Constraint Programming

Lecture 5. Symmetry. Real-life problems modeling

Ruslan Sadykov

INRIA Bordeaux—Sud-Ouest

3 February 2022

Lignes directrices

Lignes directrices

Symmetry in CSPs

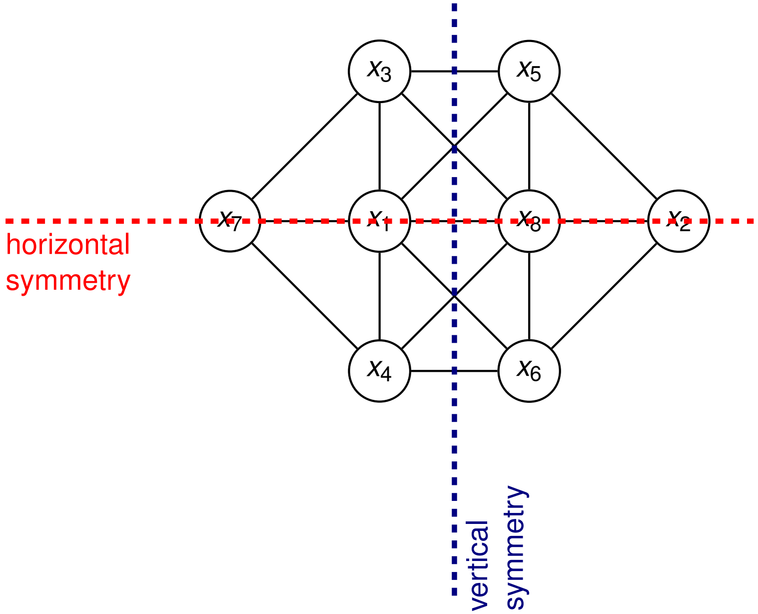
Variants of symmetry

- Variables are « interchangeable »
- Values are « interchangeable »
- Symmetry of pairs « variable-value »

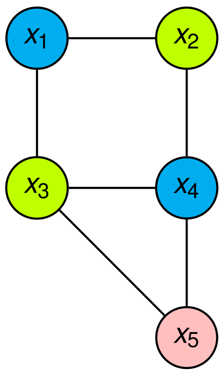
Symmetry consequences

- Enumeration (search) tree contains several equivalent sub-trees
- If one of such sub-trees does not contain a solution, the equivalent sub-trees does not contain it neither !
- If we do not recognise equivalent sub-trees, useless search will be performed

Symmetry of variables : an example



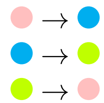
Symmetry of values : 3-colouring example



A solution

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

Mapping



Another solution

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

Symmetry of variables : 4-queens example

	x_1	x_2	x_3	x_4
1			Q	
2	Q			
3				Q
4		Q		

Symmetry of variables : 4-queens example

	x_1	x_2	x_3	x_4
1	Q			
2				
3				
4	Q			

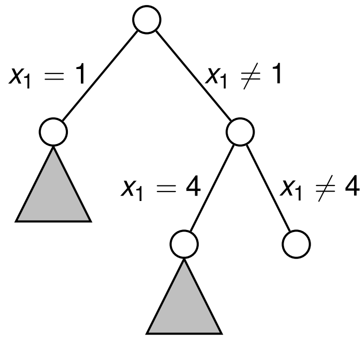
horizontal symmetry

Partial assignment

$x_1 = 1$

Symmetric assignement

$x_1 = 4$



Symmetry : formal definition

For a CSP $\langle \mathbf{X}, \mathbf{D}, \mathbf{C} \rangle$, with

- $\mathbf{X}$ set of variables $\{x_1, \ldots, x_n\}$
- $\mathbf{D}$ set of domaines $\{D_{x_1}, \ldots, D_{x_n}\}$
- Let $\mathcal{D}$ be the union of domains $\mathcal{D} = \bigcup_{x \in \mathbf{X}} D_x$ (set of values)

A symmetry of P

Is a permutation of the set $\mathbf{X} \times \mathcal{D}$ which preserves the set of solutions of P

Particular cases

- Variables symmetry $\sigma(x, v) = (\sigma'(x), v)$
- Values symmetry $\sigma(x, v) = (x, \sigma'(v))$

Variables-values symmetry : 4-queens example

	x_1	x_2	x_3	x_4
1	1	2	3	4
2	5	6	7	8
3	9	10	11	12
4	13	14	15	16

Identité

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

	x_1	x_2	x_3	x_4
1	13	9	5	1
2	14	10	6	2
3	15	11	7	3
4	16	12	8	4

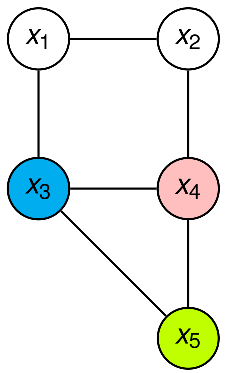
Rotation 90°

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
13	9	5	1	14	10	6	2	15	11	7	3	16	12	8	4

Symmetry elimination (decreasing)

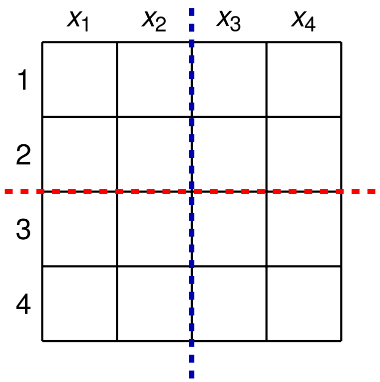
- Reformulate the model
  - Example : variables which take sets as values (packing)
- Add constraints to the model
  - At least one of symmetric solutions (assignments) should satisfy them
- Eliminating the symmetry during the search
  - Recognise and ignore dynamically the symmetric sub-trees during the search

Symmetry elimination : example I



We fix the colors of vertices which belong to a clique

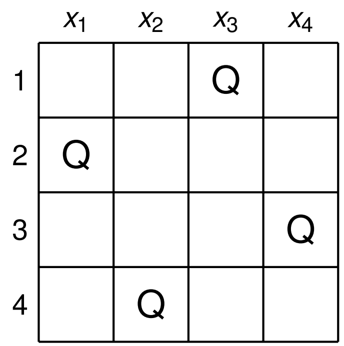
Symmetry elimination : example I



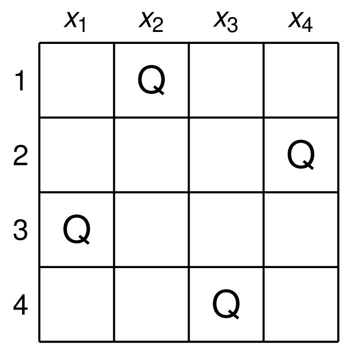
We eliminate the horizontal symmetry by adding the constraint $x_1 \leq 2$

We eliminate the vertical symmetry by adding the constraint $x_2 \leq x_3$

Elimination of several symmetries : a danger



By adding $x_2 \leq x_3$, we eliminate this solution



By adding $x_1 \leq 2$, we eliminate this solution

But there are only 2 solutions !

Lignes directrices

Symmetry

Problems modelling
- Frequency assignment
- Car sequencing
- Sports scheduling
- Timetabling
- « Job-shop »
- Cutting

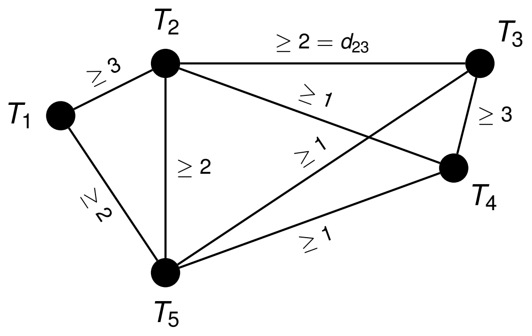
Lignes directrices

Symmetry

Problems modelling
- Frequency assignment
- Car sequencing
- Sports scheduling
- Timetabling
- « Job-shop »
- Cutting

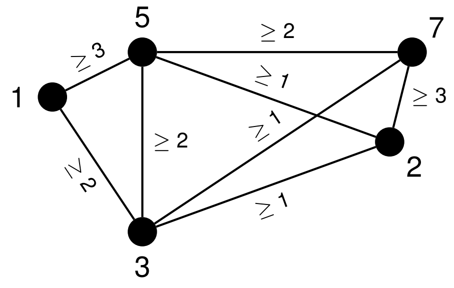
Frequency assignment : problem definition

- There are 5 transmitters and 7 possible transmission frequencies.
- We need to assign frequencies to transmitters so that parasites between nearby transmitters are avoided.
- All assigned frequencies should be different



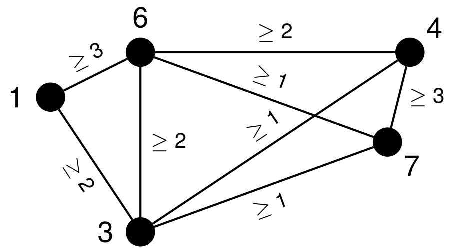
Frequency assignment : model

- Variables : F_i — frequency assigned to transmitter i.
- Domains : $D_{F_i} = \{1, \ldots, 7\}, \forall i$.
- Constraints :
  - $\mid F_i - F_j \mid \geq d_{ij}$
    ou $F_i - F_j \geq d_{ij} \vee F_i - F_j \leq -d_{ij}, \forall (i, j)$;
  - `all-different`$(F_1, \ldots, F_5)$.



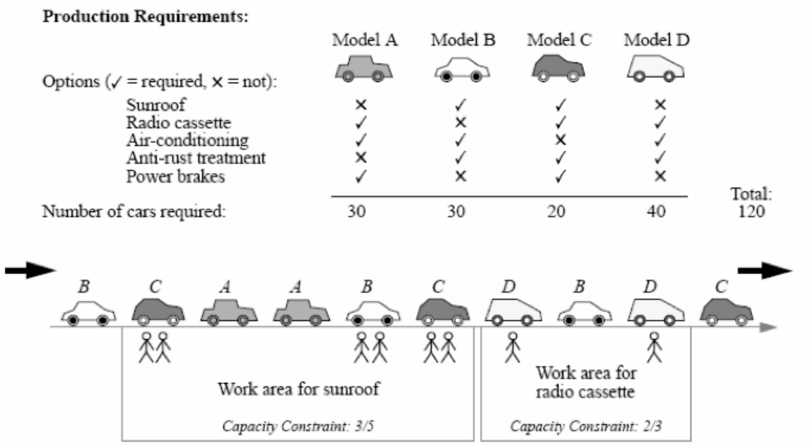
Frequency assignment : additional constraints

- There are low frequencies or VHF (1,2,3) and high frequencies or UHF (4,5,6,7).
- Exactly 2 low frequencies and 3 high frequencies should be assigned.
- Additional variables : $S_i = 0$ if low and 1 if high.
- Additional constraints :
  - `element`$(S_i, \{0, 0, 0, 1, 1, 1, 1\}, F_i), \forall i$.
  - `gcc`$(\{S_i\}_{\forall i}, \{0, 1\}, 2, 3, 2, 3)$.



Lignes directrices

Car sequencing : definition



Source : Alan M. Frisch

Car sequencing : model

- Data :
  - n options, m vehicle types.
  - d_i vehicles of type i should be produced, $1 \leq i \leq m$, $T = \sum_{i=1}^{m} d_i$.
  - $a_{ij} = 1$ if type i requires option j, otherwise $a_{ij} = 0$, $1 \leq i \leq m$, $1 \leq j \leq n$.
  - For each subsequence of q_j vehicles, option j can be installed on at most p_j, $1 \leq j \leq n$.
- Variables :
  - X_k — number of vehicle type in position k in the sequence, $1 \leq k \leq T$.
  - $O_{kj} = 1$ if the vehicle in position k requires option j, otherwise $O_{kj} = 0$, $1 \leq k \leq T$, $1 \leq j \leq n$.
- Domains :
  - $D_{X_k} = \{1, \ldots, m\}$, $\forall k$.
  - $D_{O_{kj}} = \{0, 1\}$, $\forall k, j$.

Car sequencing : constraints

- The demand for each vehicle type should be satisfied :

$$\texttt{gcc}\left(\{X_k\}_{\forall k}, \{1, \ldots, m\}, \{d_i\}_{\forall i}, \{d_i\}_{\forall i}\right).$$

- Link between variables X and O :

$$\texttt{element}\left(O_{kj}, \{a_{ij}\}_{\forall i}, X_k\right), \quad \forall k, j.$$

- Sequence constraints :

$$\sum_{k'=k}^{k+q_j} O_{k'j} \leq p_j, \quad \forall j, \quad 1 \leq k \leq T - q_j + 1.$$

Global sequencing constraint

The last two constrains can be replaces by the global sequencing constraint (Source : Puget et Régin) :

$$\texttt{gsc}(X_1, \ldots, X_n, \mathcal{V}, q, p)$$

This constraints requires that in each sub-sequence of X of size q the total number of taken values in $\mathcal{V}$ should be at most p.

For our problem :

$$\texttt{gsc}\left(\{X_k\}_{\forall k}, \{i\}_{a_{ij}=1}, q_j, p_j\right), \quad 1 \leq j \leq n.$$

Lignes directrices

Symmetry

Problems modelling

Frequency assignment

Car sequencing

Sports scheduling

Timetabling

« Job-shop »

Cutting

Sports scheduling : definition

- n teams, $n-1$ weeks, $\frac{n}{2}$ periods.
- Each pair of teams plays exactly one time.
- Each team plays one match per week.
- Each team plays at most two times in each period.

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
Period 1	0 vs 1	0 vs 2	4 vs 7	3 vs 6	3 vs 7	1 vs 5	2 vs 4
Period 2	2 vs 3	1 vs 7	0 vs 3	5 vs 7	1 vs 4	0 vs 6	5 vs 6
Period 3	4 vs 5	3 vs 5	1 vs 6	0 vs 4	2 vs 6	2 vs 7	0 vs 7
Period 4	6 vs 7	4 vs 6	2 vs 5	1 vs 2	0 vs 5	3 vs 4	1 vs 3

Source : Jean-Charles Régin

Sports scheduling : variables

- For each cell, 2 variables represent the playing teams :

$$T^h_{pw} \text{ et } T^a_{pw}, \quad p \in [1, \dots, \tfrac{n}{2}], w \in [1, \dots, n-1].$$
$$D(T^h_{pw}) = D(T^a_{pw}) = \{0, \dots, n-1\}, \quad T^h_{pw} < T^a_{pw}, \ \forall p, w.$$

- For each cell, one variable represents the match :

$$M_{pw}, \quad p \in [1, \dots, \tfrac{n}{2}], w \in [1, \dots, n-1].$$
$$D(M_{pw}) = \{1, \dots, \tfrac{n(n-1)}{2}\}, \quad M_{pw} = n \cdot T^h_{pw} + T^a_{pw}, \ \forall p, w.$$

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
Period 1	M11	M12	M13	M14	M15	M16	M17
Period 2	M21	M22	M23	M24	M25	M26	M27
Period 3	M31	M32	M33	M34	M35	M36	M37
Period 4	M41	M42	M43	M44	M45	M46	M47

Sports scheduling : constraints

- `all-different`$(\{M_{pw}\}_{1 \le p \le n/2, 1 \le w \le n-1})$;
- `all-different`$(\{T^h_{pw}, T^a_{pw}\}_{1 \le p \le n/2}), w \in [1, \dots, n]$;
- `gcc`$(\{T^h_{pw}, T^a_{pw}\}_{1 \le w \le n-1}, \{k, 2, 2\}_{0 \le k \le n-1}), p \in [1, \dots, \frac{n}{2}]$.
- implicit constraints ;
- symmetry (very important) : elimination of permutations.

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Dummy
Period 1	0 vs 1	0 vs 2	4 vs 7	3 vs 6	3 vs 7	1 vs 5	2 vs 4	5 vs 6
Period 2	2 vs 3	1 vs 7	0 vs 3	5 vs 7	1 vs 4	0 vs 6	5 vs 6	2 vs 4
Period 3	4 vs 5	3 vs 5	1 vs 6	0 vs 4	2 vs 6	2 vs 7	0 vs 7	1 vs 3
Period 4	6 vs 7	4 vs 6	2 vs 5	1 vs 2	0 vs 5	3 vs 4	1 vs 3	0 vs 7

Sports scheduling : results

Using Constraint Programming, we can find a scheduling for 40 teams in 6 hours — real-life size !

Today, scheduling for Major League Baseball (US) with hundrends of constraints is produced by Operations Research (MIP, CP, heuristics) Source : Michael A. Trick

Lignes directrices

Symmetry

Problems modelling

- Frequency assignment
- Car sequencing
- Sports scheduling
- Timetabling
- « Job-shop »
- Cutting

Timetabling : definition

- 4 employees, 7-days week.
- 3 periods of work each day :
  day (D, difficulty 1.0), evening (E, 0.8), night (N, 0.5).
- In each period, exactly one employee should be present ⇒ each day 3 employees work, and one has a day-off.
- The total difficulty should not exceed $\geq$ 3.0.

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
M. Green	J						
M. Blue	S						
M. Red	N						
M. Brown	–						

Source : Gilles Pesant

Timetabling : modeling

- Variables : Job_{ij}, $1 \leq i \leq 4$, $1 \leq j \leq 7$, $Charge_{ij}$, $1 \leq i \leq 4$, $1 \leq j \leq 7$.
- Domains : $D_{Job_{ij}} = \{\mathrm{D}, \mathrm{E}, \mathrm{N}, -\}$, $\forall i, j$.
- Constraints :
  - `all-different`($Job_{\cdot j}$), $\forall j$.
  - `element`($Charge_{ij}, \{1.0, 0.8, 0.5, 0\}, Job_{ij}$), $\forall i, j$.
  - $\sum_{j=1}^{7} Charge_{ij} \geq 3.0$, $\forall i$.

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
M. Green	D	–	D	–	D	–	D
M. Blue	–	N	N	N	N	N	N
M. Red	N	D	–	D	E	D	–
M. Brown	E	E	E	E	–	E	E

Timetabling : series length

- Additional constraint : the length of a series should be inside an interval.
- Modeling :
  $\mathtt{stretch}(Job_{i.}, \{2, 1, 1, 1\}, \{4, 4, 4, 7\})$.

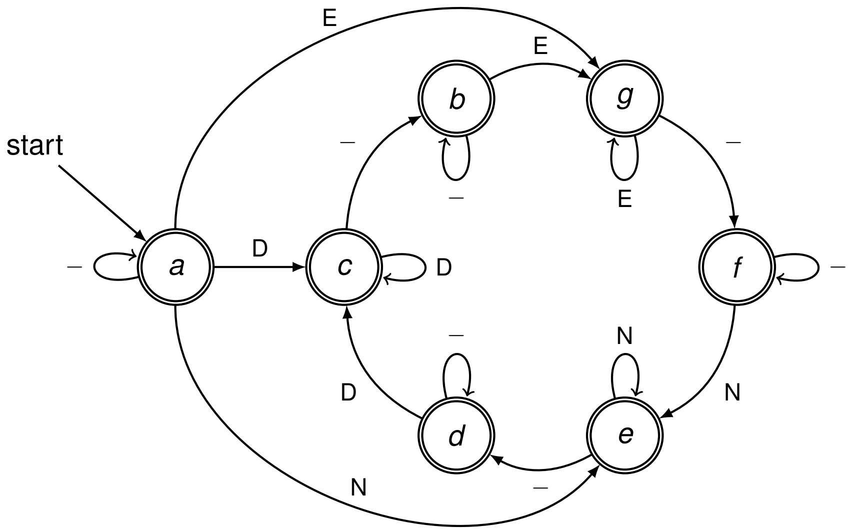
	Mon	Tue	Wed	Thu	Fri	Sat	Sun
M. Green	D	D	–	N	E	D	D
M. Blue	N	N	N	–	N	N	N
M. Red	–	–	D	D	D	–	–
M. Brown	E	E	E	E	–	E	E

Timetabling : constraint `Pattern`

- Additional constraint :
  - No period change without a day-off.
  - Forward rotation : D... E... N... D...
- Modelling : $\mathtt{pattern}(Job_{i.}, \mathcal{A}), \forall i$.
  These constraints are satisfied if every sequence (« word ») $(Job_{i1}, \ldots, Job_{i7})$ is satisfied by a finite automaton $\mathcal{A}$.

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
M. Green	D	D	–	E	E	E	E
M. Blue	E	E	E	–	N	N	N
M. Red	N	N	N	N	–	D	D
M. Brown	–	–	D	D	D	–	–

Finite automaton for our problem



Timetabling : a real-life solution

∘	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S
23796	-	-	-	-	D	D	N	N	-	-	D	-	-	-	-	D	-	D	D	-	D	D	-	-	-	-	-	-
603042	D	D	D	D	E	-	-	-	D	D	D	D	-	D	D	D	D	D	E	-	-	-	D	D	D	D	-	D
12310	D	D	-	-	-	-	-	-	-	-	-	-	-	D	D	D	-	-	-	-	-	-	-	-	-	-	-	D
511811	D	D	D	-	D	D	-	-	D	D	-	-	-	D	D	D	D	-	D	D	-	-	D	D	-	-	-	D
60324	-	-	D	D	D	-	D	D	-	D	D	D	-	-	-	-	-	D	D	-	D	D	D	-	D	D	D	-
603095	E	-	-	E	E	E	-	-	-	-	-	-	E	E	E	-	-	E	E	E	-	-	-	-	E	-	-	E
603230	-	D	D	D	D	-	D	D	D	D	-	D	D	-	-	D	D	D	D	-	D	D	D	-	D	D	D	-
510723	D	D	D	-	-	D	-	-	D	D	D	-	-	D	D	D	D	-	-	D	-	-	D	D	D	-	-	D
511104	-	R	R	R	R	R	-	-	R	R	R	R	R	-	-	-	-	E	E	-	E	E	-	-	E	E	E	-
34108	-	D	D	D	D	-	D	D	D	D	-	-	-	-	-	R	R	R	R	R	D	D	-	-	D	-	-	-
11866	-	D	-	D	D	D	E	E	-	-	D	-	-	-	-	D	-	D	D	D	E	E	-	D	-	-	-	-
35022	-	R	R	R	R	R	D	D	-	-	-	-	-	-	-	-	-	-	D	-	D	D	D	-	D	D	D	-
512287	E	E	E	-	D	D	E	E	-	-	-	-	-	E	E	E	E	-	D	-	E	E	-	-	E	-	-	E
56507	D	D	-	D	D	D	-	-	D	-	-	-	-	D	D	D	-	D	D	D	-	-	D	-	-	-	-	D
512281	-	E	-	D	D	-	D	D	E	-	-	-	-	-	-	E	-	D	D	-	D	D	E	-	-	-	-	-
511066	-	D	D	-	-	-	D	D	-	-	-	-	D	-	-	-	-	-	-	-	D	D	-	-	D	D	D	-
600955	D	D	-	D	D	-	-	-	-	-	-	-	-	D	D	D	-	D	D	-	-	-	-	-	-	-	-	D
602576	D	D	-	D	D	D	-	-	-	-	-	-	-	D	D	D	-	D	D	D	-	-	-	-	-	-	-	D
600315	-	-	T	T	-	-	T	T	-	T	-	T	T	-	-	-	T	-	-	T	T	T	-	-	T	T	T	-
511865	-	-	-	-	-	-	T	T	-	T	T	T	T	-	-	-	-	-	-	-	-	-	R	R	R	R	R	T

Lignes directrices

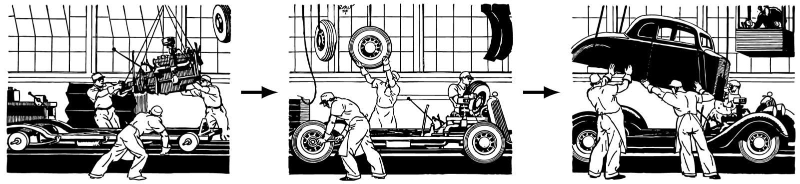
Symmetry

Problems modelling

- Frequency assignment
- Car sequencing
- Sports scheduling
- Timetabling
- « Job-shop »
- Cutting

Shop scheduling

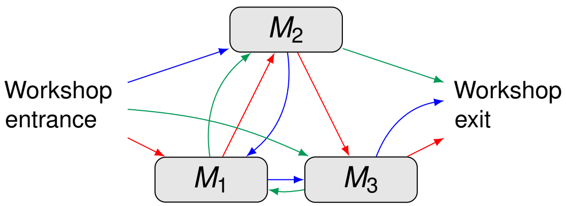
Shop scheduling models problems where jobs consist of operations which require specific machines (ressources).



Application examples

- Assembly workshops.
- Conveyor belt production.

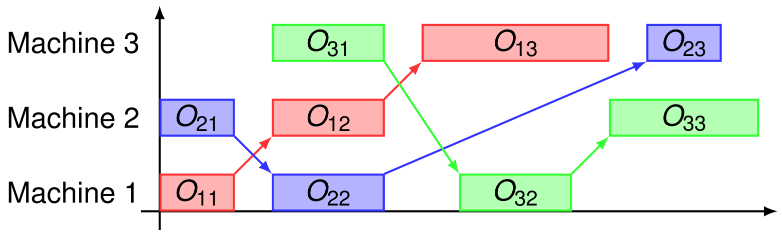
Job-shop



- Operations of each job form a chain : $O_{i1} \to O_{i2} \to \cdots \to O_{in_i}$.
- Jobs follow their own sequences on machines.

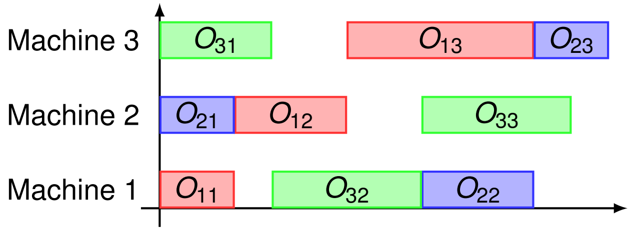
« Job-shop » scheduling : definition

- n jobs, each job J_i consists of a chain of n_i operations $(O_{i1}, \ldots, O_{i,n_i})$.
- m available machines.
- Each operation O_{ij} has duration p_{ij} and should be executed on machine $a_{ij} \in \{1, \ldots, m\}$.
- Aim : find a scheduling of length not exceeding T such that, on each machine, operations do not overlap.



« Job-shop » scheduling : modeling

- Variables : S_{ij} — stating time of execution of operation O_{ij}, $1 \leq i \leq n$, $1 \leq j \leq n_i$.
- Domains : $D_{S_{ij}} = [0, T - p_{ij}]$, $\forall i, j$.
- Constraints :
  - precedence : $S_{ij} + p_{ij} \leq S_{i,j+1}$, $\forall i$, $1 \leq j \leq n_i - 1$;
  - non-overlapping : $\texttt{disjunctive}(\{S_{ij}\}_{a_{ij}=k}, \{p_{ij}\}_{a_{ij}=k})$, $1 \leq k \leq m$.



Job-shop : example

In the company « Doeverything », some products are labeled befor being packaged, while for others the label is placed on the packaging. How long does it take to prepare the following batches ?

lot	*A*	*B*	*C*	*D*	*E*	*F*
packaging duration	10	16	14	4	8	4
labeling duration	12	10	12	0	6	8
packaging before labeling ?	oui	oui	oui		no	no

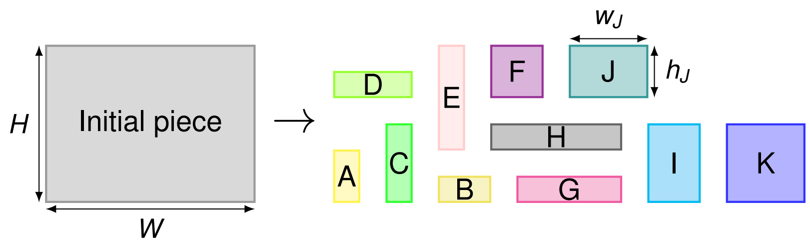
Source : François Vanderbeck

Lignes directrices

Cutting problem : definition

One needs to cut a rectangular piece (wooden, steel,...) in small pieces.

Rotations are not allowed.



Cutting problem : modelling

- Variables : X_i, Y_i — x and y coordinates of the lower left corner of piece i.
- Domains : $D_{X_i} = [0, W - w_i]$, $D_{Y_i} = [0, H - h_i]$, $\forall i$.
- Constraints for each pair (i, j) of pieces :

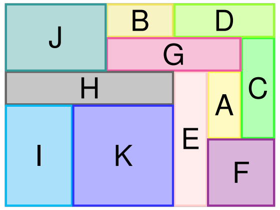
$$\underbrace{X_i + w_i \leq X_j}_{i \text{ is on the left of } j} \quad \underbrace{\bigvee}_{\text{or}} \quad \underbrace{X_i \geq X_j + w_j}_{i \text{ is on the right of } j} \quad \underbrace{\bigvee}_{\text{or}}$$

$$\underbrace{Y_i + h_i \leq Y_j}_{i \text{ is below } j} \quad \underbrace{\bigvee}_{\text{or}} \quad \underbrace{Y_i \geq Y_j + h_j}_{i \text{ is above } j}$$

- Constraints are very « loose » and local !

Cutting problem : redundant constraints

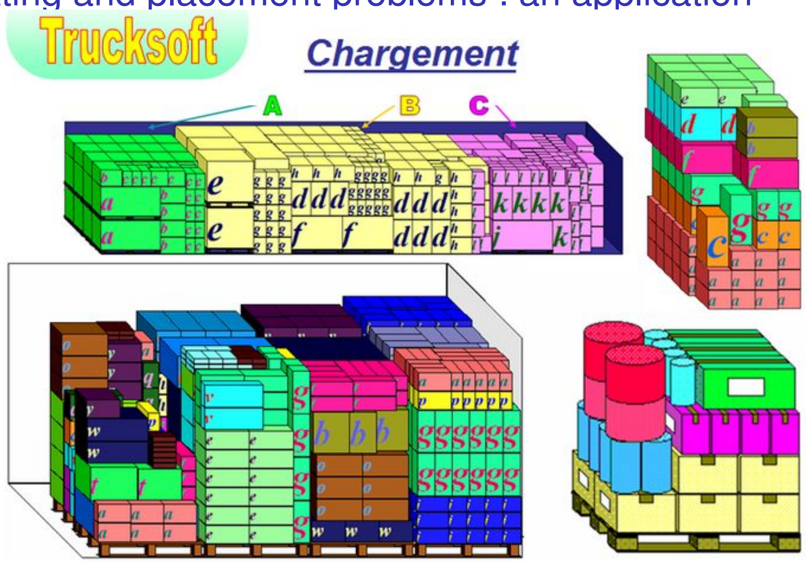
- We need a « global point of view » on our problem.
- We add some more constraints :
  - $\texttt{cumulative}(\{X_i\}_{\forall i}, \{w_i\}_{\forall i}, \{h_i\}_{\forall i}, H)$;
  - $\texttt{cumulative}(\{Y_i\}_{\forall i}, \{h_i\}_{\forall i}, \{w_i\}_{\forall i}, W)$.
- These constraints are redundant but useful !
- Results (Source : Pedro Barahona) :



Without `cumulative` : 24 seconds.

With `cumulative` : 40 milliseconds.

Cutting and placement problems : an application



Source : www.terciel.eu

Fundamentals of « efficient » modelling

- Try to use more global constraints and less local constraints (there is *Global Constraint Catalog* on the Internet).
- Determine and eliminate all symmetries you can.
- Use redundant constraints (but useful).
- Try different models.
- Try different heuristics for instantiation of variables and values.