Advanced Process Systems Engineering
Lecture on Constraint Programming

Sylvain Mouret
Instructor: Ignacio E. Grossmann
Department of Chemical Engineering

March 2, 2009
http://www.andrew.cmu.edu/user/smouret/CP-Slides.pdf

Introduction to Constraint Programming

- Constraint Programming (CP) is a logic-based approach to optimization that makes use of implicit functions and logic inference
  $\Rightarrow$ Van Hentenryck (1989), Puget (1994)
- Commercial solvers:
  - Ilog CP
  - Dash Xpress-Kalis
- Free solvers:
  - Choco: http://choco-solver.net/
  - Gecode: http://www.gecode.org/

CP basic elements

Model

- No canonical form
- Constraint Satisfaction Problem (CSP)
$$\text{Model} = \{\text{Variables}, \text{Constraints}\}$$
- Constraint Optimization Problem (COP)
$$\text{Model} = \{\text{Variables}, \text{Constraints}, \text{Objective}\}$$

Variables with domains

- Continuous variables
$$z \in [l_1, u_1] \cup \ldots \cup [l_n, u_n], l_i, u_i \in \mathbb{R}$$
- Boolean variables $y \in \{\texttt{true}, \texttt{false}\}$
- Integer variables $x \in \{x_1, \ldots, x_n\}, x_i \in \mathbb{Z}$

CP basic elements

Constraints

- Algebraic constraints $h(x) \leq 0$
- Disjunctions $[A_1 x \leq b_1] \vee [A_2 x \leq b_2]$
- Conditional constraints $x = y \Rightarrow zy \leq w$
- Global constraints
  - $\texttt{alldifferent}(x_1, \ldots, x_n)$ — x_i integer
  - $\texttt{cumulative}((t_1, \ldots, t_n), (d_1, \ldots, d_n), (c_1, \ldots, c_n), C)$

    t_i start time of job i

    d_i duration of job i

    c_i capacity required by job i

    C total capacity limit
- Meta-constraints $(x \neq y) + (y \neq z) + (x \neq z) \geq 2$

Example of CSP model

- Find x, y, z
  - Continuous: $x \geq -5, y \geq 0$
  - Integer: $z = 0, 1, 2, 3, 4$
- Subject to

$$
\begin{aligned}
&(1) \quad x^3 + 10x = y^x - 2^z \\
&(2) \quad zx + 7.7y = 2.4 \\
&(3) \quad (z-1)^{y+1} \leq 10 \\
&(4) \quad \{[\ln(y + 2x + 12) \leq z + 5] \vee [y \geq z^2]\} \Rightarrow \\
&\qquad\; \{x \leq 0 \wedge y \leq 1\} \\
&(5) \quad x \leq 0 \Rightarrow z > 3
\end{aligned}
$$

- Solution
  - $x = -1.28505$
  - $y = 0.979$
  - $z = 4$

Reformulation as GDP

(4) $\{[\ln(y+2x+12) \leq z+5] \vee [y \geq z^2]\} \Rightarrow \{x \leq 0 \wedge y \leq 1\}$

(5) $x \leq 0 \Rightarrow z > 3$

<u>Note</u>: the CP model can be reformulated as a GDP

(4) $A \vee B \Rightarrow C \wedge D$ $\qquad (\neg A \wedge \neg B) \vee (C \wedge D)$

(5) $E \Rightarrow F$ $\qquad \neg E \vee F$

<u>GDP</u>: minimize 0 subject to

(1) $x^3 + 10x = y^x - 2^z$

(2) $zx + 7.7y = 2.4$

(3) $(z-1)^{y+1} \leq 10$

(4) $$\begin{bmatrix} Y_1 \\ \ln(y+2x+12) - z - 5 \geq \varepsilon \\ y - z^2 \geq \varepsilon \end{bmatrix} \vee \begin{bmatrix} Y_2 \\ x \leq 0 \\ y \leq 1 \end{bmatrix}$$

(5) $$\begin{bmatrix} W_1 \\ x \geq \varepsilon \end{bmatrix} \vee \begin{bmatrix} W_2 \\ z \geq 4 \end{bmatrix}$$

Traveling salesman problem

Let integer variables y_k represent the city visited in k^{th} position of tour, e.g. $y_1 = 3$ if city 3 is visited first.
Let c_{ij} be the cost from city i to j.
TSP:

- Select $y_k, k = 1 \ldots n$
- Minimize $\sum_{k=1}^{n} c_{y_k, y_{k+1}}$ (variable indices)
- Subject to $\texttt{alldifferent}(y_k, k = 1 \ldots n)$

Search in CP

Search in CP

- Tree search - implicit enumeration (no LP relaxation)
- Domain reduction at each node

Elements of search

- Depth first search
- Lower bound: partial solutions / open nodes
- Upper bound: best feasible solution

Domain reduction / Constraint propagation

- No general purpose algorithms
- Specific algorithms for classes of constraints

$\Rightarrow$ remove "inconsistent" values from each variable domain

Filtering algorithm for linear constraints

$$L \leq \sum_j c_j x_j \leq U \qquad\qquad c_j > 0, x_j \in [l_j, u_j]$$

Lower/Upper bound reasoning for x_i:

$$\begin{aligned}\sum_j c_j x_j \geq L &\Rightarrow x_i \geq L/c_i - \sum_{j \neq i} c_j x_j / c_i \\ &\Rightarrow x_i \geq L/c_i - \sum_{j \neq i} c_j u_j / c_i\end{aligned}$$

$$\begin{aligned}\sum_j c_j x_j \leq U &\Rightarrow x_i \leq U/c_i - \sum_{j \neq i} c_j x_j / c_i \\ &\Rightarrow x_i \leq U/c_i - \sum_{j \neq i} c_j l_j / c_i\end{aligned}$$

Lower bound $l_i := \max\{l_i, L/c_i - \sum_{j \neq i} c_j u_j / c_i\}$
Upper bound $u_i := \min\{u_i, U/c_i - \sum_{j \neq i} c_j l_j / c_i\}$

Filtering algorithm for linear constraints

Example:

$$2x + 3y = 11 \qquad x \geq 0, y \geq 0$$

a) Continuous x, y $\qquad L = U = 11$

$$x = \frac{11 - 3y}{2} \Rightarrow x^U = \frac{11 - 3y^L}{2} = 5.5$$

$$y = \frac{11 - 2x}{3} \Rightarrow y^U = \frac{11 - 2x^L}{3} = 3.666$$

$\Rightarrow x \in [0, 5.5], y \in [0, 3.666]$

b) Integer $x, y \Rightarrow x \in [0, 5], y \in [0, 3]$

Note: Filtering is more difficult for nonlinear constraints

Job-shop scheduling

Edge finding algorithm for disjunctive scheduling:

- Time decisions are discretized

Job i: start time st_i, duration d_i (fixed).
Job j: start time st_j, duration d_j (fixed).

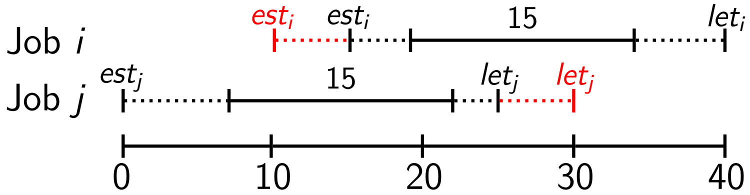
Disjunction: $\underset{i \text{ before } j}{[st_i + d_i \leq st_j]} \bigvee \underset{j \text{ before } i}{[st_j + d_j \leq st_i]}$

"Edge finding conditions", Applegate and Cook (1991).
Let est = earliest start time, let = latest end time.

$$est_i + d_i + d_j > let_j \Rightarrow j \text{ before } i$$

Job-shop scheduling

Example:



$$
\begin{aligned}
10 + 15 + 15 > 30 &\Rightarrow est_i + d_i + d_j > let_j \\
&\Rightarrow j \text{ before } i \\
&\Rightarrow st_j + d_j \leq st_i \\
&\Rightarrow \begin{cases} est_i & := 15 \\ let_j & := 25 \end{cases}
\end{aligned}
$$

Tree search

- Implicit enumeration (depth first search) with domain reduction / constraint propagation at each node
- Constraint propagation is used to iteratively execute the filtering procedure for each constraint in the model

Example:

- Find integers x, y, z
- $x \in \{1, 2, 3\}, y \in \{1, 2, 3\}, z \in \{1, 2, 3\}$
- Subject to

(1) $y < z$
(2) $x - y = 1$
(3) $x \neq z$

Summary

- Compact syntax for modeling
- Primarily aimed at feasibility / discrete problems
- Optimization is performed by keeping track of objective bounds, or using dichotomic search
- Online guide to CP by R. Barták:
  http://kti.mff.cuni.cz/~bartak/constraints/index.html
- Slides:
  http://www.andrew.cmu.edu/user/smouret/CP-Slides.pdf