HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

GRADUATION THESIS

Developing a solution to the Teaching Assignment Problem using Reinforcement Learning

LÊ ĐỨC ANH
anh.ld194416@sis.hust.edu.vn

Major: Data Science
Specialization: Data Science And Artificial Intelligence

Supervisor: Associate Professor Huỳnh Quyết Thắng _____

Signature

School: School of Information and Communications Technology

HANOI, 08/2023

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my family, friends, and teachers for their unwavering support and encouragement throughout the completion of my thesis. Their belief in me and their constant motivation have been crucial in my diligent and determined efforts to achieve the best results in my research. I am particularly grateful to my thesis advisors, Ph.D. Huynh Quyet Thang and Ms. Le Phuong Chi, for their invaluable guidance, expertise, and mentorship. Their insightful advice, constructive feedback, and dedication have played a significant role in shaping the outcome of my work. I am truly thankful for their contributions to my academic and personal growth. Lastly, I would like to thank myself for the patience, effort, and determination invested in achieving a favorable outcome. This thesis has provided me with invaluable experiences and learning opportunities.

ABSTRACT

In this Graduation Thesis, I address the Teaching Assignment Problem in an educational institution. The Teaching Assignment Problem arises in educational institutions when the task of allocating classes to instructors needs to be optimized for efficiency and fairness. This problem is prevalent in universities, schools, and other educational organizations where multiple classes are offered across different subjects and time slots, and instructors have preferences and constraints regarding the classes they can teach. Recent studies have highlighted that Reinforcement Learning (RL) can be a viable approach, leading to a growing trend of applying RL to optimize various problems. Inspired by these findings, I aim to explore and experiment with RL in tackling the Teaching Assignment Problem.. The chosen approach leverages Q-learning, a model-free, off-policy reinforcement learning technique that enables instructors to learn from experiences and make informed decisions. My proposed solution involves creating a custom environment that models the Teaching Assignment Problem and initializing a Q-table to facilitate Q-learning. The Q-learning algorithm is then applied iteratively to update the Q-values, refining the course-instructor assignments over multiple episodes. The main contributions of this Graduation Thesis lie in the successful application of Q-learning to the Teaching Assignment Problem and the achievement of improved results. The proposed approach enhances the efficiency of the assignment process, ensuring a fair distribution of classes among instructors and meeting their preferences.

In conclusion, this thesis demonstrates the effectiveness of reinforcement learning, particularly Q-learning, in addressing the Teaching Assignment Problem. The successful implementation of this approach provides valuable insights into the application of artificial intelligence techniques to optimize educational processes and enhance resource allocation in educational institutions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Definition
ILP	Integer Linear Programming
MOO	Multi-Objective Optimization
RL	Reinforcement Learning
TAP	Teaching Assignment Problem

CHAPTER 1. INTRODUCTION

1.1 Problem Statement

The Teaching Assignment Problem (TAP) is a critical challenge faced by educational institutions, including universities, schools, and other academic organizations. It involves the optimal allocation of courses to instructors, considering their preferences and constraints, while ensuring fairness and efficiency in the teaching assignment process.

In a typical academic setting, there are multiple courses offered across various subjects and time slots. Additionally, instructors have preferences for specific subjects, course numbers, time slots, and teaching conditions. The objective of the TAP is to find an assignment that maximizes the overall quality of education, optimizes resource utilization, and ensures a balanced workload for instructors.

The main considerations in the Teaching Assignment Problem include:

1. **Instructor Preferences**: Each instructor may have preferences for certain subjects and course levels, as well as time slots that align with their teaching style and expertise.
2. **Course Offerings**: The educational institution offers a diverse range of courses across different subjects and levels.
3. **Time Constraints**: The assignment must adhere to specific time slots and the overall academic schedule.
4. **Fairness**: It is essential to ensure that the workload is distributed fairly among instructors, preventing overloading of some and underutilization of others.
5. **Optimal Resource Utilization**: Efficiently allocating courses to instructors optimizes the use of available resources, such as classroom capacity and instructor expertise.
6. **Dynamic Changes**: Course offerings and instructor availability may vary between semesters, necessitating flexibility in the assignment process.

While the Teaching Assignment Problem is a complex combinatorial optimization problem, it plays a crucial role in ensuring the quality of education and the satisfaction of both instructors and students.

1.2 Background and Problems of Research

1.2.1 Current Research Results for Teaching Assignment Problem

The Teaching Assignment Problem is a complex optimization problem that involves assigning instructors to classes in a way that maximizes teaching quality and satisfies various constraints. Over the years, researchers have proposed several approaches to tackle this problem. Some of the notable research results include:

- **Heuristic-Based Methods**: Many studies have utilized heuristic-based algorithms, such as greedy algorithms and genetic algorithms, to find near-optimal solutions for the Teaching Assignment Problem. These methods are computationally efficient and provide reasonably good results for small to medium-sized instances.
- **Integer Linear Programming (ILP)**: ILP formulations have been applied to model the Teaching Assignment Problem as a set of linear equations and constraints. ILP can handle larger instances of the problem and can guarantee optimal solutions under certain conditions. However, for larger problem sizes, the computational complexity of ILP increases significantly.
- **Metaheuristic Algorithms**: Various metaheuristic algorithms, such as simulated annealing, tabu search, and particle swarm optimization, have been employed to address the Teaching Assignment Problem. These approaches explore the solution space more effectively, making them suitable for larger instances with complex constraints.
- **Constraint Programming**: Researchers have explored constraint programming techniques to express and optimize the assignment problem based on different preferences and restrictions. Constraint programming can handle various constraints efficiently and is applicable to real-world scenarios.

1.2.2 Limitations of Existing Approaches

Despite the progress made in solving the Teaching Assignment Problem, several limitations persist in the current approaches:

1. **Lack of Guarantee for Optimal Solutions**: Many existing heuristic algorithms and integer linear programming methods may not always guarantee finding the absolute best solutions for the Teaching Assignment Problem. This limitation can lead to suboptimal allocations in some cases.
2. **Single-Objective Optimization**: Many existing methods focus on single-objective optimization, such as maximizing teaching quality or instructor preferences. However, in real-world scenarios, multiple conflicting objectives, such as minimizing workload imbalance or ensuring instructor preferences, need to be considered.

3. **Limited Consideration of Preferences**: Some approaches may not fully consider the complex preferences and constraints of both the academic department and instructors. As a result, the assignment may not adequately balance the interests of all parties involved.

4. **Static Nature**: Certain methods assume a fixed set of preferences and constraints, neglecting the possibility of dynamic changes in instructor preferences or course requirements over time.

5. **Lack of Generalization**: Some existing techniques may be tailored to specific instances or problem formulations, making it challenging to apply them to other teaching assignment scenarios.

1.3 Research Objectives and Conceptual Framework

In this section, I outline the primary objectives of this thesis and present the proposed approach to address the Teaching Assignment Problem. The main objectives of this research are as follows:

1. **Algorithm Development:** The central objective of this research is to design and develop an Reinforcement Learning-based algorithm that effectively solves the Teaching Assignment Problem. The algorithm will leverage the Q-learning technique, a model-free, off-policy Reinforcement Learning approach, to facilitate course-instructor assignments.

2. **Experimentation and Evaluation:** Another key goal is to conduct thorough experimentation using real-world and simulated data to evaluate the performance of the proposed Reinforcement Learning solver.

3. **Performance Analysis:** The research aims to analyze the results obtained from the Reinforcement Learning-based solver. The performance analysis will consider various metrics such as teaching quality, instructor preferences satisfaction, and computational efficiency.

4. **Robustness and Scalability:** The research will assess the robustness and scalability of the Reinforcement Learning-based algorithm concerning large-scale instances of the Teaching Assignment Problem. This analysis is crucial to evaluate the algorithm's feasibility and efficiency in handling real-world educational settings with a substantial number of classes, instructors, and subjects.

The conceptual framework for this research revolves around employing Reinforcement Learning as a viable approach to address the Teaching Assignment Problem. The proposed conceptual framework consists of the following stages:

1. **Problem Formulation:** The Teaching Assignment Problem will be mathematically defined, taking into account the constraints, preferences, and objectives of educational institutions when assigning classes to instructors.
2. **Reinforcement Learning Model:** The Reinforcement Learning-based solver will be developed, utilizing Q-learning as the core technique. The Q-learning model will enable instructors to learn from experiences and iteratively update Q-values to make optimal decisions in assigning classes.
3. **Experimentation:** The Reinforcement Learning-based algorithm will be implemented and evaluated with real-world data.
4. **Performance Analysis:** The obtained results will be analyzed using relevant evaluation metrics to assess the algorithm's effectiveness, efficiency, and robustness.
5. **Insights and Recommendations:** Based on the analysis, insights will be drawn regarding the advantages and limitations of the Reinforcement Learning-based approach compared to traditional heuristic methods. Recommendations for practical applications and potential improvements will be provided.

By following this conceptual framework, the research aims to demonstrate the feasibility and effectiveness of Reinforcement Learning in optimizing the teaching assignment process in educational institutions, contributing valuable insights to the field of educational resource allocation and optimization.

1.4 Contributions

In this section, I present the specific and concise contributions of this thesis. The main contributions are as follows:

1. Implementation of the Q-learning algorithm to optimize the assignment decisions in the Teaching Assignment Problem. The system learns from experience and adapts its assignment strategy based on rewards obtained from previous assignments, leading to a more efficient and effective allocation process.
   (a) Customized Q-learning Framework: As a fundamental contribution, I designed and built a customized Q-learning framework that caters specifically to the unique characteristics of the Teaching Assignment Problem. The framework includes a well-defined state representation, action space, and reward function, which allows the RL agent to learn optimal assignment strategies iteratively.
   (b) Integration of Domain Knowledge: To enhance the learning process, I integrated domain-specific knowledge into the Q-learning algorithm. This includes incorporating subject preferences, instructor expertise, and time slot availability into the state and reward representations. The incorporation

of such domain knowledge ensures that the RL agent can make informed and contextually relevant assignment decisions.

(c) Efficient Exploration-Exploitation Trade-off: One of the significant challenges in RL is the exploration-exploitation trade-off. I meticulously crafted a balance between exploration and exploitation to ensure that the RL agent efficiently explores the state space while exploiting the learned knowledge to make optimal assignment decisions.

(d) Convergence Analysis and Hyperparameter Tuning: I conducted extensive convergence analysis and hyperparameter tuning to ensure the Q-learning algorithm's stability and efficiency. This process involved fine-tuning the learning rate, discount factor, and exploration strategy to achieve fast convergence and optimal performance.

2. Evaluation of the proposed teaching assignment system using real-world datasets. The experimental results demonstrate the effectiveness and efficiency of the proposed approach in achieving better assignment outcomes while maintaining fairness and meeting individual preferences.

3. Potential for Educational Process Optimization: The thesis highlights the potential of Reinforcement Learning-based approaches in optimizing other aspects of educational processes beyond teaching assignment. The successful application of Reinforcement Learning in this context paves the way for exploring Reinforcement Learning's efficacy in solving various resource allocation problems in educational institutions.

These contributions collectively form the foundation of this thesis and contribute to the advancement of teaching assignment optimization using reinforcement learning techniques.

1.5 Organization of Thesis

The remainder of this thesis is organized as follows:

Chapter 2: In this chapter, I conduct a comprehensive review of the existing literature related to the teaching assignment optimization problem, explore the scope of research in this field and identify key aspects addressed by previous studies. I analyze various approaches, methods, and algorithms proposed in the literature to tackle the challenges of teaching assignment. By examining prior works, I aim to build a strong foundation for my research and identify potential gaps that can be addressed in this thesis.

Chapter 3: This chapter presents the methodology employed in addressing

the teaching assignment optimization problem. Building on the literature review conducted in Chapter 2, I outline the key steps and approaches undertaken to develop an effective and equitable teaching assignment system. The primary objective of this chapter is to provide a clear and comprehensive overview of the research methodology, which includes problem formulation, the utilization of game theory concepts like Nash equilibrium, and the application of reinforcement learning techniques, particularly the Q-learning algorithm. Through this detailed methodology, I aim to lay a strong foundation for the subsequent analysis, evaluation, and numerical results presented in the following chapters.

Chapter 4: In this chapter, I present the numerical results obtained from the implementation and experimentation of the teaching assignment system. I discuss the evaluation parameters used to assess the system's performance, such as teaching quality rates of different subjects and satisfaction rates of instructors. I describe the simulation methods employed to validate the system's outcomes and analyze its performance under various conditions. By examining the numerical results, I aim to demonstrate the effectiveness and efficiency of my proposed teaching assignment system and highlight its advantages over traditional approaches.

Chapter 5: This chapter is to provide a summary of the key findings, contributions, and achievements of this thesis, highlight the main outcomes of the research, including the successful implementation of the teaching assignment optimization system using Reinforcement Learning approach. Then I discuss the system's performance in achieving fair and balanced teaching assignments while considering the preferences and workload constraints of both academic departments and instructors. This chapter also reflect on the limitations and potential future extensions of the proposed system. In conclusion, this thesis contributes valuable insights and methodologies to the field of teaching assignment optimization, paving the way for more efficient and equitable teaching assignment processes in educational institutions.

I hope that this organization provides a clear and structured presentation of the research and contributes to the overall coherence of this thesis.

CHAPTER 2. LITERATURE REVIEW

2.1 Scope of Research

The scope of this research encompasses the development and application of a novel approach to solve the Teaching Assignment Problem in the context of a university environment. The focus is on efficiently allocating classes to instructors, taking into account various constraints and preferences while optimizing the overall teaching quality and instructor satisfaction.

The research primarily explores the utilization of Reinforcement Learning techniques to address the Teaching Assignment Problem. The research leverages insights and modeling techniques obtained from existing studies in the field. The problem formulation and modelling with NASH equilibrium used in this thesis are derived from relevant research that explores resource allocation and optimization problems in educational institutions. The Nash equilibrium ensures a balanced and fair distribution of classes, considering the diverse preferences and expertise of the instructors. Reinforcement Learning, particularly the Q-learning algorithm, is employed to train an intelligent agent capable of making optimal assignment decisions in a dynamic and evolving teaching environment.

To evaluate the effectiveness and efficiency of the proposed approach, extensive simulations and numerical experiments are conducted using real-world data or realistic synthetic datasets. The evaluation focuses on various performance metrics, such as teaching quality rates, instructor satisfaction levels, and resource utilization efficiency, to gauge the effectiveness of the solution.

The research acknowledges that the Teaching Assignment Problem can be highly complex and may vary across different universities or educational institutions. While the proposed approach aims to provide a general and adaptable solution, it may require fine-tuning or adjustments to fit specific institutional requirements and constraints.

It is important to note that this research does not delve into the broader field of educational administration or curriculum planning. Instead, it specifically targets the optimization of the teaching assignment process, addressing the challenges and complexities inherent in this particular domain.

In summary, the research aims to contribute to the field of educational optimization by proposing an innovative and data-driven approach that leverages Reinforcement Learning to solve the Teaching Assignment Problem effectively. Through empirical

evaluations and analyses, this research aims to shed light on the potential benefits and limitations of the proposed approach and its applicability in real-world educational settings.

2.2 Related Works

Several studies have been conducted to address the Teaching Assignment Problem in educational institutions. These works have explored various methods and techniques to optimize the allocation of courses to instructors. I summarize some of the prominent approaches and discuss their advantages and disadvantages below.

2.2.1 Heuristic-Based Methods

Some early works in teaching assignment have relied on heuristic-based approaches, where administrators manually assign courses to instructors based on their expertise and availability. The advantage of such methods lies in their simplicity and ease of implementation. They can quickly produce reasonable assignments, especially in small-scale scenarios. However, these methods are often subjective, and the quality of assignments may vary depending on the experience and judgment of the administrators. Moreover, heuristic-based approaches may struggle to handle large and complex teaching schedules efficiently.[1]

2.2.2 Mathematical Optimization Techniques

Linear programming and integer programming techniques have been employed to formulate the Teaching Assignment Problem as an optimization model. These methods aim to find the optimal assignment that maximizes certain objective functions while satisfying various constraints. One significant advantage of these techniques is their ability to guarantee an optimal or near-optimal solution under certain conditions. They provide a systematic and rigorous approach to handle the assignment problem. However, the computational complexity of solving large-scale optimization models can be prohibitive, and real-time adaptability to changing requirements may be challenging.

2.2.3 Metaheuristic Algorithms

In recent years, metaheuristic algorithms, such as genetic algorithms and simulated annealing, have been applied to tackle the Teaching Assignment Problem. These methods offer a powerful and flexible way to explore the search space and find near-optimal solutions. They can handle complex and dynamic scenarios and are less likely to get stuck in local optimal. Additionally, metaheuristic algorithms can incorporate various objective functions and constraints, making them suitable for multi-objective optimization. However, they often require extensive parameter tuning, and the computational time can be significant, especially for large-scale

problems.

2.2.4 Multi-Objective Optimization

To address the conflicting objectives in teaching assignment, some researchers have employed multi-objective optimization techniques. These methods consider multiple criteria, such as instructor preferences, teaching quality, and class coverage, as separate objectives. The advantage of multi-objective optimization lies in its ability to generate a set of Pareto-optimal solutions, representing the trade-offs between different objectives. This allows decision-makers to explore different assignment scenarios and choose the most suitable one based on their preferences. However, multi-objective optimization can be computationally intensive and may require additional effort to interpret and select the final assignment.

In conclusion, the Teaching Assignment Problem has been approached from various perspectives, each with its advantages and disadvantages. Heuristic-based methods are simple but lack objectivity, while mathematical optimization techniques guarantee optimality but may be computationally demanding. Metaheuristic algorithms offer flexibility but require parameter tuning, and multi-objective optimization enables trade-offs but can be complex. In this thesis, I propose an approach based on Nash equilibrium and Reinforcement Learning to address the challenges and limitations of existing methods. By combining the strengths of game theory and intelligent decision-making algorithms, my approach aims to provide efficient and fair teaching assignments in dynamic educational environments.

2.3 Multi-Objective Optimization Problem

Multi-objective optimization (MOO) is a mathematical optimization paradigm that deals with problems involving multiple, conflicting objectives. In contrast to single-objective optimization, where a single optimal solution is sought, MOO aims to find a set of solutions called the Pareto-optimal front, representing the best trade-offs among the objectives.

Problem Formulation: Let $\mathbf{X} \in \mathbb{R}^n$ be the decision variables vector, and $f_i(\mathbf{X})$ be the i-th objective function to be optimized. The multi-objective optimization problem can be formulated as follows:

$$
\begin{aligned}
\text{Minimize (or Maximize)} \quad & f_i(\mathbf{X}) \quad \text{for} \quad i = 1, 2, \ldots, m \\
\text{Subject to} \quad & g_j(\mathbf{X}) \leq 0 \quad \text{for} \quad j = 1, 2, \ldots, p \\
& h_k(\mathbf{X}) = 0 \quad \text{for} \quad k = 1, 2, \ldots, q \\
& \mathbf{X} \in \Omega
\end{aligned}
$$

Here, m is the number of objective functions, p and q are the number of inequality and equality constraints, respectively, and Ω represents the feasible region defined by the constraints.

Pareto-Optimal Solutions: A solution $\mathbf{X}^*$ is said to be Pareto-optimal if there does not exist any other feasible solution that simultaneously improves all objectives without worsening any other objective. The set of all Pareto-optimal solutions is known as the Pareto-optimal front.[2]

Solving Multi-Objective Optimization: There are several approaches to solving multi-objective optimization problems, including evolutionary algorithms, genetic algorithms, particle swarm optimization, and mathematical programming-based methods like the weighted sum method and ϵ-constraint method. Each of these methods has its strengths and weaknesses, and the choice of method depends on the nature of the problem and the characteristics of the objectives and constraints.

In this thesis, I explore the application of Nash equilibrium to tackle the multi-objective Teaching Assignment Problem. The Nash equilibrium is well-suited for handling multi-objective problems and provides a way to change from multi-objective optimization problem to single objective optimization problem.

The next section will detail the Nash Equilibrium and its adaptation to the Teaching Assignment Problem, showcasing the potential of Nash equilibrium in multi-objective optimization.

2.4 Nash Equilibrium

Nash Equilibrium is a fundamental concept in game theory that involves finding a stable state in a non-cooperative game, where no player has an incentive to change their strategy unilaterally. In the context of multi-objective optimization, Nash Equilibrium can be used to transform a multi-objective problem into a single-objective problem, enabling the application of Reinforcement Learning techniques.[3]

Definition: Consider a non-cooperative game with N players, each having a set of strategies denoted as $\mathbf{S}_i$, and utility functions $U_i(\mathbf{s})$ representing the payoffs to each player i when they choose a strategy profile $\mathbf{s} = (s_1, s_2, \ldots, s_N)$, where $s_i \in \mathbf{S}_i$. A strategy profile $\mathbf{s}^* = (s_1^*, s_2^*, \ldots, s_N^*)$ is said to be a Nash Equilibrium if, for each player i, the following condition holds:

$$U_i(s_i^*, s_{-i}^*) \geq U_i(s_i, s_{-i}^*) \quad \text{for all} \quad s_i \in \mathbf{S}_i$$

where s_{-i}^* represents the strategies chosen by all players except player i.

Transforming Multi-Objective Problem: In the context of multi-objective optimization, Nash Equilibrium can be leveraged to combine the multiple objectives into a single objective function. By considering each objective as a player and their corresponding strategies as different weighting factors, we can use Nash Equilibrium in the game to determine the optimal weighting factors for combining the objectives.

Reinforcement Learning with Nash Equilibrium: By transforming the multi-objective optimization problem into a single-objective form using Nash Equilibrium, we can apply Reinforcement Learning techniques, such as the Q-learning algorithm, to search for an optimal solution. In this approach, the Q-learning algorithm learns to adjust the weighting factors to find the Nash Equilibrium, which represents the optimal trade-off between conflicting objectives.

The next section will delve into the application of Reinforcement Learning in the context of the Teaching Assignment Problem.

2.5 Reinforcement Learning

Reinforcement Learning (RL) is a subfield of machine learning that focuses on training agents to make decisions in an environment to achieve specific goals. Unlike supervised learning, where the agent is provided with labeled data, and unsupervised learning, where there is no explicit guidance, RL relies on trial and error to learn optimal actions through interactions with the environment. RL has found applications in various domains, including robotics, gaming, finance, and autonomous systems. [4]

Key Concepts: The core components of a Reinforcement Learning system are as follows:

- **Agent:** The learner or decision-maker that interacts with the environment and takes actions.
- **Environment:** The external system with which the agent interacts. It provides feedback to the agent in the form of rewards or penalties based on the agent's actions.
- **State:** A representation of the current situation of the environment that the agent observes. It serves as the input for the agent's decision-making process.
- **Action:** The set of possible moves or decisions that the agent can take in response to a given state.
- **Policy:** The strategy that the agent uses to map states to actions. It defines the agent's behavior and is the main focus of learning in RL.

- **Reward Function:** A function that assigns a numerical reward to the agent based on the action taken in a particular state. The agent's goal is to maximize the cumulative reward over time.
- **Value Function:** A function that estimates the expected cumulative reward from a particular state under a given policy. It helps the agent evaluate the desirability of different states.
- **Q-Function:** A function that estimates the expected cumulative reward from taking a specific action in a particular state under a given policy. It is used to guide the agent's decision-making process.
- **Q-Learning Algorithm:** Q-learning is a popular RL algorithm used for environments with discrete state and action spaces. It learns the optimal Q-function, which represents the expected cumulative reward for taking a particular action in a specific state. The Q-learning process involves updating Q-values based on the rewards received and the new information obtained during agent-environment interactions.
- **Exploration and Exploitation:** One of the fundamental challenges in RL is the trade-off between exploration and exploitation. Exploration involves trying out different actions to gather information about the environment, while exploitation focuses on taking actions that are known to yield high rewards based on past experiences. Balancing exploration and exploitation is crucial to discovering optimal policies.
- **Value Iteration and Policy Iteration:** Value Iteration and Policy Iteration are classic dynamic programming approaches used to solve Markov Decision Processes (MDPs). Value Iteration involves iteratively updating the value function for each state until convergence, while Policy Iteration alternates between policy evaluation and policy improvement steps to find an optimal policy.
- **Deep Reinforcement Learning:** Deep Reinforcement Learning combines RL with deep neural networks to tackle problems with large and continuous state spaces. Deep RL algorithms, such as Deep Q Networks (DQNs) and Deep Deterministic Policy Gradients (DDPG), use neural networks as function approximators to represent value functions and policies.
- **On-Policy vs. Off-Policy Learning:** In RL, there are two primary learning approaches: on-policy and off-policy. On-policy methods use the current policy to collect data and improve the policy over time. In contrast, off-policy methods use a different policy for data collection and can learn from data generated by

other policies.

- **Temporal Difference Learning:** Temporal Difference (TD) Learning is a technique widely used in RL. It updates the value function based on the difference between the expected reward and the observed reward at each time step. TD methods, such as Q-learning and SARSA, are essential for learning in environments with incomplete knowledge of transition probabilities.
- **Function Approximation:** Function approximation is employed in RL to deal with large state and action spaces. By using function approximators like neural networks, RL algorithms can generalize across similar states and actions and handle continuous state and action spaces efficiently.
- **Convergence and Stability:** In RL, ensuring convergence and stability is critical to guarantee that the learning process converges to an optimal policy. Techniques like learning rate schedules and experience replay are used to achieve stable learning.

Application to Teaching Assignment Problem: In the context of the Teaching Assignment Problem, Reinforcement Learning with Q-learning can be applied to find an optimal assignment of courses to instructors, where states represent different course assignments, actions correspond to instructor choices, and rewards reflect the quality of the assignments, the Q-learning algorithm can be used to discover the best assignment strategy.

The subsequent chapter will dig into how these concepts are applied in solving the Teaching Assignment Problem using Reinforcement Learning with Q-learning and Nash Equilibrium.

CHAPTER 3. METHODOLOGY

3.1 Overview

This chapter presents the comprehensive methodology employed to address the Teaching Assignment Problem (TAP). The methodology involves the application of a Reinforcement Learning Approach, which incorporates various components such as data import, environment modeling, and training of the Q-learning agent. The problem and NASH equilibrium modelling are the result of a recent research. [5]

Problem formulation defines and describes the research problem that I intend to address in this study. This step is crucial as it ensures that the research question or objective is clearly defined and precisely stated. It helps avoid ambiguity and ensures that everyone involved in the project understands what needs to be accomplished.

In modelling the problem with NASH equilibrium, Nash equilibrium is applied to change the multi-objective optimization problem into a single-objective optimization problem with weighted sum approach. This involves assigning weights (importance factors) to each objective and transforming the multiple objectives into a single aggregated objective function. The aggregated objective function is a linear combination of the individual objectives with their respective weights.

After transforming the multi-objective optimization problem into a single-objective optimization problem using Nash equilibrium, I use reinforcement learning (RL) to solve the problem. The RL agent will be responsible for making decisions on how to assign courses to instructors based on the transformed single-objective optimization function. The RL agent will learn from its interactions with the environment and receive feedback in the form of rewards or penalties based on the quality of its assignments.

3.2 Problem Formulation

The Teaching Assignment Problem can be defined as follows:

- Let N_s be the number of subjects taught in the university.
- Let N_c be the number of classes that need to be assigned to instructors.
- Let N_t be the number of time slots.
- Let N_i be the number of instructors.
- Matrix $X = [x_{c,s}]$, $1 \leq c \leq N_c$, $1 \leq s \leq N_s$, in which $x_{c,s} = 1$ if class c studies subject s, and $x_{c,s} = 0$ if class c does not study subject s.

- Matrix $Y = [y_{c,t}]$, $1 \leq c \leq N_c$, $1 \leq t \leq N_t$, in which $y_{c,t} = 1$ if class c is taught during time slot t, and $y_{c,t} = 0$ otherwise.
- Matrix $P = [p_{i,s}]$, $1 \leq i \leq N_i$, $1 \leq s \leq N_s$, in which $p_{i,s}$ indicates the level of preference of instructor i for subject s.
- Matrix $E = P * X^{-1}$, $1 \leq i \leq N_i$, $1 \leq c \leq N_c$, in which $e_{i,c}$ indicates the level of preference of instructor i for class c.
- Matrix $R = [r_{i,s}]$, $1 \leq i \leq N_i$, $1 \leq s \leq N_s$, in which $r_{i,s}$ indicates the teaching quality of instructor i for subject s.
- Matrix $F = R * X^{-1}$,$1 \leq i \leq N_i$, $1 \leq c \leq N_c$, in which $f_{i,c}$ indicates the teaching quality of instructor i for class c.
- Matrix $T = [t_{i,t}]$, $1 \leq i \leq N_i$, $1 \leq t \leq N_t$, in which $t_{i,t}$ indicates the level of preference of instructor i for time slot t.
- Matrix $H = T * X^{-1}$, $1 \leq i \leq N_i$, $1 \leq c \leq N_c$, in which $h_{i,c}$ indicates the level of preference in term of time slots of instructor i for class c.
- Vector $V = [v_i]$, $1 \leq i \leq N_i$, in which v_i is the number of classes that instructor i wants to teach.
- Vector $MinC = [m_i]$, $1 \leq i \leq N_i$, in which m_i is the minimum number of classes that instructor i needs to be assigned.
- Vector $MaxC = [M_i]$, $1 \leq i \leq N_i$, in which M_i is the maximum number of classes that instructor i can be assigned.

A teaching schedule is a matrix $D = [d_{c,i}]$, $1 \leq c \leq N_c$, $1 \leq i \leq N_i$, where $d_{c,i} = 1$ if instructor i is assigned to teach class c, and $d_{c,i} = 0$ if instructor i is not assigned to teach class c. A selected teaching schedule must ensure the rights of all parties involved and cannot violate any constraints.

The Teaching Assignment Problem is subject to the following constraints:

1. All classes must be assigned to an instructor.
2. Each class can only be assigned to one instructor.
3. An instructor cannot be assigned to teach different classes in the same time slot.
4. Do not assign instructors to teach the subjects they have no interest in teaching.
5. Do not assign an instructor to teach a subject they cannot ensure the quantity.
6. Do not assign an instructor to teach in time slots they cannot teach.

7. Do not assign an instructor to teach fewer classes than the minimum required.
8. Do not assign an instructor to teach more classes than the maximum allowed.

These constraints ensure that the teaching assignment is feasible and satisfies the preferences and requirements of both classes and instructors.

The objective of the Teaching Assignment Problem is to find the optimal teaching schedule D that maximizes the overall teaching quality while satisfying the constraints on instructor preferences and workload.

3.3 NASH equilibrium Modeling

To find the optimal teaching schedule D that maximizes the overall teaching quality while satisfying the constraints on instructor preferences and workload, the TAP are modelled as a non-cooperative game. In this game, each instructor i aims to maximize their individual payoff, which is a function of their teaching assignments in the schedule. The overall teaching quality of the system is represented by the special player Γ, and its payoff function is defined as follows:

$$\text{Payoff function of player } \Gamma : \Phi = \sum_{i=1}^{N_c} \sum_{j=1}^{N_i} (d_{i,j} * f_{j,i})$$

where $f_{j,i}$ is the teaching quality of class i for instructor j.

Each instructor i has their own payoff function, denoted as Φ_i, which is calculated based on their preferences for subjects, time slots, and teaching quantity. Specifically, the payoff function of instructor i is given by:

$$\text{Payoff function of player } i\text{: } \Phi_i = w_1 \cdot \Psi_{i,\text{subject}} + w_2 \cdot \Psi_{i,\text{time slot}} + w_3 \cdot \Psi_{i,\text{teaching quantity}}$$

where w_1, w_2, and w_3 are coefficients determined based on the desires of instructors, and $\Psi_{i,\text{subject}}$, $\Psi_{i,\text{time slot}}$, and $\Psi_{i,\text{teaching quantity}}$ are the level of preference of instructor i for subjects, time slots, and teaching quantity, respectively, in the assigned schedule D.

The Nash equilibrium point in this game corresponds to the teaching schedule that maximizes the overall teaching quality while ensuring that no instructor can unilaterally improve their payoff without worsening the payoff of another instructor. To find this Nash equilibrium, we aim to maximize the overall payoff function, which is the sum of the payoffs of the special player Γ and all instructors i. The overall payoff function is defined as follows:

$$\text{Overall Payoff function: } \Psi = w_4 \cdot \Phi + w_5 * \sum_{i=1}^{Ni} \Phi_i$$

where w_4 and w_5 is a coefficient determined to balance between the benefits of instructors and department.

The solution that maximizes the overall payoff function Ψ represents the optimal teaching schedule that satisfies all constraints and ensures fairness and efficiency in the teaching assignment process. We can use optimization techniques such as Reinforcement Learning to find this solution.

In the next section, I will detail the implementation of the Reinforcement Learning for finding the optimal teaching schedule.

3.4 Reinforcement Learning Approach

The Teaching Assignment Problem involves assigning teachers to various classes based on their preferences, qualifications, and availability. The goal is to optimize the assignment process to ensure that teachers are allocated to classes in a way that maximizes overall satisfaction and teaching quality.

Solution Approach:

Import Data: Import data to create the environment.

Environment Modelling: Create the environment for the Teaching Assignment Problem, where the states represent the classes to be assigned, and the actions represent the assignment of teachers to those classes.

Initialize Q-Table: Set up the Q-table to store the Q-values for each state-action pair and initialize the initial values.

Parameters Settings: Adjust reinforcement learning parameters such as the learning rate, discount factor, and exploration rate to control the learning process.

Reward Function: Design a reward system to measure the quality of the teaching assignment based on various factors such as teacher preferences, teaching quality, and other relevant criteria.

Epsilon-Greedy Algorithm: an effective way to balance exploration and exploitation in reinforcement learning by allowing the agent to explore new actions and exploit the best-known actions simultaneously.

Q-Learning Algorithms: Update the Q-values using the Q-learning algorithm based on the received reward and the expected future rewards.

Training the Q-learning Agent: Train the reinforcement learning model, updating the Q-table after each episode.

Evaluation and Optimization: Evaluate the performance of the reinforcement learning model and optimize the parameters based on the evaluation results.

A block diagram illustrating the flow of the proposed solution for the Teaching Assignment Problem with reinforcement learning will be provided in the subsequent part.

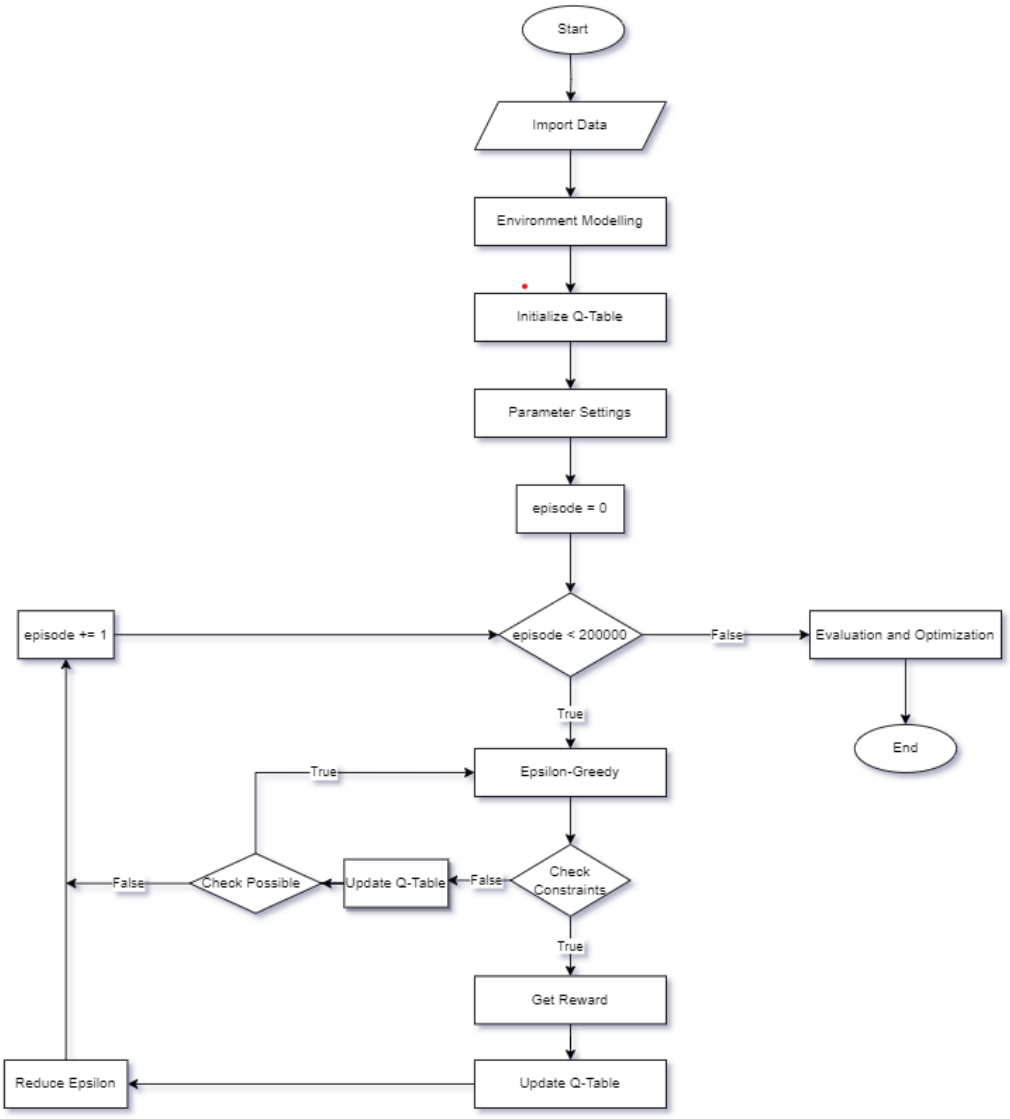


Figure 3.1: Q-Learning Approach

3.4.1 Import Data

The data used in this research was collected from FPT University - Ha Noi during the Spring 2022 semester. The dataset includes information about 153 classes, 25 instructors, 13 subjects, and 10 time slots.

a, Data Sources

The following Excel files were used to collect the necessary data:

- `data_subjects_information.xlsx`: Contains information about the subjects offered in the semester. An example of this dataset is shown below:

Id	Subject
0	PRJ301
1	PRF192
2	OSG202
3	PRO192
4	CSD201
5	DBI202
6	DBW301
7	PRN211
8	PFP191
9	PRN221
10	PRU211M
11	JFE301
12	CSD301

Figure 3.2: Subjects Ìnformation

The "Id" column represents the unique identifier of subject and "Subject" column represents the name of subjects.

- `data_instructors_timeslot.xlsx`: Provides the availability of instructors for each time slot.An example of this dataset is shown below:

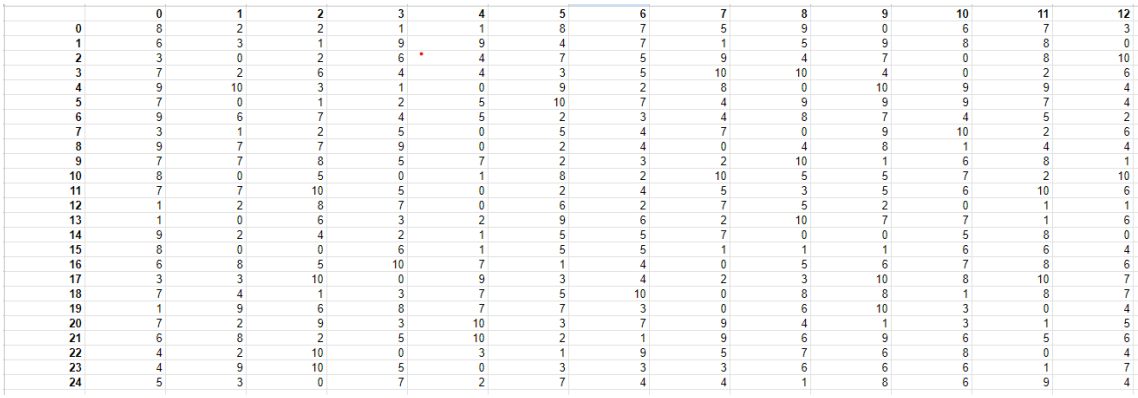
	M1	M2	M3	E1	E2	E3	M4	M5	E4	E5
0	7	7	10	5	7	1	5	8	8	8
1	5	3	8	10	4	1	5	1	1	3
2	0	5	1	8	1	3	10	4	9	8
3	5	8	8	5	7	1	8	3	0	4
4	10	3	9	4	9	10	1	2	10	2
5	5	10	10	2	5	2	3	4	3	5
6	4	3	4	9	4	6	6	6	4	2
7	6	2	0	7	2	0	7	10	10	5
8	9	1	8	7	6	1	1	10	3	10
9	6	7	3	8	2	0	5	8	0	6
10	7	7	9	7	7	1	1	1	2	10
11	9	8	7	8	1	6	3	8	8	8
12	1	2	6	3	10	1	1	4	1	8
13	1	4	10	0	5	5	7	7	1	7
14	9	0	5	0	4	7	10	8	0	7
15	5	9	9	2	1	1	7	2	2	6
16	6	8	10	3	6	3	6	3	4	9
17	8	4	3	3	6	5	3	1	7	9
18	2	10	0	7	0	7	8	1	10	2
19	2	7	5	5	7	1	6	3	10	5
20	8	8	9	3	0	1	8	5	5	8
21	10	5	10	9	8	5	5	7	1	3
22	8	10	8	9	6	6	3	2	1	4
23	7	7	10	0	6	0	5	1	7	1
24	0	7	2	7	7	9	9	10	8	1

Figure 3.3: Instructors Timeslot

The first column is the identifier of instructors from 0 to 24, the first row is the name of timeslots.

- `data_instructors_quality.xlsx`: Indicates the subjects that each

instructor is qualified to teach. An example of this dataset is shown below:

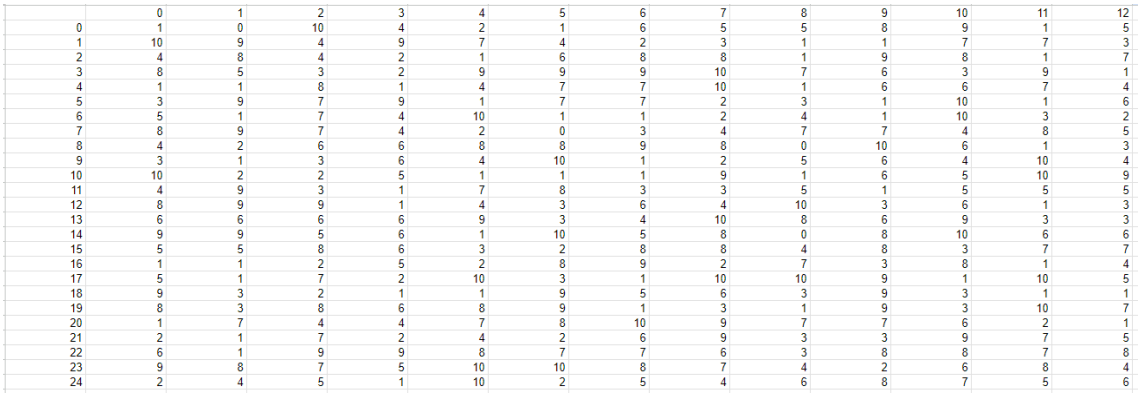


	0	1	2	3	4	5	6	7	8	9	10	11	12
0	8	2	2	1	1	8	7	5	9	0	6	7	3
1	6	3	1	9	9	4	7	1	5	9	8	8	0
2	3	0	2	6	4	7	5	9	4	7	0	8	10
3	7	2	6	4	4	3	5	10	10	4	0	2	6
4	9	10	3	1	0	9	2	8	0	10	9	9	4
5	7	0	1	2	5	10	7	4	9	9	9	7	4
6	9	6	7	4	5	2	3	4	8	7	4	5	2
7	3	1	2	5	0	5	4	7	0	9	10	2	6
8	9	7	7	9	0	2	4	0	4	8	1	4	4
9	7	7	8	5	7	2	3	2	10	1	6	8	1
10	8	0	5	0	1	8	2	10	5	5	7	2	10
11	7	7	10	5	0	2	4	5	3	5	6	10	6
12	1	2	8	7	0	6	2	7	5	2	0	1	1
13	1	0	6	3	2	9	6	2	10	7	7	1	6
14	9	2	4	2	1	5	5	7	0	0	5	8	0
15	8	0	0	6	1	5	5	1	1	1	6	6	4
16	6	8	5	10	7	1	4	0	5	6	7	8	6
17	3	3	10	0	9	3	4	2	3	10	8	10	7
18	7	4	1	3	7	5	10	0	8	8	1	8	7
19	1	9	6	8	7	7	3	0	6	10	3	0	4
20	7	2	9	3	10	3	7	9	4	1	3	1	5
21	6	8	2	5	10	2	1	9	6	9	6	5	6
22	4	2	10	0	3	1	9	5	7	6	8	0	4
23	4	9	10	5	0	3	3	3	6	6	6	1	7
24	5	3	0	7	2	7	4	4	1	8	6	9	4

Figure 3.4: Instructors Quality

The first column is the identifier of instructors from 0 to 24, the first row is the identifier of subjects from 0 to 12.

- `data_instructors_preference_subjects.xlsx`: Contains ratings of instructors for each subject. An example of this dataset is shown below:



	0	1	2	3	4	5	6	7	8	9	10	11	12
0	1	0	10	4	2	1	6	5	5	8	9	1	5
1	10	9	4	9	7	4	2	3	1	1	7	7	3
2	4	8	4	2	1	6	8	8	1	9	8	1	7
3	8	5	3	2	9	9	9	10	7	6	3	9	1
4	1	1	8	1	4	7	7	10	1	6	6	7	4
5	3	9	7	9	1	7	7	2	3	1	10	1	6
6	5	1	7	4	10	1	1	2	4	1	10	3	2
7	8	9	7	4	2	0	3	4	7	7	4	8	5
8	4	2	6	6	8	8	9	8	0	10	6	1	3
9	3	1	3	6	4	10	1	2	5	6	4	10	4
10	10	2	2	5	1	1	1	9	1	6	5	10	9
11	4	9	3	1	7	8	3	3	5	1	5	5	5
12	8	9	9	1	4	3	6	4	10	3	6	1	3
13	6	6	6	6	9	3	4	10	8	6	9	3	3
14	9	9	5	6	1	10	5	8	0	8	10	6	6
15	5	5	8	6	3	2	8	8	4	8	3	7	7
16	1	1	2	5	2	8	9	2	7	3	8	1	4
17	5	1	7	2	10	3	1	10	10	9	1	10	5
18	9	3	2	1	1	9	5	6	3	9	3	1	1
19	8	3	8	6	8	9	1	3	1	9	3	10	7
20	1	7	4	4	7	8	10	9	7	7	6	2	1
21	2	1	7	2	4	2	6	9	3	3	9	7	5
22	6	1	9	9	8	7	7	6	3	8	8	7	8
23	9	8	7	5	10	10	8	7	4	2	6	8	4
24	2	4	5	1	10	2	5	4	6	8	7	5	6

Figure 3.5: Instructors preference Subjects

The first column is the identifier of instructors from 0 to 24, the first row is the identifier of subjects from 0 to 12.

- `data_instructors_information.xlsx`: Includes information about the minimum, maximum, and ideal number of courses that each instructor wants to teach. An example of this dataset is shown below:

TeacherID	Name	Min_course	Max_course	Ideal_course
1	TrangNTP	3	6	5
2	TungHA	2	6	4
3	SonNX	3	5	2
4	HoangHM	3	7	6
5	HuongNT	4	7	5
6	BaoNX	1	8	4
7	KienNTT	1	8	4
8	KhangNK	2	8	6
9	ChauNV	4	7	5
10	ThuyNH	2	6	4
11	ThuyNTN	4	7	6
12	TuyNB	2	6	5
13	LuongHN	2	8	5
14	GiangNTK	2	7	5
15	ThuyNH2	2	7	2
16	PhanNT	3	7	5
17	XuanNK	1	8	6
18	TungNTH	2	6	6
19	HaNK	2	8	3
20	PhuongNTM	1	6	3
21	TrungNB	3	6	6
22	BinhNT	3	9	3
23	VanNTK	2	8	4
24	BurkeNS	2	8	4
25	JohnNS	4	6	5

Figure 3.6: Instructors Information

The first column is the instructors identifier, the second column is name of the instructors, the third column is the minimum number of courses that instructor is assigned to, the fourth column is the maximum number of courses that instructor assigned to, the fifth column is the ideal number of courses that the instructor want to teach.

- `data_courses_information.xlsx`: Contains essential course information such as Course ID, Subject ID, and Time Slot ID. An example of this dataset is shown below:

CourseId	Class	Subject	SubjectId	Slot	SlotId
0	SE1617	PRJ301	0	M5	7
1	SE1609	PRJ301	0	E5	9
2	SE1602	PRJ301	0	M1	0
3	SE1622	PRJ301	0	E1	3
4	SE1601	PRJ301	0	M2	1
5	SE1606	PRJ301	0	E2	4
6	SE1610	PRJ301	0	M3	2
7	SE1613	PRJ301	0	E3	5
8	SE1620	PRJ301	0	M4	6
9	SE1616	PRJ301	0	E4	8
10	SE1612	PRJ301	0	M5	7
11	SE1618	PRJ301	0	E5	9
12	SE1605	PRJ301	0	M1	0
13	SE1608	PRJ301	0	E1	3
14	SE1604	PRJ301	0	M2	1
15	SE1611	PRJ301	0	E2	4
16	SE1607	PRJ301	0	M3	2
17	SE1603	PRJ301	0	E3	5
18	SE1621	PRJ301	0	M4	6
19	SE1614	PRJ301	0	E4	8
20	SE1615	PRJ301	0	M5	7
21	SE1706	PRF192	1	E3	5
22	SE1705	PRF192	1	M3	2
23	SE1710	PRF192	1	E2	4
24	SE1711	PRF192	1	M2	1
25	SE1715	PRF192	1	E3	5
26	SE1713	PRF192	1	M3	2
27	SE1707	PRF192	1	E4	8
28	SE1717	PRF192	1	M4	6
29	SE1714	PRF192	1	E5	9

Figure 3.7: Courses Information

The first column is the identifier of courses, the second column is name of class, the third column is the name of subject in that course, the fourth column is identifier of subject name, the fifth column is name of time slot and the sixth column is identifier of time slot.

b, Data Preprocessing

Once the data was imported from the Excel files, further preprocessing was performed to extract relevant information. The dimensions of the problem were determined based on the extracted data:

- The number of time slots: *num_timeslots*.
- The number of instructors: *num_instructors*.
- The number of subjects: *num_subjects*.
- The number of courses: *num_courses*.

Additional data preprocessing and calculations were conducted based on the specific requirements of the problem. For instance, a matrix (*course_subject*) was created to represent the relation between courses and subjects. Furthermore, the *quality_class* was calculated based on the *instructor_rating* and *course_subject*.

c, Data Usage

The imported and processed data served as the foundation for implementing the Q-learning algorithm to solve the Teaching Assignment Problem. The data was utilized to determine the fitness and constraints for assigning instructors to courses optimally.

3.4.2 Environment Modelling

In this section, I describe the environment modeling process for the Teaching Assignment Problem using reinforcement learning. The environment is crucial for defining the problem's state space, action space, and the rewards associated with different state-action pairs. The objective is to create an environment that accurately represents the real-world teaching assignment scenario and allows the agent to learn and make optimal decisions.

a, State Space

The state space represents the possible states that the agent can be in at any given time. In the Teaching Assignment Problem, state space consists of the course which is assigned now and the state of the courses are assigned before this course

b, Action Space

The action space represents the possible actions that the agent can take in each state. In the Teaching Assignment Problem, the action space consists of all possible combinations of assigning an instructor to a course. The agent must assign a course

to an instructor. Therefore, the action space can be represented as follows:

$$\text{Action Space} = \{i \mid i \in \{0, 1, ..., 24\} \tag{3.1}$$

c, Dynamic Environment

The Teaching Assignment Problem is a dynamic environment because it evolves over time. As the agent makes decisions and assigns courses to instructors, the state of the environment changes. Therefore, the agent needs to adapt and learn from the feedback received to improve its decision-making process.

3.4.3 Initialize Q-Table

The Q-learning algorithm relies on a Q-table to store the expected rewards for each state-action pair in the environment. Before the training process begins, the Q-table needs to be initialized. In this section, we describe the process of initializing the Q-table for the Teaching Assignment Problem.

a, Q-Table Dimensions

The dimensions of the Q-table are determined by the size of the state space and the action space. In Teaching Assignment Problem, the state space consists of the course which is assigned now and the state of the courses are assigned before this course, and the action space consists of all instructors. Therefore, the number of states is massive so after testing, the Q-table will have dimensions of $153 \times 2500000 \times 25$, representing each course, the state of courses before and 25 possible actions to assign to instructors.

b, Initializing Q-Values

The initial values of the Q-table is set a random number between 0 and 1.

3.4.4 Parameter Settings

In this section, I outline the various parameters and hyperparameters used in my research to solve the Teaching Assignment Problem with Reinforcement Learning. These settings play a crucial role in shaping the behavior of my proposed algorithm and determining the quality of the assignments.

a, List of Parameters

The following are the key parameters and hyperparameters utilized in my study:

1. **Learning Rate** (α): This parameter controls the rate at which the Q-values are updated during the Q-learning process.
2. **Discount Factor** (γ): The discount factor determines the importance of future

rewards compared to immediate rewards in the Q-learning updates.

3. **Exploration Rate** (ε): The exploration rate governs the balance between exploration and exploitation in the Q-learning algorithm.
4. **Number of Episodes**: The total number of episodes the Q-learning agent undergoes during training.
5. **Reward Weights** (w_1, w_2, w_3, w_4, w_5): Weights assigned to different components of the reward function for the Q-learning agent.

b, Explanation of Parameters

Learning Rate (α)

The learning rate determines the impact of new information on the Q-values. A higher learning rate allows the agent to update its Q-values more significantly after each experience, while a lower rate results in more gradual updates.

Discount Factor (γ)

The discount factor balances immediate rewards and future rewards in the Q-learning updates. A higher discount factor values long-term rewards more, while a lower value prioritizes immediate rewards.

Exploration Rate (ε)

The exploration rate influences the agent's exploration-exploitation trade-off. A higher ε encourages more exploration to discover new strategies, while a lower ε leads to more exploitation of the currently learned Q-values.

Number of Episodes

The number of episodes defines the total iterations the agent undergoes during the training process. More episodes can improve convergence, but it may also increase the training time.

Reward Weights (w_1, w_2, w_3, w_4, w_5)

The reward weights control the contributions of various factors in the reward function. The weights are adjusted to balance preferences, assignment quality, and instructor constraints.

c, Reasoning and Justification

The selected parameter values were chosen based on prior research in the field of Reinforcement Learning and through experimentation with different combinations. I conducted preliminary experiments to determine the parameter settings that yielded stable and satisfactory results.

The parameter settings play a vital role in the performance of my proposed algorithm for solving the Teaching Assignment Problem. The chosen values strike a balance between exploration and exploitation, ensuring efficient convergence and effective assignment outcomes.

3.4.5 Reward Function

The reward function is a crucial component of the reinforcement learning algorithm used in the Teaching Assignment Problem. It plays a significant role in shaping the behavior of the learning agent, guiding it to make optimal decisions during the course scheduling process. The reward function is designed to measure the goodness of a particular action taken by the agent in a given state.

In the context of the Teaching Assignment Problem, the reward function takes into account various factors to evaluate the quality of an instructor's assignment to a specific course. Specifically, the reward function considers the teaching quality of each instructor, their preferences for time slots and subjects, and the number of courses assigned to them.

The teaching quality of an instructor is influenced by several aspects, such as their experience, expertise, and teaching history. The higher the teaching quality, the more favorable the assignment to that instructor would be. Additionally, instructors may have individual preferences for specific time slots or subjects, and these preferences are incorporated into the reward function. Assigning an instructor to their preferred time slots or subjects leads to higher rewards, reflecting their satisfaction with the assignment.

Moreover, the reward function also takes into account the workload of each instructor. Balancing the workload ensures that no instructor is overburdened with too many courses or underutilized with too few. To achieve this, the reward function incentivizes the assignment of courses to instructors in a way that aligns with their desired number of courses.

To compute the final reward for a particular assignment, the reward function aggregates these factors with respective weights. These weights can be fine-tuned to strike the right balance between teaching quality, preference fulfillment, and workload distribution, based on the specific requirements and priorities of the educational institution.

So in this problem I calculate the formula in **Overall Payoff function** as a reward for each state and action.

By optimizing the reward function through reinforcement learning, the Teaching

Assignment Problem aims to find a scheduling solution that maximizes the overall quality of the course assignments while considering the preferences and constraints of both the academic department and individual instructors.

3.4.6 Epsilon Greedy Algorithm

The epsilon-greedy algorithm is a fundamental exploration-exploitation strategy used in reinforcement learning to strike a balance between exploring new actions and exploiting the current best-known actions. In the context of the Teaching Assignment Problem, the epsilon-greedy algorithm is employed to guide the agent (scheduler) in selecting an instructor for each course assignment while considering both the immediate rewards and long-term benefits.

The algorithm introduces an exploration rate, denoted as epsilon (ϵ), which determines the probability of the agent taking a random action (exploration) versus choosing the action with the highest estimated reward (exploitation). At the beginning of the learning process, the exploration rate is typically set to a high value, encouraging the agent to explore different assignments and learn about the environment.

During each iteration of the assignment process, the agent decides whether to explore or exploit based on a random value drawn from a uniform distribution between 0 and 1. If the random value is less than epsilon, the agent explores by selecting an instructor randomly, regardless of their past performance. This exploration phase enables the agent to discover potentially better assignments and avoid prematurely converging to suboptimal solutions.

Conversely, if the random value is greater than or equal to epsilon, the agent exploits the current knowledge by selecting the instructor with the highest estimated reward based on the Q-values. The Q-values are updated as the agent gains more experience and feedback from the reward function during the learning process.

The exploration rate ϵ is an important hyperparameter in reinforcement learning algorithms, especially in strategies like epsilon-greedy. It controls the balance between exploration and exploitation during the learning process. At the beginning of the learning, a higher value of ϵ encourages the agent to explore more, i.e., take random actions to discover potentially better strategies. As the agent gains more experience, ϵ is reduced to prioritize exploitation, i.e., choosing the actions that are currently estimated to be the best.

The function `reduce_epsilon` updates ϵ using an exponential decay formula:

$$\epsilon_{\text{new}} = \text{min\_epsilon} + (\text{max\_epsilon} - \text{min\_epsilon}) \times \exp(-\text{decay\_rate} \times \text{episode})$$

where:

- ϵ_{new} is the new value of ϵ after the update.
- `min_epsilon` is the minimum value of ϵ allowed (= 0.001).
- `max_epsilon` is the initial value of ϵ (= 1).
- `decay_rate` is a hyperparameter that controls the rate at which ϵ decreases over episodes (= 0.0003).

By decreasing ϵ over time, the agent becomes more focused on exploiting the current knowledge gained from exploration, while still leaving room for occasional exploration to avoid getting stuck in suboptimal solutions. This approach helps the agent to converge to a good policy for the given problem.

This `reduce_epsilon` function is called during the training of the reinforcement learning model to gradually reduce the exploration rate as the number of episodes increases.

The epsilon-greedy algorithm is an essential component of the Q-learning process in the Teaching Assignment Problem. By intelligently combining exploration and exploitation, the algorithm enables the agent to efficiently explore the assignment space and find high-quality schedules that maximize the overall reward, while considering the conflicting interests of the academic department and instructors.

3.4.7 Q-Learning Algorithm

The Q-learning algorithm is a widely used reinforcement learning technique that enables an agent to learn an optimal policy in an unknown environment. In the context of the Teaching Assignment Problem, Q-learning can be applied to find an optimal assignment of instructors to classes, considering various preferences and constraints.

The Q-learning algorithm uses a Q-table to store Q-values, which represent the expected cumulative rewards for taking specific actions in different states. In this algorithm, we have the state space represents the current state of the system, which includes the current assignments of instructors to classes and the current schedule and $num\_actions$ actions, where each action corresponds to an instructor.

The Q-learning process consists of the following steps:

In each episode, the agent explores the environment and updates the Q-values based on the observed rewards. The exploration rate (ϵ) controls the balance between exploration and exploitation. As the algorithm progresses, the exploration rate decreases to focus more on exploiting the learned optimal policy.

After choose an action, I will check if that action is valid or not. If the action is not valid at that state, the Q value of that state-action is reduced and the agent will choose another action based on the greedy-epsilon algorithm, after several time if the action is still not valid, the agent will choose the action that have the highest Q value at that state, if action still not valid, the agent will end that episode and start new episode. If the action is valid, then the agent will receive reward and update the Q-value for the corresponding state-action pair using the Q-learning update rule:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \cdot [R(s,a) + \gamma \cdot \max_{a'} Q(s',a') - Q(s,a)]$$

where:

- $Q(s,a)$ is the Q-value for state-action pair (s, a).

- α is the learning rate, which determines the weight given to new information.

- $R(s,a)$ is the immediate reward received after taking action a in state s but the reward is just calculated at the first time update Q(s, a).

- γ is the discount factor, which determines the importance of future rewards.

- s' is the next state after taking action a.

- $\max_{a'} Q(s',a')$ is the maximum Q-value for all possible actions in state s'.

The Q-learning algorithm iteratively refines the Q-values until it converges to the optimal Q-function, which represents the optimal policy for assigning instructors to classes. The final Q-table contains the Q-values for each state-action pair, enabling the agent to make the best decisions to maximize the overall teaching quality and instructor satisfaction.

The fitness value represents the quality of the assignment, and the algorithm tries to find the assignment with the highest fitness value. The process continues for the specified number of episodes ($num\_episodes$) until the optimal assignment is obtained.

The Q-learning algorithm is particularly useful for problems with large state and action spaces, making it suitable for solving complex Teaching Assignment Problems with various preferences and constraints.

3.4.8 Training the Q-Learning Agent

The Q-learning agent is trained through multiple episodes, where each episode represents a complete run through the Teaching Assignment Problem. In each episode, the agent interacts with the environment, updating its Q-table based on

the rewards received. The training process continues until the agent's Q-values converge to an optimal policy or a predefined stopping criterion is met.

3.4.9 Evaluation and Optimization

In this part I evaluate the performance of the Q-learning agent's teaching schedule and explore potential optimization techniques to improve its results.

a, Evaluation Metrics

To assess the effectiveness of the Q-learning agent's teaching schedule, I use various evaluation metrics. These metrics include:

- **Teaching Quality Rate of Subjects:** This metric measures the overall teaching quality of the subjects in the schedule. It takes into account factors such as instructor preferences, teaching quality, and subject assignments.
- **Satisfaction Rate of Instructors:** The satisfaction rate of instructors reflects how well the teaching schedule meets the preferences and workload constraints of individual instructors. Higher satisfaction rates indicate a better allocation of classes.
- **Other Relevant Metrics:** Depending on the specific requirements of the Teaching Assignment Problem, additional evaluation metrics such as class coverage, class sizes, and instructor workload balance may also be considered.

b, Hyperparameter Tuning

During the training process, the Q-learning agent relies on various hyperparameters, such as learning rate, discount factor, and exploration rate (epsilon). To optimize the agent's performance, I conduct hyperparameter tuning experiments. I systematically vary the hyperparameters and evaluate the agent's teaching schedule with different settings to identify the best combination of hyperparameters.

In conclusion, I have presented a comprehensive methodology for addressing the Teaching Assignment Problem using Reinforcement Learning. The methodology aims to optimize the allocation of instructors to classes, considering various preferences and constraints, while maximizing the overall teaching quality. The methodology presented in this chapter offers a robust and efficient approach to tackle the complex Teaching Assignment Problem. In the next chapter, I will present the numerical results to validate the performance of my proposed approach and provide further insights into its efficiency and effectiveness.

CHAPTER 4. NUMERICAL RESULTS

4.1 Evaluation Parameters

In this section, I present the evaluation parameters used to assess the effectiveness of the Reinforcement Learning Approach in addressing the Teaching Assignment Problem (TAP).

1. **Teaching Quality of Subjects:** This parameter evaluates the teaching quality of each subject based on the assignments generated by the reinforcement learning algorithm.
2. **Instructor Satisfaction Rate:** The instructor satisfaction rate reflects the level of satisfaction with the assigned teaching assignments. This parameter considers factors such as the instructor's preferences, the number of classes assigned, and the teaching time slots. A higher satisfaction rate indicates that instructors are assigned classes and time slots that align with their preferences, expertise, and availability.
3. **Fitness Value:** the fitness value tracks the performance of the reinforcement learning agent as it undergoes training iterations. The fitness value represents the overall effectiveness of the agent in optimizing the Teaching Assignment Problem (TAP) over the course of its training process. It is a key indicator of how well the agent is learning to make informed decisions and improve its course-instructor assignments.

4.2 Simulation Method

The experiment is conducted following the methodology described in the previous section. After modeling the environment for the Teaching Assignment Problem (TAP), the Q-learning agent is trained using the following steps and parameter settings:

1. Environment Modeling: The TAP is formulated as a multi-agent game, where each player represents an instructor and their strategies correspond to different teaching assignments. The environment is constructed to include the set of instructors, classes, subjects, time slots, and their preferences. The reward functions are defined based on teaching quality, instructor satisfaction rate, and fitness value.

2. Q-learning Agent Initialization: The Q-learning algorithm is employed as the reinforcement learning approach. A Q-table is randomly initialized to store the Q-values for each state-action pair in the environment. Each Q-value represents the expected reward the agent can achieve by taking a specific action (teaching

a specific instructor) in a given state (current course and status of the previous courses).

3. Training the Q-learning Agent: The Q-learning agent undergoes training for 21,000 episodes. In each episode, the agent interacts with the environment and updates its Q-values using the Q-learning algorithm. During the training process, the agent explores the environment by taking actions based on an epsilon-greedy policy to balance exploration and exploitation.

4. Parameter Settings: The following parameter settings are used during the training of the Q-learning agent:

- Learning Rate (α): 0.7
- Discount Factor (γ): 1
- Exploration Rate (ε): Initially set to 1 and decayed linearly over episodes to encourage exploration in the beginning and exploitation later in the training process.
- Exploration Decay Rate: 0.0003 (per episode)
- Minimum Exploration Rate: 0.001 (ensuring some level of exploration even in later episodes)
- $w_1 = w_2 = w_3 = 1/3$, $w_4 = w_5 = 0.5$

The discount factor (γ) represents the importance given to future rewards compared to immediate rewards. A discount factor of 1 means that the agent considers future rewards to be just as important as immediate rewards.

w1, w2, and w3 are set to 1/3 each: These three weights represent the relative importance of three different objectives: subject preference, timeslot preference and number of courses that instructor want. Since each weight is set to 1/3, it means that each criterion is considered equally important in the evaluation. w4 and w5 are used to determine the importance of teaching quality and instructor satisfaction in the overall reward calculation. Both w4 and w5 are set to 0.5, which means that these two components are considered equally important.

5. Evaluation Metrics: During and after training, the Q-learning agent's performance is evaluated based on teaching quality of subjects, instructor satisfaction rate, and the fitness value achieved. These metrics are calculated at different stages of training to assess the agent's learning progress and convergence.

Conclusion: The simulation method involves training the Q-learning agent for 21,000 episodes using specific parameter settings to optimize the assignment of

classes to instructors in the Teaching Assignment Problem. The evaluation metrics provide insights into the effectiveness of the proposed approach in achieving teaching quality and instructor satisfaction.

4.3 Teaching Quality Rate of Subjects

After the Q-learning agent has completed its training process, it will propose a solution for the Teaching Assignment Problem. This solution will be based on the learned Q-values and the policy that the agent has developed through its exploration of the environment.

The proposed solution will involve the assignment of instructors to different classes based on their preferences, subject expertise, and time slot availability. The Q-learning agent will use the Q-values to determine the best actions (instructor assignments) to take in each state to maximize the overall reward, which is a combination of teaching quality and instructor satisfaction rate. In this chapter, the focus will be on evaluating the teaching quality rate of subjects based on the applied Reinforcement Learning (RL) approach. The quality of each course in the schedule is shown in the figure below:

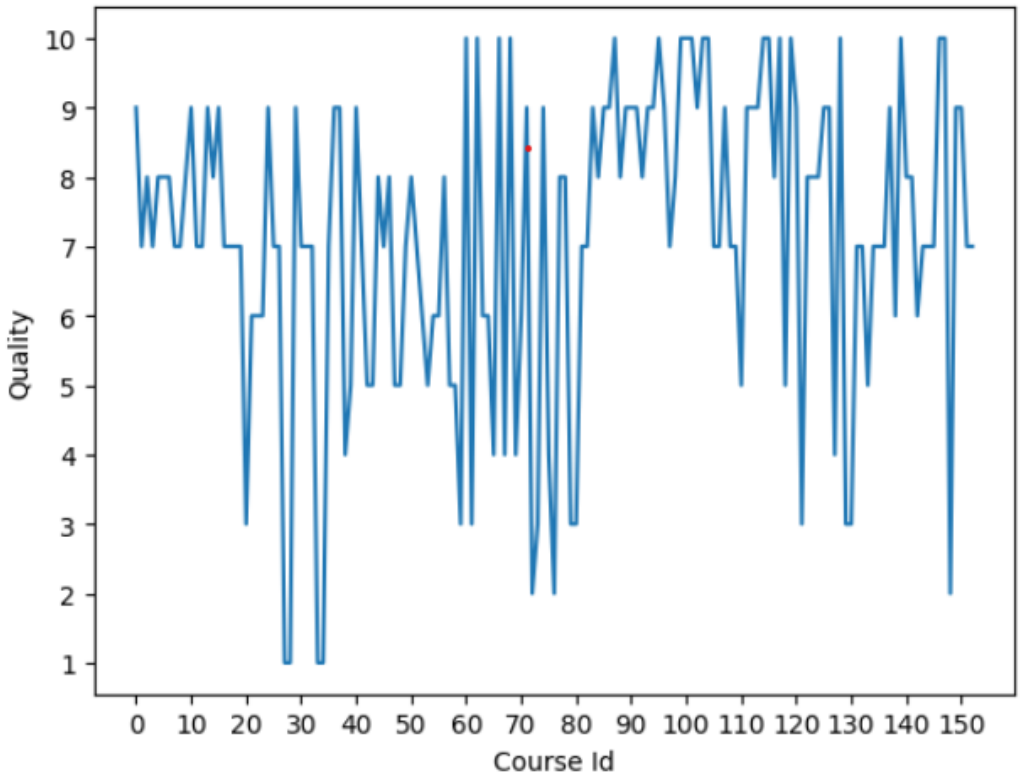


Figure 4.1: Quality of each Course

The line graph above depicts the quality of instructors for each course in the Teaching Assignment Problem. The x-axis represents the courses, while the y-axis represents the quality rating of instructors assigned to those courses. As shown in

the graph, the quality of instructors varies across different courses. Some courses have instructors with higher quality ratings, indicating their expertise and effectiveness in teaching those subjects. Conversely, other courses may have instructors with relatively lower quality ratings, suggesting areas for improvement or additional support.

Next, I calculated the overall quality rate of each subject by aggregating the quality ratings of instructors teaching courses related to that subject which is shown in figure below:

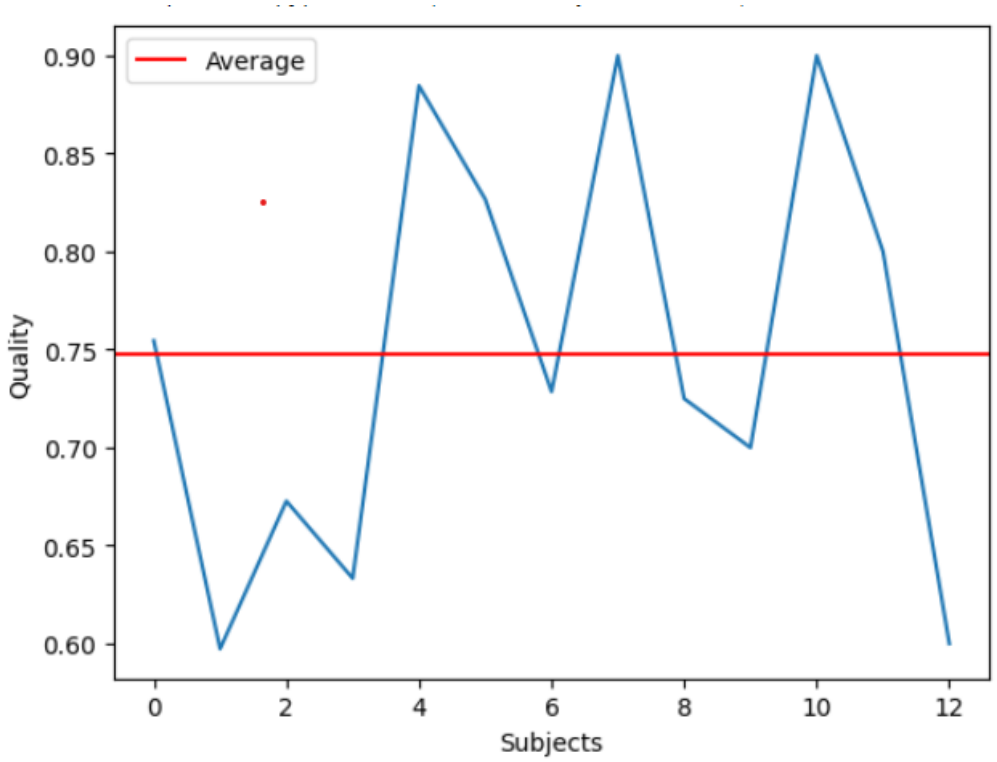


Figure 4.2: Quality of each Subject

The results demonstrated variations in the teaching quality of different subjects, with some subjects consistently achieving higher quality rates due to the allocation of instructors with better quality. The analysis of the experimental results reveals that the average teaching quality on each subject is approximately 0.75, indicating consistently high-quality course assignments to instructors with strong subject expertise.

4.4 Satisfaction Rate of Instructors

Upon evaluating the satisfaction rate of instructors, the experimental findings indicate a positive outcome. The implemented Reinforcement Learning approach successfully assigns instructors to courses that align with their preferences and constraints, leading to a high level of instructor satisfaction throughout the teaching assignment process.

To assess the satisfaction rate of instructors, I first present the preference timeslot distribution for each teacher in the following figure. It provides insights into the preferred time slots of individual instructors, highlighting their availability and willingness to teach at specific time periods.

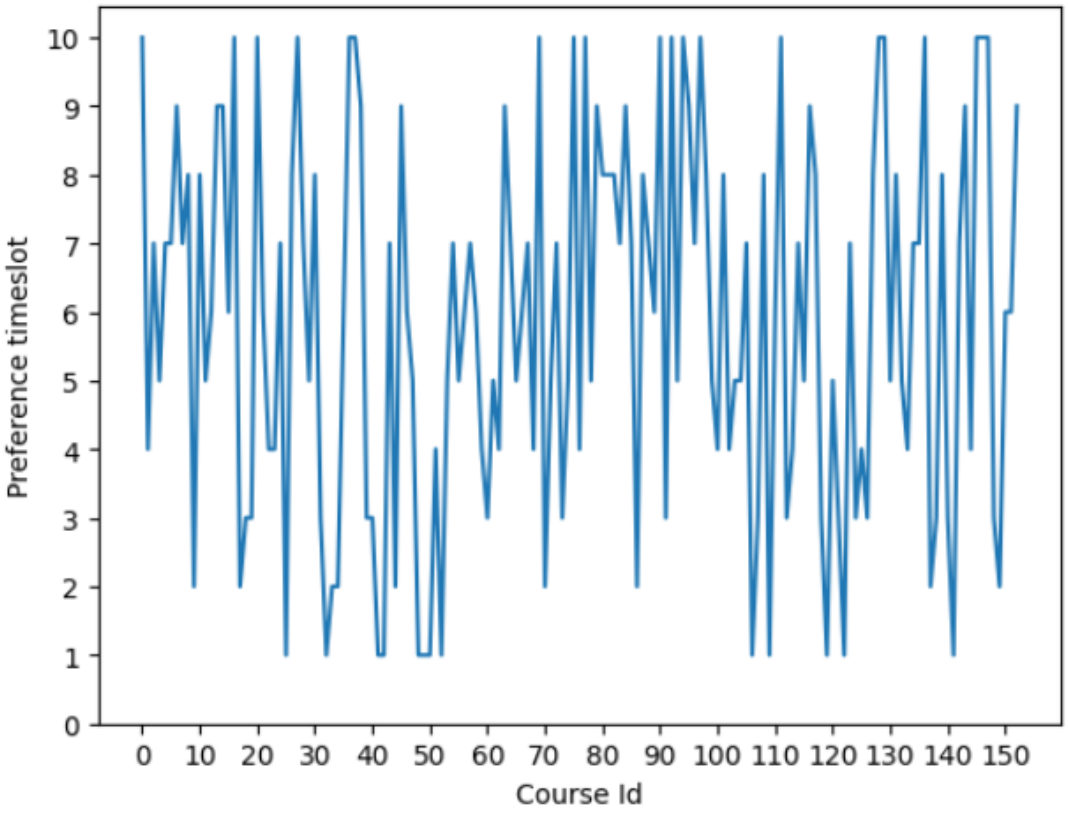


Figure 4.3: Preference timeslot

In the preference timeslot figure, each point on the graph represents an instructor's preference for a specific timeslot of a particular course. The x-axis corresponds to the course ID, indicating the different courses offered in the Teaching Assignment Problem. The y-axis represents the preference timeslot of the instructor for that specific timeslot of the corresponding course.

After analyzing the preference timeslot of instructors, the next step is to analyze the preference subject. The figure below show about it:

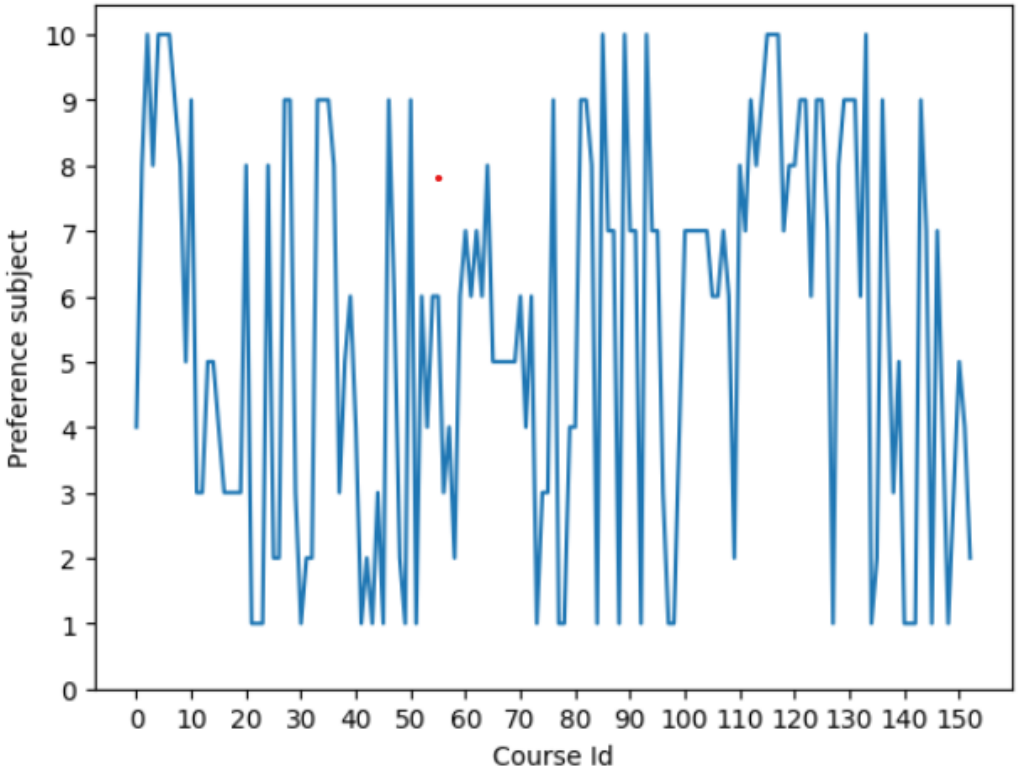


Figure 4.4: Preference subject

The preference subject figure illustrates the relationship between the course ID (representing different courses) and the preference level of each instructor for teaching the corresponding subject. A higher value for a specific course indicates that the instructor has a stronger preference for teaching that subject. Conversely, a lower value suggests that the instructor might not be as interested in teaching that particular subject.

The next thing I want to show is about the number of courses assigned to each instructor and compare it with their ideal course load.

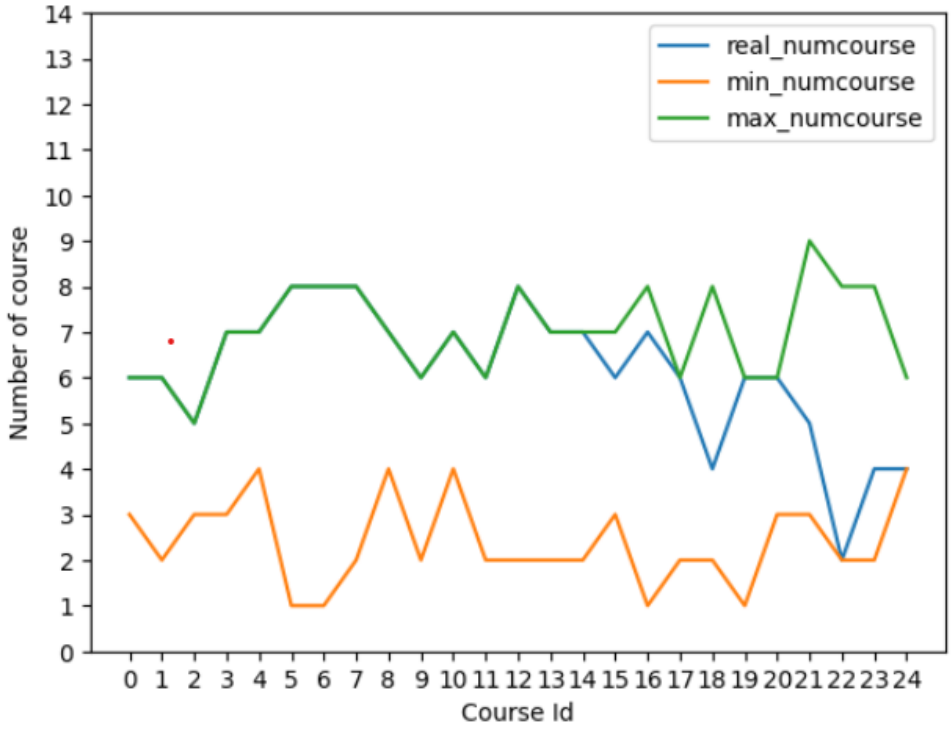


Figure 4.5: Number of course

The figure above illustrating the number of courses assigned to each instructor, the x-axis represents the instructor ID, and the y-axis represents the number of courses assigned to each instructor.

The green line on the graph indicates the maximum number of courses that each instructor can teach. This value represents the upper limit or capacity of the workload that the instructor is capable of handling. In other words, it is the maximum number of courses that the instructor is willing and able to teach.

The orange line represents the minimum number of courses that each instructor must teach. This value represents the lower limit or the minimum workload requirement that the instructor needs to fulfill.

The blue line on the graph represent the actual number of courses assigned to each instructor. By comparing the blue line with the green and orange lines, we can assess how well the teaching assignment aligns with the instructors' preferences and workload constraints.

If the blue line for an instructor falls within the range defined by the green and orange lines, it indicates a satisfactory course assignment that meets the instructor's workload preferences. On the other hand, if the blue line exceeds the green line, the instructor may be overburdened with courses beyond their capacity. If the blue line falls below the orange line, the instructor may be underutilized and not assigned

enough courses.

The detail information of number of course assigned to each instructor is shown in the figure below:

TeacherID	Name	Min_course	Max_course	Ideal_course	Number of assigned courses	Lệch so với ideal
1	TrangNTP	3	6	5	6	-1
2	TungHA	2	6	4	6	-2
3	SonNX	3	5	2	5	-3
4	HoangHM	3	7	6	7	-1
5	HuongNT	4	7	5	7	-2
6	BaoNX	1	8	4	8	-4
7	KienNTT	1	8	4	8	-4
8	KhangNK	2	8	6	8	-2
9	ChauNV	4	7	5	7	-2
10	ThuyNH	2	6	4	6	-2
11	ThuyNTN	4	7	6	7	-1
12	TuyNB	2	6	5	6	-1
13	LuongHN	2	8	5	8	-3
14	GiangNTK	2	7	5	7	-2
15	ThuyNH2	2	7	2	7	-5
16	PhanNT	3	7	5	6	-1
17	XuanNK	1	8	6	8	-2
18	TungNTH	2	6	6	6	0
19	HaNK	2	8	3	3	0
20	PhuongNTM	1	6	3	6	-3
21	TrungNB	3	6	6	6	0
22	BinhNT	3	9	3	3	0
23	VanNTK	2	8	4	6	-2
24	BurkeNS	2	8	4	2	2
25	JohnNS	4	6	5	4	1

Figure 4.6: Detail number of course

This analysis helps identify instructors who are receiving an appropriate number of courses based on their preferences and workload constraints, as well as those who may need adjustments in their course assignments to achieve a more balanced workload distribution. It ensures that the teaching assignment process respects instructors' preferences while maximizing their satisfaction and overall teaching quality.

After considering all three factors mentioned earlier, now I analyze the satisfaction rate of each instructor. The satisfaction rate is an essential metric that reflects how well the teaching assignment aligns with the preferences and workload constraints of individual instructors.

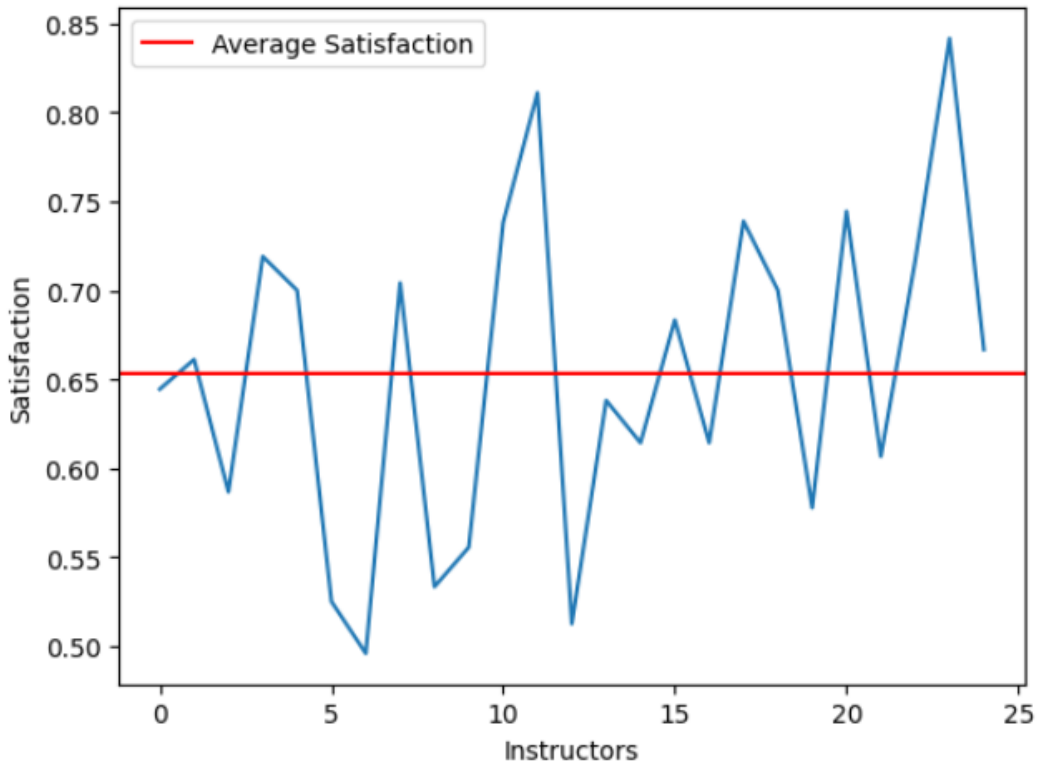


Figure 4.7: Satisfaction of instructors

The satisfaction rate of instructors is presented on the figure above. A higher satisfaction rate implies that the instructor's teaching assignment aligns well with their preferences and workload constraints, leading to a higher level of satisfaction. On the contrary, a lower satisfaction rate may indicate that the instructor's assignment does not fully align with their preferences, potentially leading to dissatisfaction. The average satisfaction rate of 0.65 indicates a moderately positive level of overall contentment and fulfillment among the instructors with their teaching assignments.

4.5 Fitness Value

The fitness value serves as a comprehensive metric that combines both teaching quality and instructor satisfaction, providing a holistic assessment of the proposed teaching assignment solutions. By integrating these two critical factors, the fitness value offers valuable insights into the effectiveness and optimality of the assignments made by the Q-learning agent.

A high fitness value suggests that the teaching assignment solution is optimal, with instructors assigned to courses that match their expertise and preferences, resulting in both high teaching quality and overall satisfaction. Conversely, a lower fitness value indicates potential areas for improvement in the assignment process.

Throughout the experimentation and analysis, the fitness value is continuously

monitored to evaluate the performance of the Q-learning agent and its ability to generate effective teaching assignments. This measure plays a crucial role in guiding the iterative refinement of the Q-learning algorithm and fine-tuning the parameter settings to achieve improved fitness values and, consequently, better teaching assignment solutions.

The fitness value when training the Q-learning agent is depicted in the below figure:

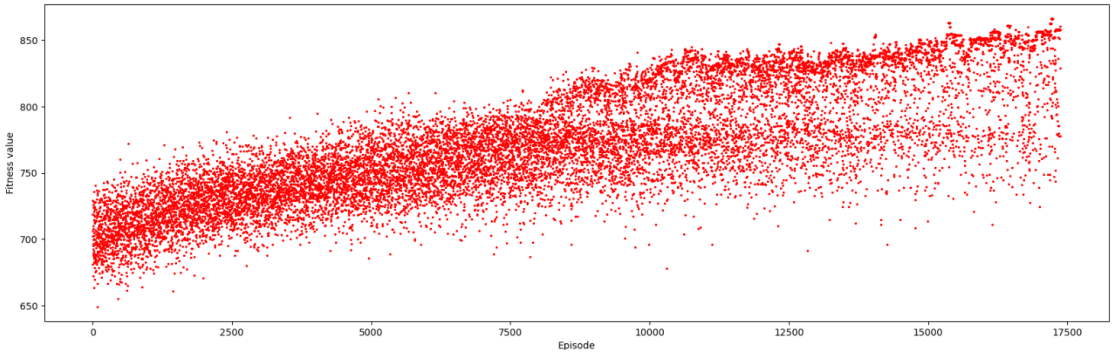


Figure 4.8: Fitness value

The x-axis represents the training episodes, while the y-axis represents the fitness value obtained at each episode. As the Q-learning agent undergoes training, the fitness value is continuously updated, reflecting the progress of the optimization process.

In the early stages of training, the fitness value may fluctuate as the agent explores different actions and policies. However, as the training progresses, the fitness value tends to stabilize and converge to a higher value. This convergence indicates that the Q-learning agent is effectively learning to make better teaching assignments that balance teaching quality and instructor satisfaction.

The fitness value's upward trend in the figure demonstrates the successful adaptation of the Q-learning agent to the Teaching Assignment Problem. As the agent gains more experience and updates its Q-values iteratively, it becomes more adept at selecting optimal actions that lead to higher fitness values. This improvement in fitness value signifies the agent's ability to achieve more favorable teaching assignments over time.

The observed trend in the fitness value serves as an important indicator of the Q-learning agent's performance. By monitoring the fitness value during training, we can assess the effectiveness of the reinforcement learning approach and track the agent's learning progress. The ultimate goal is to achieve a high and stable fitness

value, which indicates the Q-learning agent's ability to propose well-optimized teaching assignments that enhance both teaching quality and instructor satisfaction.

CHAPTER 5. CONCLUSIONS

5.1 Summary

In summary, this Graduation Thesis addressed the Teaching Assignment Problem (TAP) in an educational institution using a Reinforcement Learning (RL) approach. The main achievements and contributions of this work can be summarized as follows:

1. **Proposed RL Approach:** The thesis successfully applied a Q-learning agent to optimize the assignment of instructors to classes in an educational setting. By leveraging RL, the agent learned from experiences and made informed decisions, resulting in improved teaching assignments.
2. **Enhanced Teaching Quality:** The RL approach contributed to enhancing the overall teaching quality by considering instructors' preferences, subject expertise, and time slot availability. This resulted in more efficient and effective teaching assignments.
3. **Improved Instructor Satisfaction:** By incorporating instructor preferences and workload constraints, the RL agent achieved better instructor satisfaction rates in the course assignments. The proposed approach ensured fair distribution and optimization of workload for each instructor.
4. **Optimization of Teaching Assignment:** The RL agent's training process and Q-value updates led to the discovery of optimal teaching assignments that balanced teaching quality and instructor satisfaction. The agent continuously learned and adapted to maximize the fitness value over the training episodes.

However, certain challenges and open issues remain in the Teaching Assignment Problem domain:

1. **Scalability:** While the proposed RL approach demonstrated promising results, further research is needed to investigate its scalability to larger educational institutions with more complex teaching assignment scenarios.
2. **Fine-tuning Parameters:** The choice of hyperparameters in the RL algorithm can significantly impact the learning process and final results. More in-depth exploration and fine-tuning of these parameters could lead to even better performance.
3. **Generalization to Other Institutions:** The application of the RL approach may vary across different educational institutions with diverse organizational structures and requirements. Future research should assess the generalizability

of the proposed approach to various educational settings.

In conclusion, this Graduation Thesis contributes to the optimization of teaching assignment in an educational institution using a Reinforcement Learning approach. The proposed solution showcases the potential of Reinforcement Learning techniques to improve educational processes and resource allocation. While the current work achieved promising results, continued research and exploration will further advance the field and address the remaining challenges in teaching assignment optimization.

5.2 Suggestion for Future Works

In the pursuit of further advancing the field of Teaching Assignment Problem (TAP) optimization, there are several potential directions for future research and development:

1. **Advanced Reinforcement Learning Techniques:** Exploring and implementing more sophisticated RL algorithms, such as Deep Q Networks (DQNs) or Proximal Policy Optimization (PPO), could potentially enhance the learning capabilities and performance of the agent. These advanced techniques might offer better convergence and scalability for large-scale educational institutions. [6]
2. **Dynamic Environment Modeling:** Consideration of dynamic factors, such as changes in student enrollment, instructor preferences, or curriculum updates, could be integrated into the environment modeling. This would allow the RL agent to adapt and optimize teaching assignments in response to evolving conditions.
3. **Incorporating Human Feedback:** Investigating methods to incorporate human feedback and domain expert knowledge into the RL process could lead to more interpretable and reliable teaching assignments. Combining human expertise with RL algorithms can bridge the gap between artificial intelligence and human decision-making.
4. **Real-World Deployment and Validation:** Validating the proposed RL approach in real-world educational institutions would provide practical insights into its effectiveness and feasibility. Collaborating with educational institutions to deploy the solution in live settings could offer valuable feedback and improvements.
5. **Hybrid Approaches:** Exploring hybrid approaches that combine RL with other optimization techniques, such as Genetic Algorithms or Simulated Annealing, may offer novel solutions to the TAP. These hybrid methods can leverage the strengths of each approach and mitigate their respective weaknesses.

In conclusion, the future works in Teaching Assignment Problem optimization

are ripe with exciting possibilities. Continual research and development in these areas have the potential to significantly impact educational institutions' efficiency and resource allocation. By addressing the outlined suggestions, researchers and practitioners can further improve teaching assignment processes and contribute to the overall advancement of educational systems.

REFERENCE

[1] C. Reeves, "Heuristic search methods: A review," **january** 1996.

[2] S. Sharma **and** V. Chahar, "A comprehensive review on multi-objective optimization techniques: Past, present and future," *Archives of Computational Methods in Engineering*, **jourvol** 29, **page** 3, **july** 2022. DOI: 10.1007/s11831-022-09778-9.

[3] N. Giocoli, "Nash equilibrium," *History of Political Economy*, **jourvol** 36, **december** 2004. DOI: 10.1215/00182702-36-4-639.

[4] A. Hammoudeh, "A concise introduction to reinforcement learning," **february** 2018. DOI: 10.13140/RG.2.2.31027.53285.

[5] L. P. Chi, "Teaching assignment based on nash equilibrium and genetic algorithm,"

[6] K. Arulkumaran, M. P. Deisenroth, M. Brundage **and** A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, **jourvol** 34, **number** 6, **pages** 26–38, 2017. DOI: 10.1109/msp.2017.2743240. **url**: https://doi.org/10.1109%2Fmsp.2017.2743240.