Course guide
270611 - CPS - Combinatorial Problem Solving

Last modified: 02/02/2024

Unit in charge: Barcelona School of Informatics
Teaching unit: 723 - CS - Department of Computer Science.

Degree: MASTER'S DEGREE IN INNOVATION AND RESEARCH IN INFORMATICS (Syllabus 2012). (Optional subject).

Academic year: 2023 **ECTS Credits:** 6.0 **Languages:** English

LECTURER

Coordinating lecturer: ENRIC RODRIGUEZ CARBONELL

Others: Segon quadrimestre:
ENRIC RODRIGUEZ CARBONELL - 10

PRIOR SKILLS

Basic knowledge on the Linux operating system and the C++ programming language.
Basic knowledge on linear algebra, graph algorithms and logics.

DEGREE COMPETENCES TO WHICH THE SUBJECT CONTRIBUTES

Specific:
CEE3.2. Capability to use a wide and varied spectrum of algorithmic resources to solve high difficulty algorithmic problems.
CEE3.3. Capability to understand the computational requirements of problems from non-informatics disciplines and to make significant contributions in multidisciplinary teams that use computing.

Generical:
CG1. Capability to apply the scientific method to study and analyse of phenomena and systems in any area of Computer Science, and in the conception, design and implementation of innovative and original solutions.
CG3. Capacity for mathematical modeling, calculation and experimental designing in technology and companies engineering centers, particularly in research and innovation in all areas of Computer Science.

Transversal:
CTR6. REASONING: Capacity for critical, logical and mathematical reasoning. Capability to solve problems in their area of study. Capacity for abstraction: the capability to create and use models that reflect real situations. Capability to design and implement simple experiments, and analyze and interpret their results. Capacity for analysis, synthesis and evaluation.

Basic:
CB6. Ability to apply the acquired knowledge and capacity for solving problems in new or unknown environments within broader (or multidisciplinary) contexts related to their area of study.

TEACHING METHODOLOGY

The main feature of the teaching methodology is the use of materials accessible through the web, specifically designed for a self-learning course. These materials allow reformulating teaching in such a way that the traditional model of classes largely disappears.

Thus:

1. It regards the class as a baseline for work, which the student must continue and deepen on his/her own.

2. It builds upon high quality materials (slides, lists of problems, solved problems, examples of laboratory practical work, LP/SAT/CP software, bibliographic references).

3. It aims at motivating students, with examples, discussions, comments, etc... The intuitions behind the definitions, properties and techniques are discussed in group.

The laboratory will encourage independent work by the students. The role of the teacher will be mainly to assist and evaluate the students, who should work mostly autonomously.

LEARNING OBJECTIVES OF THE SUBJECT

1.Modelling problems arising from computer science and other disciplines in the solving paradigms considered in the course: constraint programming, linear integer programming, propositional satisfiability.
2.Becoming familiar with state-of-the-art tools for the solving paradigms considered in the course: constraint programming, linear integer programming, propositional satisfiability.
3.Understanding the algorithmic foundations of each of the solving paradigms considered in the course: constraint programming, linear integer programming, propositional satisfiability.

STUDY LOAD

Type	Hours	Percentage
Self study	96,0	64.00
Hours large group	54,0	36.00

Total learning time: 150 h

CONTENTS

Combinatorial Problems.

Description:
Informal definition. NP-complete problems vs. polynomial-time problems. Some examples and applications: propositional satisfiability, graph coloring, knapsack, bin packing, etc. Approaches to problem solving.

Constraint Programming.

Description:
Basic definitions. Constraint Satisfaction Problems. Examples. Local consistency: arc consistency, directional arc consistency, bounds consistency. Constraint propagation for global constraints: all different. Search algorithms: basic backtracking, forward checking, partial/full lookahead. Variable and value ordering heuristics. Constraint Optimization Problems. Modeling and solving problems with CP.

Linear Programming.

Description:
Review of linear programming: The simplex algorithm. Duality and the dual simplex. Modelling and solving problems with linear programming. Mixed integer linear programming. Branch & bound, cutting planes, branch & cut. Totally unimodular matrices. Network simplex algorithm. Modelling and solving problems with mixed integer linear programming.

SAT solving and extensions.

Description:
Propositional logic. The satisfiability (SAT) problem. DPLL algorithm. Resolution. Conflict-Driven Clause Learning SAT solvers. Modeling and solving problems with SAT: cardinality constraints, pseudo-boolean constraints. Satisfiability Modulo Theories.

ACTIVITIES

Introduction to Combinatorial Problems

Description:
Introduction to Combinatorial Problems

Specific objectives:
1, 2, 3

Related competencies :
CG3. Capacity for mathematical modeling, calculation and experimental designing in technology and companies engineering centers, particularly in research and innovation in all areas of Computer Science.
CG1. Capability to apply the scientific method to study and analyse of phenomena and systems in any area of Computer Science, and in the conception, design and implementation of innovative and original solutions.
CEE3.2. Capability to use a wide and varied spectrum of algorithmic resources to solve high difficulty algorithmic problems.
CEE3.3. Capability to understand the computational requirements of problems from non-informatics disciplines and to make significant contributions in multidisciplinary teams that use computing.
CTR6. REASONING: Capacity for critical, logical and mathematical reasoning. Capability to solve problems in their area of study. Capacity for abstraction: the capability to create and use models that reflect real situations. Capability to design and implement simple experiments, and analyze and interpret their results. Capacity for analysis, synthesis and evaluation.
CB6. Ability to apply the acquired knowledge and capacity for solving problems in new or unknown environments within broader (or multidisciplinary) contexts related to their area of study.

Full-or-part-time: 2h
Theory classes: 2h

Constraint Programming

Description:
Modelling and solving with Constraint Programming

Specific objectives:
1, 2, 3

Related competencies :
CG3. Capacity for mathematical modeling, calculation and experimental designing in technology and companies engineering centers, particularly in research and innovation in all areas of Computer Science.
CG1. Capability to apply the scientific method to study and analyse of phenomena and systems in any area of Computer Science, and in the conception, design and implementation of innovative and original solutions.
CEE3.2. Capability to use a wide and varied spectrum of algorithmic resources to solve high difficulty algorithmic problems.
CEE3.3. Capability to understand the computational requirements of problems from non-informatics disciplines and to make significant contributions in multidisciplinary teams that use computing.
CTR6. REASONING: Capacity for critical, logical and mathematical reasoning. Capability to solve problems in their area of study. Capacity for abstraction: the capability to create and use models that reflect real situations. Capability to design and implement simple experiments, and analyze and interpret their results. Capacity for analysis, synthesis and evaluation.
CB6. Ability to apply the acquired knowledge and capacity for solving problems in new or unknown environments within broader (or multidisciplinary) contexts related to their area of study.

Full-or-part-time: 26h 48m
Theory classes: 8h
Laboratory classes: 6h
Self study: 12h 48m

Linear Programming

Description:
Modelling and solving with Linear Programming

Specific objectives:
1, 2, 3

Related competencies :
CG3. Capacity for mathematical modeling, calculation and experimental designing in technology and companies engineering centers, particularly in research and innovation in all areas of Computer Science.
CG1. Capability to apply the scientific method to study and analyse of phenomena and systems in any area of Computer Science, and in the conception, design and implementation of innovative and original solutions.
CEE3.2. Capability to use a wide and varied spectrum of algorithmic resources to solve high difficulty algorithmic problems.
CEE3.3. Capability to understand the computational requirements of problems from non-informatics disciplines and to make significant contributions in multidisciplinary teams that use computing.
CTR6. REASONING: Capacity for critical, logical and mathematical reasoning. Capability to solve problems in their area of study. Capacity for abstraction: the capability to create and use models that reflect real situations. Capability to design and implement simple experiments, and analyze and interpret their results. Capacity for analysis, synthesis and evaluation.
CB6. Ability to apply the acquired knowledge and capacity for solving problems in new or unknown environments within broader (or multidisciplinary) contexts related to their area of study.

Full-or-part-time: 34h 36m
Theory classes: 10h
Laboratory classes: 4h
Self study: 20h 36m

SAT and Extensions

Description:
Modelling and solving with SAT and extensions

Specific objectives:
1, 2, 3

Related competencies :
CG3. Capacity for mathematical modeling, calculation and experimental designing in technology and companies engineering centers, particularly in research and innovation in all areas of Computer Science.
CG1. Capability to apply the scientific method to study and analyse of phenomena and systems in any area of Computer Science, and in the conception, design and implementation of innovative and original solutions.
CEE3.2. Capability to use a wide and varied spectrum of algorithmic resources to solve high difficulty algorithmic problems.
CEE3.3. Capability to understand the computational requirements of problems from non-informatics disciplines and to make significant contributions in multidisciplinary teams that use computing.
CTR6. REASONING: Capacity for critical, logical and mathematical reasoning. Capability to solve problems in their area of study. Capacity for abstraction: the capability to create and use models that reflect real situations. Capability to design and implement simple experiments, and analyze and interpret their results. Capacity for analysis, synthesis and evaluation.
CB6. Ability to apply the acquired knowledge and capacity for solving problems in new or unknown environments within broader (or multidisciplinary) contexts related to their area of study.

Full-or-part-time: 24h 48m
Theory classes: 8h
Laboratory classes: 4h
Self study: 12h 48m

Final Exam

Description:
The exam covers the topics of modelling and solving with constraint programming, linear programming and propositional satisfiability.

Specific objectives:
1, 2, 3

Related competencies :
CG3. Capacity for mathematical modeling, calculation and experimental designing in technology and companies engineering centers, particularly in research and innovation in all areas of Computer Science.
CG1. Capability to apply the scientific method to study and analyse of phenomena and systems in any area of Computer Science, and in the conception, design and implementation of innovative and original solutions.
CEE3.2. Capability to use a wide and varied spectrum of algorithmic resources to solve high difficulty algorithmic problems.
CEE3.3. Capability to understand the computational requirements of problems from non-informatics disciplines and to make significant contributions in multidisciplinary teams that use computing.
CTR6. REASONING: Capacity for critical, logical and mathematical reasoning. Capability to solve problems in their area of study. Capacity for abstraction: the capability to create and use models that reflect real situations. Capability to design and implement simple experiments, and analyze and interpret their results. Capacity for analysis, synthesis and evaluation.
CB6. Ability to apply the acquired knowledge and capacity for solving problems in new or unknown environments within broader (or multidisciplinary) contexts related to their area of study.

Full-or-part-time: 21h 48m
Guided activities: 2h
Self study: 19h 48m

Practical Work of Constraint Programming

Description:
The project consists in modelling and solving a combinatorial problem with constraint programming

Specific objectives:
1, 2, 3

Related competencies :
CG3. Capacity for mathematical modeling, calculation and experimental designing in technology and companies engineering centers, particularly in research and innovation in all areas of Computer Science.
CG1. Capability to apply the scientific method to study and analyse of phenomena and systems in any area of Computer Science, and in the conception, design and implementation of innovative and original solutions.
CEE3.2. Capability to use a wide and varied spectrum of algorithmic resources to solve high difficulty algorithmic problems.
CEE3.3. Capability to understand the computational requirements of problems from non-informatics disciplines and to make significant contributions in multidisciplinary teams that use computing.
CTR6. REASONING: Capacity for critical, logical and mathematical reasoning. Capability to solve problems in their area of study. Capacity for abstraction: the capability to create and use models that reflect real situations. Capability to design and implement simple experiments, and analyze and interpret their results. Capacity for analysis, synthesis and evaluation.
CB6. Ability to apply the acquired knowledge and capacity for solving problems in new or unknown environments within broader (or multidisciplinary) contexts related to their area of study.

Full-or-part-time: 10h
Self study: 10h

Practical Work of Linear Programming

Description:
The project consists in modelling and solving a combinatorial problem with linear programming

Specific objectives:
1, 2, 3

Related competencies :
CG3. Capacity for mathematical modeling, calculation and experimental designing in technology and companies engineering centers, particularly in research and innovation in all areas of Computer Science.
CG1. Capability to apply the scientific method to study and analyse of phenomena and systems in any area of Computer Science, and in the conception, design and implementation of innovative and original solutions.
CEE3.2. Capability to use a wide and varied spectrum of algorithmic resources to solve high difficulty algorithmic problems.
CEE3.3. Capability to understand the computational requirements of problems from non-informatics disciplines and to make significant contributions in multidisciplinary teams that use computing.
CTR6. REASONING: Capacity for critical, logical and mathematical reasoning. Capability to solve problems in their area of study. Capacity for abstraction: the capability to create and use models that reflect real situations. Capability to design and implement simple experiments, and analyze and interpret their results. Capacity for analysis, synthesis and evaluation.
CB6. Ability to apply the acquired knowledge and capacity for solving problems in new or unknown environments within broader (or multidisciplinary) contexts related to their area of study.

Full-or-part-time: 10h
Self study: 10h

Practical Work of SAT

Description:
The project consists in modelling and solving a combinatorial problem with SAT

Specific objectives:
1, 2, 3

Related competencies :
CG3. Capacity for mathematical modeling, calculation and experimental designing in technology and companies engineering centers, particularly in research and innovation in all areas of Computer Science.
CG1. Capability to apply the scientific method to study and analyse of phenomena and systems in any area of Computer Science, and in the conception, design and implementation of innovative and original solutions.
CEE3.2. Capability to use a wide and varied spectrum of algorithmic resources to solve high difficulty algorithmic problems.
CEE3.3. Capability to understand the computational requirements of problems from non-informatics disciplines and to make significant contributions in multidisciplinary teams that use computing.
CTR6. REASONING: Capacity for critical, logical and mathematical reasoning. Capability to solve problems in their area of study. Capacity for abstraction: the capability to create and use models that reflect real situations. Capability to design and implement simple experiments, and analyze and interpret their results. Capacity for analysis, synthesis and evaluation.
CB6. Ability to apply the acquired knowledge and capacity for solving problems in new or unknown environments within broader (or multidisciplinary) contexts related to their area of study.

Full-or-part-time: 10h
Self study: 10h

GRADING SYSTEM

50% of the final grade corresponds to theory. This grade will be obtained by means of a written exam at the end of the course.

50% of the final grade corresponds to laboratory. This grade will be obtained as the mean of three successive projects (one for CP, another one for LP, and another one for SAT) that the students will have to hand in.

BIBLIOGRAPHY

Basic:
- Biere, A. [et al.] (eds.). Handbook of satisfiability. 2nd ed. Amsterdam: IOS Press, 2021. ISBN 9781643681610.
- Cormen, T.H. [et al.]. Introduction to algorithms. 4th ed. Cambridge: MIT Press, 2022. ISBN 9780262046305.
- Rossi, F.; Beek, P. van ; Walsh, T. (eds.). Handbook of constraint programming. Amsterdam: Elsevier, 2006. ISBN 0444527264.
- Williams, H.P. Model building in mathematical programming. 5th ed. Chichester: Wiley & Sons, 2013. ISBN 9781118443330.

Complementary:
- Maros, I. Computational techniques of the simplex method. Boston: Kluwer Academic Publishers, 2003. ISBN 1402073321.

RESOURCES

Hyperlink:
- http://www.cs.upc.edu/~erodri/cps.html