Constraint Programming
A Programming Paradigm on the Rise

Pierre Flener

ASTRA Research Group
Department of Information Technology
Uppsala University
Sweden

www.it.uu.se/research/group/astra

Int'l Workshop on Constraint Programming
for ATC & ATM, Brétigny (France), 2 December 2008

Constraint Problems
Constraint Program-ming (CP)
Constraint Modelling & Solving
History & Success Stories & Op-portunities of CP
Bibliography on CP

Outline

1. Constraint Problems
2. Constraint Programming (CP)
3. Constraint Modelling & Solving
4. History & Success Stories & Opportunities of CP
5. Bibliography on CP

Outline

1 Constraint Problems

2 Constraint Programming (CP)

3 Constraint Modelling & Solving

4 History & Success Stories & Opportunities of CP

5 Bibliography on CP

Constraint Problems

Example (The Airline-of-the-Year Problem)

	Abder	Frank	Marc	Nic	Phil	Steve	Uli
AF							
BA							
EasyJet							
KLM							
LH							
RyanAir							
SAS							

1. Constant jury: Every airline is tested by 3 judges.
2. Constant load: Every judge tests 3 airlines.
3. Equity: Every airline pair is tested by 1 common judge.

Constraint Problems

Example (The Airline-of-the-Year Problem)

	Abder	Frank	Marc	Nic	Phil	Steve	Uli
AF	✓	✓	✓				
BA	✓			✓	✓		
EasyJet	✓					✓	✓
KLM		✓		✓		✓	
LH		✓			✓		✓
RyanAir			✓	✓			✓
SAS			✓		✓	✓	

1. Constant jury: Every airline is tested by 3 judges.
2. Constant load: Every judge tests 3 airlines.
3. Equity: Every airline pair is tested by 1 common judge.

Constraint Problems

In a constraint problem, decisions have to be made so that:

- Some constraints are satisfied.
- Optionally: Some cost/benefit is minimised/maximised.

Many real problems **must** be solved by intelligent search:

- Rostering, scheduling, time-tabling
- Planning, collaborative decision making
- Configuration, design
- RNA structure prediction, alignment, sequencing, . . .
- Financial investment instrument design
- VLSI circuit layout
- Hardware / software specification verification
- . . .

Outline

1 Constraint Problems

2 Constraint Programming (CP)

3 Constraint Modelling & Solving

4 History & Success Stories & Opportunities of CP

5 Bibliography on CP

Constraint Programming

Constraint Programming (CP) offers methods and tools for:

- Effectively modelling constraint problems
- Efficiently solving constraint problems

Slogan of CP:

Constraint Program = Model + Search

CP works in a way orthogonal and complementary to

- Operations Research (OR):
  linear programming (LP), integer LP (ILP),
  mixed integer programming (MIP), . . .
- Propositional satisfiability (SAT)
- . . .

leading to hybridised satisfaction/optimisation technologies!

Scope of Constraint Programming

Constraint Programming handles:

- Satisfaction problems and optimisation problems
- Discrete variables and continuous variables
- Linear constraints and non-linear constraints

(in principle) in any combinations thereof, by:

- Global search, if optimality more important than speed
- Local search, if speed is more important than optimality

Support for:

- Exploiting multicore CPUs (without rewriting the model)
- Explanations
- Soft constraints
- …

Outline

1 Constraint Problems

2 Constraint Programming (CP)

3 Constraint Modelling & Solving

4 History & Success Stories & Opportunities of CP

5 Bibliography on CP

Constraint Modelling & Solving

Constraint Problems
Constraint Programming (CP)
Constraint Modelling & Solving
History & Success Stories & Opportunities of CP
Bibliography on CP

Example (Sudoku)

	2			3		9		7
	1							
4		7				2		8
		5	2				9	
			1	8		7		
	4				3			
				6			7	1
	7							
9		3		2		6		5

range $N = 1..9$
var s : **matrix**$[N, N]$ **of** N
solve
  $s[3, 1] = 4; \ \ldots$ // *load clues*
  forall r **in** N : $AllDifferent(s[r, *])$
  forall c **in** N : $AllDifferent(s[*, c])$
  forall r, c **in** N **by** 3 :
    $AllDifferent(s[r + 0..2, c + 0..2])$

Fill in the grid so that every row, every column, and every highlighted 3×3 box contains the digits 1 through 9.

Human beings solve Sudoku puzzles, Kakuro puzzles, etc, the way (global) CP solvers do: by propagation + search!

UPPSALA UNIVERSITET

Constraint Problems
Constraint Programming (CP)
Constraint Modelling & Solving
History & Success Stories & Opportunities of CP
Bibliography on CP

Constraint Modelling & Solving

Example (Sudoku)

	2			3		9		7
	1							
4		7				2		8
		5	2				9	
			1	8		7		
	4				3			
				6			7	1
	7							
9		3		2		6		5

range $N = 1..9$
var s : **matrix**$[N, N]$ **of** N
solve
  $s[3,1] = 4;$ … // *load clues*
  forall r **in** N : $\mathit{AllDifferent}(s[r, *])$
  forall c **in** N : $\mathit{AllDifferent}(s[*, c])$
  forall r, c **in** N **by** 3 :
    $\mathit{AllDifferent}(s[r + 0..2, c + 0..2])$

Fill in the grid so that every row, every column, and every highlighted 3×3 box contains the digits 1 through 9.

Human beings solve Sudoku puzzles, Kakuro puzzles, etc, the way (global) CP solvers do: by propagation + search!

Constraint Modelling & Solving

Example (Sudoku)

	2			3		9		7
	1							
4		7				2		8
		5	2				9	
			1	8		7		
	4				3			
				6			7	1
	7							
9		3		2		6		5

range $N = 1..9$
var s : **matrix**$[N, N]$ **of** N
solve
  $s[3, 1] = 4; \ \ldots$ // *load clues*
  forall r **in** N : $AllDifferent(s[r, *])$
  forall c **in** N : $AllDifferent(s[*, c])$
  forall r, c **in** N **by** 3 :
    $AllDifferent(s[r + 0..2, c + 0..2])$

Fill in the grid so that every row, every column, and every highlighted 3×3 box contains the digits 1 through 9.

Human beings solve Sudoku puzzles, Kakuro puzzles, etc, the way (global) CP solvers do: by propagation + search!

Constraint Solving: Propagation

Example (*AllDifferent*)

Consider the n-ary constraint *AllDifferent*, with $n = 4$ here:

$$AllDifferent([a, b, c, d]) \qquad (1)$$

Declaratively, (1) is equivalent to $\frac{n(n-1)}{2}$ binary constraints:

$$a \neq b,\ a \neq c,\ a \neq d,\ b \neq c,\ b \neq d,\ c \neq d \qquad (2)$$

Operationally, (1) prunes much stronger than (2). Example:

$$a \in \{2, 3\},\ \ b \in \{2, 3\},\ \ c \in \{1, 3\},\ \ d \in \{1, 2, 3, 4, 5\}$$

No pruning by (2). But perfect pruning by (1). Propagator suspended as constraint not surely true. If other propagator infers $a = 2$, then *AllDifferent* propagator is awakened, to infer $b = 3$ and be killed, as its constraint then surely true.

Constraint Solving: Propagation

Example (*AllDifferent*)

Consider the n-ary constraint *AllDifferent*, with $n = 4$ here:

$$AllDifferent([a, b, c, d]) \tag{1}$$

Declaratively, (1) is equivalent to $\frac{n(n-1)}{2}$ binary constraints:

$$a \neq b,\ a \neq c,\ a \neq d,\ b \neq c,\ b \neq d,\ c \neq d \tag{2}$$

Operationally, (1) prunes much stronger than (2). Example:

$$a \in \{2, 3\},\ b \in \{2, 3\},\ c \in \{1, \cancel{3}\},\ d \in \{\cancel{1}, \cancel{2}, \cancel{3}, 4, 5\}$$

No pruning by (2). But perfect pruning by (1). Propagator suspended as constraint not surely true. If other propagator infers $a = 2$, then *AllDifferent* propagator is awakened, to infer $b = 3$ and be killed, as its constraint then surely true.

Constraint Solving: Propagation

Example (*AllDifferent*)

Consider the n-ary constraint *AllDifferent*, with $n = 4$ here:

$$AllDifferent([a, b, c, d]) \quad (1)$$

Declaratively, (1) is equivalent to $\frac{n(n-1)}{2}$ binary constraints:

$$a \neq b,\ a \neq c,\ a \neq d,\ b \neq c,\ b \neq d,\ c \neq d \quad (2)$$

Operationally, (1) prunes much stronger than (2). Example:

$$a \in \{2, 3\},\ b \in \{2, 3\},\ c \in \{1, \not{3}\},\ d \in \{\not{1}, \not{2}, \not{3}, 4, 5\}$$

No pruning by (2). But perfect pruning by (1). Propagator suspended as constraint not surely true. If other propagator infers $a = 2$, then *AllDifferent* propagator is awakened, to infer $b = 3$ and be killed, as its constraint then surely true.

Constraint Modelling: Global Constraints

Global constraints, such as *AllDifferent*, *Knapsack*, etc are a much admired key feature of CP: they allow the

preservation of combinatorial sub-structures

of a constraint problem, while modelling and while solving it.

Dozens of global constraints have been implemented so far, declaratively encapsulating advanced propagation algorithms from combinatorics, graph theory, flow theory, matching theory, geometry, automata theory, etc.

Constraint Modelling & Solving

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

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

Constraint Modelling & Solving

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

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

Outline

1. Constraint Problems
2. Constraint Programming (CP)
3. Constraint Modelling & Solving
4. **History & Success Stories & Opportunities of CP**
5. Bibliography on CP

History of CP

Stand-alone languages:

- ALICE by Jean-Louis Laurière, France, 1976
- CHIP at ECRC (D), marketed by Cosytec (F)
- OPL, by P. Van Hentenryck (B), marketed by ILOG (F): front-end to both ILOG CP Optimizer and ILOG CPLEX
- Comet, by P. Van Hentenryck and L. Michel (B, USA): front-end to CP, CBLS, and MILP solvers

Libraries (the ones before semi-colon ';' are open-source!):

- C++: Gecode; ILOG CP Optimizer (ex Ilog Solver), ...
- Java: Choco, Gecode/J, JaCoP; Koalog, ...
- Prolog: ECLiPSe, GNU Prolog; SICStus Prolog, ...

The *Association for Computing Machinery* (ACM) has identified CP as a strategic direction in computing research.

CP Success Stories I

Since the 1990s, CP is deployed by many industry leaders:

Alcatel·Lucent

CHRYSLER ENGINEERED BEAUTIFULLY

Planning of satellite missions

Vehicle production optimisation

CISCO

Routing

france telecom

Cabling

i2 The Supply Chain Results Company

Supply chain management

JEPPESEN A BOEING COMPANY

Crew rostering

MONSANTO

Production scheduling

UPPSALA UNIVERSITET

Constraint Problems
Constraint Programming (CP)
Constraint Modelling & Solving
History & Success Stories & Opportunities of CP
Bibliography on CP

CP Success Stories II

ORACLE	Scheduling
SAP	Logistics software
SIEMENS	Control software validation, circuit verification
SNCF	Resource allocation
Whirlpool	Manufacturing
XEROX	Copier component specification

⋮

CP has become the technology of choice in short-term scheduling, rostering, timetabling, and configuration.

Opportunities for CP

- Rapid prototyping (with high solving performance) if:
  - Constraints (still / again) subject to experiments.
  - Partition into hard & soft constraints not determined yet.
- Combinatorial structure impure by side constraints.
- Time to consider all / more problem constraints.
- Domain knowledge usable for problem-specific search.
- It is a (short-term) scheduling problem.
- It is a rostering problem.
- It is a time-tabling problem.
- It is a configuration problem.

Outline

1. Constraint Problems
2. Constraint Programming (CP)
3. Constraint Modelling & Solving
4. History & Success Stories & Opportunities of CP
5. **Bibliography on CP**

UPPSALA UNIVERSITET

Constraint Problems

Constraint Programming (CP)

Constraint Modelling & Solving

History & Success Stories & Opportunities of CP

Bibliography on CP

Introductory CP Bibliography I

- 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.

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

- E. C. Freuder and M. Wallace.
  Constraint programming.
  In: Burke and Kendall (editors), *Search Methodologies*, chapter 9, pages 239–272. Springer-Verlag, 2005.

Introductory CP Bibliography II

- 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.
- P. Van Hentenryck, V. Saraswat, et al.
  Strategic directions in constraint programming.
  ACM Computing Surveys, 28(4):701–726, 1996.
  Also see `http://www.acm.org/surveys/sdcr/`.
- M. Wallace.
  Constraint programming – The paradigm to watch.
  Constraint Programming Letters, 1:7–13, 2007.