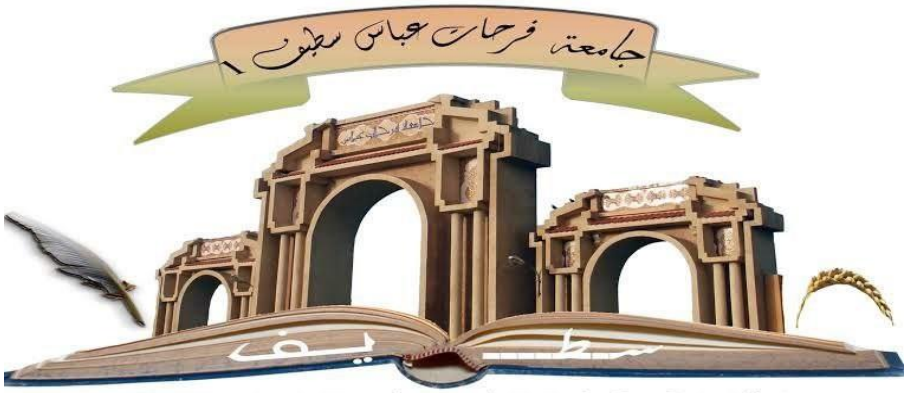
PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
UNIVERSITY FERHAT ABBAS SETIF 1



Faculty of Sciences
Department of Computer Science
Speciality: Data Engineering and Web Technologies (IDTW)

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Master's degree in Computer Science

---

Subject : AI-Based Timetable Generation Using Constraint Programming and Machine Learning with Synthetic Preference Modeling

---

Presented by :
Louaileche Oussama
Sellaoui Idris

Supervised by :
Dr. HAMDI Skander

2025-2026

Contents

List of Figures **15**

List of Tables **16**

1 Introduction **17**
1.1 Introduction to UTP: 17
1.2 Background and context of University Timetabling: 17
1.2.1 Definition of UTP: 17
1.2.2 Traditional Approaches for timetable generation: 18
1.2.3 Artificial intelligence Approaches for Timetabling: 18
1.3 Problem Statement and Motivation: 18
1.4 Dissertation Objective: 19
1.5 Research Questions: 19
1.6 Methodology Overview: 19
1.6.1 Modeling Phase: 20
1.6.2 Learning and Data Synthetic Phase: 20
1.6.3 Hybrid Solver Integration Phase: 20
1.6.4 Validation Phase: 20
1.7 Dissertation Outline: 20

2 State of the Art **22**
2.1 Timetabling problem overview: 22
2.2 Classification of constraints: 23
2.2.1 Hard Constraints: 23
2.2.2 Soft Constraints: 24
2.3 Classical approaches for UTP resolution: 24
2.3.1 Heuristics: 24
2.3.2 Metaheuristics: 25
2.4 Constraints programming: 26
2.4.1 Constraint Propagation (Filtering): 26
2.4.2 Systemic Search: 26
2.5 Machine learning in scheduling: 27
2.6 Hybrid approaches (Constraint Programming and Machine Learning): 28
2.7 Limitation of existing methods: 29
2.8 Machine learning foundations for UTP 30
2.8.1 Introduction: 30
2.8.2 Supervised learning principles: 30
2.8.3 Modeling soft constraints: 32
2.8.4 Machine learning models for preference learning: 32
2.8.5 Optimization and loss functions: 34
2.8.6 Integration with Constraint programming: 36
2.9 Conclusion 36

3 Design and Implementation of the Proposed System 37
3.1 Introduction: 37
3.2 Global Pipeline Architecture: 37
3.2.1 Overview of Python modules: 37
3.2.2 Data Flow Between Modules: 38
3.2.3 System Inputs and Outputs: 39
3.3 Dataset Generation: 39
3.3.1 Structure of levels, modules, and teachers: 39
3.3.2 Room Generation: 40
3.3.3 Timeslot Generation: 40
3.3.4 Session and teacher Assignment Generation: 40
3.3.5 Teacher preferences and unavailability: 41
3.3.6 Structure of Generated CSV files: 41
3.3.7 Data validation: 42
3.3.8 Limitations of the Dataset 42
3.4 CP-SAT Modeling: 43
3.4.1 CP-SAT scheduling process overview: 43
3.4.2 Decision variables: 44
3.4.3 Compatible Room Filtering: 44
3.4.4 Hard constraints: 45
3.4.5 Teacher Unavailability Constraints: 45
3.4.6 Quality Objective function: 45
3.4.7 Generating a feasible timetable: 46
3.5 Feature Extraction: 47
3.5.1 Role of feature engineering: 47
3.5.2 Extracted features: 47
3.5.3 Group gap calculation: 48
3.5.4 Teacher Gap calculation: 48
3.5.5 Room utilization rate: 48
3.5.6 Average room capacity usage: 49
3.5.7 Preference respected and violation: 49
3.6 Machine Learning model: 50
3.6.1 Training data generation: 50
3.6.2 Rule-based penalty function: 50
3.6.3 Diversification of CP-SAT objectives: 51
3.6.4 Random forest training: 51
3.6.5 ML model evaluation metrics: 52
3.6.6 Feature importance analysis: 52
3.6.7 Comparison with an alternative model (Gradient Boosting) 53
3.7 Hybrid solver: 54
3.7.1 Principle of the Hybrid approach: 54
3.7.2 Generation of feasible solution via CP-SAT: 54

3.7.3 Randomized objective function: 55
3.7.4 Candidate evaluation via machine learning model: 56
3.7.5 Best solution selection: 56
3.8 Automated Pipeline execution: 56
3.8.1 Role of the automated execution: 56
3.8.2 Pipeline step order: 57
3.8.3 File produced by each step: 58
3.8.4 Limitations and potential errors: 58
3.9 Conclusion: 58

4 Experiments, Evaluation, and Results 60
4.1 Introduction: 60
4.2 Experimental Environment: 60
4.2.1 Programming language and libraries: 60
4.2.2 Project file structure: 61
4.3 Experimental protocol: 61
4.4 Evaluation metrics: 61
4.4.1 Number of early classes: 61
4.4.2 Group gaps: 61
4.4.3 Teacher gaps 62
4.4.4 Room utilization rate: 62
4.4.5 Average room capacity usage: 62
4.4.6 Preferred slots respected: 62
4.4.7 Avoided slots violated: 63
4.4.8 Unavailable slots violated 63
4.4.9 Predicted penalty score: 63
4.5 Results of the simple CP-SAT solver: 63
4.6 Results of the ML model: 64
4.6.1 Random Forest performance: 64
4.6.2 Feature Importance: 64
4.7 Results of the Hybrid solver: 65
4.7.1 Best solution obtained: 65
4.7.2 Evolution of penalty across attempts: 65
4.7.3 Interpretation of the final score: 66
4.8 Comparison: CP-SAT and Hybrid: 66
4.8.1 Gap comparison: 66
4.8.2 Preference Comparison: 67
4.8.3 Room utilization comparison: 67
4.8.4 Discussion of Trade-offs: 67
4.9 Analysis of generated plots: 67
4.9.1 Metrics comparison plot: 67
4.9.2 Penalty by the hybrid attempt plot: 68
4.9.3 Feature Importance plot: 69
4.9.4 Code for plot Generation: 70

4.9.5 General interpretation of the graphs: 71
4.10 User Interface: 71
4.10.1 Role of the HTML interface: 71
4.10.2 Display structure: 72
4.10.3 Data export: 73
4.11 Discussion: 73
4.11.1 Contributions of the hybrid approach: 73
4.11.2 Limitations of the approach: 73
4.11.3 Threats to validity: 74
4.11.4 Possible Improvements: 74
4.12 Conclusion: 74

5 General Conclusion and Future Works: 76
5.1 General Conclusion 76
5.2 Future Works 77

Bibliography 79

تلخيص :

تتناول هذه المذكرة واحدة من أعقد المسائل في مجال البحوث العملياتية والذكاء الاصطناعي، ألا وهي مشكلة إعداد الجداول الزمنية الجامعية (University timetabling problem) ، حيث يُعَدُّ بناء جدول دراسي صالح وعالي الجودة تحدّياً حقيقياً نظراً لتشابك القيود الكثيرة، من توافر القاعات والأساتذة، إلى تجنّب التعارضات الزمنية، واحترام المتطلبات المؤسساتية ورغبات المستخدمين.

ولطالما عجزت المقاربات التقليدية، اليدوية منها والآلية، عن تحقيق التوازن المنشود بين الجدوى الصارمة من جهة، وتحسين الرضا عن التفضيلات من جهة أخرى. ومن هنا تنبثق فكرة هذا العمل.

تقترح هذه الدراسة مقاربة هجينة تجمع بين قوّتين متكاملتين: البرمجة بالقيود(ConstraintProgramming) التي تضمن احترام جميع القيود الصارمة دون استثناء، والتعلّم الآلي (Machine Learning)الذي يتعلّم القيود المرنة ويتنبّأ بها انطلاقاً من بيانات تفضيلات اصطناعية. وقد جرى دمج النموذجين ضمن إطار موحّد، يقود فيه التعلّمُ الآلي عمليةَ البحث لدى المحلّل القائم على القيود نحو حلول أكثر جودة.

وقد تمّ تقييم الأداء باستخدام بيانات اصطناعية وسيناريوهات محاكاة لجدولة أكاديمية واقعية. وأظهرت النتائج أن المقاربة الهجينة قادرة على توليد جداول زمنية قابلة للتطبيق، مع تحسين ملموس في مدى احترام تفضيلات المستخدمين مقارنةً بالطرائق التقليدية. غير أنّ بعض التحدّيات تظلّ قائمة، لا سيّما في ما يتعلّق بقابلية التوسّع والتعميم عند الانتقال إلى بيئات أوسع نطاقاً وأكثر ديناميكية.

الكلمات المفتاحية: مشكلة إعداد الجداول الزمنية الجامعية (UTP) ، البرمجة بالقيود، التعلّم الآلي، المقاربات الهجينة، الجدولة، القيود المرنة، القيود الصارمة، الذكاء الاصطناعي.

Résumé :

Ce mémoire aborde le problème de l'établissement des emplois du temps universitaires, l'un des problèmes d'optimisation combinatoire les plus complexes et les plus étudiés en recherche opérationnelle et en intelligence artificielle. Générer des emplois du temps académiques réalisables et de qualité est une tâche ardue en raison du grand nombre de contraintes, notamment la disponibilité des ressources, les conflits d'horaires et les exigences institutionnelles. Les approches manuelles et automatisées traditionnelles peinent souvent à concilier la stricte faisabilité et l'optimisation des préférences des utilisateurs.

Dans ce contexte, cette étude propose une approche hybride combinant la programmation par contraintes et l'apprentissage automatique afin d'améliorer l'efficacité et la qualité de la génération des emplois du temps. Un modèle de programmation par contraintes est développé pour garantir le respect de toutes les contraintes strictes, tandis qu'un modèle d'apprentissage automatique est conçu pour apprendre et prédire les contraintes souples à partir de données de préférences synthétiques. Le système proposé intègre les deux modèles dans un cadre unifié, où l'apprentissage automatique guide le processus de recherche du solveur de contraintes.

Les performances sont évaluées à l'aide de données synthétiques et de scénarios simulés d'organisation d'emplois du temps académique. Les résultats obtenus montrent que l'approche hybride est capable de générer des emplois du temps réalisables tout en améliorant la satisfaction des préférences des utilisateurs par rapport aux méthodes traditionnelles. Toutefois, des défis subsistent en termes d'extensibilité et de généralisation lorsqu'elles sont appliquées à des environnements à grande échelle et hautement dynamiques.

Mots-clés : Problème d'emploi du temps universitaire (UTP), Programmation par contraintes, Apprentissage automatique, Approches hybrides, Ordonnancement, Contraintes souples, Contraintes strictes, Intelligence artificielle (IA).

Abstract :

This dissertation addresses the University timetabling problem, which is one of the most complex and widely studied combinatorial optimization problems in the field of operations research and AI. Generating feasible and high-quality academic timetables is a challenging task due to the large number of constraints involved, including resource availability, scheduling conflicts, and institutional requirements. Traditional manual and automated approaches often struggle to balance strict feasibility with the optimization of user preferences.

In this context, this study proposes a hybrid approach that combines constraint programming and machine learning techniques to improve the efficiency and quality of timetable generation. A constraint programming model is developed to ensure the satisfaction of all the hard constraints, while a machine learning model component is designed to learn and predict soft constraints based on synthetic preference data. The proposed system integrates both models into a unified framework, where machine learning guides the search process of the constraint solver.

The performance is evaluated using synthetic data and simulated academic scheduling scenarios. The experimental results show that the hybrid approach is capable of generating feasible timetables while improving the satisfaction of user preferences compared to traditional methods. However, challenges remain in terms of scalability and generalization when applied to large-scale and highly dynamic environments.

Keywords: University Timetabling Problem (UTP), Constraint programming, Machine learning, Hybrid approaches, Scheduling, Soft constraints, Hard Constraints, AI.