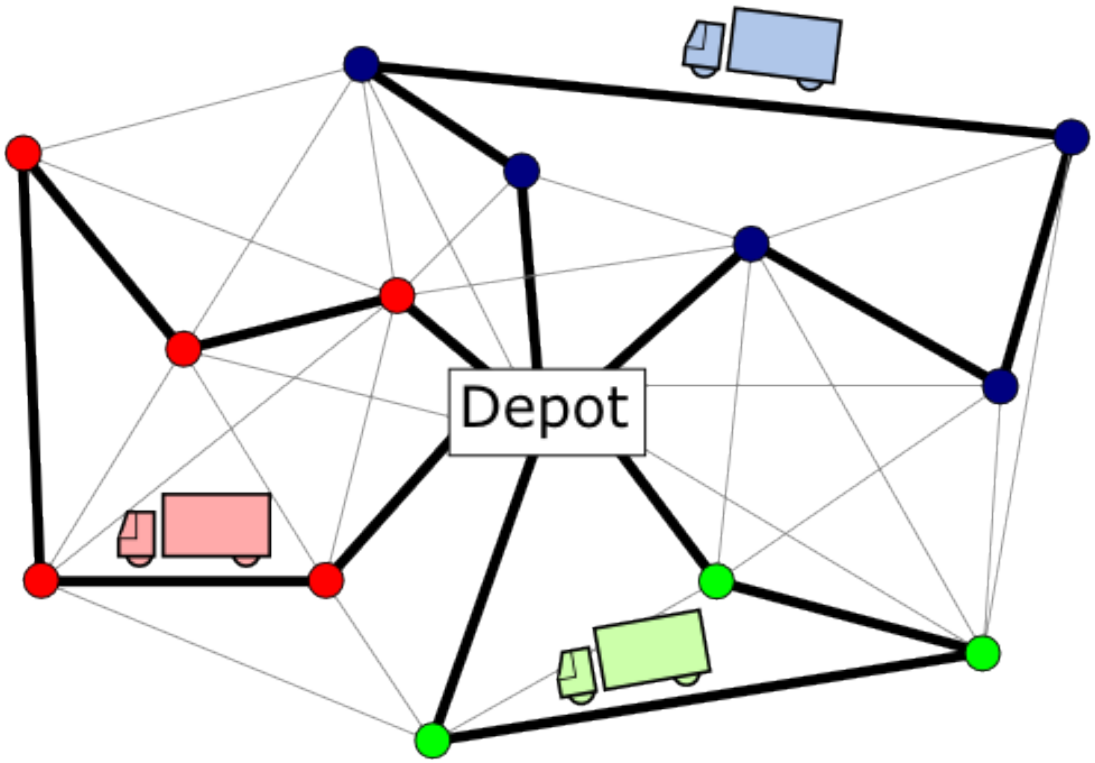
Constraint Programming in a Nutshell

Pierre Flener

ASTRA Research Group
on Combinatorial Optimisation
Department of Information Technology
Uppsala University
Sweden
www.it.uu.se/research/group/astra

NordConsNet 2017
2017-05-22

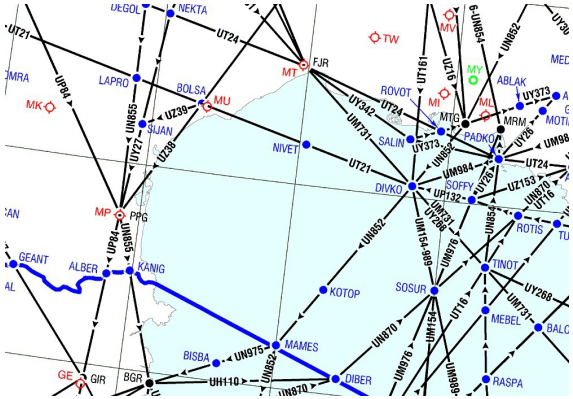
Optimisation



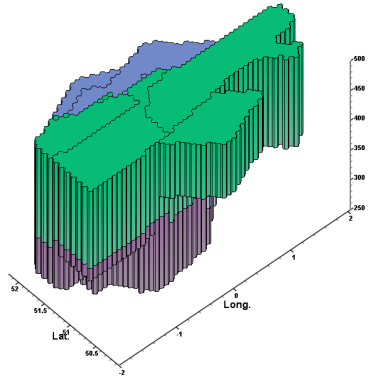
Optimisation is a science of service:
to scientists, to engineers, to artists, and to society.

Applications in Air Traffic Management

Demand vs capacity



Airspace sectorisation



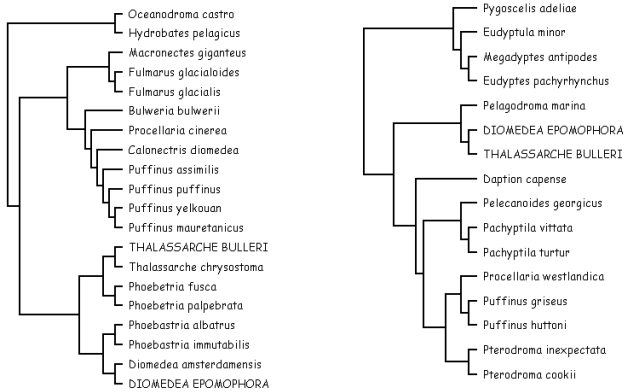
Contingency planning

Flow	Time Span	Hourly Rate
From: Arlanda To: west, south	00:00 – 09:00 09:00 – 18:00 18:00 – 24:00	3 5 2
From: Arlanda To: east, north	00:00 – 12:00 12:00 – 24:00	4 3
...	...	...

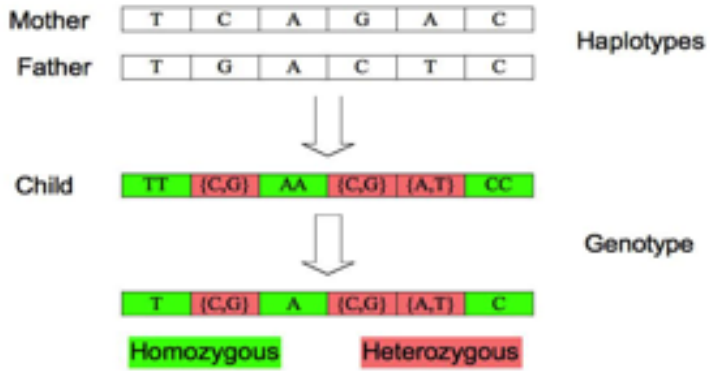
Workload balancing

Applications in Biology and Medicine

Phylogenetic supertree



Haplotype inference



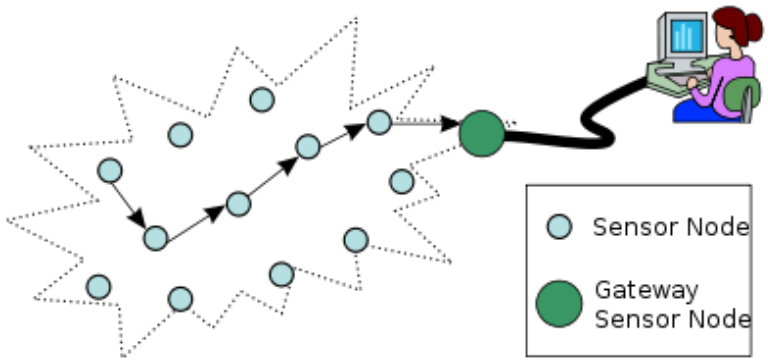
Medical image analysis

Doctor rostering

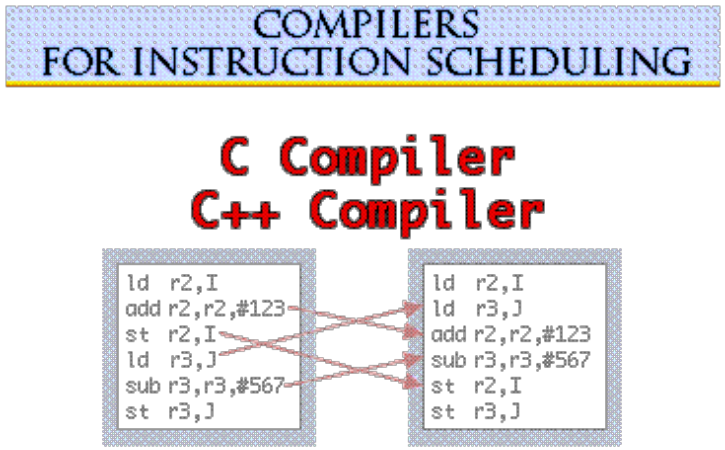
Applications in Programming and Testing

Robotic task sequencing

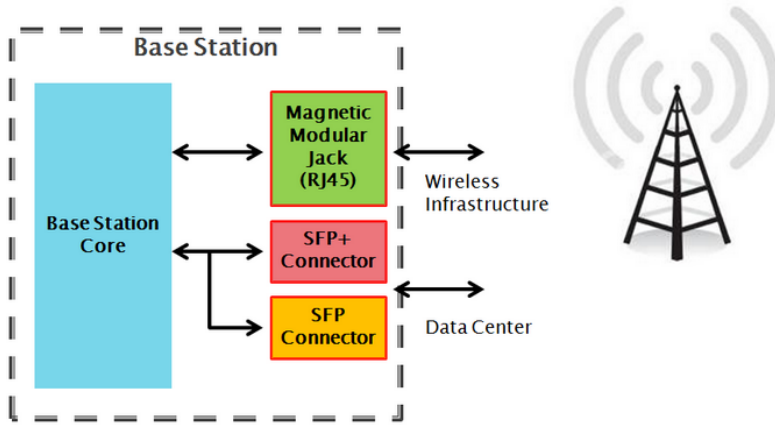
Sensor net configuration



Compiler design



Base station testing



Other Application Areas

School timetabling

	Monday	Tuesday	Wednesday	Thursday	Friday
9.00	MT2202 Ordinary Differential Equations FTh1		LABC S2072 Computer Graphics (G) Dual	MT2282 Numerical Analysis II Williamson, G03	
10.00	XMT2202 Ordinary Differential Equations NG15 / Roscoe, 2.3		LABC S2072 Computer Graphics (G) Dual	XMT2202 Ordinary Differential Equations Simon Engineering, Basement Theatre 3A XMT2282 Numerical Analysis II LG20	XMT2202 Ordinary Differential Equations NG15
11.00	CS2012 Algorithms and Data Structures 1.1		XMT2212 Further Linear Algebra 1.5		MT2202 Ordinary Differential Equations Stopford, Theatre 1
12.00	MT2212 Further Linear Algebra Roscoe, Theatre A	MT2282 Numerical Analysis II Williamson, G03	CS2072 Computer Graphics 1.1		MT2212 Further Linear Algebra Stopford, Theatre 1
1.00			PASS Peer Assisted Study I407 / LF15 / LF17 / I406		XMT2212 Further Linear Algebra Simon Engineering, Basement Theatre 4A
2.00	CS2072 Computer Graphics 1.1			XMT2212 Further Linear Algebra NG17	
3.00		CSTUT Tutorial			
4.00		CS2012 Algorithms and Data Structures 1.1			

Sports tournament design

Security: SQL injection?

Container packing

UPPSALA UNIVERSITET

Constraint Programming Technology

Constraint programming (CP) offers methods & tools for:

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

Constraint Programming Technology

Constraint programming (CP) offers methods & tools for:

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

and

how: **Solving** constraint problems intelligently by:

Constraint Programming Technology

Constraint programming (CP) offers methods & tools for:

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

and

how: **Solving** constraint problems intelligently by:

- either default search upon pushing a button

UPPSALA UNIVERSITET

Constraint Programming Technology

Constraint programming (CP) offers methods & tools for:

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

and

how: **Solving** constraint problems intelligently by:

- either default search upon pushing a button
- or systematic search guided by user-given heuristics

Constraint Programming Technology

Constraint programming (CP) offers methods & tools for:

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

and

how: **Solving** constraint problems intelligently by:

- either default search upon pushing a button
- or systematic search guided by user-given heuristics

plus inference, called propagation, but little relaxation.

UPPSALA UNIVERSITET

Constraint Programming Technology

Constraint programming (CP) offers methods & tools for:

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

and

how: **Solving** constraint problems intelligently by:

- either default search upon pushing a button
- or systematic search guided by user-given heuristics

plus inference, called propagation, but little relaxation.

A solver is a software that takes a model as input and tries to solve the modelled problem.

Constraint Programming Technology

Constraint programming (CP) offers methods & tools for:

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

and

how: **Solving** constraint problems intelligently by:

- either default search upon pushing a button
- or systematic search guided by user-given heuristics

plus inference, called propagation, but little relaxation.

A solver is a software that takes a model as input and tries to solve the modelled problem.

Slogan of CP:

Constraint Program = Model [+ Search]

Example (Doctor rostering)

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Doctor A							
Doctor B							
Doctor C							
Doctor D							
Doctor E							

Constraints to be **satisfied**:

1. #doctors-on-call / day $= 1$
2. #operations / workday ≤ 2
3. #operations / week ≥ 7
4. #appointments / week ≥ 4
5. day off after operation day
6. ...

Objective function to be **minimised**:

- Cost: ...

Example (Doctor rostering)

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Doctor A	call	–	oper	–	oper	–	–
Doctor B	app	call	–	oper	–	–	call
Doctor C	oper	–	call	app	app	call	–
Doctor D	app	oper	–	call	oper	–	–
Doctor E	oper	–	oper	–	call	–	–

Constraints to be **satisfied**:

1. #doctors-on-call / day $= 1$
2. #operations / workday ≤ 2
3. #operations / week ≥ 7
4. #appointments / week ≥ 4
5. day off after operation day
6. ...

Objective function to be **minimised**:

- Cost: ...

Example (Doctor rostering)

```
1  set of int: Days = 1..7;
2  set of int: Mon2Fri = 1..5;
3  set of int: Doctors = 1..5;
4  enum: ShiftTypes = {app, call, oper, none};
5
6  array[Doctors,Days] of var ShiftTypes: Roster;
7
8  solve minimize ...;  % objective function
9
10 constraint forall(d in Days)
11   (count(Roster[..,d],call) = 1);
12 constraint forall(w in Mon2Fri)
13   (count(Roster[..,w],oper) <= 2);
14 constraint count(Roster,oper) >= 7;
15 constraint count(Roster,app) >= 4;
16 constraint forall(d in Doctors)
17   (regular(Roster[d,..], (oper none|app|call|none)*);
18 ...  % other constraints
```

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
- . . .

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, ...; ...
- Scala: OscaR; ...
- ...

CP Users and Contributors:

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

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

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.


C. Allignol, N. Barnier, P. Flener, and J. Pearson.
Section 2 of Constraint programming [. . .].
Knowledge Engineering Review, 27(3):361–392, 2012.


K. R. Apt.
Principles of Constraint Programming.
Cambridge University Press, 2003.


Ph. Baptiste, C. Le Pape, and W. Nuijten.
Constraint-Based Scheduling.
Kluwer Academic Publishers, 2001.


P. Flener, M. Carlsson, and Ch. Schulte.
Constraint programming in Sweden.
IEEE Intelligent Systems, 24(2):87–89, 2009.


E. C. Freuder and M. Wallace.
Constraint technology and the commercial world.
IEEE Intelligent Systems, 15(1):20–23, 2000.


M. Milano, P. Van Hentenryck, et al.
The Future of Constraint Programming.
Constraints, special issue, 19(2), 2014.


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


P. Van Hentenryck and L. Michel.
Constraint-Based Local Search.
The MIT Press, 2005.


M. Wallace.
Constraint programming – The paradigm to watch.
Constraint Programming Letters, 1:7–13, 2007.