Topic 18: Conclusion
(Version of 1st December 2018)

Pierre Flener

Optimisation Group
Department of Information Technology
Uppsala University
Sweden

Course 1DL441:
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 & Opportunities
Outline

1. Constraint Problems
2. Constraint Programming Technology
3. Constraint-Based Modelling
4. History & Success Stories & Opportunities
Many important real-life problems are NP-hard or worse and can only be solved exactly & fast enough by intelligent search, unless $P = NP$:

- Personnel rostering, scheduling, time-tableing, . . .
- Transportation logistics: vehicle routing, . . .
- Packing: container or truck loading, carpet cutting, . . .
- Configuration, design, experiment set-up, . . .
- Alignment of bio-molecules, phylogeny, . . .
- Financial investment instrument design, . . .
- . . .

**Definition**

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

- All the given constraints on the 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 & Opportunities
Constraint programming (CP) offers methods & tools for:

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

**how:** Solving constraint problems intelligently,
by strategy-guided systematic search plus inference, or
by (meta-)heuristic-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), modulo theories (SMT)
- . . .

This leads to hybrid optimisation technologies!

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

CP has a wide scope, as it addresses:

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

in principle in any combinations thereof, by:

- systematic search, if optimality 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 & Opportunities
The constraint predicates (circuit, extensional, distinct, ...) and structured variable types (sets, ...) allow us both to model the structure of a constraint problem and to exploit that structure when solving it.

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 & Opportunities
Stand-alone languages:

- **ALICE** by Jean-Louis Laurière, France, 1976
- **CHIP** at ECRC, Germany, 1987 – 1990, then marketed by Cosytec, France
- **OPL**, by P. Van Hentenryck, USA, and ILOG, France: front-end to both ILOG CP Optimizer and ILOG CPLEX
- **Comet**, by P. Van Hentenryck and L. Michel, USA
- **MiniZinc**, at U. of Melbourne and Monash U., Australia
- ...  

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

- Prolog: **ECLiPSe**, ...; **SICStus Prolog**, ...
- C++: **Gecode**, **OR-Tools**; **IBM CP Optimizer**, **CHIP**, ...
- Java: **Choco**, **Google OR-Tools**, **JaCoP**, **MiniCP**, ...; ...
- Scala: **OscaR**; ...
- ...

...
Success Stories by CP Users and Contributors:

Success stories: CP = technology of choice in scheduling, configuration, personnel rostering, timetabling, . . .
Opportunities for CP

Rapid prototyping (with high solving performance) when:

- Constraints are, still or again, subject to experiments
- Partition into hard & soft constraints yet undetermined

Combinatorial structure is impure, due to side constraints.

It is time to consider all or more problem constraints.

Domain knowledge exploitable for problem-specific search.

It is a configuration problem.

It is a personnel rostering problem.

It is a scheduling problem.

It is a time-tabling problem.