DM877

Discrete Optimization

Lecture 1

Course Introduction
Constraint Programming

Combination

Simplification

Contradiction

Redundancy

Outline

You

- Background
  study program
  programming skills
  other optimization course?
- Expectations

Outline

1. Motivation

2. Course Organization

Main Aim of the Course

To enable the student to solve
discrete satisfaction and optimization problems
that arise in practical applications
by means of constraint programming

Discrete and Combinatorial Optimization

- Discrete optimization emphasizes the difference to continuous optimization, solutions are described by integer numbers or discrete structures
- Combinatorial optimization is a subset of discrete optimization.
- Combinatorial optimization is the study of the ways discrete structures (eg, graphs) can be selected/arranged/combined: Finding an optimal object from a finite set of objects.
- Discrete/Combinatorial Optimization involves finding a way to efficiently allocate resources in mathematically formulated problems.
- We will assume discrete variables with finte domains.

Discrete Optimization Problems

Discrete Optimization problems

They arise in many areas of
Computer Science, Artificial Intelligence, Operations Research...:

- allocating register memory
- planning, scheduling, timetabling
- Internet data packet routing
- protein structure prediction
- auction winner determination
- portfolio selection
- ...

Discrete Optimization Problems

Simplified models are often used to formalize real life problems

- finding models of propositional formulae (SAT)
- finding variable assignment that satisfy constraints (CSP)
- partitioning graphs or digraphs
- partitioning, packing, covering sets
- finding shortest/cheapest round trips (TSP)
- coloring graphs (GCP)
- finding the order of arcs with minimal backward cost
- ...

Example Problems

- They are chosen because conceptually concise, intended to illustrate the development, analysis and presentation of algorithms
- Although real-world problems tend to have much more complex formulations, these problems capture their essence

Elements of Combinatorial Problems

Combinatorial problems are characterized by an input,
i.e., a general description of conditions (or constraints) and parameters,
and a question (or task, or objective) defining
the properties of a solution.

They involve finding a grouping, ordering, or assignment
of a discrete, finite set of objects that satisfies given conditions.

Candidate solutions are combinations of objects or solution components that need not satisfy all given conditions.

Feasible solutions are candidate solutions that satisfy all given conditions.

Optimal Solutions are feasible solutions that maximize or minimize some criterion or objective function.

Approximate solutions are feasible candidate solutions that are not optimal but good in some sense.

Applied Character

Optimization problems are very challenging, seldom solvable exactly in polynomial time and no single approach is likely to be effective on all problems.
Solving optimization problems remains a very experimental endeavor: what will or will not work in practice is hard to predict. [HM]

Hence the course has applied character:

- We will learn the theory
- but also implement some models $\rightsquigarrow$ programming in MiniZinc
- and solve them with solvers that implement what we learn in the theory

Expected prerequisites

Students taking the course are expected to:

- Be able to use algorithms and data structures
- Be able to assess the complexity of the algorithms with respect to runtime and space consumption
- Be able to program

Outline

Course Organization

- Modeling Problems in CP
- Local Consistency
- Constraint Propagation
- Search
- Symmetry Breaking

Schedule

- Class schedule:
  - See course web page.
  - mitsdu.sdu.dk
- Working load:
  - Intro phase (Introfase): 28 hours, 14 classes
  - Skills training phase (Træningsfase): 14 hours, 7 classes
  - Study phase: (Studiefase) ?? hours

  We have 3 classes for 7 weeks scheduled (42 hours).

COVI19 Situation

Online vs on Campus?

COVID19 is transmitted through aerosols. We have enough evidence.
We need to:

- Be outdoors
- Wear fitted, quality masks
- Improve indoor ventilation

Assessment

- Obligatory Assignments:
  Two preparation assignments
  One final
- Preparation assignments must be passed.
- Final assignment graded with 7-grade scale + internal censor grade is weighted(?) average of midterm and final assignments.
- Preparation assignments can be prepared in pairs but individual submission $\rightsquigarrow$ Feedback
- Final assignments is individual and only limited communication is allowed.

Learning Objectives

For a top perfomance the student must demonstrate ability to:

- model a problem similar in nature to the ones seen in the course within the framework of constraint programming
- argue about the different modeling choices arising from the theory behind the components of constraint programming, including global constraints, propagators, search and branching schemes.
- develop a solution prototype in a constraint programming system
- undertake an experimental analysis, report the results and draw sound conclusions based on them
- describe the work done in an appropriate language including mathematical formalism

Content of the Graded Assignments

- Modeling
- Implementation (deliverable and checkable source code)
- Written description
- (Analytical) and experimental analysis
- Performance counts!

Competences wrt Degree

- plan and carry out scientific projects at the high professional level including managing work and development situations that are complex, unpredictable and require new solutions
- describe, analyze and solve advanced computational problems using the learned models
- analyze the advantages and disadvantages of various algorithms, especially in terms of resource consumptions
- elucidate the hypotheses of qualified theoretical background and critically evaluate own and others' research and scientific models
- develop new variants of the methods learned where the specific problem requires
- communicate through a written report research based knowledge and discuss professional and scientific problems with peers
- give expertise in discrete optimization and solution methods from the international research front

Communication media

- Public Web Page [WWW] $\Leftrightarrow$ BlackBoard e-learn.sdu.dk [BB] (link from http://www.imada.sdu.dk/~marco/DM877/)
- Announcements in BlackBoard
- Course Documents in [BB] (unless linked from [WWW])
- Discussion Board (anonymous) in [BB]
- Personal email marco@imada.sdu.dk
- Office visits
- (A-bit-earlier-than) Mid term evaluation in class

Literature

RBW F. Rossi, P. van Beek and T. Walsh (ed.), Handbook of Constraint Programming, Elsevier, 2006

SMT Peter J. Stuckey, Kim Marriott, Guido Tack. MiniZinc Handbook. 2020

- Coursera's "Basic Modeling for Discrete Optimization"
- Coursera's "Advanced Modeling for Discrete Optimization"

- Other sources: articles, slides, lecture notes

Agreement for the Exercise Sessions

- Read the text before meeting at the class
- If you encounter difficulties then take note of the question and bring it in the class; it may be useful also for others
- The meaning with the exercise classes is for you to get feedback, not to deliver new material
- All questions and comments are welcome
- There is not stupid/wrong question and by the way we all learn from mistakes.
- I can ask questions to everybody and it is not to punish someone. You can well say pass.

Class format

Be prepared for:

- Flipped classes: learn content at home, engage with material in class
- Problem solving in class
- Hands on experience with modeling and programming
- Experimental analysis of performance
- Discussion on exercises for home

These activities will be announced

They require study phase (= work outside the classes)