Constraint Programming
Global constraints

Hadrien Cambazard

(ROSP)

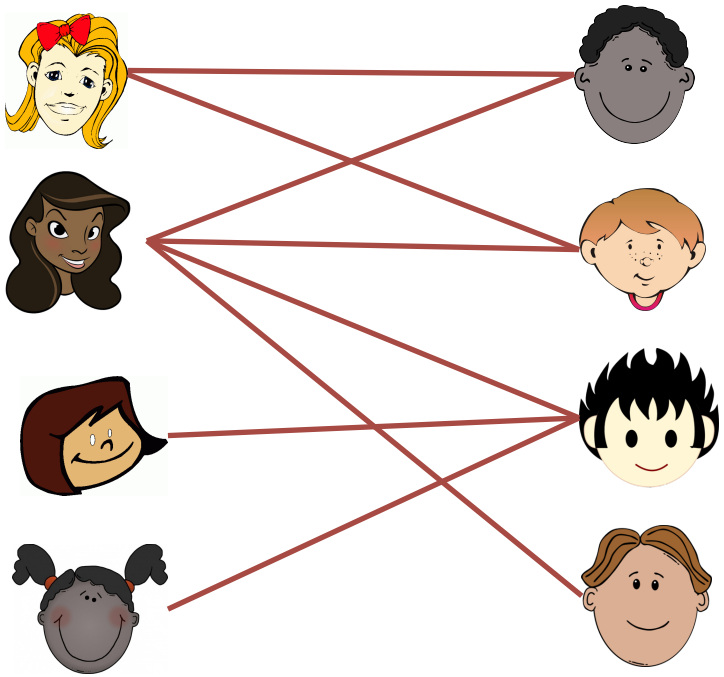
AllDifferent

- Hall's Marriage theorem (Hall, 1935):

  (formulated as "Marriage theorem" by H. Weyl, 49)

"If a group of men and women marry only if they have been introduced to each other previously, then a complete set of marriages is possible ***if and only if every subset of men*** *has collectively been introduced to* ***at least as many women****, and vice versa."*

Can they all get married ?



AllDifferent

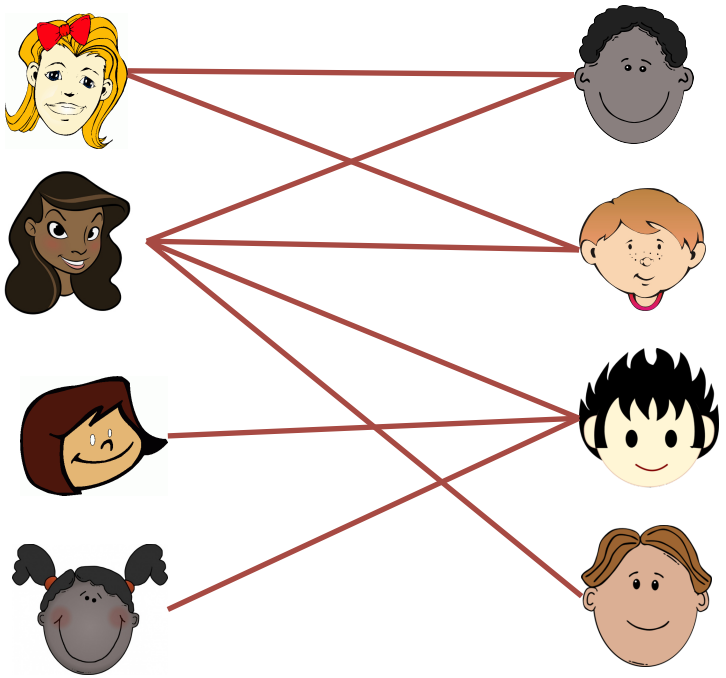
- Hall's Marriage theorem (Hall, 1935):

  (formulated as "Marriage theorem" by H. Weyl, 49)

*"If a group of men and women marry only if they have been introduced to each other previously, then a complete set of marriages is possible **if and only if every subset of men** has collectively been introduced to **at least as many women**, and vice versa."*

Can they all get married ?

Women = variables
Men = values



D( ) = { , }

AllDifferent(, , ,)

AllDifferent

Tradeoff between level of consistency and complexity

Generalized Arc consistency	Régin, 94	$\mathcal{O}(n^{\frac{3}{2}}d)$
Range Consistency	Leconte, 96	$\mathcal{O}(n^2)$
Bounds Consistency	Puget, 98	$\mathcal{O}(nlog(n))$
Bounds Consistency	Mehlhorn et al, 2000	$\mathcal{O}(n)$
Bounds Consistency	Quimper et al, 2003	$\mathcal{O}(n)$

- Applications for GAC: timetabling, combinatorial design, sometimes scheduling, …
- Most of the time BC is great (and much better than the decomposition)

Global cardinality

$$\mathrm{GCC}(X = [x_1, \ldots, x_n], [l_1, \ldots, l_m], [u_1, \ldots, u_m])$$

$$x_1 \in \{2\}, x_2 \in \{1,2\}, x_3 \in \{2,3\}, x_4 \in \{2,3\}, x_5 \in \{1,2,3,4\}, x_6 \in \{3,4\}$$

$$\mathrm{GCC}(X = \{x_1, x_2, x_3, x_4, x_5, x_6\}, [0,1,1,2], [3,2,1,3])$$

	1	2	3	4
l_j	0	1	1	2
u_j	3	2	1	3

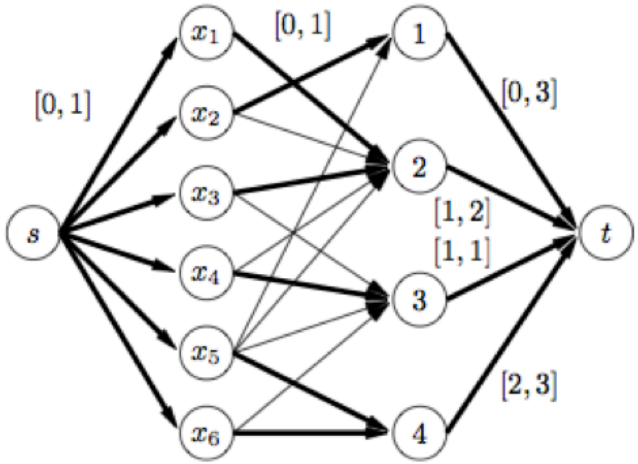
Global cardinality

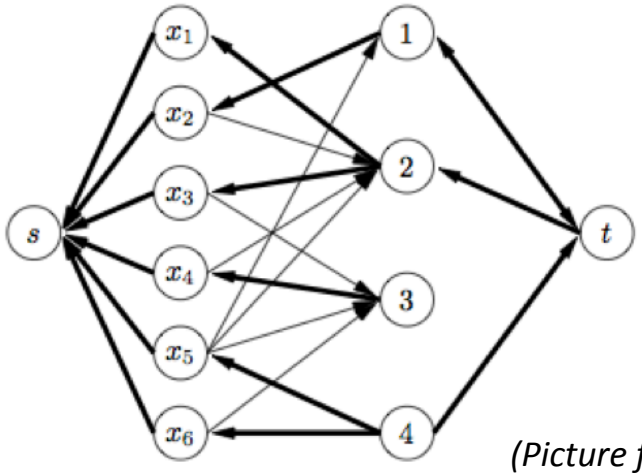
$$\mathrm{GCC}(X = [x_1, \ldots, x_n], [l_1, \ldots, l_m], [u_1, \ldots, u_m])$$

$$x_1 \in \{2\}, x_2 \in \{1,2\}, x_3 \in \{2,3\}, x_4 \in \{2,3\}, x_5 \in \{1,2,3,4\}, x_6 \in \{3,4\}$$

$$\mathrm{GCC}(X = \{x_1, x_2, x_3, x_4, x_5, x_6\}, [0,1,1,2], [3,2,1,3])$$

	1	2	3	4
l_j	0	1	1	2
u_j	3	2	1	3





(Picture from Claude-Guy Quimper's phd)

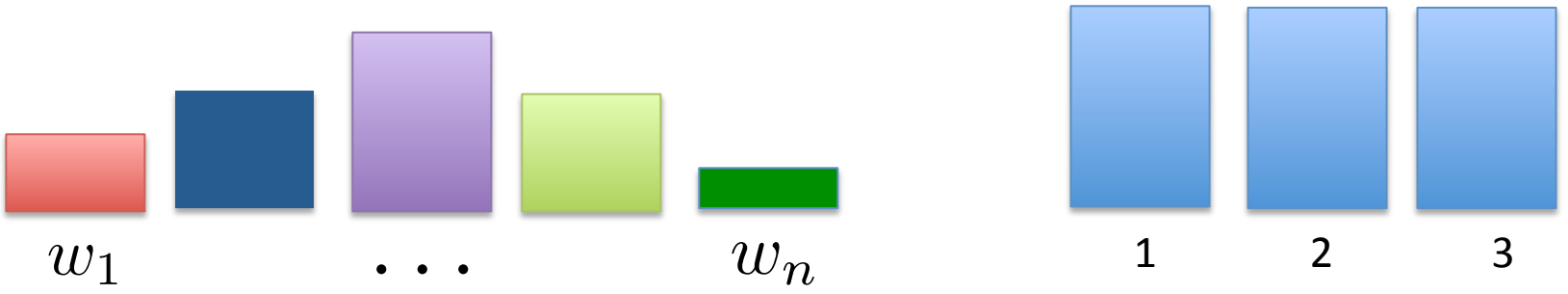
The value network

The residual graph

Bin-Packing

$$\textsc{BinPacking}([x_1, \ldots, x_n], [l_1, \ldots, l_m], [w_1, \ldots, w_n])$$

Variables:
x: items (domain is the possible bins)
l : the loads of the bins
Constants:
w: the size of the items



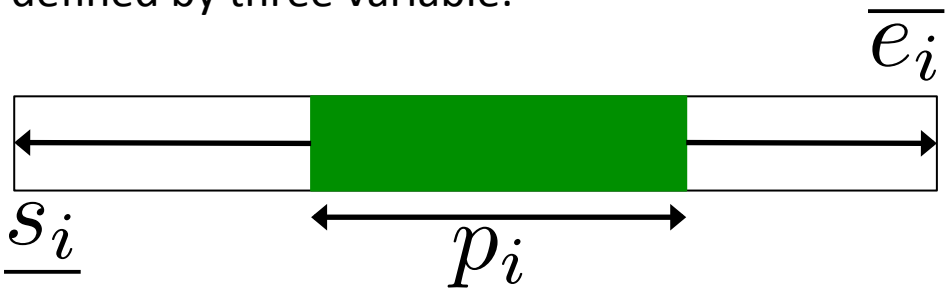
$$x_1 = 1, x_2 = 1, x_3 = 2, x_4 = 3, x_5 = 3$$

Disjunctive

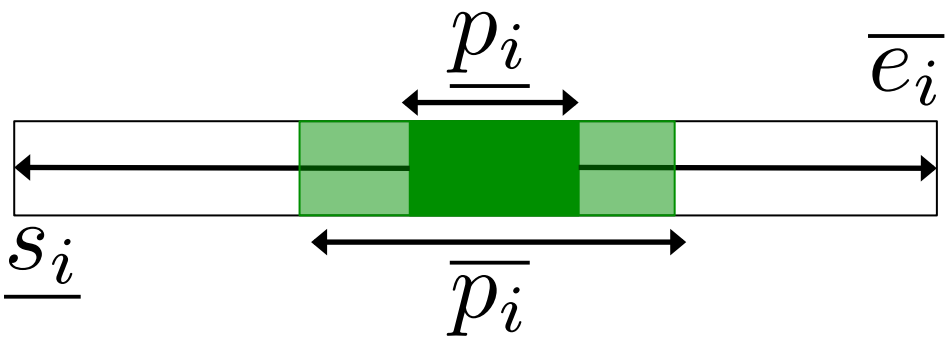
$$\text{DISJUNCTIVE}([s_1, \ldots, s_n], [e_1, \ldots, e_n], [p_1, \ldots, p_n])$$

$$s_i + p_i = e_i$$

Task i defined by three variable:



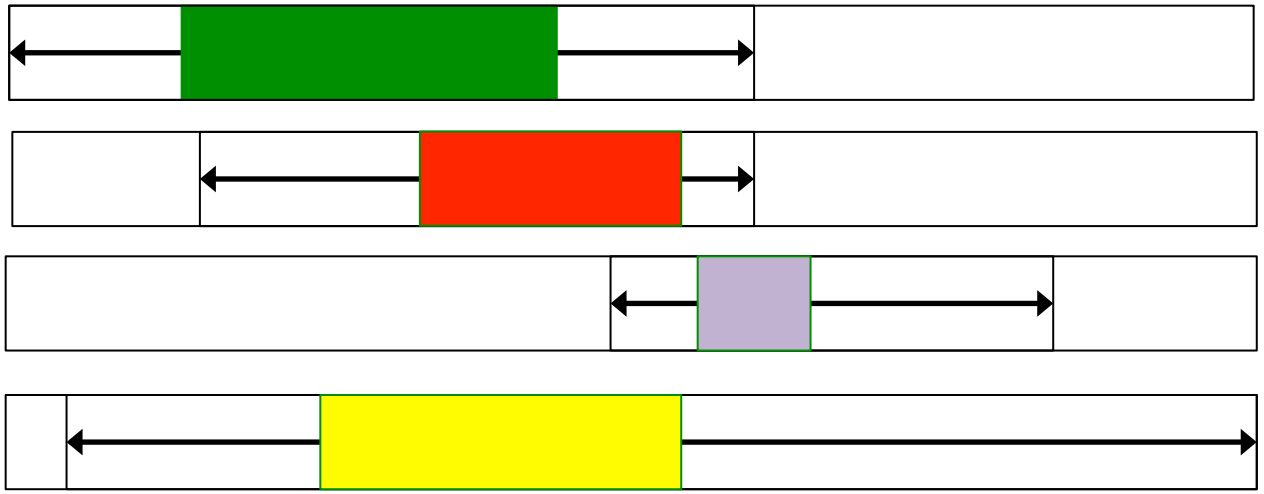
Constant duration



Variable duration

Disjunctive

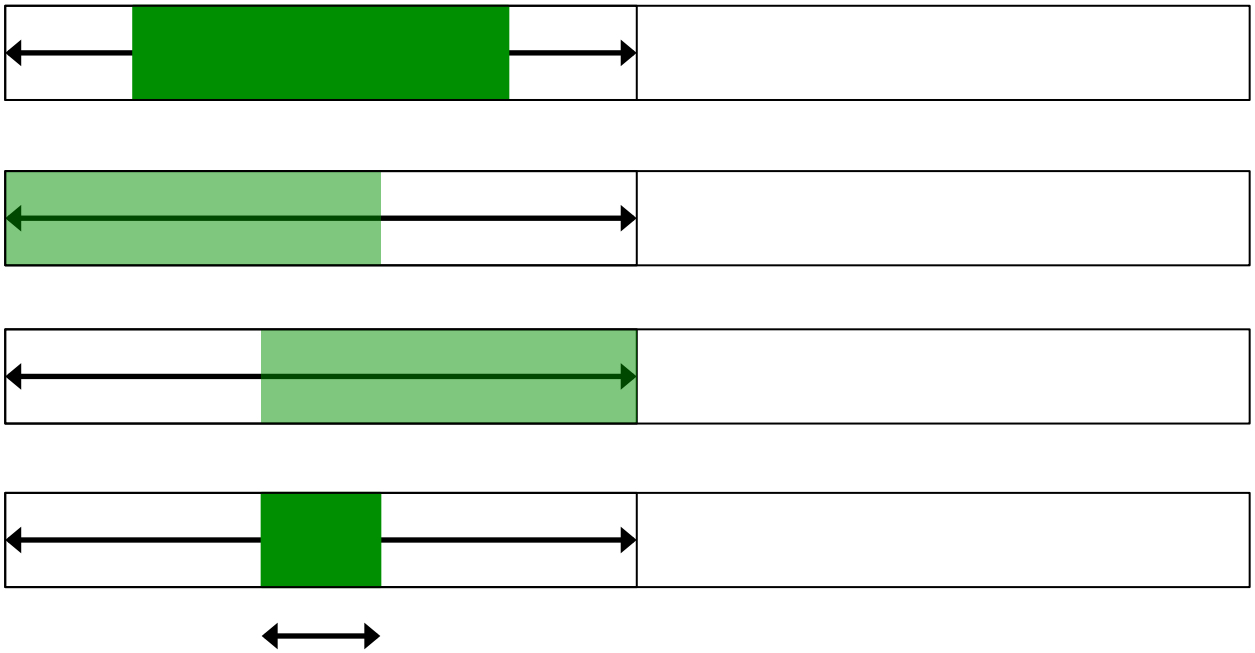
$$\textsc{Disjunctive}([s_1, \ldots, s_n], [e_1, \ldots, e_n], [p_1, \ldots, p_n])$$



The disjunctive make sure the tasks don't overlap in time

Disjunctive

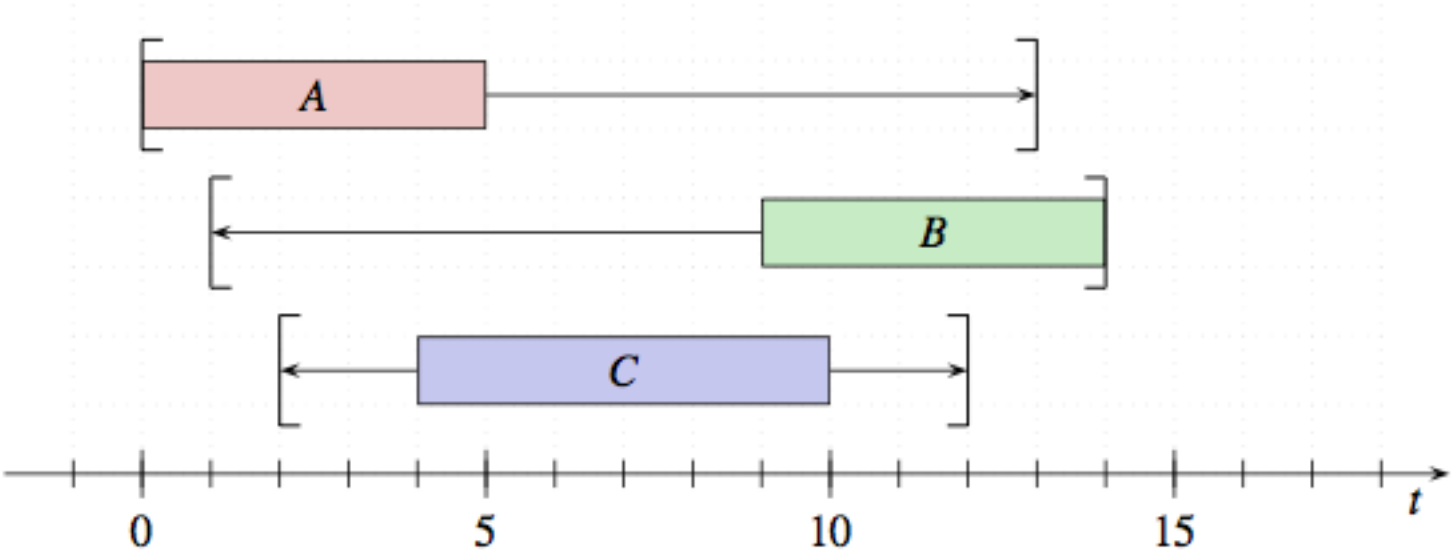
- Mandatory parts (filtering rule)



mandatory part : we can filter other tasks to make sure they will not overlap this interval

Disjunctive

- Overload checking (failure detection)

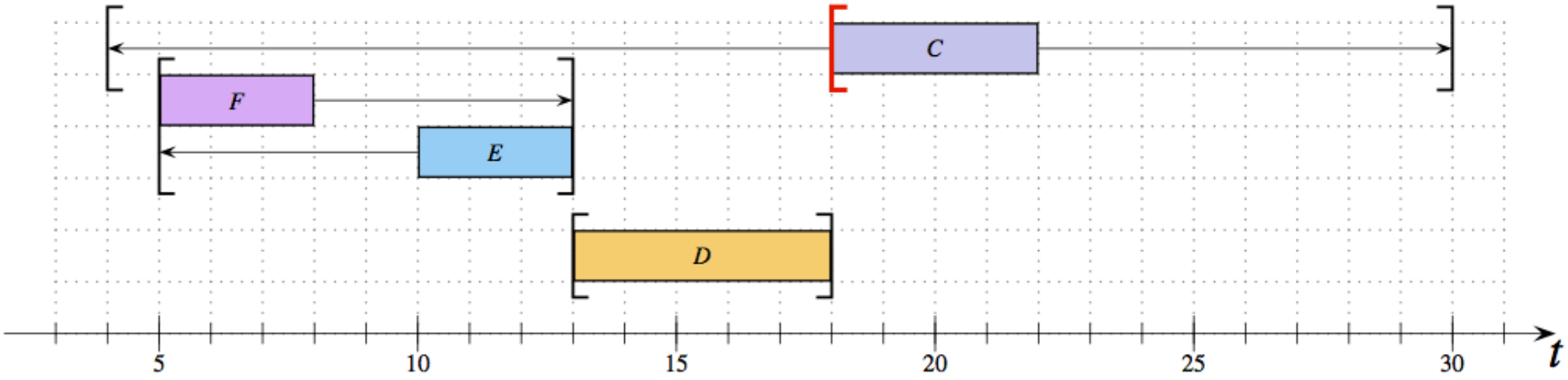


(Picture from **Petr's Villim** phd thesis)

The constraint fails (not enough space for all the three tasks)

Disjunctive

- Edge finding (filtering rule)



(Picture from **Petr's Villim** phd thesis)

The constraint infers that C should be after {F,E,D} and update the lower bound of the starting date of C.