Pierre Flener

Optimisation Group
Department of Information Technology
Uppsala University
Sweden

Course 1DL442:
Combinatorial Optimisation and Constraint Programming,
whose part 1 is Course 1DL451:
Modelling for Combinatorial Optimisation
Outline

1. Constraint Problems

2. Constraint Programming Technology

3. Constraint-Based Modelling

4. History, Success Stories, and Opportunities
Outline

1. Constraint Problems

2. Constraint Programming Technology

3. Constraint-Based Modelling

4. History, Success Stories, and Opportunities
Many important real-life problems are NP-hard or worse and can only be solved exactly and fast enough by intelligent search, unless P = NP:

- Assignment: personnel rostering, resource allocation, . . .
- Configuration of products, design, experiment set-up, . . .
- Packing: container or vehicle loading, carpet cutting, . . .
- Routing of robots, vehicles, . . .
- Scheduling, planning, . . .
- . . . hybrid problems, such as time-tabling and transportation logistics . . .

**Definition**

In a constraint problem, values have to be found for all the decision variables within their given domains so that:

- All the given constraints on the decision variables are satisfied.
- Optionally: A cost is minimal, or a benefit is maximal.

Search spaces are often larger than the universe!
NP-hardness is not where the fun ends, but where it begins!
Outline

1. Constraint Problems

2. Constraint Programming Technology

3. Constraint-Based Modelling

4. History, Success Stories, and Opportunities
Constraint programming (CP) offers languages, methods, and tools for:

**what:** Modelling constraint problems in a **high-level** declarative language.

**how:** **Solving** constraint problems **intelligently**, 
   either by strategy-guided **systematic search** plus **inference**,
   or by strategy-guided **local search** plus **inference**.

**Slogan of CP:** Constraint Program = Model [ + Search ]

CP solvers are **complementary** in strength to those of:

- Operations Research (OR): linear programming (LP), integer LP (ILP),
  mixed integer programming (MIP), . . .
- Boolean satisfiability (SAT), satisfaction modulo theories (SMT), . . .
- . . .

This leads to **hybrid** solving technologies!

In *Algorithms and Data Structures 3 (1DL481)*, taught in period 3 (January to March), there are assignments on local search and MIP, SAT, SMT modelling.
Scope of Constraint Programming

CP has a wide scope, because it addresses:

- satisfaction problems and optimisation problems
- discrete decision variables and continuous decision variables
- linear constraints and non-linear constraints

in principle in any combinations thereof, by:

- systematic search, if optimality is more crucial than speed
- local search, if speed is more crucial than optimality
Outline

1. Constraint Problems
2. Constraint Programming Technology
3. Constraint-Based Modelling
4. History, Success Stories, and Opportunities
The constraint predicates (AllDifferent, Circuit, Table, ...) and structured variable types (sets, ...) allow us both to model the structure of a problem and to exploit that structure when solving the problem.

Dozens of constraint predicates (see the Catalogue) declaratively encapsulate complex inference algorithms.

There is no standardised CP modelling language: distinct CP solvers may support distinct predicates, possibly under distinct names and signatures, as well as distinct types.
Pride:

*Constraint programming represents one of the closest approaches computer science has yet made to the Holy Grail of programming: the user states the problem, the computer solves it.*

— Eugene Freuder, a CP pioneer
**Pride:**

*Constraint programming represents one of the closest approaches computer science has yet made to the Holy Grail of programming: the user states the problem, the computer solves it.*
— Eugene Freuder, a CP pioneer

**Prejudice:**

*The contribution of the article should be the reduction of an engineering problem to a known optimization format. [...] showcases pseudo code [...] submit this work to a journal interested in code semantics [...].*
— Reviewer of a paper of ours at a prestigious OR journal
Prejudice:

Constraint programming represents one of the closest approaches computer science has yet made to the Holy Grail of programming: the user states the problem, the computer solves it.

— Eugene Freuder, a CP pioneer

Pride:

The contribution of the article should be the reduction of an engineering problem to a known optimization format. […] showcases pseudo code […] submit this work to a journal interested in code semantics […].

— Reviewer of a paper of ours at a prestigious OR journal
Outline

1. Constraint Problems

2. Constraint Programming Technology

3. Constraint-Based Modelling

4. History, Success Stories, and Opportunities
Stand-Alone Languages and Solvers:

- **ALICE** by Jean-Louis Laurière, France, 1976
- **OPL**, by P. Van Hentenryck, USA, and ILOG, France: modelling language for both IBM ILOG CP Optimizer and IBM ILOG CPLEX Optimizer
- **Comet**, by P. Van Hentenryck and L. Michel, USA
- **MiniZinc**, at Monash University, Australia
- ... 

Libraries (the ones listed before “;” are open-source):

- **Prolog**: ECLiPSe, ...; SICStus Prolog, ...
- **C++**: Gecode, Google CP-SAT; IBM ILOG CP Optimizer, CHIP, ...
- **Java**: Choco, Google CP-SAT, JaCoP, MiniCP, ...; ...
- **Objective-C**: Objective-CP; ...
- **Scala**: OscaR.cp, OscaR.cbls; ...
- ...
Success Stories by CP Users and Contributors:

Success stories: CP is the technology of choice in routing, configuration, rostering, scheduling, timetabling, ...
You're watching

The Americans

Season 6: Ep. 4

Mr. and Mrs. Teacup

Oleg presses Philip to reveal what Elizabeth is working on. Paige joins her mom on an operation to steal a sensor. Philip breaks bad news to Henry.

And two: the concept of the propagation of constraints...
Opportunities for CP

Rapid prototyping (with high solving performance) when:
- The constraints are, still or again, subject to experiments.
- The partition into hard and soft constraints is not yet determined.

The combinatorial structure is impure, due to side constraints.

It is time to consider all or more problem constraints.

Domain knowledge is exploitable for problem-specific search.

It is a configuration problem.

It is a personnel rostering problem.

It is a scheduling or timetabling problem.