Constraint Programming

Dr Rong Qu
School of Computer Science
Automated Scheduling, Optimization and Planning (ASAP) Group
University of Nottingham, Nottingham, NG8 1BB, UK
rxq@cs.nott.ac.uk

Overview of the Course
Research in ASAP Group

- **Meta-heuristics**
  - evolutionary algorithm, tabu search, simulated annealing, variable neighbourhood search, particle swarm optimization, immune systems, hyper-heuristics, etc
  - multi-objective, fitness landscape analysis

- **Decision support systems**
  - case based reasoning, knowledge based systems, etc

- **Constraint Programming**

- **OR optimization**
  - integration of IP/ILP/MIP with meta-heuristics
  - modelling
Research in ASAP Group

- Personnel scheduling (nurse rostering, etc)
- Educational timetabling
- Bin packing
- Multicast network routing
- Computational finance (portfolio optimization)
- Graph colouring
- Job shop scheduling
- Space allocation
- ...

Constraint Programming - Introduction
Research in ASAP Group

- Multi-Disciplinary International Scheduling conference: Theory and Applications (MISTA bi-annual conferences)
- Practice and Theory of Automated Timetabling (PATAT bi-annual conferences)
- EURO Working Group on Automated Time-Tabling (WATT)
- *Journal of Scheduling*, impact factor: 1.0
- IEEE Congress on Evolutionary Computation (CEC’2010), Barcelona
- 2011 IEEE Symposium on Computational Intelligence in Scheduling, Singapore
Course Schedule

- Lectures 1 & 2  **Introduction**
  - Introduction to constraint programming, constraint satisfaction problems, and constraint based scheduling

- Lectures 3 & 4  **Basics**
  - Modelling scheduling problems in CP, basic techniques

- Lectures 5 & 6  **Constraint Propagation**
  - Constraint propagation and its application in constraint based scheduling

- Lectures 7 & 8  **Advanced techniques**
  - Consistency checks, search strategies, ordering heuristics
Course Schedule

- Lectures 9 & 10  
  Research in CBS  
  - Search in constraint based scheduling, current research

- Lectures 11 & 12  
  Modeling & COP  
  - Modelling AI puzzles, demos, and constraint optimization problems

- Course web page
  - [http://www.cs.nott.ac.uk/~rxq/xjtu.htm](http://www.cs.nott.ac.uk/~rxq/xjtu.htm)
  - Lecture slides
  - Textbooks
  - Extra reading materials
Constraint Logic Programming

Logic Programming

+ Constraint Solving

= Constraint (Logic) Programming

CLP or CP

These ideas were unified under a common conceptual and practical framework, constraint programming.

Constraint programming is not restricted to CLP.
Logic Programming

- Procedural: Sequences of instructions
  - how problem is solved and what the solution is
  - languages: C; Pascal; C++

- Declarative: State the logic of the problem
  - interested in only what the solution is
  - languages: Logic (Prolog); CP

- PROgramming in LOGic: PROLOG
  - brings logic into computer programs
  - express specifications for *problem solving* in formal logic:
    - relations + logical variables
  - problems expressed in terms of high level descriptions;
  - emphasis on "what is true", "what needs to be done" rather than "how to do"
Constraint Solving

- Arose from AI & computer graphics in 1960s, 1970s

- Constraint satisfaction
  - Solve problems by exploring constraints which must be satisfied by the solution
  - Set of variables taking values within domains satisfying constraints
Constraint Logic Programming

- 1987 **CLP** Jaar & Lassez
  - Realised that logic programming is a type of constraint programming
- 1990 **CHIP** (Constraint Handling in Prolog)
  - First commercial tool
- 1990 **Prolog III** Colmerauer
- 1994 **ILOG Solver**
  - C++ library
  - Millions dollars revenue in 2005

```plaintext
enum Letters {S, E, N, D, M, O, R, Y};
var int l[Letters] in 0..9;
solve {
  alldifferent(l) onDomain;
  // l[S] <> 0;
  // l[H] <> 0;
  1000*l[S] + 100*l[E] + 10*l[N] + l[O]
  + 1000*l[H] + 100*l[R] + l[E]
  = 10000*l[I] + 1000*l[O] + 100*l[I][N] + 10*l[E] + l[Y];
};
```

Send + More = Money
Constraint Programming

“... the user states the problem, the computer solves it.”

Eugene C. Freuder

- **User states the problem**
  - N-queen problem, Sudoku, Cryptarithmetic
  - Graph coloring, Scheduling, Timetabling, Knapsack
  - Real world examples:
    - Lufthansa (Daysy): personnel planning after air traffic disturbances
    - Nokia: automatic configuration of mobile phone software
    - Renault: production planning
Constraint Programming

"... the user states the problem, the computer solves it."

Eugene C. Freuder

- **Computer** uses pre-defined algorithms to *solve the problem*
  - Understand how algorithms work
  - Aim of this course!
  - Improve the search efficiency
    - Constraint propagation to assist decision making
    - Efficient search techniques, provided by constraint library or language
Textbooks

- Foundations of Constraint Satisfaction (*Tsang*), 1993
  - Online http://www.bracil.net/edward/FCS.html
  - Well written & easy to read

- Constraint Processing (*Dechter*), 2003
  - Online slides http://www.ics.uci.edu/~dechter/books/
  - Good for both beginners and advanced readers
References/Resources


- Eclipse Prolog http://eclipse.crosscoreop.com/

- SICStus prolog http://www.sics.se/isl/sicstuswww/site/bibliography.html