e) = \hat{m}(\theta_i^w(e))$,
- $k = \theta_i^w(\phi)$.

So in this enlarged space $Z = W \times M \times \{1, 2, \dots\}$ each measure μ on $W \times M$ extends to a σ-finite measure $\mu* := \mu \times \mathtt{count}$ on Z, where $\mathtt{count}$ is the counting measure on $\{1, 2, 3, \dots\}$ that is defined as follows:

$$\mathtt{count}(A) = \begin{cases} |A| & \text{if } A \text{ is finite} \\ \infty & \text{if } |A| = \infty \end{cases} \tag{3.1}$$

In this way $\mu*$ is able to describe a random matching on the PWIT with a specific distinguished vertex.

This extension is essential for the definition of spatial invariance for a random matching $\mathcal{M}$ on the PWIT. In fact spatial invariance can be defined as follows, utilizing the probability distribution μ of $((\mathcal{M}(e), W(e)), e \in E)$ and the σ-finite measure $\mu^* := \mu \times \mathtt{count}$ on Z.

Definition 4. *A random matching M on the PWIT, with distribution μ on $W \times M$, is spatially invariant if $\mu*$ is invariant under θ, that is if $\mu * (\cdot) = \mu(\theta^{-1}(\cdot))$ as σ-finite measures on Z.*

Informally, what the definition is saying is that a spatially invariant matching is a random matching whose distribution in the extended space Z is not affected by a relabeling of a vertex. If an arbitrary rule is used to distinguish some vertex k (where the rule depends on the realization of $\mathcal{M}$ and $W(e)$), then the μ^*-distribution of the configuration without the vertex being distinguished shouldn't change.

It is important to recall that Theorem 11 considers only spatially invariant matchings on the PWIT, which are the only ones to have the same cost as the limit of the expected value of the solution of the random assignment problem. In this way, many matchings on the PWIT are not considered by the theorem. This notion explains why the greedy matching $\mathcal{M}_{\text{greedy}}$ introduced in the previous chapter couldn't prove the existence of a smaller bound than the desired one of $\frac{\pi^2}{6}$ because it is, in fact, not spatially invariant.

To prove this, it is necessary to consider: $B \doteq \{W(1,11) < W(\phi,1) < W(1,12)\}$. Then:

$$P(W(1,11) < W(\phi,1)) = \frac{1}{2} = P(W(1,11) > W(\phi,1)),$$

and similarly,

$$P(W(\phi,1) < W(1,2)) = \frac{1}{2} = P(W(\phi,1) > W(1,2)).$$

Because the events are independent, the probability of B is equal to $\frac{1}{4}$. So the event $B* \doteq B \cap \{1 \text{ is distinguished }\}$ has $\mu*$-measure $\frac{1}{4}$, because the counting measure defined as 3.1 gives equal weight of 1 to each single vertex. The inverse image $\tilde{B} = \theta^{-1}(B*)$ is:

$$\tilde{B} = \{W(\phi,2) < W(2,21), 2 \text{ is distinguished }\}.$$

This has $\mu*$-measure $\frac{1}{4}$ because $P(W(\phi,2) < W(2,21)) = \frac{1}{4}$. This equality of $\mu*$-measure is a consequence of the fact that the distribution of edge-weights is spatially invariant. Now for a random matching $\mathcal{M}$ to be spatially invariant, the event:

$$B * \cap \{\overrightarrow{\mathcal{M}}(\phi) = 1\}$$

must have an identical $\mu*$-measure to its inverse image under θ, which is represented by the event:

$$\tilde{B} \cap \{\overrightarrow{\mathcal{M}}(\phi) = 2\}.$$

However, the greedy matching always has $\overrightarrow{\mathcal{M}}(\phi) = 1$, so the first event has $\mu*$-measure $\frac{1}{4}$ while the second event has $\mu*$-measure 0, leading to a contradiction.

On the contrary, $\mathcal{M}_{\text{opt}}$ is spatially invariant, as proved in the following result:

Lemma 5. *$\mathcal{M}_{opt}$ is spatially invariant.*

Proof. $\mathcal{M}_{opt}$ is determined by the $W(e)$ and the $X(\overrightarrow{e})$ that satisfy the recursion 2.8 which is not changed by the relabeling of vertices. Moreover $X(\overrightarrow{e})$ has a joint distribution that is determined by the fact (Lemma 3) that the random variables $(X(\overrightarrow{e}), \overrightarrow{e} \in \overrightarrow{E}_h)$ are independent logistic. So in order to prove that the matching is spatially invariant is sufficient to show that the random variables $(X(\overrightarrow{e}), \overrightarrow{e} \in \overrightarrow{E}_h)$ are independent logistic, even after the application of a relabeling θ.

Fix k and write:

$$B = \left\{ X(\overrightarrow{e}, \overrightarrow{e}\ in \overrightarrow{E}_h) \in C, k \text{ is distinguished} \right\},$$

for arbitrary C in the appropriate range space. Write:

$$A_l = \{W(l, k-1) < W(\phi, l) < W(l, k)\}.$$

Then:

$$\theta^{-1}(B) = \cup_l \left[A_l \cap \{l \text{ is distinguished }\} \cap \{(X(\overrightarrow{e}), \overrightarrow{e} \in \overrightarrow{E}_{h,l}) \in C\} \right],$$

where $\overrightarrow{E}_{h,l}$ comprises edges (v, v_j) with $j \geq 1$ and with $v = j_1 j_2 \dots j_{h-1}$ for $j_1 \neq l$, or with $v = l j_1 j_2 \dots j_h$. Given that $\sum_l P(A_l) = 1$ and the $X(\overrightarrow{e})$ under consideration are independent of A_l, demonstrating $\mu * (B) = \mu * (\theta^{-1}(B))$ reduces to establishing the independence of $\{X(\overrightarrow{e}), \overrightarrow{e} \in \overrightarrow{E}_{h,l}\}$, which can be inferred from the argument outlined in Lemma 1. □

3.1.2 Deduction of Theorem 11

Now that was discussed the concept of a spatially invariant matching, there are all the necessary components to deduce a proof of Theorem 11 with the addition of some technical details. In the preceding section, was denoted a random matching in the $n \times n$ random assignment problem as π_n, but it can alternatively be represented as an $n \times n$ matrix $\mathcal{M} = (m(i, j))$ with entries in $0, 1$. A matching, represented in this manner, is considered spatially invariant if the joint distribution $((c(i, j), m(i, j)), 1 \leq i, j \leq n)$ remains invariant under various automorphisms of the complete bipartite graph, including permutations of i, permutations of j, and complete interchange of i and j. By applying a uniform random automorphism to any random n-matching, is possible to obtain a spatially invariant n-matching with the same distribution of average cost per edge A_n. Consequently, it suffices to consider only spatially invariant n-matchings without any loss of generality.

Because is possible to consider only spatial invariant matching is also possible to see that the following result, that was proven in the previous work of Aldous [25], imply Theorem 11.

Theorem 13. *Let $\mathcal{M}$ be a spatially invariant matching on the PWIT with $\mathbb{E}W(\phi, \overrightarrow{\mathcal{M}}(\phi)) < \infty$. Then there exists spatially invariant n-matchings, $\mathcal{M}_n$ such that $\mathcal{M}_n$ locally converges to $\mathcal{M}$ and $(c(1, \overrightarrow{\mathcal{M}}_n(1)), 1 \leq n < \infty)$ is uniformly integrable.*

To better clarify the statement, will be given in (3.3) a more precise notion of local convergence for $\mathcal{M}_n \rightarrow^{d}_{local} \mathcal{M}$ that will also imply that:

$$c(1, \overrightarrow{\mathcal{M}}_n(1)) \rightarrow^{d} W(\phi, \overrightarrow{\mathcal{M}}(\phi)). \tag{3.2}$$

Theorem 13 along with the equation (3.2), imply Theorem 11. This can be deduced as follows: First apply Theorem 13 considering as spatial invariant matching $\mathcal{M}_{opt}$ stated in section 2.3. Then the theorem implies that there is a sequence of n-matchings, which essentially represents the solution of the random assignment problem , $\mathcal{M}_n$ that converges locally to $\mathcal{M}_{opt}$ and that the cost of $(1, \mathcal{M}_n(1))$ has finite and well defined expected value. Subsequently, from the local convergence follows the property 3.2. Then starting from this, by taking the expected value and recalling that $\mathcal{M}_{opt}$ achieve the infimum over $\{\mathbb{E}\{W(\phi, \overrightarrow{\mathcal{M}}(\phi))\}\}$, is possible to obtain the statement of Theorem 11.

At this point is only needed further notation to formalize local convergence and get the equation (3.2). To begin let's fix n, define $V^{(n)}$ as the set of vertices $v = v_1 v_2 \ldots v_l$ satisfying $v_1 < n$ and $v_i \leq n - 1$ for $i \geq 2$ and denote the corresponding subtree of T as $T^{(n)} = (V^{(n)}, E^{(n)})$. Then, let G_{nn} represent the complete bipartite graph with vertex-set $\{1, 2, \ldots n\} \times \{1, 2\}$.

Afterwards, to establish a connection between $T^{(n)}$ and G_{nn} with a given realization $c = (c(i, j))$ of the cost matrix, it is necessary to define a graph homomorphism $\psi = \psi_c$ from $T^{(n)}$ onto G_{nn}. A graph homomorphism is a function from the vertex set of one graph to the vertex set of another, which preserves edges. Formally, it can be stated as:

Definition 5. *A graph homomorphism f from a graph $G = (V(G), E(G))$ to a graph $H = (V(H), E(H))$, denoted as $f : G \rightarrow H$, is a function from $V(G)$ to $V(H)$ that preserves edges. Specifically, if $(u, v) \in E(G)$, then $(f(u), f(v)) \in E(H)$ for all pairs of vertices u, v in $V(G)$.*

In this context, a graph homomorphism ψ maps each vertex v in $T^{(n)}$ to a vertex $\psi(v)$ in G_{nn}, ensuring that the edges in $T^{(n)}$ are preserved in G_{nn} under the mapping ψ. This homomorphism ψ can be defined as follows:

- $\psi(\phi) = (1,1)$.

- For $i \in \{1,2,3,\ldots,n\}$, define $\psi(i) = (j,2)$, for the j such that $c(1,j)$ is the i'th smallest of $\{c(1,u), 1 \leq u \leq n\}$.

- For $i \in \{1,2,3,\ldots,n-1\}$ and $i' \in \{1,2,\ldots,n\}$ define $\psi(i',i) = (k,1)$, for the k such that $c(k,1)$ is the $i'$the smallest of $\{c(\psi(v),u), 1 \leq u \leq n, (u,2) \neq \psi(v^-)\}$ where v^- is the parent of v.

- For $v = v_1v_2\ldots v_{2m+1}$, for $i \in \{1,2,3,\ldots,n-1\}$, define $\psi(v_i) = (k,1)$, for the k such that $c(k,\psi(v))$ us the $i'$the smallest of $\{c(u,\psi(v)), 1 \leq u \leq n, (u,1) \neq \psi(v^-)\}$.

This *folding* map ψ_c induces an *unfolding* map which uses the matrix c to define edge-weights $(W^{(n)}(e))$ on the edge-set $E^{(n)}$:

$$W^{(n)}(v,vk) = c(i,j) \text{ if } \psi_c(v) = (i,1) \text{ and } \psi_c(vk) = (j,2)$$
$$\text{or if } \psi_c(v) = (j,2) \text{ and } \psi_c(vk) = (i,1).$$

Now, for $h \geq 1$, let's define $E_{(h)} \subset E$ as the set of edges, each of which has both end-vertices in the form $v_1v_2\ldots v_l$, where $l \leq h$ and $\max_i v_i \leq h$. Thus, $E_{(h)}$ constitutes a finite set of edges. Upon fixing h, it becomes possible to see that:

$$\left(W^{(n)}(e), e \in E_{(h)}\right) \xrightarrow{d} (W(e), e \in E_{(h)}),$$

where the limits correspond to the edge-weights in the PWIT. The random n-matchings are denoted as $0,1$-valued random variables $\mathcal{M}n(e)$, which are indexed by the edges $e \in Gnn$. Utilizing the homomorphism ψ, can be defined:

$$\tilde{\mathcal{M}}_n(e) = \mathcal{M}_n(\psi(e)), e \in E^{(n)}.$$

Now can be stated also the concept of *local convergence* $\mathcal{M}_n \xrightarrow[local]{d} \mathcal{M}$ that means: for each fixed h,

$$((W^{(n)}(e), \tilde{\mathcal{M}}_n(e)), e \in E_{(h)}) \xrightarrow{d} ((W(e), \mathcal{M}(e)), e \in E_{(h)}). \tag{3.3}$$

3.2 Proof of proposition 2

The proof of this proposition requires two different steps. In the first part it will be shown that proving the proposition is equivalent to proving Lemma 8 and then will be given a proof of the last result.

3.2.1 Equivalent result to Proposition 2

Consider $\mathcal{M}$ as a spatially invariant matching, and denote A as the event $\overrightarrow{M}(\phi) \neq \overrightarrow{M}_{opt}(\phi)$. On A write:

$$(v_{-1}, v_0, v_1) = (\overrightarrow{M}_{opt}(\phi), \phi, \overrightarrow{M}(\phi)). \tag{3.4}$$

Then on this event it can be defined a doubly-infinite path $\dots, v_{-2,-1}, v_0, v_1, v_2, \dots$ by: $\forall -\infty < m < \infty$

$$(v_{2m-1}, v_{2m}) \text{ is an edge of } \mathcal{M}_{opt},$$
$$(v_{2m}, v_{2m+1}) \text{ is an edge of } \mathcal{M}.$$

Now, utilizing the concept of spatial invariance is possible to prove the following result.

Lemma 6. *Conditional on A, the distributions of $X(v_{-2}, v_{-1})$ and $X(v_0, v_1)$ are the same.*

Proof. The core point of this proof lies in the equivalence between the distribution of $X(v_0, v_1)$ with $\phi = v_0$ as the distinguished vertex (as explained in section 3.1.1) and the distribution of $X(v_0, v_1)$ with v_2 as a distinguished vertex. By a relabel is possible to see also that the distribution of $X(v_0, v_1)$ with v_2 as a distinguished vertex is the distribution of $X(v_{-2}, v_{-1})$ as seen from $v_0 = \phi$. This series of equivalences can be formally expressed as follows:

$$P\left(A, X(v_0, v_1) \in \cdot\right) = P(\overrightarrow{\mathcal{M}}(\phi)) = P\left(\overrightarrow{\mathcal{M}}(\phi) \neq \overrightarrow{\mathcal{M}_{opt}}(\phi), X(\phi, \overrightarrow{\mathcal{M}}(\phi)) \in \cdot\right)$$

$$= \mu * \left(\overrightarrow{\mathcal{M}}(\phi) \neq \overrightarrow{\mathcal{M}}_{opt}(\phi), X(\phi, \overrightarrow{\mathcal{M}}(\phi)) \in \cdot, \overrightarrow{\mathcal{M}}(\phi) \text{ is distinguished}\right)$$
$$= \mu * \left(\overrightarrow{\mathcal{M}}(\phi) \neq \overrightarrow{\mathcal{M}}_{opt}(\phi), X(\overrightarrow{\mathcal{M}}(\phi), \phi) \in \cdot, \overrightarrow{\mathcal{M}}(\phi) \text{ is distinguished}\right).$$

Using spatial invariance to change the root from ϕ to $\overrightarrow{\mathcal{M}}(\phi)$, after noticing that the event $\overrightarrow{\mathcal{M}}(\phi) \neq \overrightarrow{\mathcal{M}}_{opt}(\phi)$ is identical to the event $\overrightarrow{\mathcal{M}}(\overrightarrow{\mathcal{M}}(\phi)) \neq \overrightarrow{\mathcal{M}}_{opt}(\overrightarrow{\mathcal{M}}(\phi))$, is possible to rewrite the last member of the previous equation as:

$$\mu * \left(\overrightarrow{\mathcal{M}}(\phi) \neq \overrightarrow{\mathcal{M}}_{opt}(\phi), X(\overrightarrow{\mathcal{M}}(\phi), \phi) \in \cdot, \overrightarrow{\mathcal{M}}(\phi) \text{ is distinguished }\right).$$

This can be done because the $\mu*$-measure doesn't depend on the rule for distinguishing a vertex. Then, using spatial invariance to switch the root from ϕ to $\overrightarrow{\mathcal{M}}_{opt}(\phi)$ it can be rewritten as:

$$\mu * (\overrightarrow{\mathcal{M}}(\overrightarrow{\mathcal{M}}_{opt}(\phi)) \neq \overrightarrow{\mathcal{M}}_{opt}(\overrightarrow{\mathcal{M}}_{opt}(\phi)), X(\overrightarrow{\mathcal{M}}(\overrightarrow{\mathcal{M}}_{opt}(\phi)), \overrightarrow{\mathcal{M}}_{opt}(\phi)) \in \cdot,$$
$$\overrightarrow{\mathcal{M}}_{opt}(\phi) \text{ is distinguished}).$$

Which is equal to:

$$= \mu * \left(\overrightarrow{\mathcal{M}}(\phi) \neq \overrightarrow{\mathcal{M}}_{opt}(\phi), X(\overrightarrow{\mathcal{M}}(\overrightarrow{\mathcal{M}}_{opt}(\phi)), \overrightarrow{\mathcal{M}}_{opt}(\phi)) \in \cdot \overrightarrow{\mathcal{M}}_{opt}(\phi) \text{ is distinguished}\right),$$

because the events are identical.

Then because $v_{-1} = \overrightarrow{\mathcal{M}}_{opt}(\phi)$ and $v_{-2} = \overrightarrow{\mathcal{M}}(v_{-1})$:

$$= P\left(\overrightarrow{\mathcal{M}}(\phi) \neq \overrightarrow{\mathcal{M}}_{opt}(\phi), X(\overrightarrow{\mathcal{M}}(\overrightarrow{\mathcal{M}}_{opt}(\phi)), \overrightarrow{\mathcal{M}}_{opt}(\phi)) \in \cdot\right)$$

$$= P\left(A, X(v_{-2}, v_{-1}) \in \cdot\right).$$

□

At this stage, further reasoning begins by considering that:

$$X(v_{-2}, v_{-1}) = \min_{y \sim v_{-1}, y \neq v_{-2}} \left(W(v_{-1}, y) - X(v_{-1}, y)\right), \text{by (2.8)}.$$

which simplifies to:

$$X(v_{-2}, v_{-1}) = W(v_{-1}, v_0) - X(v_{-1}, v_0),$$

by (2.9), because (v_{-1}, v_0) is in $\mathcal{M}_{opt}$. Additionally, thanks to (2.8):

$$X(v_{-1}, v_0) \leq W(v_0, v_1) - X(v_0, v_1), \tag{3.5}$$

so that:

$$D \doteq W(v_0, v_1) - X(v_0, v_1) - X(v_{-1}, v_0) \geq 0. \tag{3.6}$$

Combining (3.5) with this definition of D gives:

$$W(v_0, v_1) - W(v_0, v_{-1}) = D + X(v_0, v_1) - X(v_{-2}, v_{-1}).$$

Thus,

$$\begin{aligned}
&\mathbb{E}W(\phi, \overrightarrow{\mathcal{M}}(\phi)) - \mathbb{E}W(\phi, \overrightarrow{\mathcal{M}}_{opt}(\phi)) \\
&= \mathbb{E}(W(v_0, v_1) - W(v_0, v_{-1}))\mathbb{1}_A \\
&= \mathbb{E}D\mathbb{1}_A + \mathbb{E}X(v_0, v_{-1}\mathbb{1}_A - \mathbb{E}X(v_{-2}, v_{-1})\mathbb{1}_A) \\
&= \mathbb{E}D\mathbb{1}_A \text{ by Lemma 6.}
\end{aligned} \tag{3.7}$$

Since $D \geq 0$ this is enough to establish the weak inequality:

$$\mathbb{E}W(\phi, \overrightarrow{\mathcal{M}}(\phi)) \geq EW(\phi, \overrightarrow{\mathcal{M}}_{opt}(\phi)).$$

To prove that the strict inequality also holds, suppose for contradiction that $\mathbb{E}W(\phi, \overrightarrow{\mathcal{M}}(\phi)) = EW(\phi, \overrightarrow{\mathcal{M}}_{opt}(\phi))$. Then (3.7) would imply that $\mathbb{E}D\mathbb{1}_A = 0$, and then (3.6) that on A is true that:

$$v_1 = \arg\min_i^{[2]}(W(\phi, i) - X(\phi, i)),$$

where $\min^{[2]}$ denotes the second-smallest value. This equation comes from the fact that $\overrightarrow{\mathcal{M}}(\phi) = v_{-1} = \arg\min_i(W(\phi, i) - X(\phi, i))$.

So without restricting to A:

$$P\left(\overrightarrow{\mathcal{M}}(\phi) = \arg\min_i(W(\phi, i) - X(\phi, i)) \text{ or } \arg\min_i^{[2]}(W(\phi, i) - X(v, i))\right) = 1.$$

By an use of spatial invariance comes that under that hypothesis the same property should hold for every $v \in V$, which can be written as:

$$P\left(\overrightarrow{\mathcal{M}}(v) = \arg\min_i(W(v, i) - X(v, i)) \text{ or } \arg\min_i^{[2]}(W(v, i) - X(v, i))\right) = 1. \quad (3.8)$$

To conclude the argument, it is essential to demonstrate the impossibility of this scenario, which would be implied by the following Proposition:

Proposition 3. *The only spatially invariant random matching on the PWIT satisfying* (3.8) *is* $\overrightarrow{\mathcal{M}}_{opt}$.

Demonstrating this result equals to complete the proof of Proposition 2, so the remaining part of the section will be dedicated to its proof.

The demonstration starts by defining a path $\phi = w_0, w_{-1}, w_{-2}, w_{-3}, \ldots$ inductively by: for $m = 1, 2, \ldots$

$$\begin{aligned} w_{-2m+1} &= \arg\min_{y \sim w_{-2m+2}}(W(w_{-2m+2}, y) - X(w_{-2m+2}, y)), \\ w_{-2m} &= \arg\min^{[2]}(W(w_{-2m+1}, y) - X(w_{-2m+1}, y)). \end{aligned}$$

Then:

$$(w_{-2m+1}, w_{-2m+2}) \text{ is an edge of } \mathcal{M}_{opt}, \text{ each } m = 1, 2, 3, \ldots$$

Suppose $\mathcal{M}$ is a spatially invariant matching that satisfies equation (3.8), with the set $A \doteq$

$\overrightarrow{\mathcal{M}}(\phi) \neq \overrightarrow{\mathcal{M}}_{opt}(\phi)$ having $P(A) > 0$. Then, on the set A, the following property holds:

$$(w_{-2m}, w{-}2m+1) \text{ is and edge of } \mathcal{M}, \text{ each } m = 1, 2, 3, \ldots.$$

So, by (3.8), for $v = w_{-2m}$ is also true that:

$$A \subseteq B_m \doteq \left\{ w_{-2m+1} = \arg\min\nolimits^{[2]}_{y \sim w_{-2m}} (W(w_{-2m}, y) - X(w_{-2m}, y)) \right\},$$

and so:

$$A \subseteq B \doteq \cap_{m=1}^{\infty} B_m.$$

Writing

$$\overline{B}_q \doteq \cap_{m=1}^{q} B_m,$$

then:

$$\text{if } P(A) > 0, \text{ then } \lim_{q \to \infty} P(\overline{B}_{q+1}) / P(\overline{B}_q) = 1. \tag{3.9}$$

Now, to continue the proof, the following Lemma is necessary:

Lemma 7. $P(\overline{B}_{q+1}) = P(\overline{B}_q \cup B*)$, *where:*

$$B* = \left\{ \phi = \arg\min\nolimits^{[2]}_{y \sim w_1} (W(w_1, y) - X(w_1, y)) \right\},$$

with:

$$w_1 \doteq \arg\min\nolimits^{[2]}_{y \sim \phi} \left(W(\phi, y) - X(\phi, y) \right).$$

Proof. The argument of the proof is similar to the one used for Lemma 6. The central idea is that the probability of $\overline{B}_q \cap B*$ with $\phi = w_0$ as the distinguished vertex is the same, by spatial invariance, as the probability of $\overline{B}_q \cap B*$ with $w_2 = \arg\min_{w \sim w_1}(W(w_1, w) - X(w_1, w))$, which by relabeling is the probability of $\overline{B}_{q+1}$ observed from $w_0 = \phi$. □

Then using Lemma 7:

$$\lim_{q \to \infty} \frac{P(\overline{B}_{q+1})}{P(B_q)} = \lim \frac{P(\overline{B}_q \cap B*)}{P(\overline{B}_q)} = \frac{P(B \cap B*)}{P(B)} = P(B * | B).$$

Thus, by equation (3.9), to complete the proof of Proposition 3 and so of Proposition 2, it is sufficient to prove the following result.

Lemma 8. *If* $P(B) > 0$ *then* $P(B * | B) < 1$.

Lemma 8 appears to be clear for the following reason. Event B depends solely on what occurs along the branch from ϕ through w_{-1}, while $B*$ depends only on what happens along the branch through w_1. Although there is some dependency between these branches, this dependency is not expected to be strong enough to make $B*$ conditionally certain to happen. To formalize this idea, is required a study of the effect of conditioning the PWIT, that will be explained in the next section.

3.2.2 Proof Lemma 8

The proof of Lemma 8 requires a further construction that will be synthetized here. First, it is essential to define what is a bi-infinite tree. Recall that λ represents the distribution of the edge-weights of the PWIT, and that λ is a probability measure on $W = (0, \infty)^E$. To include a distinguished neighbour of ϕ, consider the state space $w \times \{1, 2, 3, \dots\}$ and introduce the σ-finite measure $\lambda\times$ `count`.

A different representation of this structure, useful for certain calculations with $\mathcal{M}_{opt}$ is now described. Take two copies of the PWIT, labeled T^+ and T^-, with their vertices denoted as $+v$ and $-v$. Then construct a new "bi-infinite" tree $T^{\leftrightarrow}$ by connecting the roots $+\phi$ and $-\phi$ of T^+ and T^- with a distinguished edge $(+\phi, -\phi)$. This new tree has an edge set denoted $E^{\leftrightarrow}$.

Now the edge weights in each of T^+ and T^- follow the distribution of the PWIT independently on both sides of $T^{\leftrightarrow}$. Then define a σ-finite measure $\lambda^{\leftrightarrow}$ on $W^{\leftrightarrow} \doteq (0, \infty)^{E^{\leftrightarrow}}$ by ensuring that the weight $W(-\phi, +\phi)$ on the distinguished edge has a uniform distribution over $(0, \infty)$, independent of the other edge-weights.

This construction is useful because there is a natural bijection between $W \times \{1, 2, 3, \dots\}$ and $W^{\leftrightarrow}$ which transforms $\lambda\times$ `count` to $\lambda^{\leftrightarrow}$. Specifically, if k is the distinguished vertex of T, the vertices can be relabeled according to the following rules:

1. Relabel k as $-\phi$.

2. Relabel j as $+j$ for $j \leq k - 1$ and as $+(j - 1)$ for $j \geq k + 1$.

3. Relabel descendants accordingly.

This relabeling process generates a map $\psi : W \times 1, 2, 3, \dots \to W^{\leftrightarrow}$ that is invertible. The same relabeling used to define ψ can be utilized to establish a family $(X(\overrightarrow{e}), \overrightarrow{e}$ being a directed edge of $T^{\leftrightarrow})$ based on the $X(\overrightarrow{e})$ on the directed edges of the PWIT constructed in Lemma 3. It can be verified that the joint distribution of $(W(e), X(\overrightarrow{e}); e, \overrightarrow{e}$ being edges of $T^{\leftrightarrow})$ obtained in this manner matches the distribution obtained by applying the construction in Lemma 3 to

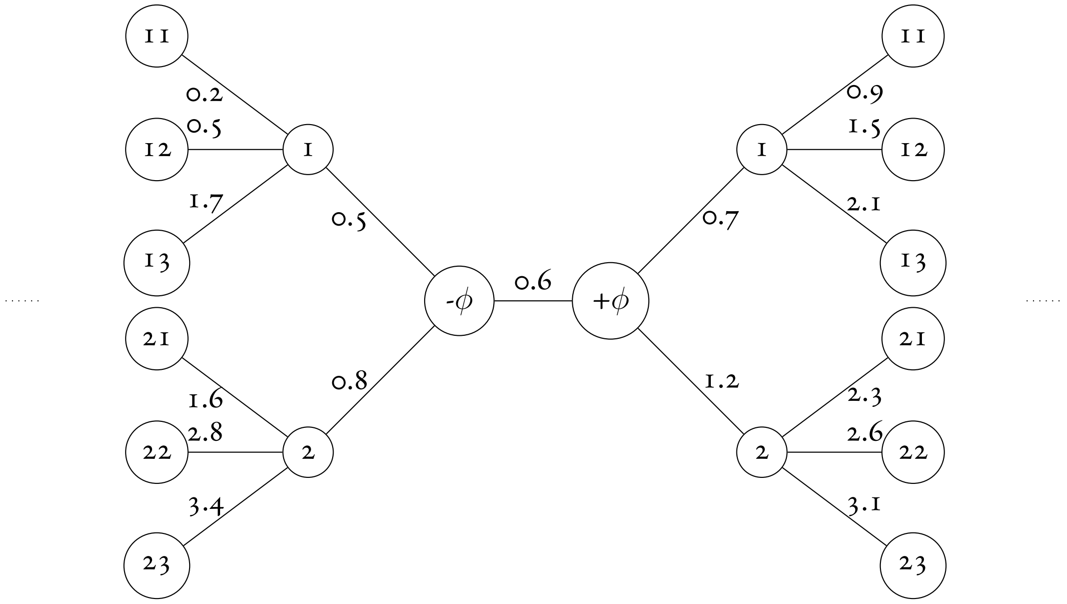


Figure 3.1: A bi-infinite tree with edge weights assigned according to a Poisson point process.

$(W(e), e \in E^{\leftrightarrow})$, substituting $\overrightarrow{E_h}$ in the construction with:

$$\overleftrightarrow{bE_h} = \{\overrightarrow{e} = (+v, +vj) : |v| = h, j \geq 1\} \cup \{\overrightarrow{e} = (-v, -vj) : |v| = h, j \geq 1\}\,. \tag{3.10}$$

Then the matching $\mathcal{M}_{opt}$ can be defined on $T^{\leftrightarrow}$ in the same way as on T:

$$e \text{ is an edge of } \mathcal{M}_{opt} \iff W(e) < X(\overrightarrow{e}) + X(\overleftarrow{e}).$$

The following lemma illustrates the usefulness of working with the bi-infinite tree for calculations. Informally, it describes the distribution of $\mathcal{M}_{opt}$ as viewed from a typical edge within $\mathcal{M}_{opt}$, and demonstrates a conditional independence property for the restrictions of $\mathcal{M}_{opt}$ on the two sides of the tree defined by that edge.

On the PWIT define $X^{\downarrow} = \min_{i \geq 1}(W(\phi, i) - X(\phi, i))$ and denote by ν_x the conditional distribution of the family $(W(e), e \in E; X(\overrightarrow{e}), \overrightarrow{e} \in \overrightarrow{E}, \overrightarrow{e}$ directed away from ϕ), given $X^{\downarrow} = x$. Returning to the bi-infinite tree, let λ^1 be the measure obtained by restricting $\lambda^{\leftrightarrow}$ to the set $\{W(-\phi, +\phi) < X(-\phi, +\phi) + X(+\phi, -\phi)\}$. So under λ^1, in $\mathcal{M}_{opt}$ the vertex $+\phi$ is almost surely matched with vertex $-\phi$.

Lemma 9. *λ^1 is a probability measure. Under λ^1 we have:*

1. *the joint density of $(W(-\phi, +\phi), X(-\phi, +\phi), X(+\phi, -\phi))$ at (w, x_1, x_2) is equal to $f(x_1)f(x_2)\mathbb{1}_{(0<w<x_1+x_2)}$, where f is the logistic density;*

2. *conditional on* $(W(-\phi,+\phi), X(-\phi,+\phi), X(+\phi,-\phi)) = (w, x_1, x_2)$*, the distribution of the family:*

$$\Big(W(e), e \in E^+; X(\overrightarrow{e}), \overrightarrow{e} \in \overrightarrow{E}^+, \overrightarrow{e} \textit{ directed away from } +\phi\Big),$$

is the image of ν_{x_1} *under the natural embedding* $T \to T^+ \subset T^{\leftrightarrow}$*; the distribution of the family:*

$$\Big(W(e), e \in E^-; X(\overrightarrow{e}), \overrightarrow{e} \in \overrightarrow{E}^-, \overrightarrow{e} \textit{ directed away from } -\phi\Big),$$

is the image of ν_{x2} *under the natural embedding* $T \to T^- \subset T^{\leftrightarrow}$*; and these two families are conditionally independent.*

An important observation is that because $+\phi$ is matched to $-\phi$, $X(+j,+\phi) = W(+\phi,-\phi) -X(+\phi,-\phi)$, one can recursively construct $X(\overrightarrow{e})$ for $\overrightarrow{e}$ directed toward $(-\phi,+\phi)$. So Lemma 9 is sufficient to specify the joint distribution of all the $X(\overrightarrow{e})$ and, consequently, of $\mathcal{M}_{opt}$ under λ^1.

Proof of Lemma 9. The joint density takes the form given in the first point of the lemma by construction, and so its total mass is $\int\int (x_1+x_2)^+ f(x_1) f(x_2) dx_1 dx_2$ which is equal to 1 as shown in Lemma 2. Moreover, using the construction based on $(X(\overrightarrow{e}), \overrightarrow{e} \in \overleftrightarrow{bE_h})$ at 3.10, it is possible to see that under $\lambda^{\leftrightarrow}$ the families:

$$\Big\{W(e), e \in E^+; X(\overrightarrow{e}), \overrightarrow{e} \in \overrightarrow{E}^+, \overrightarrow{e} \text{ directed away from } +\phi\Big\} \cup X(-\phi,+\phi),$$

and

$$\Big\{W(e), e \in E^-; X(\overrightarrow{e}), \overrightarrow{e} \in \overrightarrow{E}^-, \overrightarrow{e} \text{ directed away from } -\phi\Big\} \cup X(+\phi,-\phi),$$

are independent from each other and from $W(-\phi,+\phi)$. Since λ^1 is defined by an event that depends only on $\{X(+\phi,-\phi), X(-\phi,+\phi), W(+\phi,-\phi)\}$ the desired conditional independence property is obtained. Each family under $\lambda^{\leftrightarrow}$ is the distributed as the image of the corresponding family on the PWIT (with $X(-\phi,+\phi)$ corresponding to $X^{\downarrow}$), and therefore, due to the independence under $\lambda^{\leftrightarrow}$, the conditional distribution under λ^1 depends only on x_1 (respectively x_2). □

Now recall the map $\psi : W \times \{1,2,3,\dots\} \to W^{\leftrightarrow}$ which takes $\lambda \times$ `count` to $\lambda^{\leftrightarrow}$. The inverse image of the event $\{W(-\phi,+\phi) < X(-\phi,+\phi) + X(+\phi,-\phi)\}$, under this map, is:

$$\psi^{-1}\left\{W(-\phi,+\phi) < X(-\phi,+\phi) + X(+\phi,-\phi) = \mathcal{M}_{opt}(\phi) \text{ is distinguished }\right\}.$$

Thus the inverse image of the probability measure λ^1 is obtained by restricting $\lambda\times$ count to the set where the vertex $\mathcal{M}_{opt}(\phi)$ is distinguished. After removing this distinction, what remains is the probability distribution λ on the PWIT.

In summary what was done was establishing the following "relabeling principle".

> Given $(Z(\overrightarrow{e}))$ and hence $\mathcal{M}_{opt}$ on the PWIT, map the whole structure to the bi-infinite tree by relabeling $(\phi, \overrightarrow{\mathcal{M}}_{opt}(\phi))$ as $(+\phi,-\phi)$ and relabeling other vertices accordingly; then the resulting distribution on the bi-infinite tree is λ^1.

Then on the bi-infinite tree define:

$$C* \doteq \left\{+\phi = \arg\min^{[2]}_{y\sim +I}(W(+I,y) - X(+I,y))\right\},$$

where:

$$I = \arg\min_{i\geq 1}\left(W(+\phi,+i) - X(+\phi,+i)\right).$$

$$\mathcal{F}^{+} = \sigma\left(X(\overrightarrow{e}), W(e) : e, \overrightarrow{e} \text{ edges of } T^{+}\right),$$
$$\mathcal{F}^{-} = \sigma\left(X(\overrightarrow{e}), W(e) : e, \overrightarrow{e} \text{ edges of } T^{-}\right),$$
$$\mathcal{F}^{\phi} = \sigma\left(X(+\phi,-\phi), X(-\phi,+\phi), W(+\phi,-\phi)\right).$$

Now consider the two following results, each provided with its corresponding proof:

Lemma 10. *Define:*

$$X^{\downarrow} = \min_{i\geq 1}(W(\phi,i) - X(\phi,i)),$$
$$I = \arg\min_{i\geq 1}(W(\phi,i) - X(\phi,i)).$$

For $-\infty < b < a < \infty$ *define:*

$$g(a,b) = P(W(\phi,I)) = \min_{k\geq 1}(W(I,Ik) - X(I,Ik)).$$

Then $g(a,b) > 0$.

Proof. Given $X(\phi,I) = x$, the remaining values of $\{W(I,I_k) - X(I,Ik), k \geq 1\}$ form an inhomogeneous Poisson process on (x,∞), and so:

$$P\left(\min^{[2]}_{k\geq 1}\left(W(I,Ik) - X(I,Ik)\right) \in [y,y+dy] | X(\phi,I) = x\right) = \beta_x(y)dy, \quad (3.11)$$

for a certain function $\beta_x(\cdot)$ such that $\beta_x(y) > 0$ for all $y > x$. The quantities in (3.11) are independent of $W(\phi, I)$. Thus:

$$\begin{aligned}\tilde{g}(a,b,x) &\doteq P(W(\phi,I) - b \\ &\qquad > \min\nolimits^{[2]}_{k\geq 1}(W(I,Ik) - X(I,Ik))|X(\phi,I) = x, W(\phi,I) = a + x),\end{aligned}$$

satisfies $\tilde{g}(a,b,x) > 0$ for all $-\infty < b < a < \infty$ and $-\infty < x < \infty$. Since $X^{\downarrow} = W(\phi,I) - X(\phi,I)$, it is possible to see:

$$g(a,b) = \mathbb{E}\left(\tilde{g}(a,b,X(\phi,I))|X^{\downarrow} = a\right) > 0,$$

as required. □

Lemma 11. $\lambda^1\{C*^C|\mathcal{F}^-,\mathcal{F}^\phi\} = g(X(-\phi,+\phi), W(+\phi,-\phi) - X(+\phi,-\phi))$ *for g defined in the previous Lemma.*

Proof. $C*$ is $\mathcal{F}^+$-measurable, so by the conditional independence assertion of Lemma 9 comes that:

$$\lambda^1\{C*^C|\mathcal{F}^-,\mathcal{F}^\phi\} = \lambda^1\{C*^C|\mathcal{F}^\phi\}.$$

Therefore, it is necessary to demonstrate that:

$$\lambda^1\left\{C*^C|(W(-\phi,+\phi), X(-\phi,+\phi), X(+\phi,-\phi)) = (w, x_1, x_2)\right\} = g(x_1, w - x_2). \tag{3.12}$$

Under this conditioning, Lemma 9 implies that the family $(W(e), e \in E^+; X(\overrightarrow{e}), \overrightarrow{e} \in \overrightarrow{E}, \overrightarrow{e}$ directed away from $+\phi)$ is distributed as the image of the family $(W(e), e \in E; X(\overrightarrow{e}), \overrightarrow{e}$ directed away from $\phi)$ conditioned on $\{X^{\downarrow} = x_1\}$. According to the definition of $g(a,b)$ in Lemma 10,

$$\begin{aligned}g(x_1, w - x_2) = \lambda^1\{&W(+\phi,+I) - (w - x_2) \\ &> \min^{[2]}_{y\sim +I}(W(+I,y) - X(+I,y)) \\ &|(W(-\phi,+\phi), X(-\phi,+\phi), X(+\phi,-\phi)) = (w, x_1, x_2)\}.\end{aligned}$$

However, under this conditioning

$$\begin{aligned}W(+\phi,+I) - (w - x_2) &= W(+\phi,+I) - (W(+\phi,-\phi) - X(+\phi,-\phi)) \\ &= W(+I,+\phi) - X(+I,+\phi),\end{aligned}$$

by (2.8). □

Because under λ^1 the vertex $+\phi$ is always matched to $-\phi$. Substituting into (3.12), is possible to see that the event in (3.12) is precisely the event $C{*}^C$ as required. So finally, it is possible to complete the proof of Proposition 2 by proving Lemma 8.

Proof of Lemma 8. The relabeling principle shows that $P(B * |B)$ can be rewritten as $\lambda^1\{C * |C\}$, for a certain event C which is $\mathcal{F}^\phi$-measurable and such that $P(B) = \lambda^1\{C\}$. Now:

$$\begin{aligned}\lambda^1\{C *^C |C\} &= \mathbb{E}_{\lambda^1}\mathbb{1}_C g(X(-\phi,+\phi), W(+\phi,-\phi) - X(+\phi,-\phi)) \text{ by Lemma 11}\\ &> 0 \text{ if } \lambda^1\{C\} > 0 \text{ by Lemma 10,}\end{aligned}$$

establishing Lemma 8. □

4

Privacy preserving wireless communication using Bipartite Matching

To emphasize the significance of the random assignment problem and the importance of estimating algorithmic costs, this chapter will explore a practical application related to the problem analyzed in this thesis. Specifically, it will focus on the article: "Privacy preserving wireless communication using bipartite matching in social big data" [2] which explain how to employ a bipartite matching method to address a privacy protection issue. The first section of this chapter will introduce the privacy protection problem, detailing the proposed model and providing an illustrative example to clarify the construction process. Following that, in Section 4.2, will be presented the algorithm, its application to the initial example and briefly explained the experimental results presented in the article.

4.1 Introduction to the problem

Recent research has explored the development of advanced privacy protection measures in wireless communications, a crucial factor for boosting the performance of electronic infrastructure to satisfy the varying demands across fields like social networks and big data. Many technologies in this field apply multi-channel communication systems, which possess the flexibility to switch channels based on immediate needs and real-time channel conditions.

The primary benefit of using multi-channel communication over static channel configurations lies in its ability to efficiently connect service requesters with providers, avoiding traffic congestion and minimizing channel conflicts. In addition to that, multi-channel systems optimize interconnections by striking a balance between privacy protection and response times. However, in practical scenarios, communication can be disrupted due to inefficient response times, often coming from poor channel scheduling. This challenge becomes more significant as data volumes expand within social networks, and the complexity of channel scheduling is further compounded when integrating privacy considerations with the need for dynamic, adjustable communication. Achieving higher levels of privacy protection within the same time constraints generally results in increased energy consumption. This issue becomes more pronounced as data volumes increase in social networks, and the complexity of channel scheduling is even worsened when privacy considerations and the need for dynamic, adjustable communication are taken into account.

To address this issue, the analyzed paper proposes a novel approach that optimizes privacy protection in wireless communications through an efficient task scheduling method. The focus is on achieving privacy-preserving scheduling by enhancing the execution of Channel Scheduling Controller (CSC) systems. The primary strategy involves dynamically selecting different communication channels across various social networks to enhance privacy protection. The authors introduced an optimal task scheduling method, focusing on the social networks scenario where multiple communication channels are needed between users. The goal of their model is to allocate tasks across heterogeneous nodes, which in this context represent connection points that can both transmit wireless signals and perform optional processing computations.

The proposed model utilizes channel scheduling techniques to enhance privacy protections while adhering to timing constraints and employs a heuristic method called Task Scheduling by Bipartite Matching (TSBM). This method obtains a near-optimal solution with minimal computational cost. Experimental results demonstrate that the TSBM strategies can reduce privacy protection costs by 23.37% with four nodes when the time constraint is relaxed.

4.1.1 Overview of the Channel scheduling Controller model

The CSC model introduced in the study deals with a network of M nodes, denoted as a set $N = \langle N_1, N_2, \ldots, N_M \rangle$, each responsible for managing both computational tasks and communication processes. In the context of social networks, these nodes correspond to physical devices that handle the distribution of information and execute certain computations. Each node serves as an interconnecting point for various media, where task scheduling is arranged by CSC systems.

The architecture of a CSC model is relatively straightforward. In its simplest form it can be represented as two nodes, N_1 and N_2, connected by a direct communication link. This link represents the point-to-point communication between the nodes and, in some cases, this communication link can be thought as a separate node with an associated weight that quantifies the communication delay associated with the link.

The Task Data Flow Graph (TDFG) is modeled using Directed Acyclic Graphs (DAGs), denoted as $G = \langle V, E, T \rangle$ where each node in V represents a task, and the set E comprises directed edges that define the precedence relationships between tasks. For example, an edge $E(u \rightarrow v)$ indicates that task v can only start after task u finishes, with the edge weight reflecting the communication cost between these tasks. The set T contains the execution times for tasks, where $Tv_i = \langle T(i,1), T(i,2), \ldots, T(i,M) \rangle$ specifies the computation time for task v_i on each node N_x. The TDFG model, exemplified by a graph with 7 tasks in Figure 4.1, shows how tasks from social networking applications are partitioned into coarse-grained units, such as functions or code blocks. These coarse-grained tasks have higher computation costs compared to finer tasks, but their communication overhead is null when scheduled on the same node due to a high Computation-to-Communication Ratio (CCR).

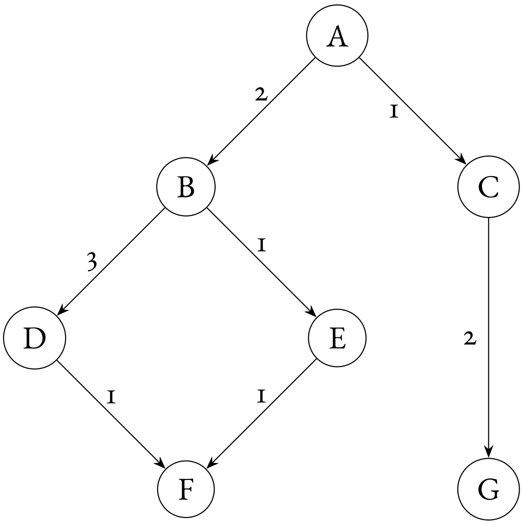


Figure 4.1: Example of a Task Data Flow Graph (TDFG), where nodes represent tasks, directed edges indicate task precedence, and edge weights reflect communication costs between tasks.

4.1.2 The privacy-preserving model

The privacy-preserving model described aims to maximize privacy protection in CSC systems. However, before evaluating the system's privacy protection level, it is necessary to first develop a task schedule for an application across nodes in the target system. An illustrative example of task scheduling is illustrated in Figure 4.2 involving 7 task nodes and 3 communication nodes.

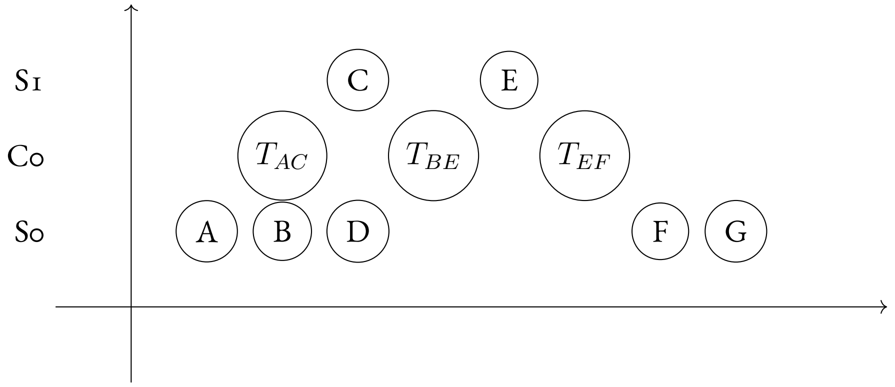


Figure 4.2: A simple task scheduling plot that provides a visual representation of the task dependencies and execution flow.

An important assumption here is that the task completion failure rate is zero, ensuring that tasks are completed without any failures in the computation components.

Now can be defined the privacy protection component as follows: Consider a system with M node and define F_i and $T(i, j)$ as the failure rate of the component S_i and the computation time of task v_i on the component j, respectively. The failure rate refers to the likelihood of the protection failure attached to a type of defining method, caused either by a processor or by an interconnection between processors. It is denoted as F_i and in this analysis the hypothesis is that each component in the system experiences failures that adhere to a Poisson distribution so that the time until a failure occurs is modeled by an exponential distribution. The computational time $T(i, j)$ is counted in the unit of the time depending on the implementation scenario. The cost of the privacy protection is associated with $T(i, j)$ using the same timing counts. Based on the system's scheduling mechanisms, the level of privacy protection, denoted as PPL, can be formulated as indicated in the following equation, based on the scheduling.

$$\begin{aligned} PPL &= \prod_{i=1}^{N}\prod_{j=1}^{M}(1 - F_j)^{T(i,j)} \\ \implies \lim_{F_j \to 0} PPL &= \prod_{i=1}^{N}\prod_{j=1}^{M}(e)^{-F_j \cdot T(i,j)} \text{ when } F_j \text{ is very small.} \end{aligned} \tag{4.1}$$

Then it is possible to prove that the problem of maximizing the overall privacy protection level can be rewritten as minimizing the privacy protection cost, that can be defined as:

$$PC = \sum_{i=1}^{N}\sum_{j=1}^{M} F_j \cdot T(i, j).$$

Proof. Equation (4.1) implies:

$$
\begin{aligned}
(1-F_j)^{T(i,j)} = e^{T(i,j)\ln(1-F_j)} \text{ and } \lim_{F_j \to 0} \ln(1+F_j) = F_j \\
\implies \lim_{F_j \to 0} (1-F_j)^{T(i,j)} = e^{-F_j \cdot T(i,j)}.
\end{aligned}
$$

The expression of the privacy protection in $\lim_{F_j \to 0}(1-F_j)^{T(i,j)} = e^{-F_j \cdot T(i,j)}$ is nonlinear. It is possible to obtain its maximum by utilizing an equivalent linear form as (4.2) through a logarithmic operation. Its important to notice that $\lim_{F_J \to 0}(1-F_j)^{T(i,j)} = e^{-F_j \cdot T(i,j)}$ and (4.2) has the same optimal solution for maximization, although they may have different function values.

$$
\begin{aligned}
\sum_{i=1}^{N}\sum_{j=1}^{M} \ln(e^{-F_j \cdot T(i,j)}) &= \sum_{i=1}^{N}\sum_{j=1}^{M}(-F_j \cdot T(i,j)) \\
\implies PC &= \sum_{i=1}^{N}\sum_{j=1}^{M} F_j \cdot T(i,j) \\
\implies PC_{i,j} &= F_j \cdot T(i,j).
\end{aligned}
\tag{4.2}
$$

In order to maximize the overall system privacy protection level PPL is sufficient to maximize the value of (4.2). Since PPL is a decreasing function, the maximal PPL can be obtained by minimizing $\sum_{i=1}^{N_{DAG}} \sum_{j=1}^{M} F_j \cdot T(i,j)$. This expression is denoted as privacy cost PC of the target CSC system. Then a variable $PC_{i,j}$ is introduced to denote the privacy protection cost of allocating task v_i to node j. For a specific task, the privacy protection cost can be represented by the step of equation 4.2. The overall system privacy protection cost PC for a CSC is calculated as the summation of the privacy protection cost incurred by allocating each task in the system formulated by the following equation.

$$
PC = \sum_{i=1}^{N_{DAG}} \sum_{j=1}^{M} PC_{i,j}.
$$

□

Therefore, the problem of maximizing the system's privacy protection level is equivalent to minimize the privacy protection cost of the system. The privacy protection cost indicates the expense incurred by the system when a set of real-time tasks is assigned to it.

4.1.3 Example of Task Scheduling and its Impact on Privacy Protection Costs

In this section is presented an example to illustrate the problem and demonstrate the impact of different task scheduling outcomes on the privacy protection cost of a system. Consider a CSC with two nodes, namely N_0 and N_1, each with low failure rates assumed to be 0.6 and 0.2, respectively. The communication costs between inter-task connections are randomly generated and represented by edge weights in Figure 4.1. Additionally, the failure probability of the communication link between N_0 and N_1 is assumed to be 0.1. A task DAG representing a simple application in social networks is depicted in Figure 4.1, and the execution times for each task on N_0 and N_1 are detailed in Table 4.1.Suppose the given time constraint for this application is 16 time units. In the figure 4.3 there are some examples of valid schedules, where the arrows represent the communication links between 2 nodes. All three schedules meet the time constraint which is based on the given DAG in Figure 4.1. The schedule lengths of these three scheduling methods are 16, 11 and 16 respectively. The privacy protection costs can be computed as the sum of three different components, which for the first scheduling method are:

Data	A	B	C	D	E	F	G
N_0	2	2	3	3	5	3	4
N_1	2	4	2	4	3	3	5

Table 4.1: Example of data. In this table, based on the network in figure 4.1 are represented the execution time of each task in N_0 and N_1.

- Privacy protection costs of the operations scheduled on N_0, which is the product of the failure rate of N_0 with the sum of the time for operations A,B,D,F,G if scheduled on N_0: $0.6 \times (2 + 2 + 3 + 3 + 4) = 8.4$.

- Privacy protection costs of the operations scheduled on N_1, which is the product of the failure rate of N_1 with the sum of the time for operations c and E if scheduled on N_1: $0.2 \times (2 + 3) = 1$.

- Privacy protection costs of the communication links, which can be computed as the failure rate of the communication link, multiplied by the times of the communication between operations on different nodes, that in this case are between A and C, B and E, E and F and C and G. The result of this operation is : $0.1 \times (1 + 1 + 1 + 2) = 0.5$

By summing all these costs is possible to see that the privacy protection cost of the first scheduling in figure 4.3 is $8.4 + 1 + 0.5 = 9.9$ and in a similar way can be computed that for the second one is 8.7 and 7.5 for the third one. By comparison of these scheduling can be

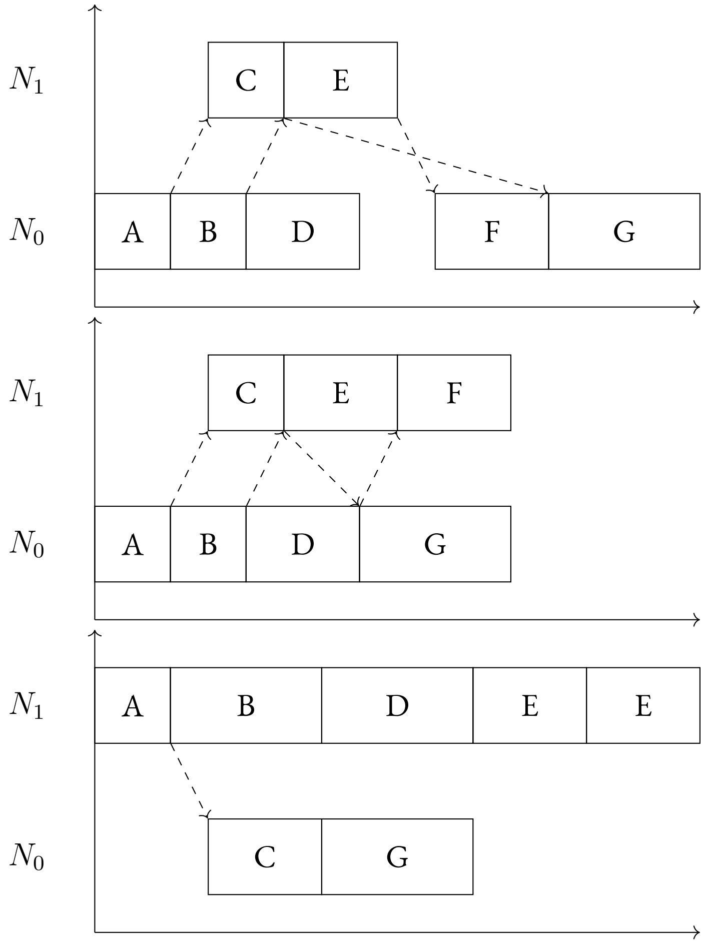


Figure 4.3: Plot of three valid scheduling, based on the DAG in figure 4.1.

seen that the second one leads to a reduction in schedule length by 31.25% and a reduction in privacy protection cost by 12.12%, while the third one achieves a reduction in the privacy protection cost by 24.24% within the time constraint.

This example gives two optimal scheduling for different metrics, the second scheduling in figure 4.3 has the shortes scheduling time and the third one has the lowest privacy protection cost. Therefore there are two results that follows:

1. Scheduling strategies significantly impact on the privacy protection cost and the schedule length of an application.

2. The shortest scheduling does not always yield the smallest privacy protection cost.

4.2 Proposed algorithm for task-scheduling

This section will outline the formulation of the privacy protection optimization problem within a CSC system. The problem is quite complex, in fact task scheduling with real-time constraint is classified as an NP-complete problem and as a consequence is also true that optimizing privacy protection costs for heterogeneous CSC systems with real-time requirements is NP-complete. Due to this complexity, the introduction of an heuristic algorithm based on a bipartite matching strategy is highly beneficial because this approach can closely approximate the optimal solution without excessive computational time, which is crucial for practical applications in such systems.

4.2.1 Problem definition

The privacy protection cost optimization problem for CSCs is defined as follows:

- A target CSC system with M nodes represented by a the set $N = \langle N_1, N_2, \ldots, N_M \rangle$.
- A set of N tasks organized in a DAG G= $\langle V, E \rangle$.
- The execution times of each task at different nodes.
- A timing constraint TC that can result in the minimum privacy protection costs for the system.

The proposed heuristic algorithm to analyze this complex issue incorporates a bipartite matching technique which re -schedules tasks based on an initial scheduling arrangement. This Task Scheduling by Bipartite Matching (TSBM) strategy specifically focuses on efficiently realigning tasks to balance privacy protection costs against real-time constraints. Its key steps can be explained as follows.

1. Using an *As Last As Possible* (ALAP) algorithm to schedule the given DAG and gain the *Latest Start Time* (LST) for each task.
2. Constructing a bipartite matching graph by putting the schedulable tasks in the task set.
3. Rescheduling the tasks to different nodes by using a bipartite-matching algorithm to minimize the privacy protection cost.

These three stages, summarized here, are detailed in Algorithm 4.1, Algorithm 4.2, and Algorithm 4.3.

Algorithm 4.1 Initial scheduling for TSBM algorithm.

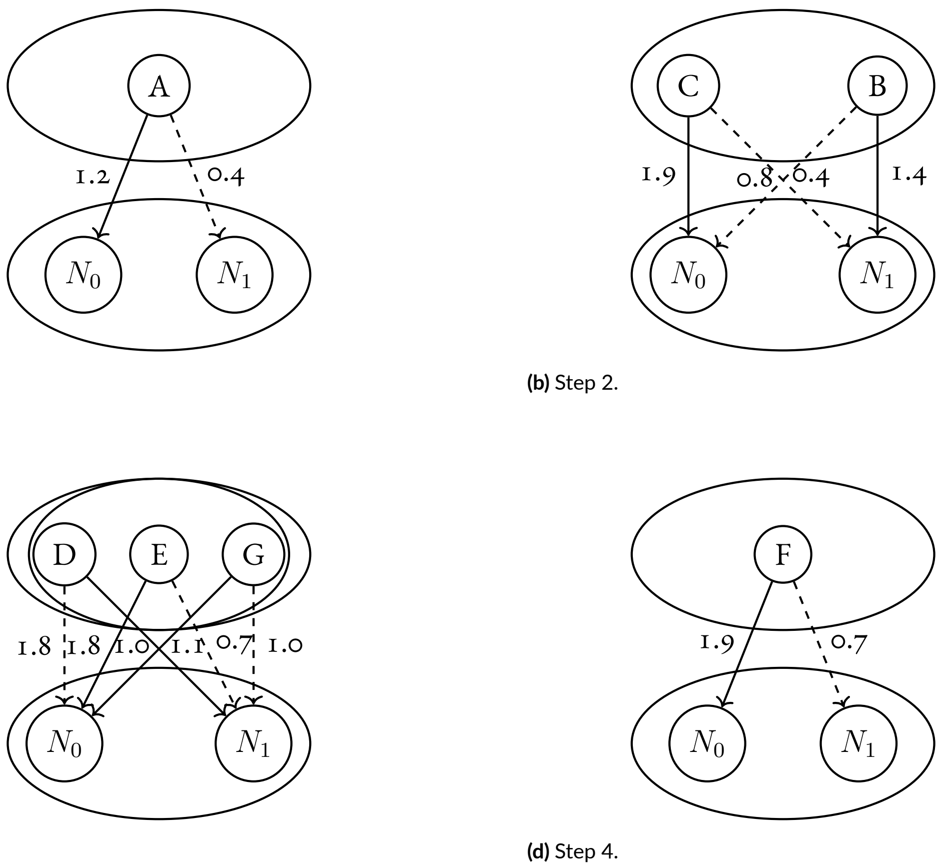
```
Require: A DAG $G = \langle V, E \rangle$.
Ensure: An initial scheduling of the DAG $G$.
Using as soon as possible scheduling to gain the earliest star-time for each task.
Using as late as possible scheduling to gain the latest start-time and the finish-time for each task.
for all $N_j \in N$
    SL($N_j \leftarrow 0$);
end for
Set the overall privacy level: $OPC \leftarrow 0$;
return The initial scheduling;
```

Algorithm 4.2 Construction of the bipartite matching graph.

```
Require: A DAG $G = \langle V, E \rangle$, a node set $V_p$, a task set $V_T$
Ensure: A bipartite matching graph $GB = \langle VB, EB \rangle$
for all $v_i \in V_T$
    for all $N_j \in V_p$
        $tmpSL_j \leftarrow SL(N_j)$
        Use equation (4.2) to calculate $PC_{i,j}$
        Update $EB$ by adding $e(v_i, N_j)$ between $v_i$ and $N_j$
        if $\max(tmpSL_j, EST(v_i)) + T(i,j) \leq FT(v_i)$
            $W(EB(v_i, N_j)) \leftarrow PC_{i,j}$
        else
            $W(EB(v_i, N_j)) \leftarrow \infty$
        end if
    end for
end for
$VB \leftarrow V_T \cup V_p$
return $GB$
```

Algorithm 4.3 Task scheduling by bipartite matching (TSBM)

```
Require: A DAG $G = \langle V, E \rangle$, task set $V_T$, processor set $V_P$, and task constraints $TC$
Ensure: A schedule with the overall privacy protection cost $OPC$
Use Algorithm 4.1 to obtain an initial scheduling
while not AllTasksVisited()
    for all $N_j \in N$
        if a schedulable task $v_i$ with the earliest start-time $EST(v_i)$ is found at node $N_j$
            Insert task $v_i$ into $V_T$
        end if
        Insert node $N_j$ into $V_s$
    end for
    Using Algorithm 4.2 to construct $GB = \langle VB, EB \rangle$
    for all $i = 1 \rightarrow |EB|$
        $e(v_i, S_j) \leftarrow$ minimal weight edge $W(EB(v_i, P_j))$
        $TmpSL \leftarrow \max(EST(v_i), SL(S_j)) + T(i, S(v_i))$ where $SL$ denote the schedule length
        $W(EB(v_i, S_j)) \leftarrow$ minimal weight edge $W(EB(v_i, P_j))$
        $TmpSL \leftarrow \max(EST(v_i), SL(S_j)) + T(i, S(v_i))$
        if $TmpSL \geq FT(v_i)$
            $W(EB(v_i, S_j)) \leftarrow \infty$
            continue
        end if
        Set $v_i$ to be *visited*; $N(v_i) \leftarrow N_j$
        $OPC \leftarrow W(EB(v_i, N_j)) + OPC$
        $SL(N(v_i)) \leftarrow TmpSL$
    end for
    for all $v_k \in Succ(v_i)$
        Delete precedence dependency between $v_k$ and $v_i$
        if $EST(v_k) < SL(N(v_i))$
            $EST(v_k) \leftarrow SL(N(v_i))$
        end if
    end for
end while
return $OPC$
```



(a) Step 1.

(b) Step 2.

(c) Step 3.

(d) Step 4.

Figure 4.4: Visual representation of the procedure described in Algorithm 4.3.

4.2.2 Example of application

To show the utility of the proposed algorithm this section will present its application to the example introduced in section 4.1.3. In this case, an initial scheduling is obtained using the ALAP scheduling method showed in algorithm 4.1. Based on this initialization it can be subsequently achieved a scheduling that assigns tasks A, B, C, E and F to N_0 and task D and G to N_1. Meanwhile is also possible to get that the finish times of task A, B, C, D, E, F and G are $2, 6, 11, 15, 14, 18$ and 20. According to the inital schedule, it can be employed the bipartite matching strategy to get a tighter and more reliable schedule for the target CSC system by first applying algorithm 4.2 to construct the bipartite matching associated with the initial scheduling and the utilize algorithm 4.3 to improve the scheduling, as showed in figure 4.4. The algorithm 4.3 starts by comparing the weights of the edges in the constructed bipartite matching graph for the node A. This evaluation determines that assigning task A to node N_1 results in the lowest privacy protection cost. Consequently, task A is reassigned to N_1 following the TSBM scheduling method. Following this reassignment, task A is marked as visited, and are updated the related information for its descendant tasks. These updates include revising the Earliest Start Times (ESTs) and adjusting for precedence constraints, as detailed in lines 18 to 28 of Algorithm 4.3. As a result, the earliest start times are adjusted, setting task B to start at time 4 on node N_0 and task C to begin at time 2 on node N_1. Once these tasks become schedulable, task A, along with tasks B and C, are added to the set V_T. Following this, is constructed a bipartite matching graph for tasks B and C, as illustrated in Figure 4.4b, by repeating the same steps. From this newly formed graph, it becomes apparent that the edge $e(C, S_1)$ represents the lowest privacy protection cost. Consequently, task C is allocated to node N_1, marked as completed, and the schedule length for N_1 is updated to 4 time units. In the subsequent scheduling phase, although the edge $e(B, N_1)$ initially appears to offer the lowest privacy protection cost, scheduling task B to node N_1 would result in a schedule length of 8. This length surpasses the required finish time of task B, which is 6. Therefore, to ensure proper execution, task B must be allocated to node N_0. Once this decision is made, tasks D, E, and G become eligible for execution. The the same scheduling strategies are applied until all tasks are marked as visited. The scheduling for the remaining tasks is illustrated in figures 4.4c and 4.4d.

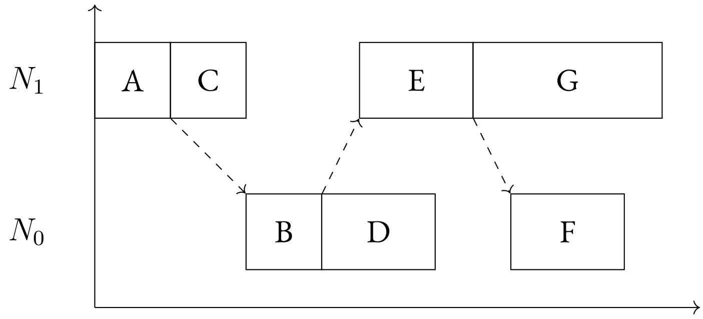


Figure 4.5: Final task scheduling, obtained utilizing the proposed algorithm.

The outcome of applying the TSBM algorithm to the Directed Acyclic Graph (DAG) depicted in 4.1 is presented in Fig. 4.5. Through this scheduling the privacy protection cost is 7.6, which closely aligns with the optimal value of 7.5 showed in Section 4.1.3. This demonstrates the algorithm's capability to efficiently reduce privacy protection costs in CSC systems.

The key insight from the algorithm is that the bipartite matching process for a task is constrained by its finish time as defined by the As Late As Possible (ALAP) scheduling. This constraint ensures that, in the worst-case scenario, a task can be assigned to the same node as it would be in the ALAP scheduling. Consequently, TSBM consistently produces a valid schedule for tasks in a DAG, provided that a valid ALAP schedule exists. This robustness and efficiency in achieving near-optimal scheduling for privacy protection costs underscore the algorithm's effectiveness for systems requiring both efficient task scheduling and privacy considerations.

4.2.3 Experimental results

In the concluding section of the article, were presented the experimental results of the proposed TSBM algorithm. Its performances were evaluated by comparing it against the Integral Linear Programming (ILP) algorithm, which was introduced in the same article, and the existing *Highest Level First with Estimated Times* (HLFET) algorithm [28]. When comparing TSBM with the ILP algorithm, it's crucial to recognize that ILP is designed to find an optimal solution to the scheduling problem, though it requires substantial computational time. In contrast, the TSBM algorithm is directed towards finding an approximate solution more quickly. Consequently, while superior accuracy is expected from the ILP method, a comparison with the HLFET algorithm is sufficient to demonstrate the practical applicability of the TSBM approach. The HLFET algorithm provides a reasonable benchmark because it focuses on efficiently scheduling tasks based on their priority levels without the extensive computational overhead associated with ILP.

As a metric was defined the *Tight Time Constraint* (TTC) as the sum of the average minimal execution time of each NDAG tasks on M processors.

$$MinT() = \min\{T(i,1), T(i,2), \dots, t(i,M)\},$$

$$TTC = \frac{\sum_{i=1}^{N} MinT(i)}{M}.$$

Next, was applied the *Relax Time Constraint* (RTC) with a relax coeffiecient α.

$$RTC = \alpha \cdot TTC.$$

In this final equation, the parameter α served as an input control to modulate the strictness of the timing constraints. By adjusting the value of α these constraints can be tightened or relaxed. The α values were set to vary within a range from 1.2 to 2.0. The provided experiments included two different heterogeneous configurations for the target systems: The first configuration involved 2 nodes, while the second one utilized 4 nodes. The failure rates for each processor and the communication links between processors were modeled using the Poisson distribution.

The experimental results were measured using the *Success Rate*, defined as the ratio $\frac{N_{succ}}{N}$, where N_{succ} is the number of successfully executed tasks, and N is the total number of tasks. The performances of the three algorithms: ILP (Integer Linear Programming), TSBM (Task Scheduling with Bipartite Matching), and HLFET (Highest Level First with Estimated Times), were evaluated under different timing constraints, represented by the relaxation coefficient α for which the obtained results can be summarized as follows:

- Under tight timing constraints ($\alpha = 1.2$), ILP and TSBM outperformed HLFET, with success rates of about 84% for ILP, 78% for TSBM, and 47% for HLFET. ILP's optimal scheduling and TSBM's effective task reorganization using ALAP led to their higher success rates compared to HLFET, which struggled under strict timing constraints.

- When the timing constraint was moderately relaxed ($\alpha = 1.6$), the success rates for both the ILP and TSBM algorithms improved significantly. In contrast, the HLFET algorithm saw only a modest increase in performance and remained less effective overall. This suggests that ILP and TSBM were better able to accommodate and schedule tasks under less stringent timing constraints.

- Under extensively relaxed timing constraints ($\alpha = 2.0$), all three algorithm achieved similarly high success rates, successfully scheduling almost all tasks. This indicates that with wider time range, the performance differences between the algorithms diminish, and each can meet the scheduling requirements effectively.

The analysis also investigated the privacy protection costs of each algorithm. In systems with only two nodes, the HLFET algorithm exhibited the lowest privacy protection costs under tight timing constraints ($\alpha = 1.2$ and 1.6). This was due to fewer scheduled tasks, resulting in reduced overall costs. In contrast, TSBM incurred higher costs compared to ILP, despite having a slightly lower success rate.

With a further relaxation of the timing constraint ($\alpha = 2.0$), both ILP and TSBM algorithms demonstrated improved efficiency in reducing privacy protection costs compared to HLFET. On average, ILP reduced costs by 28.33%, and TSBM by 21.33%. In systems with four nodes, overall privacy protection costs were higher across all algorithms. However, ILP and TSBM still significantly outperformed HLFET. Under relaxed constraints, ILP and TSBM achieved average reductions in privacy costs of 32.62% and 23.37%, respectively. TSBM demonstrated increasingly noticeable performance benefits as timing constraints tightened. It enhanced system privacy protection in many cases by leveraging ALAP scheduling, although its effectiveness was slightly less pronounced compared to ILP under strict constraints. However, TSBM proved to be more computationally efficient, requiring less time to generate scheduling solutions compared to ILP.

In summary, TSBM achieved a balanced approach between efficient scheduling and cost management, particularly beneficial under moderate timing constraints. HLFET, although less effective under tight scheduling conditions, demonstrated lower privacy protection costs in scenarios with fewer scheduled tasks. As timing constraints relaxed, the performance distinctions between the algorithms diminished, with all algorithms demonstrating effective task scheduling and management of privacy protection costs.

5

Conclusions

This thesis focused on the analysis of the Random Assignment Problem and its bounds, a continuous challenge in combinatorial optimization. For many years, this problem remained unresolved until improvement were achieved through connections with other fields of study. Surprisingly the most significant advancement came from physicists Giorgio Parisi and Marc Mézard that utilizing the replica method and the theory of spin glasses, proposed a conjecture for the bound of the bipartite matching problem. The first chapter of this work was in fact focused on the historical and theoretical foundations of the problem. In this beginning part was illustrated the evolution of the problem from early contributions by Monge, Frobenius, and König to modern interpretations, culminating in Parisi and Mézard result, with also a replica of their experiment to prove the validity of the bound. The central focus of this thesis, comprehensively covered in Chapters 2 and 3, is the detailed explanation of Aldous's proof of the bound derived using the theory of spin glasses. This proof specifically addresses the expected value of the optimal allocation in the random assignment problem. The main discovery that came from the analysis of Aldous's work is that the bipartite matching problem is equivalent to find a matching on the Poisson Weighted Infinite Tree, as developed in Chapter 2, and that the estimation of the bound can be derived from the properties of this optimal matching. Chapter 3 explained deeper the mathematical relationships between these two problems by proving the equivalence between the bipartite matching problem and the PWIT model.

Following the theoretical analysis, the thesis moved to a practical applications of the Bipartite matching Problem which focused on enhancing privacy and efficiency within wireless communication systems. In Chapter 4 was examined how the principles of bipartite matching can be used to develop algorithms for privacy-preserving task scheduling in social big data environments. This investigation led to the use of the Channel Scheduling Controller model, a system which is a good example of how theoretical advancements can be effectively adapted to address modern challenges in data security within social networks.

This work showed both the theoretical and practical dimensions of the Random Assignment Problem. Aldous's proof, with its rigor, stands as an important contribution to the field, while the application in privacy-preserving wireless communication underscores the versatility of these theoretical models, demonstrating how they can be translated into effective solutions for real-world problems. For this reasons several future directions could emerge from this study. Firstly, Aldous's research, besides solving a problem that has been uncomplete for many years, highlighted how the analysis of mathematical problems can benefit from statistical physics concepts such as spin glasses and the cavity method. This study serves as a crucial reference point for these approaches. Some examples of other problems where statistical physics could provide the foundations for mathematical studies are reported in the articles: "Statistical Physics of Hard Optimization Problems" [29], "Statistical Mechanics of Deep Learning"[30], and "The Number of Matchings in Random Graphs" [31], which utilize the cavity method to obtain an estimation of matchings in random graphs. These examples illustrate the potential for future theoretical advancements through the application of statistical physics to other mathematical problems.

From a practical point of view, the flexibility of bipartite matching extends to a wide range of applications. This thesis has mentioned some uses of bipartite matching, demonstrating its adaptability to diverse challenges. Other recent relevant applications include its use in machine learning, as seen in the studies: "A Bipartite Matching-Based Feature Selection for Multi-Label Learning" [32] and "Conflict-Aware Weighted Bipartite B-Matching and Its Application to E-Commerce" [33]. These works show how the theory of bipartite matching can be applied to modern issues, providing effective solutions to contemporary problems.

In conclusion, this thesis showed that the Random Assignment Problem not only represents a significant theoretical challenge but also serves as a foundational basis for future advancements, providing substantial contributions to both theoretical and practical fields.

References

[1] D. J. Aldous, "The $\zeta(2)$ limit in the random assignment problem," *Random Structures & Algorithms*, vol. 18, no. 4, pp. 381–418, 2001.

[2] M. Qiu, K. Gai, and Z. Xiong, "Privacy-preserving wireless communications using bipartite matching in social big data," *Future Generation Computer Systems*, vol. 87, pp. 772–781, 2018.

[3] D. Aldous and J. M. Steele, "The objective method: probabilistic combinatorial optimization and local weak convergence," in *Probability on discrete structures.* Springer, 2004, pp. 1–72.

[4] L. Lovász and M. D. Plummer, *Matching theory.* American Mathematical Soc., 2009, vol. 367.

[5] A. Schrijver, "On the history of combinatorial optimization (till 1960)," *Handbooks in operations research and management science*, vol. 12, pp. 1–68, 2005.

[6] G. Monge, "Mémoire sur la théorie des déblais et des remblais," *Mem. Math. Phys. Acad. Royale Sci.*, pp. 666–704, 1781.

[7] G. Frobenius, F. G. Frobenius, F. G. Frobenius, F. G. Frobenius, and G. Mathematician, "Über matrizen aus nicht negativen elementen," 1912.

[8] D. König, "Line systems and determinants," *Math. Termész. Ert*, vol. 33, pp. 221–229, 1915.

[9] D. Konig, "Graphok es matrixok (hungarian)[graphs and matrices]," *Matematikai és Fizikai Lapok*, vol. 38, pp. 116–119, 1931.

[10] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[11] M. M. Flood, "The traveling-salesman problem," *Operations research*, vol. 4, no. 1, pp. 61–75, 1956.

[12] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the society for industrial and applied mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[13] J. M. Kurtzberg, "On approximation methods for the assignment problem," *Journal of the ACM (JACM)*, vol. 9, no. 4, pp. 419–439, 1962.

[14] J. Błażewicz, P. Formanowicz, M. Kasprzak, P. Schuurman, and G. J. Woeginger, "Dna sequencing, eulerian graphs, and the exact perfect matching problem," in *Graph-Theoretic Concepts in Computer Science: 28th International Workshop, WG 2002 Český Krumlov, Czech Republic, June 13–15, 2002 Revised Papers 28*. Springer, 2002, pp. 13–24.

[15] Y.-Q. Cheng, V. Wu, R. Collins, A. R. Hanson, and E. M. Riseman, "Maximum-weight bipartite matching technique and its application in image feature matching," in *Visual Communications and Image Processing'96*, vol. 2727. SPIE, 1996, pp. 453–462.

[16] D. W. Walkup, "On the expected value of a random assignment problem," *SIAM Journal on Computing*, vol. 8, no. 3, pp. 440–442, 1979.

[17] J. M. Steele, *Probability theory and combinatorial optimization*. SIAM, 1997.

[18] R. M. Karp, "An upper bound on the expected cost of an optimal assignment," in *Discrete algorithms and complexity: Proceedings of the Japan-US Joint Seminar*, vol. 15, 1987, pp. 1–4.

[19] A. J. Lazarus, "Certain expected values in the random assignment problem," *Operations research letters*, vol. 14, no. 4, pp. 207–214, 1993.

[20] M. E. Dyer, A. M. Frieze, and C. J. McDiarmid, "On linear programs with random costs," *Mathematical Programming*, vol. 35, pp. 3–16, 1986.

[21] B. Olin, *Asymptotic properties of random assignment problems*. Royal Inst. of Technology, Department of Mathematics, Division of ..., 1992.

[22] D. Coppersmith and G. B. Sorkin, "Constructive bounds and exact expectations for the random assignment problem," *Random Structures & Algorithms*, vol. 15, no. 2, pp. 113–144, 1999.

[23] M. Mézard and G. Parisi, "On the solution of the random link matching problems," *Journal de Physique*, vol. 48, no. 9, pp. 1451–1459, 1987.

[24] R. Brunetti, W. Krauth, M. Mézard, and G. Parisi, "Extensive numerical simulations of weighted matchings: Total length and distribution of links in the optimal solution," *Europhysics Letters*, vol. 14, no. 4, p. 295, 1991.

[25] D. Aldous, "Asymptotics in the random assignment problem," *Probability Theory and Related Fields*, vol. 93, no. 4, pp. 507–534, 1992.

[26] S. N. Chiu, D. Stoyan, W. S. Kendall, and J. Mecke, *Stochastic geometry and its applications*. John Wiley & Sons, 2013.

[27] S. Kotz and N. Johnson, "Logistic distribution," *Encyclopedia of Statistical Sciences*, vol. 5, pp. 123–128, 1985.

[28] T. L. Adam, K. M. Chandy, and J. Dickson, "A comparison of list schedules for parallel processing systems," *Communications of the ACM*, vol. 17, no. 12, pp. 685–690, 1974.

[29] L. Zdeborová, "Statistical physics of hard optimization problems," *arXiv preprint arXiv:0806.4112*, 2008.

[30] Y. Bahri, J. Kadmon, J. Pennington, S. S. Schoenholz, J. Sohl-Dickstein, and S. Ganguli, "Statistical mechanics of deep learning," *Annual Review of Condensed Matter Physics*, vol. 11, pp. 501–528, 2020.

[31] L. Zdeborová and M. Mézard, "The number of matchings in random graphs," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2006, no. 05, p. P05003, 2006.

[32] A. Hashemi, M. B. Dowlatshahi, and H. Nezamabadi-Pour, "A bipartite matching-based feature selection for multi-label learning," *International journal of machine learning and cybernetics*, vol. 12, pp. 459–475, 2021.

[33] C. Chen, L. Zheng, V. Srinivasan, A. Thomo, K. Wu, and A. Sukow, "Conflict-aware weighted bipartite b-matching and its application to e-commerce," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 6, pp. 1475–1488, 2016.