

DM841
Discrete Optimization

Part I
Lecture 8
Global constraints

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

1. Global Constraints

1. Global Constraints

Global Constraint: among and sequence Global Constraints

among

Let x_1, \dots, x_n be a tuple of variables, S a set of variables, and l and u two nonnegative integers

`among` ($[x_1, \dots, x_n], S, l, u$)

At least l and at most u of variables take values in S .

In Gecode: `count`

sequence

Let x_1, \dots, x_n be a tuple of variables, S a set of variables, and l and u two nonnegative integers, s a positive integer.

`sequence` ($[x_1, \dots, x_n], S, l, u, s$)

At least l and at most u of variables take values from S in s consecutive variables

Car Sequencing Problem

Car Sequencing Problem

- ▶ an assembly line makes 50 cars a day
- ▶ 4 types of cars
- ▶ each car type is defined by options: {air conditioning, sun roof}

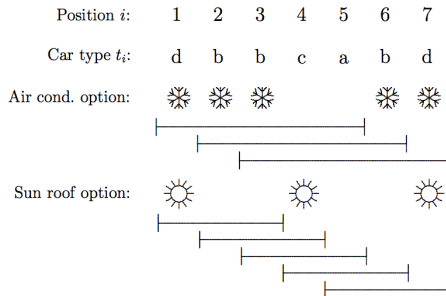
type	air cond.	sun roof	demand
a	no	no	20
b	yes	no	15
c	no	yes	8
d	yes	yes	7

- ▶ at most 3 cars in any sequence of 5 can be given air conditioning
- ▶ at most 1 in any sequence of 3 can be given a sun roof

Task: sequence the car types so as to meet demands while observing capacity constraints of the assembly line.

Car Sequencing Problem

Sequence constraints



Car Sequencing Problem: CP model

Car Sequencing Problem

Let t_i be the decision variable that indicates the type of car to assign to each position i in the sequence.

$\text{cardinality}([t_1, \dots, t_{50}], (a, b, c, d), (20, 15, 8, 7), (20, 15, 8, 7))$

$\text{among}([t_i, \dots, t_{i+4}], \{b, d\}, 0, 3), \quad \forall i = 1..46$

$\text{among}([t_i, \dots, t_{i+2}], \{c, d\}, 0, 1), \quad \forall i = 1..48$

$t_i \in \{a, b, c, d\}, i = 1, \dots, 50.$

Note: in Gecode among is count.

However, we can use sequence for the two among constraints above:

$\text{sequence}([t_1, \dots, t_{50}], \{b, d\}, 0, 3, 5),$

$\text{sequence}([t_1, \dots, t_{50}], \{c, d\}, 0, 1, 3),$

Car Sequencing Problem: MIP model

$$\left(\begin{matrix} AC_i = 0 \\ SR_i = 0 \end{matrix} \right) \vee \left(\begin{matrix} AC_i = 1 \\ SR_i = 0 \end{matrix} \right) \vee \left(\begin{matrix} AC_i = 0 \\ SR_i = 1 \end{matrix} \right) \vee \left(\begin{matrix} AC_i = 1 \\ SR_i = 1 \end{matrix} \right)$$

$$AC_i = AC_i^a + AC_i^b + AC_i^c + AC_i^d$$

$$SR_i = SR_i^a + SR_i^b + SR_i^c + SR_i^d$$

$$AC_i^a = 0, \quad AC_i^b = \delta_{ib}, \quad AC_i^c = 0, \quad AC_i^d = \delta_{id}$$

$$SR_i^a = 0, \quad SR_i^b = 0, \quad SR_i^c = \delta_{ic}, \quad SR_i^d = \delta_{id}$$

$$\delta_{ia} + \delta_{ib} + \delta_{ic} + \delta_{id} = 1$$

$$\delta_{ij} \in \{0, 1\}, \quad j = a, b, c, d$$

$$AC_i = \delta_{ib} + \delta_{id}, \quad SR_i = \delta_{ic} + \delta_{id}, \quad i = 1, \dots, 50$$

$$\delta_{ib} + \delta_{ic} + \delta_{id} \leq 1, \quad i = 1, \dots, 50$$

$$\delta_{ij} \in \{0, 1\}, \quad j = b, c, d, \quad i = 1, \dots, 50$$

$$\sum_{i=1}^{50} \delta_{ia} = 20, \quad \sum_{i=1}^{50} \delta_{ib} = 15, \quad \sum_{i=1}^{50} \delta_{ic} = 8, \quad \sum_{i=1}^{50} \delta_{id} = 7, \quad i = 1, \dots, 50$$

$$\sum_{j=i}^{i+4} AC_j \leq 3, \quad i = 1, \dots, 46$$

$$\sum_{j=j}^{i+2} SR_j \leq 1, \quad j = 1, \dots, 48$$

Global Constraint: nvalues

nvalues

Let x_1, \dots, x_n be a tuple of variables, and l and u two nonnegative integers

`nvalues`($[x_1, \dots, x_n], l, u$)

At least l and at most u different values among the variables

↪ generalization of alldifferent

In Gecode: `nvalues`

Global Constraint: stretch

stretch (In Gecode: via regular and extensional)

Let x_1, \dots, x_n be a tuple of variables with finite domains,

v an m -tuple of possible values of the variables,

l an m -tuple of lower bounds and u an m -tuple of upper bounds.

A **stretch** is a maximal sequence of consecutive variables that take the same value, i.e., x_j, \dots, x_k for v if $x_j = \dots = x_k = v$ and $x_{j-1} \neq v$ (or $j = 1$) and $x_{k+1} \neq v$ (or $k = n$).

stretch($[x_1, \dots, x_n], v, l, u$) **stretch-cycle**($[x_1, \dots, x_n], v, l, u$)

for each $j \in \{1, \dots, m\}$ any stretch of value v_j in x have length at least l_j and at most u_j .

In addition:

stretch($[x_1, \dots, x_n], v, l, u, P$)

with P set of patterns, i.e., pairs $(v_j, v_{j'})$. It imposes that a stretch of values v_j must be followed by a stretch of value $v_{j'}$.

Global Constraint: element

“element” constraint

Let y be an integer variable,

z a variable with finite domain,

and c an array of constants, i.e., $c = [c_1, c_2, \dots, c_n]$.

The element constraint states that z is equal to the y -th variable in c , or

$z = c_y$.

More formally:

$$\text{element}(y, z, [c_1, \dots, c_n]) = \{(e, f) \mid e \in D(y), f \in D(z), f = c_e\}.$$

```
IntArgs c(5, 1,4,9,16,25);  
element(home, c, x, y);
```

Global Constraint: channel

“channel” constraint

Let y be array of integer variables, and x be an array of integer variables:

$$\text{channel}([y_1, \dots, y_n], [x_1, \dots, x_n]) = \\ \{([e_1, \dots, e_n], [d_1, \dots, d_n]) \mid e_i \in D(y_i), \forall i, d_j \in D(x_j), \forall j, e_i = j \wedge d_j = i\}.$$

“channel” constraint

Let x be an array of boolean variables and y be an integer variable:

$$\text{channel}([x_1, \dots, x_n], y) = \\ \{([e_1, \dots, e_n], d) \mid e_i \in \{0, 1\}, \forall i, d \in D(y), \forall j, e_i = 1 \iff d = i\}.$$

Employee Scheduling problem

Four nurses are to be assigned to eight-hour shifts.

Shift 1 is the daytime shift, while shifts 2 and 3 occur at night.

The schedule repeats itself every week. In addition,

1. Every shift is assigned exactly one nurse.
2. Each nurse works at most one shift a day.
3. Each nurse works at least five days a week.
4. To ensure a certain amount of continuity, no shift can be staffed by more than two different nurses in a week.
5. To avoid excessive disruption of sleep patterns, a nurse cannot work different shifts on two consecutive days.
6. Also, a nurse who works shift 2 or 3 must do so at least two days in a row.

Employee Scheduling problem

Feasible Solutions

Solution viewed as assigning workers to shifts.

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
Shift1	A	B	A	A	A	A	A
Shift2	C	C	C	B	B	B	B
Shift3	D	D	D	D	C	C	D

Solution viewed as assigning shifts to workers.

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
Worker A	1	0	1	1	1	1	1
Worker B	0	1	0	2	2	2	2
Worker C	2	2	2	0	3	3	0
Worker D	3	3	3	3	0	0	3

Employee Scheduling problem

Feasible Solutions

Let w_{sd} be the nurse assigned to shift s on day d , where the domain of w_{sd} is the set of nurses $\{A, B, C, D\}$.

Let t_{id} be the shift assigned to nurse i on day d , and where shift 0 denotes a day off.

1. $\text{alldiff}(w_{1d}, w_{2d}, w_{3d}), d = 1, \dots, 7$
2. $\text{cardinality}(W, (A, B, C, D), (5, 5, 5, 5), (6, 6, 6, 6))$
3. $\text{nvalues}(\{w_{s1}, \dots, w_{s7}\}, 1, 2), s = 1, 2, 3$
4. $\text{alldiff}(t_{Ad}, t_{Bd}, t_{Cd}, t_{Dd}), d = 1, \dots, 7$
5. $\text{cardinality}(\{t_{i1}, \dots, t_{i7}\}, 0, 1, 2), i = A, B, C, D$
6. $\text{stretch-cycle}((t_{i1}, \dots, t_{i7}), (2, 3), (2, 2), (6, 6), P), i = A, B, C, D$
7. $w_{t_{id}d} = i, \forall i, d, \quad t_{w_{sd}s} = s, \forall s, d$

CP Modeling Guidelines [Hooker, 2011]

1. A **specially-structured subset of constraints** should be replaced by a single **global constraint** that **captures the structure**, when a suitable one exists. This produces a more succinct model and can allow more effective filtering and propagation.
2. A global constraint should be replaced by a **more specific** one when possible, to exploit more effectively the **special structure** of the constraints.
3. The addition of **redundant constraints** (i.e., constraints that are implied by the other constraints) can improve propagation.
4. When two alternate formulations of a problem are available, **including both** (or parts of both) in the model may improve propagation. Different variables are linked through the use of **channeling** constraints.

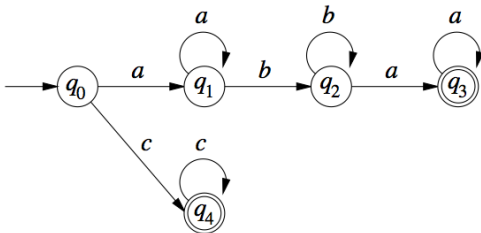
Global Constraint: regular

“regular” constraint

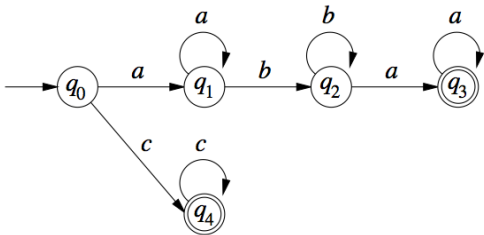
Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA and let $X = \{x_1, x_2, \dots, x_n\}$ be a set of variables with $D(x_i) \subseteq \Sigma$ for $1 \leq i \leq n$. Then

$\text{regular}(X, M) =$

$\{(d_1, \dots, d_n) \mid \forall i, d_i \in D(x_i), [d_1, d_2, \dots, d_n] \in L(M)\}$.



Global Constraint: regular



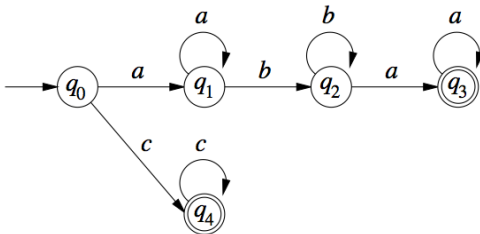
Example

Given the problem

$$x_1 \in \{a, b, c\}, \quad x_2 \in \{a, b, c\}, \quad x_3 \in \{a, b, c\}, \quad x_4 \in \{a, b, c\},$$

$$\text{regular}([x_1, x_2, x_3, x_4], M).$$

One solution to this CSP is $x_1 = a, x_2 = b, x_3 = a, x_4 = a$.



In Gecode:

```

DFA::Transition t[] = {{0, 0(a), 1}, {1, 0(a), 1}, {1, 1(b), 2}, {2, 1(b), 2},
                      {2, 0(a), 3}, {3, 0(a), 3}, {3, 0, -1},
                      {0, 2(c), 4}, {4, 2(c), 4}, {4, 0, -1}};
int f[] = {3,4,-1}; // vector of final states
DFA d(0, t, f);
BoolVarArray x(home, 4, 0(a), 3(d));
extensional(home, x, d);

```

```

REG r=(REG(0) + *REG(0) + REG(1) + *REG(1) + REG(0) + *REG(0)) | REG(2) + (*REG(2)));
DFA d(r);
extensional(home, x, d);

```

Assignment problems

The assignment problem is to find a minimum cost assignment of m tasks to n workers ($m \leq n$).

Each task is assigned to a different worker, and no two workers are assigned the same task.

If assigning worker i to task j incurs cost c_{ij} , the problem is simply stated:

$$\begin{aligned} \min \quad & \sum_{i=1, \dots, n} c_{ix_i} \\ & \text{alldiff}([x_1, \dots, x_n]), \\ & x_i \in D_i, \forall i = 1, \dots, n. \end{aligned}$$

Note: cost depends on position. Recall: with $n = m$ min weighted bipartite matching (Hungarian method)

with supplies/demands transshipment problem

Circuit problems

Given a directed weighted graph $G = (N, A)$, find a circuit of min cost:

$$\begin{aligned} \min \quad & \sum_{i=1, \dots, n} c_{x_i x_{i+1}} \\ & \text{alldiff}([x_1, \dots, x_n]), \\ & x_i \in D_i, \forall i = 1, \dots, n. \end{aligned}$$

Note: cost depends on sequence.

An alternative formulation is

$$\begin{aligned} \min \quad & \sum_{i=1, \dots, n} c_{iy_i} \\ & \text{circuit}([y_1, \dots, y_n]), \\ & y_i \in D_i = \{j \mid (i, j) \in A\}, \forall i = 1, \dots, n. \end{aligned}$$

Global Constraint: circuit

“circuit” constraint

Let $X = \{x_1, x_2, \dots, x_n\}$ be a set of variables with respective domains $D(x_i) \subseteq \{1, 2, \dots, n\}$ for $i = 1, 2, \dots, n$. Then

$$\text{circuit}(x_1, \dots, x_n) = \{(d_1, \dots, d_n) \mid \forall i, d_i \in D(x_i), d_1, \dots, d_n \text{ is cyclic}\}.$$

Circuit problems

A model with redundant constraints is as follows:

min z

$$z \geq \sum_{i=1, \dots, n} c_{x_i x_{i+1}}$$

$$z \geq \sum_{i=1, \dots, n} c_{iy_i}$$

`alldiff`($[x_1, \dots, x_n]$),

`circuit`($[y_1, \dots, y_n]$),

$$x_1 = y_{x_n} = 1, \quad x_{i+1} = y_{x_i}, \quad i = 1, \dots, n-1$$

$$x_i \in \{1, \dots, n\}, \quad \forall i = 1, \dots, n,$$

$$y_i \in D_i = \{j \mid (i, j) \in A\}, \quad \forall i = 1, \dots, n.$$

“disjunctive” scheduling

Let (s_1, \dots, s_n) be a tuple of (integer/real)-valued variables indicating the starting time of a job j . Let (p_1, \dots, p_n) be the processing times of each job.

$$\text{disjunctive}([s_1, \dots, s_n], [p_1, \dots, p_n]) = \\ \{[d_1, \dots, d_n] \mid \forall i, j, i \neq j (d_i + p_i \leq d_j) \vee (d_j + p_j \leq d_i)\}$$

In Gecode:

```
IntArgs c(4, 2, 7, 4, 11);  
unary(home, s, d);
```


Scheduling Constraints

cumulative for Resource Constrained Project Scheduling

[Aggoun and

Beldiceanu, 1993]

- ▶ r_j release time of job j
- ▶ p_j processing time
- ▶ d_j deadline
- ▶ c_j resource consumption
- ▶ C limit not to be exceeded at any point in time

Let s be an n -tuple of (integer/real) values denoting the starting time of each job

cumulative($[s_j], [p_j], [c_j], C$) :=

$$\{([d_j], [p_j], [c_j], C) \mid \forall t \sum_{i \mid d_i \leq t \leq d_i + p_i} c_i \leq C\}$$

With $c_j = 1$ for all j and $C = 1 \rightsquigarrow$ disjunctive

Others

- ▶ Sorted constraints ($\text{sorted}(x, y)$)
- ▶ Bin-packing constraints ($\text{binpacking}(l, b, s)$)
 l_j is the load variable of bin j , b_i the bin variable of item i , s_i size of item i
- ▶ Geometrical packing constraints (nooverlap)
 $\text{diffn}((x^1, \Delta x^1), \dots, (x^m, \Delta x^m))$ arranges a given set of multidimensional boxes in n -space such that they do not overlap (aka, nooverlap)
- ▶ Value precedence constraints ($\text{precede}(x, s, t)$)
- ▶ Logical implication: $\text{conditional}(\mathcal{D}, \mathcal{C})$ between sets of constraints
 $\mathcal{D} \Rightarrow \mathcal{C}$ (ite)

- ▶ Bin Packing

More (not in gecode)

- ▶ $\text{clique}(x|G, k)$ requires that a given graph contain a clique of size k
- ▶ $\text{cycle}(x|y)$ select edges such that they form exactly y directed cycles in a graph.
- ▶ $\text{cutset}(x|G, k)$ requires that for the set of selected vertices V' , the set $V \setminus V'$ induces a subgraph of G that contains no cycles.

Global Constraint Catalog

Corresponding author: **Nicolas Beldiceanu** nicolas.beldiceanu@emn.fr

Online version: **Sophie Demassey** sophie.demassey@emn.fr

 Web Catalog
 all formats html pdf

Global Constraint Catalog
html / 2009-12-16

Search by:

NAME	Keyword	Meta-keyword	Argument pattern	Graph description
		Bibliography	Index	

Keywords (ex: *Assignment, Bound consistency, Soft constraint,...*) can be searched by **Meta-keywords** (ex: *Application area, Filtering, Constraint type,...*)

About the catalogue

The catalogue presents a list of 348 global constraints issued from the literature in constraint programming and from popular constraint systems. The semantic of each constraint is given together with a description in terms of graph properties and/or automata.

- Hooker J.N. (2011). **Hybrid modeling**. In *Hybrid Optimization*, edited by P.M. Pardalos, P. van Hentenryck, and M. Milano, vol. 45 of **Optimization and Its Applications**, pp. 11–62. Springer New York.
- van Hoesel W. and Katriel I. (2006). **Global constraints**. In *Handbook of Constraint Programming*, chap. 6. Elsevier.