People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
Higher School of Computer Science and Digital Technologies

# AI 3rd Mini-Project Report

**Theme:**
Constraint Satisfaction Problem for Semester 2 Timetable Scheduling

**The Group Members:**
OUARI Kenza
GUIR Abdelhai
MENAD Abdelmadjid
BAHLOUL Rami
ATOUI Yakoub

**Supervisors:**
Mrs. LEKEHALI Soumia
Mr BECHAR amine

# Contents

# Chapter 1

# Introduction

This project aims to tackle the intricate challenge of scheduling the Semester 2 timetable for first-year Computer Science students. Crafting a timetable that satisfies both the constraints and preferences of students and faculty alike is a daunting task that requires a systematic and rigorous approach.

The primary objective of this project is to formulate the problem in a manner that defines the variables, domains, and constraints, thereby ensuring the feasibility of the generated timetable. To achieve this, we will develop an application using a Constraint Satisfaction Problem (CSP) framework, a well-established approach for solving such problems. This application will integrate backtracking algorithms, such as the AC3 algorithm, which will aid in reducing variable domains by eliminating incompatible options, as well as heuristics such as MRV (Minimum Remaining Values), MCV (Most Constraining Variable), and LCV (Least Constraining Value) to enhance solution efficiency.

The ultimate goal is to generate a timetable that satisfies all constraints imposed by the scheduling problem while striving to adhere to optional constraints as much as possible. Although optional, these constraints are crucial for optimizing aspects such as the equitable distribution of faculty workload.

In this report, we will elaborate on the methodology followed to formulate the problem, implement constraints, and develop the scheduling application. We will also address the challenges encountered along the way and the solutions devised to overcome them. Finally, we will evaluate the performance of our scheduling algorithm and analyze the results obtained in our quest to solve this complex scheduling problem.

# Chapter 2

# Problem Formulation, Variable Domains, and Constraints

## 2.1 Problem Formulation

The problem involves scheduling various courses for Semester 2 within a week consisting of five days (Sunday to Thursday). Each day has multiple work slots. The goal is to allocate these slots to different courses while satisfying several constraints.

## 2.2 Variables

The variables in this problem are the time slots for each course and tutorial. Each course requires a specific number of slots, and these slots must be assigned in a way that satisfies all constraints.

## 2.3 Variable Domains

The domains for these variables are the available time slots across the week:

- **Days**: Sunday, Monday, Tuesday, Wednesday, Thursday
- **Slots**:
  - Sunday, Monday, Wednesday, Thursday: 5 slots each day
  - Tuesday: 3 morning slots

## 2.4 Constraints

Constraints ensure that the timetable adheres to the given requirements. They are divided into hard constraints and soft constraints.

### 2.4.1 Hard Constraints

1. The week consists of five days: Sunday, Monday, Tuesday, Wednesday, and Thursday.

2. Each day has five work slots, except Tuesday which has only three in the morning.
3. A maximum of three successive slots is allowed.
4. Lectures of the same course should not be scheduled in the same slot.
5. Different courses for the same group should have different slot allocations.

### 2.4.2 Soft Constraints (Optional for Extra Grades)

- Each teacher should have a maximum of two days of work.

# Chapter 3

# Task

The task is to develop an application that:

- Formulates the problem
- Defines variable domains
- Implements constraints
- Employs backtracking algorithms, AC3, MRV, MCV, and LCV
- Generates a feasible timetable that satisfies the hard constraints and aims to satisfy as many soft constraints as possible

# Chapter 4

# Problem Description

The scheduling involves the following courses and their corresponding requirements for 1st-year Computer Science students:

- **Sécurité**: one lecture + one TD (Teacher 1)
- **Méthodes formelles**: one lecture + one TD (Teacher 2)
- **Analyse numérique**: one lecture + one TD (Teacher 3)
- **Entrepreneuriat**: one lecture (Teacher 4)
- **Recherche opérationnelle 2**: one lecture + one TD (Teacher 5)
- **Distributed Architecture & Intensive Computing**: one lecture + one TD (Teacher 6)
- **Réseaux 2**: one lecture + one TD (Teacher 7), one TP (Teachers 8, 9, 10)
- **Artificial Intelligence**: one lecture + one TD (Teacher 11), one TP (Teachers 12, 13, 14)

# Chapter 5

# Application Development

## 5.1 Approach

### 5.1.1 Problem Formulation

We identified the variables (time slots for each course) and their respective domains (available time slots in a week). The constraints were then clearly defined to guide the allocation of these slots.

### 5.1.2 Constraints Implementation

Hard constraints were implemented first to ensure the timetable's feasibility. Soft constraints were implemented to improve the quality of the timetable.

### 5.1.3 Algorithm Implementation

We used backtracking algorithms enhanced with preprocessing and heuristics:

- **AC3 Algorithm**: Used as a preprocessing step to reduce variable domains by eliminating inconsistent values.
- **Heuristics (MRV, MCV, LCV)**: Applied during backtracking to efficiently select the next variable to assign and the value to assign to it.

## 5.2 Tools and Technologies

- **Programming Language**: Python
- **Constraint Package**: python-constraint or similar packages suitable for CSP problems

# Chapter 6

# Challenges and Solutions

## 6.1 Challenges

- **Domain Reduction**: Managing the domains of variables to ensure constraints are met and the solution space is feasible.

- **Constraint Propagation**: Efficiently propagating constraints to reduce the search space and improve the algorithm's performance.

## 6.2 Solutions

- **AC3 Algorithm**: Implemented as a preprocessing step to reduce domains by removing values that are inconsistent with constraints.

- **Heuristics (MRV, MCV, LCV)**: Utilized to select variables and values that minimize the remaining search space and handle the most constrained variables first.

# Chapter 7

# Results and Discussion

The application successfully generated feasible timetables that satisfied all hard constraints and maximized the satisfaction of soft constraints. Below are screenshots of the application interface and code snippets to illustrate the functionality and workings of the developed application.

## 7.1 Screenshots

### 7.1.1 Application Interface

### 7.1.2 Code Snippets

path/to/interface/screenshot.png

Figure 7.1: Application Interface

path/to/code/snippet_ac3.png

Figure 7.2: AC3 Algorithm Implementation

path/to/code/snippet_mrv.png

Figure 7.3: MRV Heuristic Implementation

# Chapter 8

# Conclusion

This assignment successfully demonstrated the application of CSP techniques to real-world scheduling problems. The developed application efficiently generated feasible timetables using backtracking algorithms with AC3 preprocessing and heuristics, ensuring adherence to hard constraints and optimizing for soft constraints.