

Contraintes Globales : Aspects Algorithmiques et Déclaratifs

Habilitation à Diriger des Recherches

Nicolas Beldiceanu

Swedish Institute of Computer Science

Lägerhyddsvägen 5, 75237 Uppsala

nicolas@sics.se

I CURRICULUM

II CONTRAINTES GLOBALES

III CONTRIBUTIONS SUR LES ASPECTS ALGORITHMIQUES

IV CONTRIBUTIONS SUR LES ASPECTS DÉCLARATIFS

V PERSPECTIVES

I. CURRICULUM

- ▶ septembre 1985–juin 1988, Paris 6 :
doctorant puis chercheur au laboratoire MASI, (CNRS et Université Paris 6),
- ▶ septembre 1988–décembre 1990, ECRC :
chercheur à «European Computer Research Centre» à Munich en Allemagne
(centre de recherche de BULL, SIEMENS et ICL),
- ▶ janvier 1991–octobre 1999, COSYTEC :
chef de projet puis responsable de la recherche à COSYTEC à Paris en France,
- ▶ depuis novembre 1999, SICS :
responsable de l'activité contrainte à «Swedish Institute of Computer Science»
à Uppsala en Suède.

**«Combinatorial Problem Solving
and Prolog Technology» Group**
(N.Beldiceanu, M.Carlsson, P.Danielsson, P.Mildner)

- ▶ Situé au sein de **Intelligent System Lab.** à SICS (90 personnes)

- ▶ Thème programmation par contraintes sur les domaines finis :
 - **Classification** des contraintes globales,
 - Recherche de **principes essentiels** permettant de dériver de nombreux algorithmes de filtrages,
 - Développement d'**algorithmes de filtrage efficaces** s'appuyant sur la théorie des graphes, la géométrie et les automates,
 - **Prototypes** pour des industriels.

- ▶ Contrats industriels :
 - Pyrosequencing AB : **bioinformatique** (analyse de séquences)
 - Carmen Systems : **aérien** (planification de personnel)
 - Tacton : **configuration** (e-commerce)

Enseignement

- ▶ Théorie de graphes niveau licence (36 heures, à Paris 6)
C et Unix niveau maîtrise (24 heures, à Paris 6),
- ▶ Cours de programmation par contraintes tout au long de mon séjour à COSYTEC (cours sur une semaine à une fréquence d'un cours tout les trois mois).
- ▶ Interventions pour des cours de programmation par contraintes dans des écoles (Mines de Nantes), dans des universités (Paris 6, Lille, LRI) ou dans des laboratoires (LAAS),
- ▶ Cours sur une période de deux jours (16 heures de cours) sur les contraintes à des ingénieurs en Suède,
- ▶ Préparation d'une partie (18 heures de cours) du cours d'été 2003 sur les contraintes à l'Université d'Uppsala (<http://user.it.uu.se/~pierref/courses/CT/>).

Collaborations Académiques

- **Mats Carlsson** (SICS)
- **Sven Thiel** (MPI Saarbrücken, Algorithmic group)
- **Qi Guo** (Université d'Uppsala, Mathématiques)
- **Justin Pearson** (Université d'Uppsala, Informatique)
- **David Hanak et Péter Szeredi** (Université de Budapest)
- **Emmanuel Poder** (Université Blaise Pascal)

Encadrements

Thèse (CIFRE entre COSYTEC et LIPN, Université Paris 13) de **E. Bourreau** :
Traitement de contraintes sur les graphes en programmation par contraintes

Thèse (CIFRE entre COSYTEC et LIMOS) de **E. Poder** :
Programmation par contraintes et ordonnancement de tâches avec consommation variable de ressource

Stage de DEA de l'Université d'Orléans ⇒ *évaluation de contraintes*
Comparaison programmation par contraintes recuit simulé sur des problèmes de placement de tâches

Stage de fin d'étude (IMA d'Angers) ⇒ *librairie de méta heuristiques*
Introduction de méta heuristiques compatibles avec les contraintes et test sur des problèmes d'ordonnancement

Stage École Polytechnique ⇒ *ressources non-renouvelables*
Ordonnancement de tâches avec consommation variable de ressource

Stage de fin d'étude d'ingénieur de l'ISIMA de Clermont-Ferrand
Problèmes de découpe ⇒ *package métiers*

Co-encadrement de «Master Thesis» de Magnus Ågren, Université d'Uppsala
Outils de trace pour la mise au point de contraintes ⇒ *aide utilisateurs PPC*

Réalisations logicielles

Contraintes globales dans CHIP <http://www.cosytec.com>:

- Among, Cumulative*, Cycle, Diffn*, Sequence (120.000 lignes).

Contraintes globales dans SICStus <http://www.sics.se/sicstus>:

- Case, Cumulatives, Disjoint1, Disjoint2, Lex_chain, Sum_of_weights_of_distinct_val.

Reprises dans la plupart des systèmes de PPC:

- | | | | |
|----|--------------|---|----------------------------|
| ** | • BProlog | : http://www.probp.com | États-Unis |
| | • ÉCLAIR | : http://www.thalesgroup.com | France |
| * | • ECLiPSe | : http://www.icparc.ic.ac.uk | Angleterre |
| | • FaCile | : http://www.recherche.enac.fr | France |
| | • ICEBERG | : http://www.choco-constraints.net | France |
| ** | • IF/PROLOG | : http://www.ifcomputer.com | Allemagne, Japon |
| * | • ILOG Suite | : http://www.ilog.com | France, États-Unis |
| | • Koalog | : http://www.koalog.com | France |
| * | • Mozart | : http://www.mozart-oz.org | Allemagne, Suède, Belgique |

Comités de Programmes

- TRICS 2000** : Techniques foR Implementing Constraint programming Systems, 22 septembre 2000, Singapore.
- CP-AI-OR'2001** : Third International Workshop on Integration of AI and OR Techniques, Ashford, Kent UK, avril, 2001.
- CP-AI-OR'02** : Fourth International Workshop on Integration of AI and OR techniques, Le Croisic, France, mars 2002.
- TRICS 2002** : Techniques foR Implementing Constraint programming Systems, septembre 2002, Cornell.
- CP 2002** : Eighth International Conference on Principles and Practice of Constraint Programming, septembre 2002, Cornell.
- JFPLC'2002** : Dixièmes Journées Francophones de Programmation Logique et Programmation par Contraintes, Nice - mai 2002.
- CP-AI-OR 2003** : Fifth International Workshop on Integration of AI and OR Techniques, Montréal, Canada, mai 2003.
- JFPLC'2003** : Douzièmes Journées Francophones de Programmation Logique et Programmation par Contraintes, Amiens, France, juin 2003.

Projets de Recherches

Conception et développement d'outils basés sur des contraintes pour vérifier des propriétés de réseaux de Petri, Université Paris 6, 1986–1988.

Développement de contraintes dans les domaines finis dans CHIP, European Computer Research Centre, 1988–1990.

Création d'une bibliothèque de contraintes en C, projet de recherche ANVAR, 1992–1993 (projet commun COSYTEC, SLIGOS).

Disjonction de contraintes linéaires, Ministère de la recherche RNTL, 1993–1994 (projet commun COSYTEC, Université d'Orsay, Laboratoire de Recherche en Informatique, équipe Démonstration et Programmation).

Conception et réalisation d'outils de mise au point de contraintes. Projet Esprit Discipl (Long Term Research Domain, Theme 2 : Nb : 22532), 1997–1998, (projet commun COSYTEC, PrologIA, ICON, BIS, INRIA, Université de Bristol, Université de Linköping et Université de Madrid).

Contraintes globales génériques (Projet interne à SICS supporté par VINNOVA), 1999–2003.

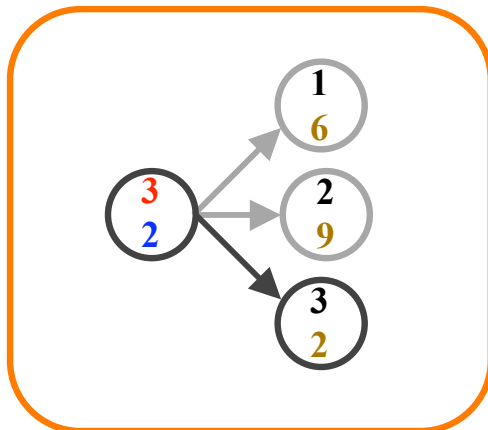
II. CONTRAINTES GLOBALES

- ASPECT DÉCLARATIF
- ASPECT ALGORITHMIQUE
 - Exemple
 - Propagation
 - Enumération

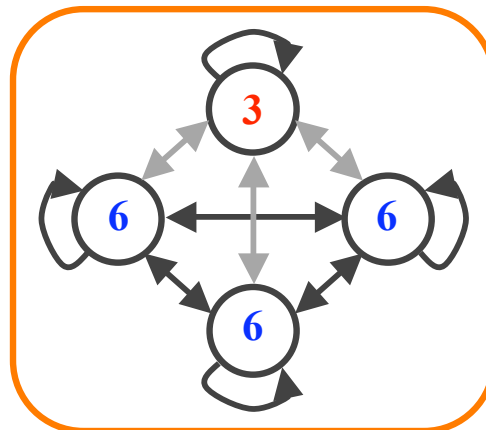
Aspect Déclaratif

BUT: Identifier des structures récurrentes
aux problèmes combinatoires discrets
(variables prenant leur valeur dans \mathbb{Z}).

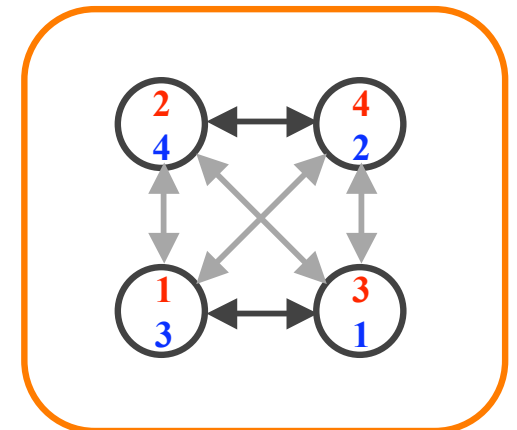
Indépendamment
de toute méthode
de résolution !



element(3, {6, 9, 2}, 2)



nvalue(2, {6, 3, 6, 6})



symetric_alldiff({3, 4, 1, 2})

- Fonction de coût,
- Certaines disjonctions.
- Limite nbre.de ressources distinctes,
- Relaxation de la contrainte *alldifferent*.
- Organisation de tournois.

Contrainte *cycle* (THÈSE DE E. BOURREAU)

Propagation et Énumération

DÉFINITION

Couvrir un graphe orienté par un ensemble de N circuits de manière à ce que chaque sommet appartienne à un circuit et un seul.

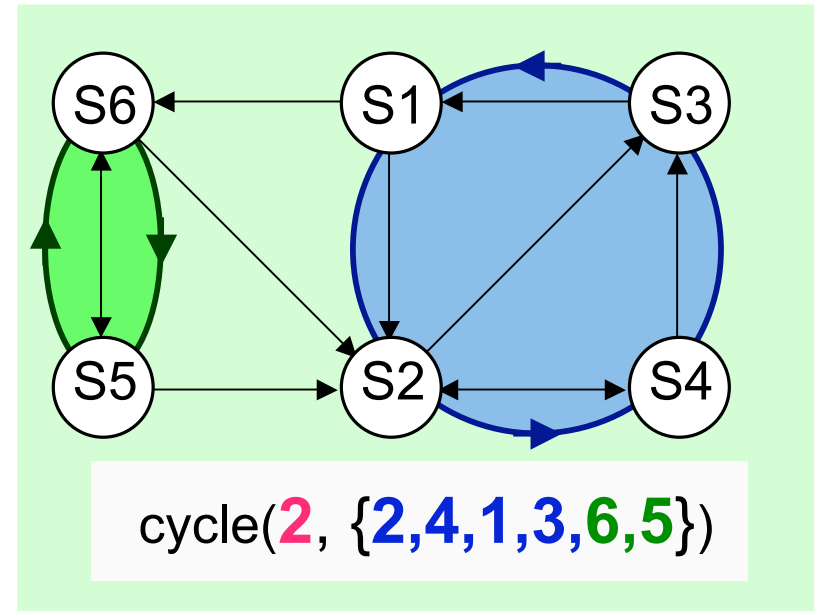
Nombre N de cycles d'une permutation (autre interprétation)

CONTRAİNTE

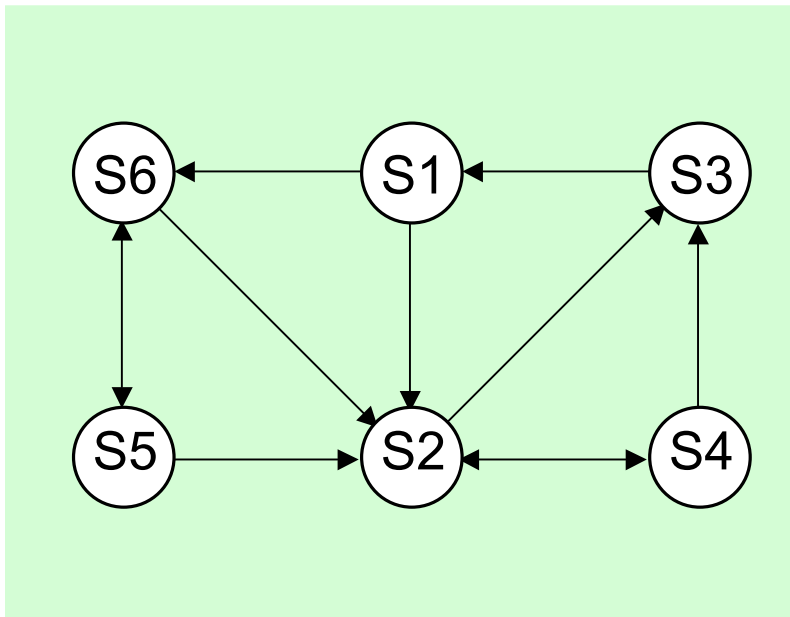
$\text{cycle}(N, \text{Successeurs})$

Extensions, contraintes sur :

- Poids de chaque circuit,
- Nœuds \notin même circuit,
-



Représentation du graphe



S1 :: {2,6}

S2 :: {3,4}

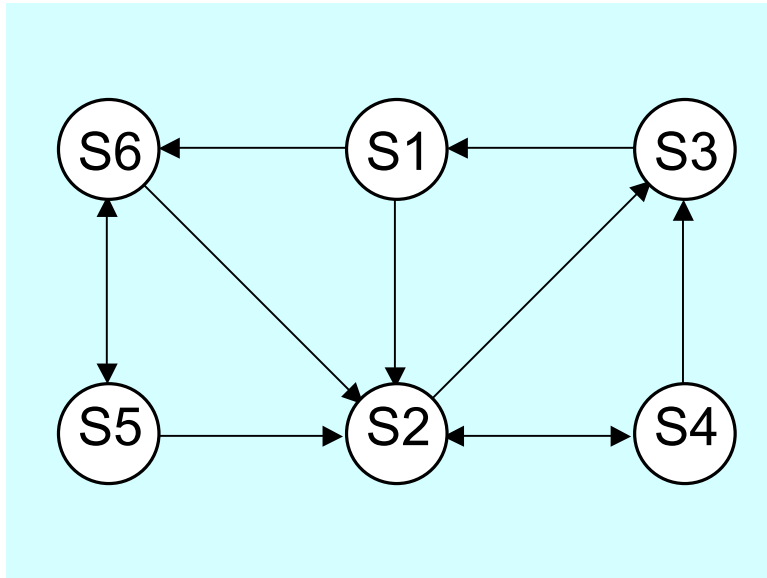
S3 :: {1}

S4 :: {2,3}

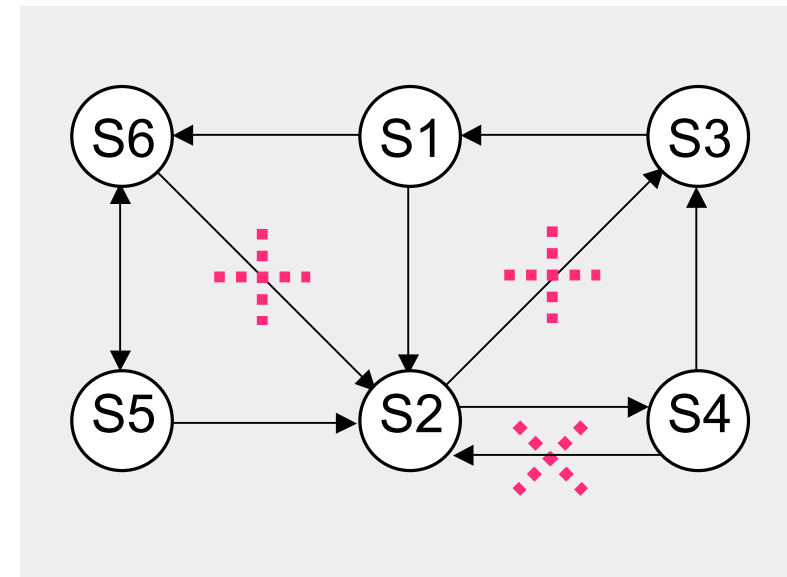
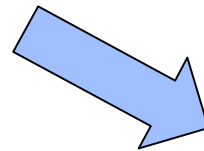
S5 :: {2,6}

S6 :: {2,5}

Propagation de Contraintes



$\text{cycle}(N, \{S1, S2, S3, S4, S5, S6\})$

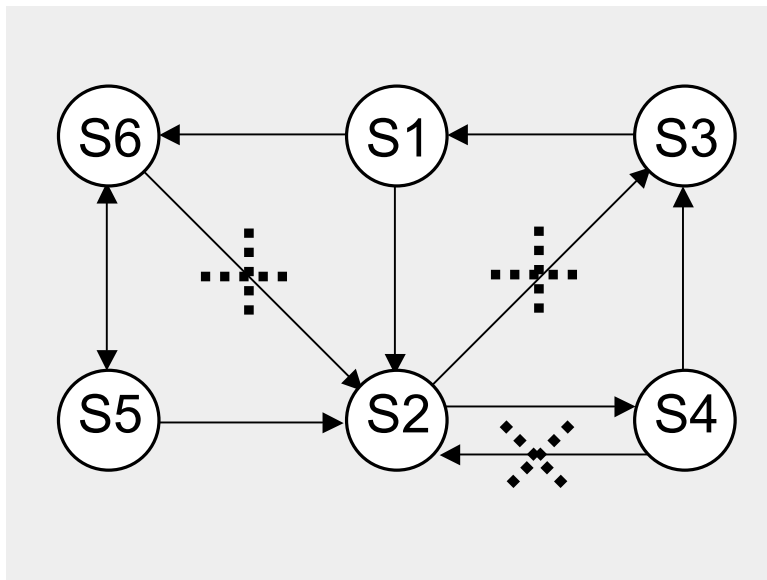


$N \leq 2$

$S6 \neq 2 \quad S2 \neq 3 \quad S4 \neq 2$

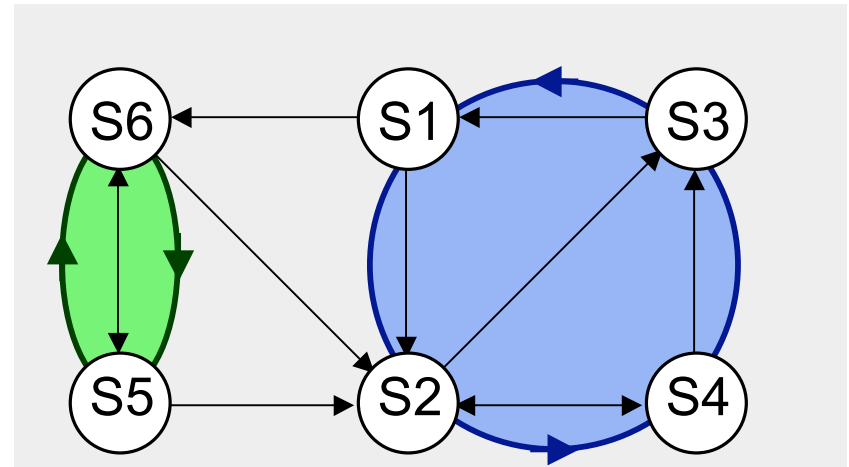
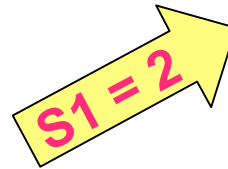
- composantes fortement connexe
- couplage
- règles de déduction
(*couplage et poids minimum*)

Enumération

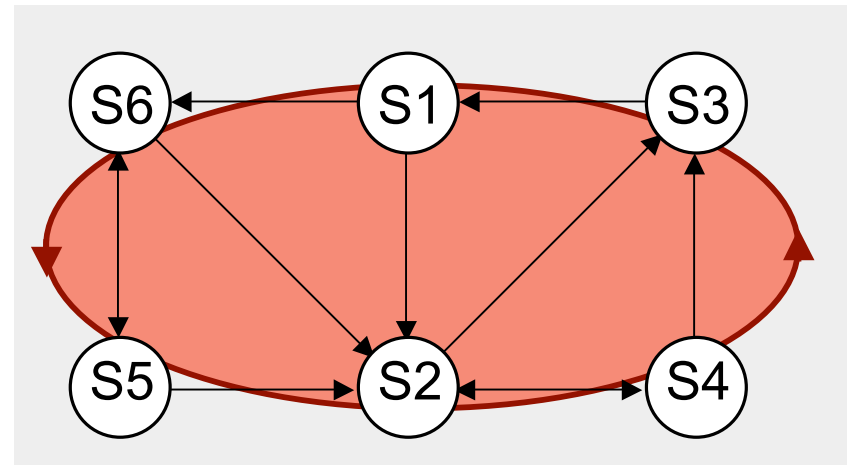
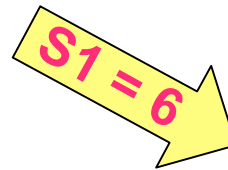


$N \leq 2$

$S6 \neq 2$ $S2 \neq 3$ $S4 \neq 2$



cycle(2, {2,4,1,3,6,5})



cycle(1, {6,4,1,3,2,5})

III. CONTRIBUTIONS SUR LES ASPECTS ALGORITHMIQUES

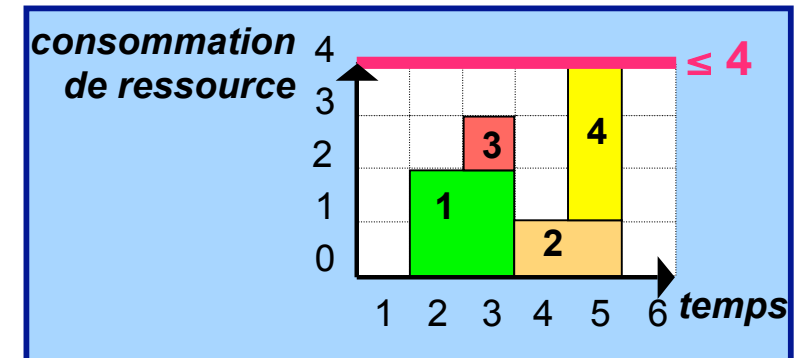
- **ORDONNANCEMENT AVEC CONTRAINTES DE CAPACITÉ**
 - Contraintes «*cumulative*» et «*cumulatives*»
 - Contrainte «*cumulative_trapèze*»
- **GÉOMETRIE**
 - Utilisation de la méthode de balayage pour différents patrons
 - Contrainte de non-recoupement
- **OPTIMISATION**
 - «*somme des poids associés aux valeurs distinctes*»
- **AUTOMATES POUR LA CONCEPTION D'ALGORITHMES DE FILTRAGE**
 - Contraintes d'ordre lexicographique

Contraintes cumulatives

La contrainte «**cumulative**»

[AGGOUN & BELDICEANU 92] :

- **Restreint** la consommation de ressource à chaque instant.



Généralisations: «**cumulatives**»

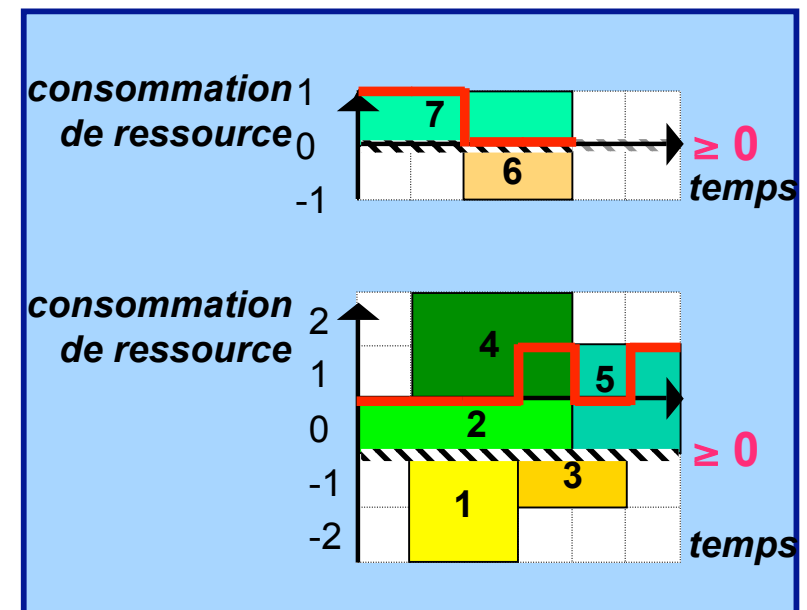
[BELDICEANU & CARLSSON 01] :

- Un **pool** de ressources,
- La consommation d'une tâche peut être **négative**,
- Contraint également le niveau **minimum**.



Algorithmes de filtrage pour cette contrainte:

- Généralisation *partie obligatoire*,
- Généralisation *intervalles de tâches*,
- Utilisation de la *construction disjonctive*.



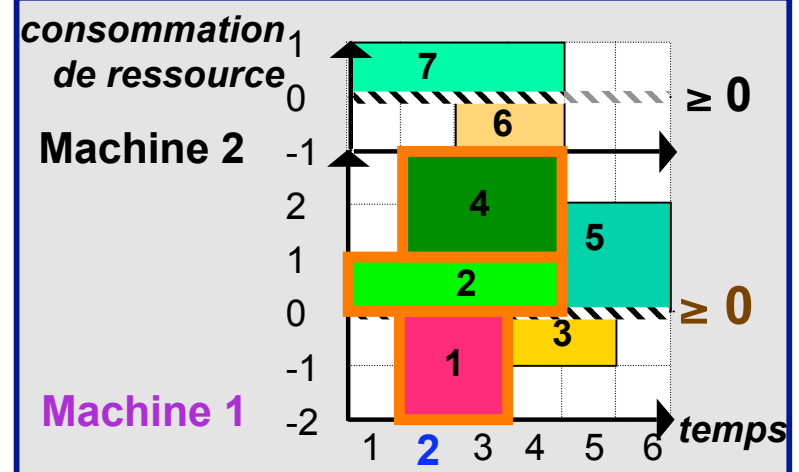
Pour toutes tâches t :

Pour tout instant i coupé par une tâche t :

Soit m la machine sur laquelle t est affectée ,

Soit S l'ensemble des tâches coupant i , et affectées à m :

La somme des consommations des tâches S n'est pas supérieure (inférieure) à la **capacité** de m .



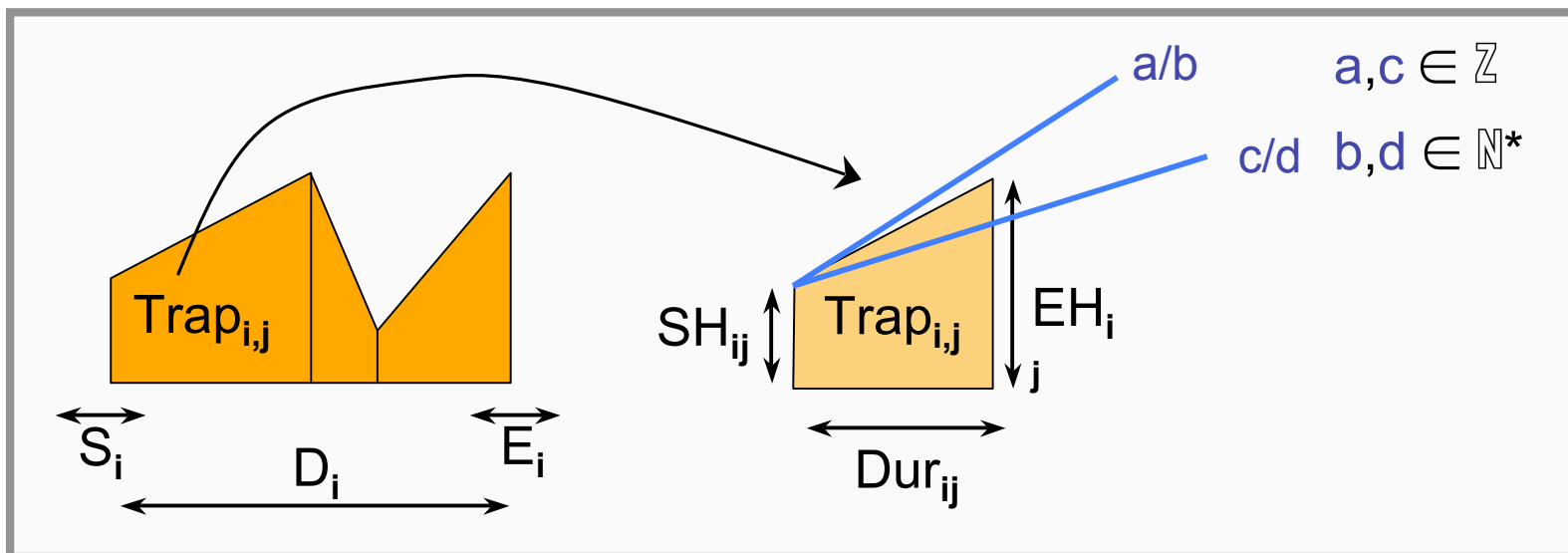
La Contrainte *cumulative_trapeze* (THÈSE DE E.PODER)

Nouveau modèle de tâche

Une tâche est définie par:

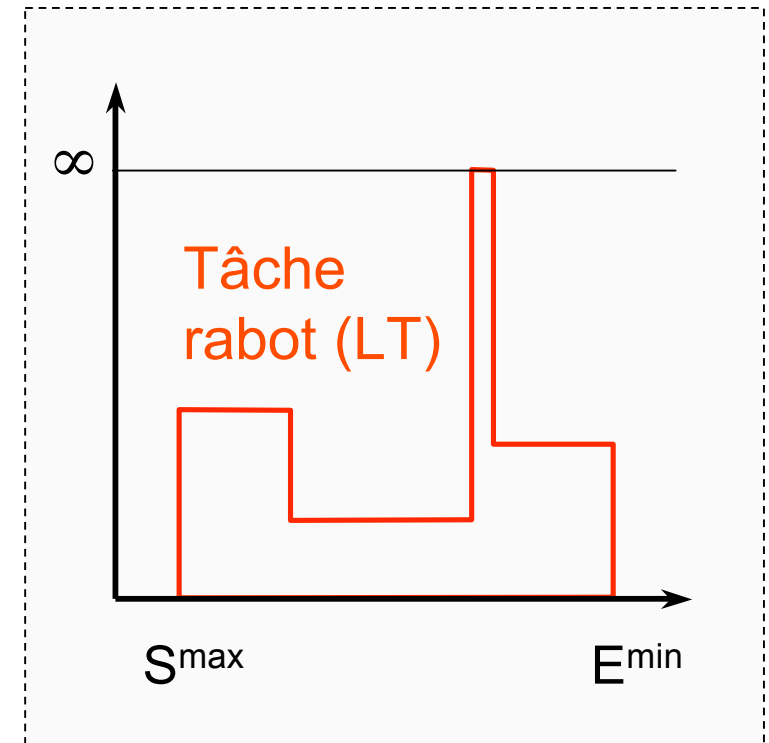
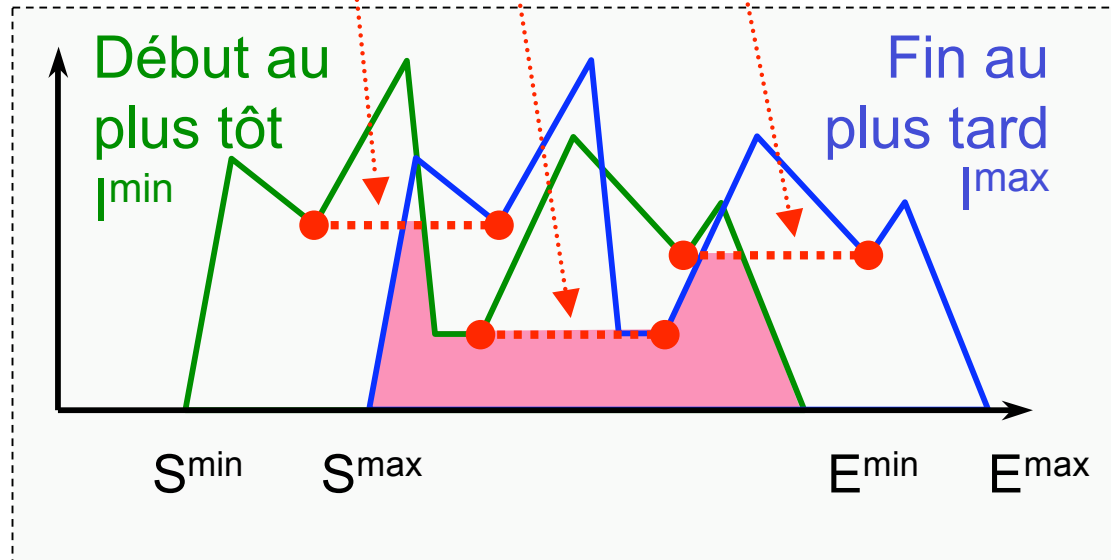
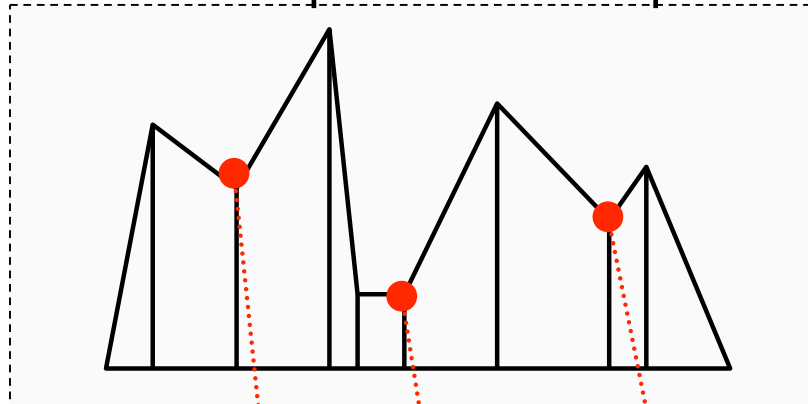
- un début S_i , une durée D_i , une fin E_i

😊 - une consommation **variant au cours du temps** représentée par une séquence de trapèzes contigus :



Calcul de la **Partie Obligatoire** d'une Tâche

Tâche composée de n trapèzes



$$\mathbf{PO} = |^{\min} \cap |^{\max} \cap \text{LT}(\text{complexité } O(n \log n), \text{ [PODER, BELDICEANU \& SANLAVILLE 02]})$$

- ↳ ORDONNANCEMENT AVEC CONTRAINTES DE CAPACITÉ
- ↳ GÉOMETRIE
 - Utilisation de la méthode de balayage pour différents patrons
 - Une conjonction de contraintes ayant deux variables en commun
 - Synchronisation de plusieurs conjonctions de contraintes
 - L'opérateur de cardinalité
- ↳ OPTIMISATION
- ↳ AUTOMATES POUR LA CONCEPTION D'ALGORITHMES DE FILTRAGE

Plus de 100 références dans la «**Geometry Literature Database**»
(*intersections de segments, intersections de rectangles, triangulations, ...*)

Mais **inutilisé** en programmation par contraintes alors que l'on n'arrive pas à bien propager les **contraintes conditionnelles**.

Une contrainte peut être vue comme un **ensemble d'objets géométriques**, chaque variable correspondant à une dimension.

L'art de propager consiste à **faire passer des informations d'un ensemble de dimensions vers un autre ensemble de dimensions**.

Régions Interdites

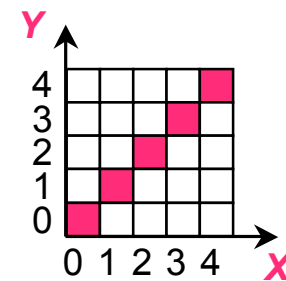
DÉFINITION *région interdite* par rapport à une contrainte **Ctr** et deux variables **X, Y** de Ctr :

Deux intervalles $\text{inf}_x..sup_x$ et $\text{inf}_y..sup_y$ tels que:

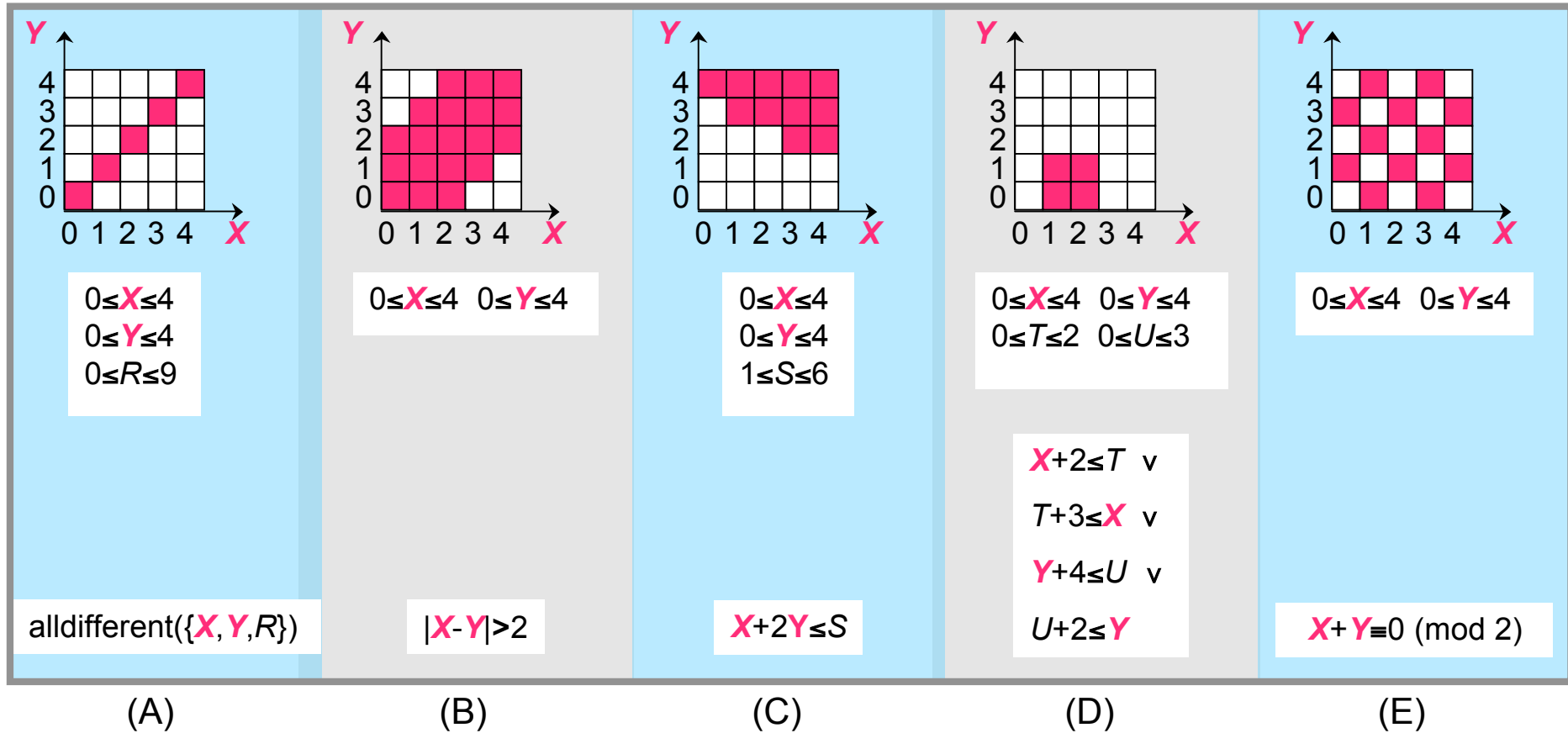
Pour tout $x \in \text{inf}_x..sup_x$,

$y \in \text{inf}_y..sup_y$: **Ctr** avec l'affectation **X=x, Y=y** est **fausse**.

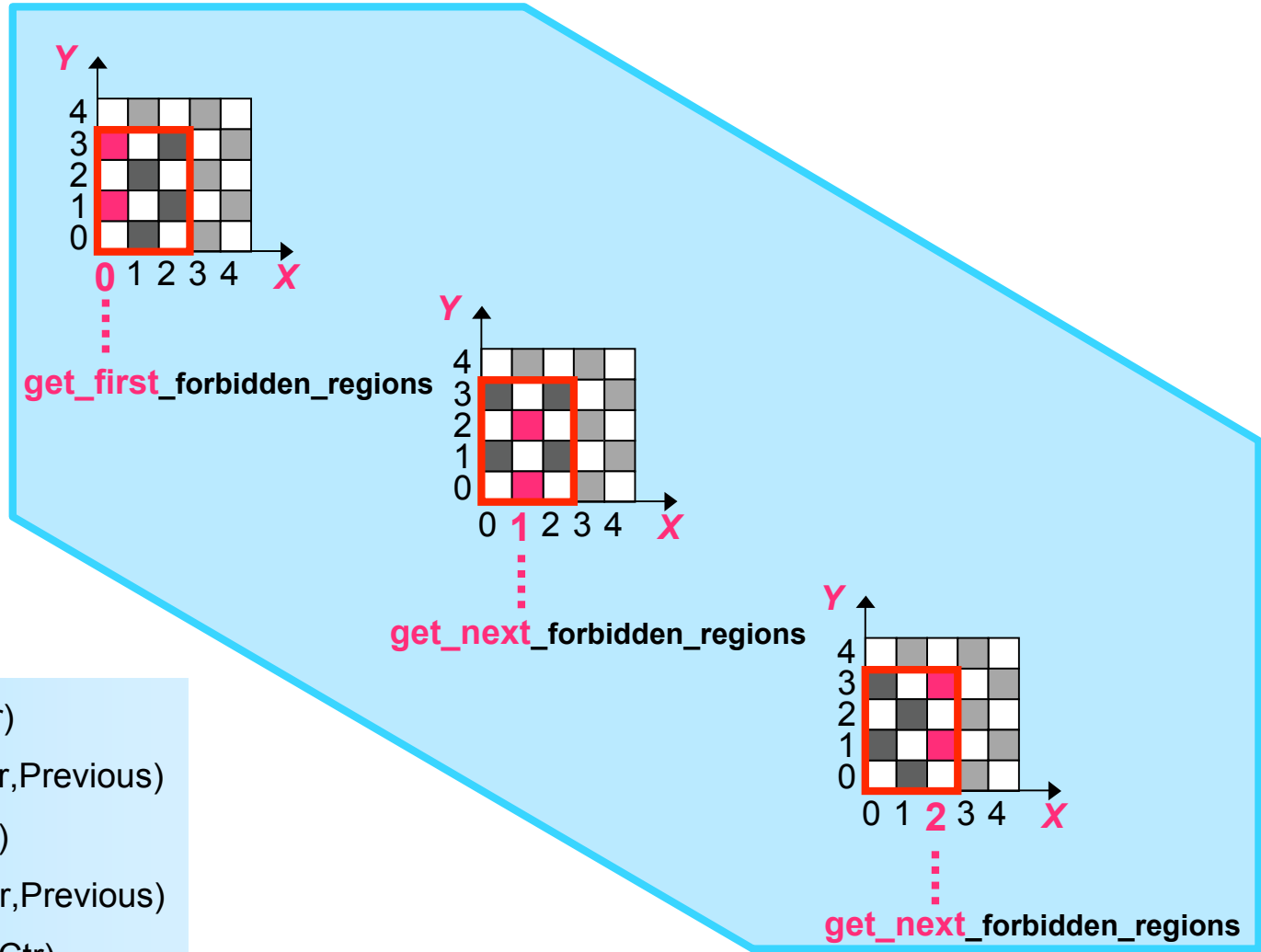
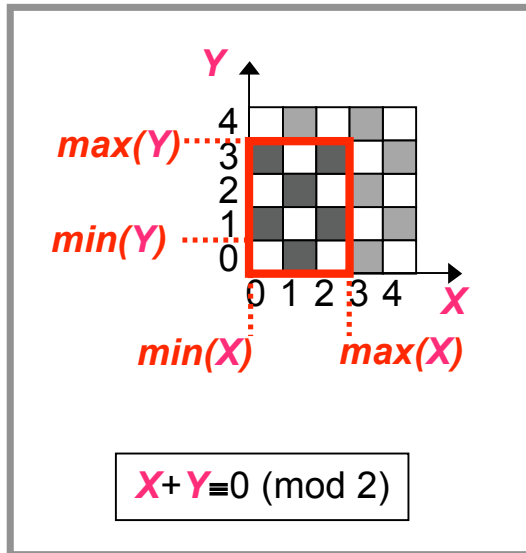
EXEMPLE régions interdites de
alldifferent({**X, Y, R**})
 en fonction de **X** et **Y** :



Généralité d'Expression des Régions Interdites



Extraction **Dynamique** des Régions Interdites à la **Demande**



```

get_first_forbidden_regions(X,Y,Ctr)
get_next_forbidden_regions(X,Y,Ctr,Previous)
get_last_forbidden_regions(X,Y,Ctr)
get_prev_forbidden_regions(X,Y,Ctr,Previous)
check_if_in_forbidden_regions(x,y,Ctr)

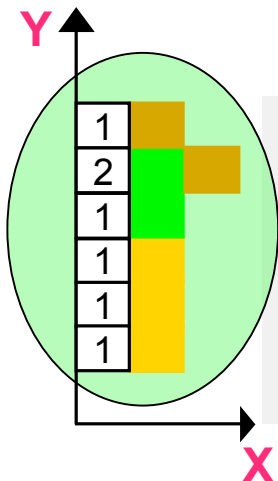
```

Idée Clef de l'Élagage par Balayage

Accumuler les régions interdites
provenant des différentes contraintes

- CTR₁(X, Y, ...)
- CTR₂(X, Y, ...)
-
- CTR_n(X, Y, ...)

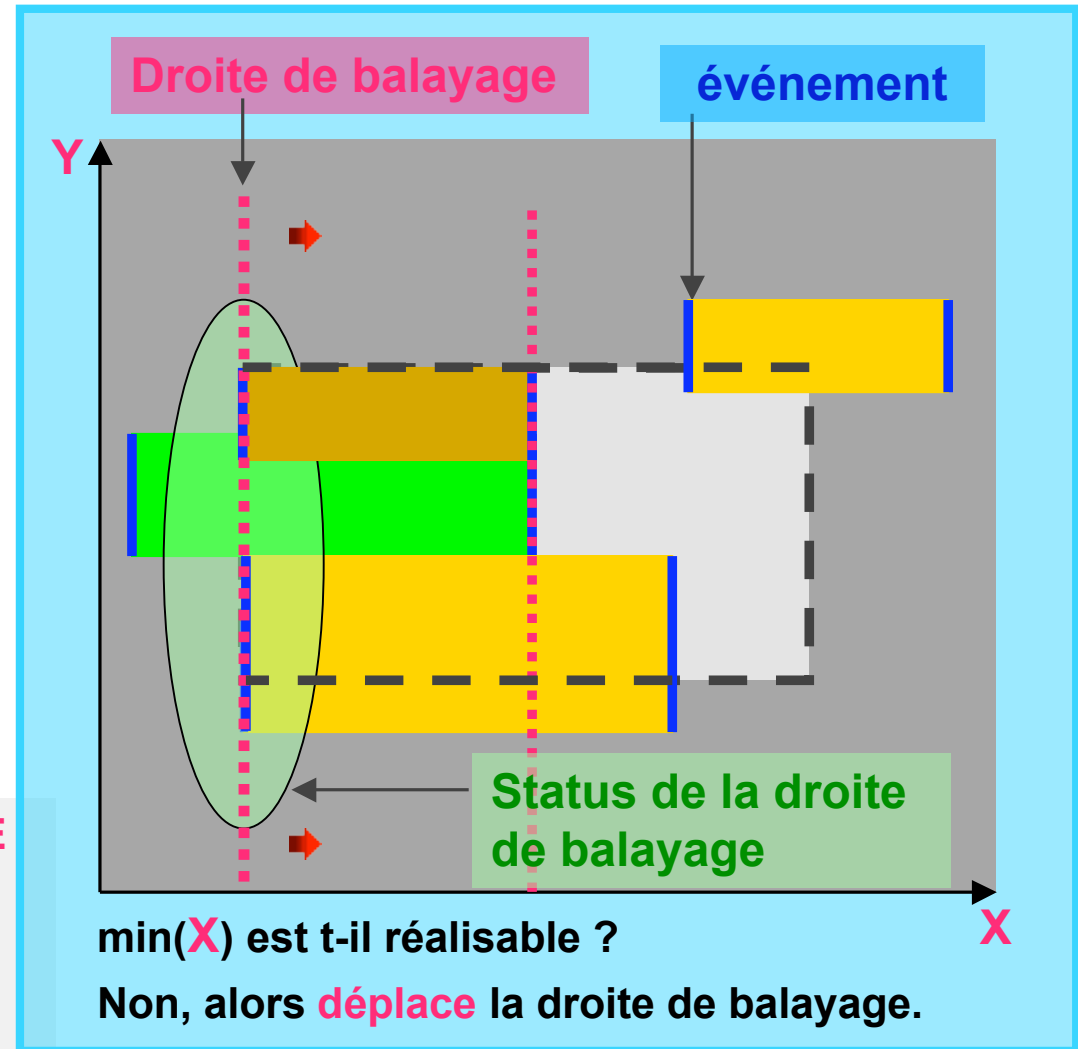
mentionnant 2 variables données X, Y



STATUS DROITE DE BALAYAGE

Pour tout $y \in \text{dom}(Y)$:
nombre de régions interdites
contenant le point (Δ, y)

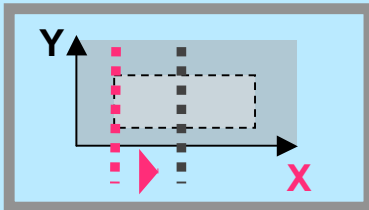
a-b tree [MEHLHORN]



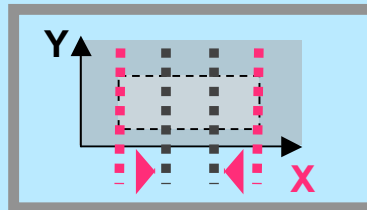
[BELDICEANU 00a] [BELDICEANU & CARLSSON 01b]

Utilisations Possibles

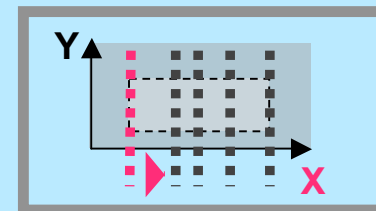
- Test de **réalisabilité**



- **Ajustement** de bornes



- **Élagage** de valeurs



Exemple

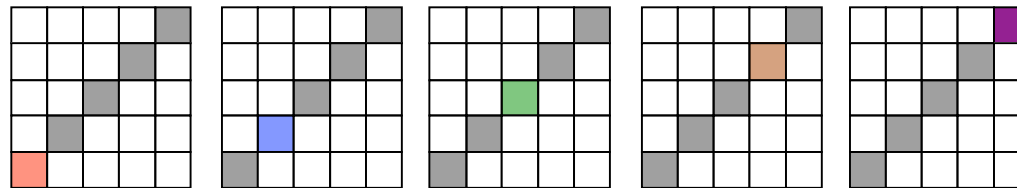
PROBLÈME:

Ajuste minimum de **X**

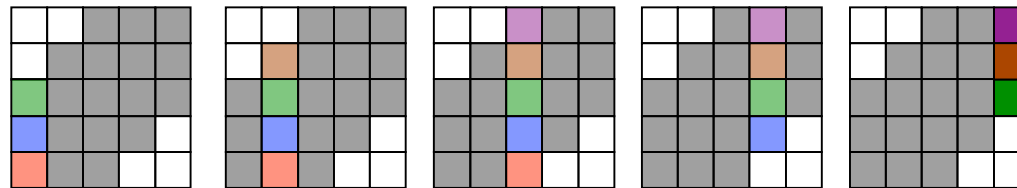
en fonction de **Y**

et de **toutes** les contraintes:

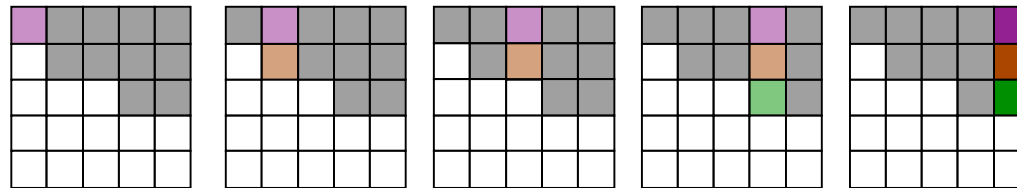
- $0 \leq X \leq 4$ $0 \leq Y \leq 4$
- $1 \leq S \leq 6$ $0 \leq T \leq 2$ $0 \leq U \leq 3$
- alldifferent({**X**, **Y**, **R**})
- $|X - Y| > 2$
- $X + 2Y \leq S$
- $X + 2 \leq T \vee T + 3 \leq X \vee$
- $Y + 4 \leq U \vee U + 2 \leq Y$
- $X + Y \equiv 0 \pmod{2}$



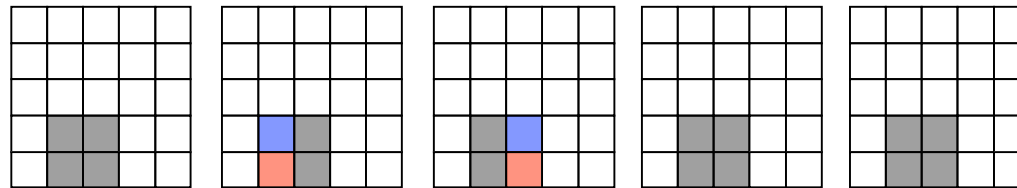
alldifferent({**X**, **Y**, **R**})



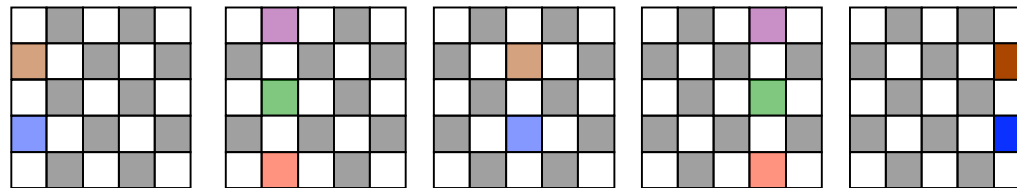
$|X - Y| > 2$



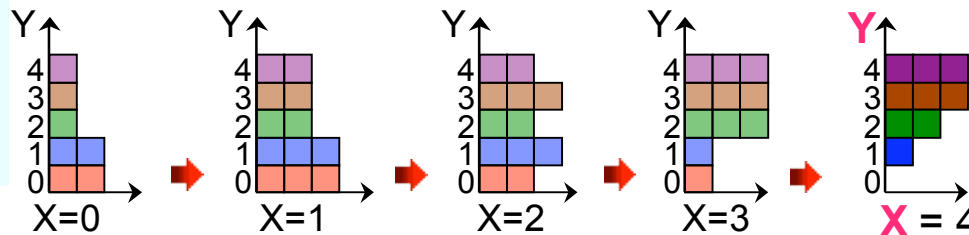
$X + 2Y \leq S$



$X + 2 \leq T \vee T + 3 \leq X \vee$
 $Y + 4 \leq U \vee U + 2 \leq Y$



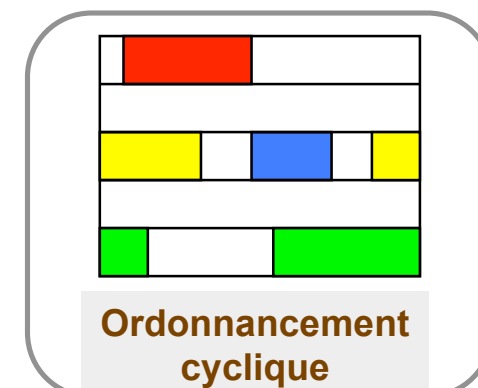
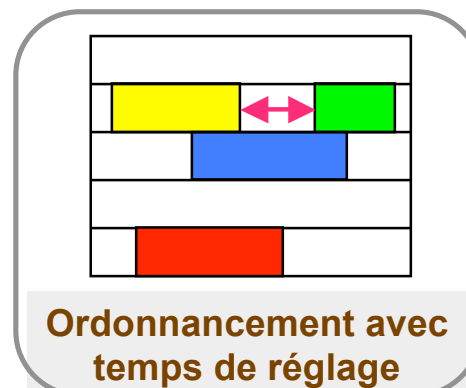
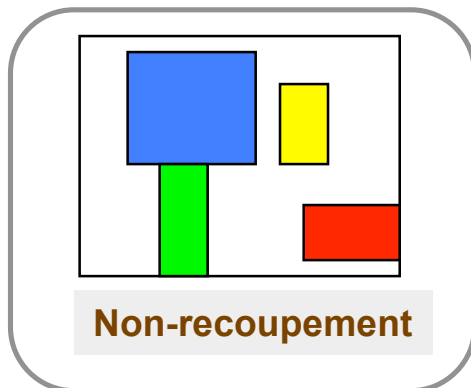
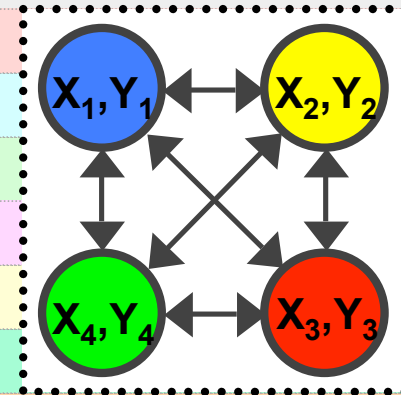
$X + Y \equiv 0 \pmod{2}$



Déduction:
X > 3

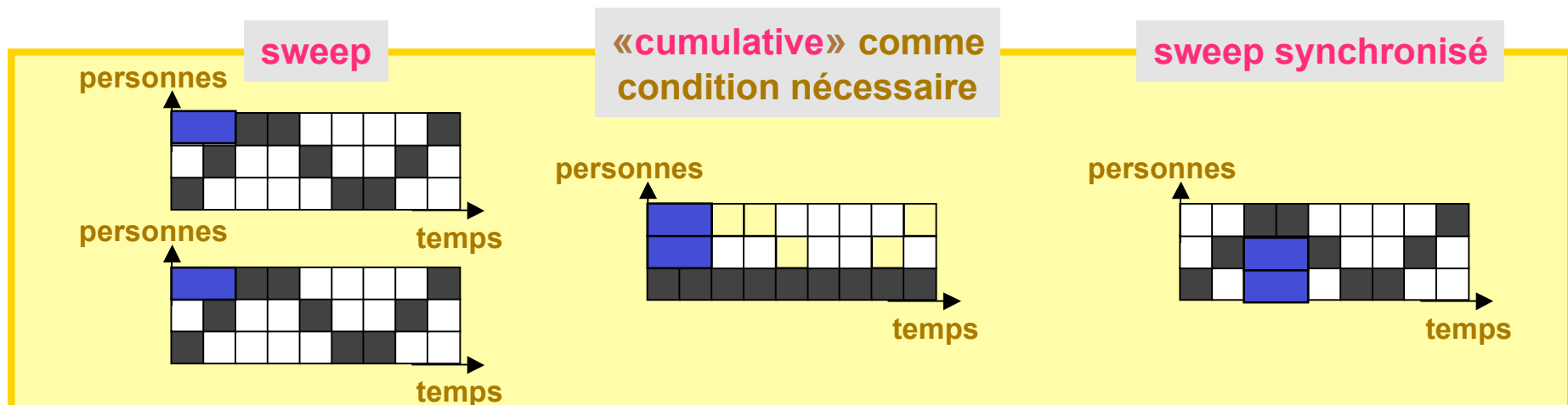
Patron des Contraintes Globales où s'Applique la Méthode

• ARGUMENT	: OBJECTS: collection(X -dvar, Y -dvar, ...)
• RESTRICTION (S)	: required(OBJECTS.X, OBJECTS.Y)
• VERTEX INPUT	: OBJECTS
• VERTEX GENERATOR	: IDENTITY
• EDGE INPUT	: OBJECTS
• EDGE GENERATOR	: CLIQUE (\neq)
• EDGE ARITY	: 2
• EDGE CONSTRAINT	:
• GRAPH PROPERTY	: NEDGE = OBJECTS . OBJECTS - OBJECTS



- ↳ ORDONNANCEMENT AVEC CONTRAINTES DE CAPACITÉ
- ↳ GÉOMETRIE
 - Utilisation de la méthode de balayage pour différents patrons
 - Une conjonction de contraintes ayant deux variables en commun
 - – Synchronisation de plusieurs conjonctions de contraintes
 - L'opérateur de cardinalité
 - Contrainte de non-recoupement
- ↳ OPTIMISATION
- ↳ AUTOMATES POUR LA CONCEPTION D'ALGORITHMES DE FILTRAGE

Motivation initiale: constituer et placer des **groupes** de personnes en tenant compte des indisponibilités des personnes.



Synchronisation de Droites de Balayage

[BELDICEANU, CARLSSON & THIEL 03]

PROBLÈME:

Ajuste minimum de X par rapport au fait que cette conjonction de contraintes est satisfaite:

$$0 \leq X \leq 4$$

$$0 \leq Y1 \leq 4 \quad 0 \leq Y2 \leq 4$$

$$0 \leq W \leq 9 \quad 2 \leq Z \leq 3$$

$$1 \leq S \leq 6 \quad 0 \leq T \leq 0 \quad 1 \leq U \leq 2$$

$$\text{alldifferent}(\{X, Y1, W\})$$

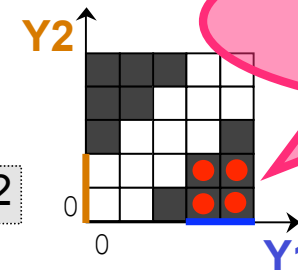
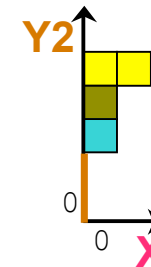
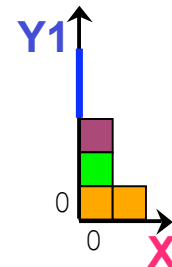
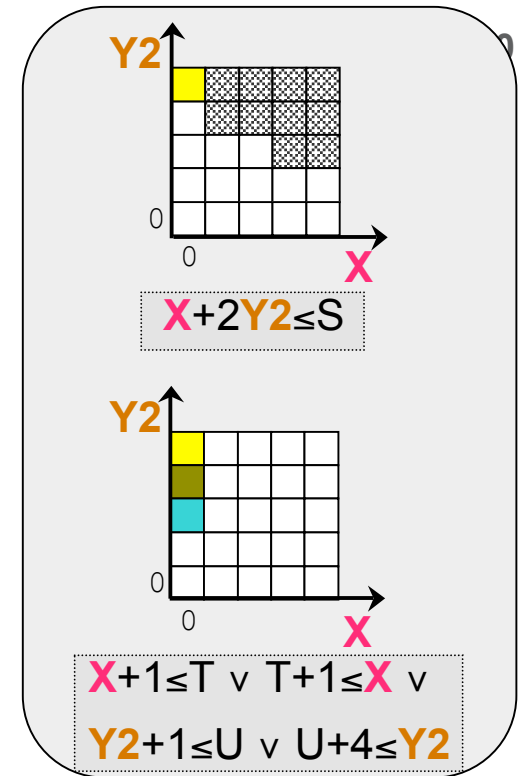
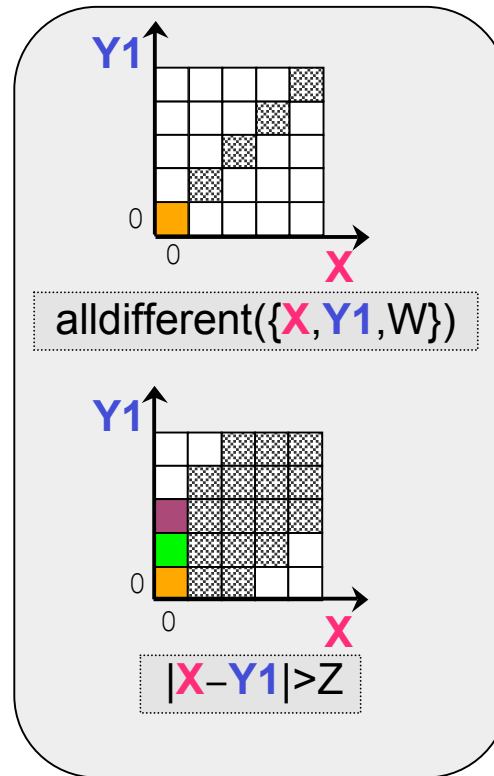
$$|X - Y1| > Z$$

$$X + 2Y2 \leq S$$

$$X + 1 \leq T \vee T + 1 \leq X \vee$$

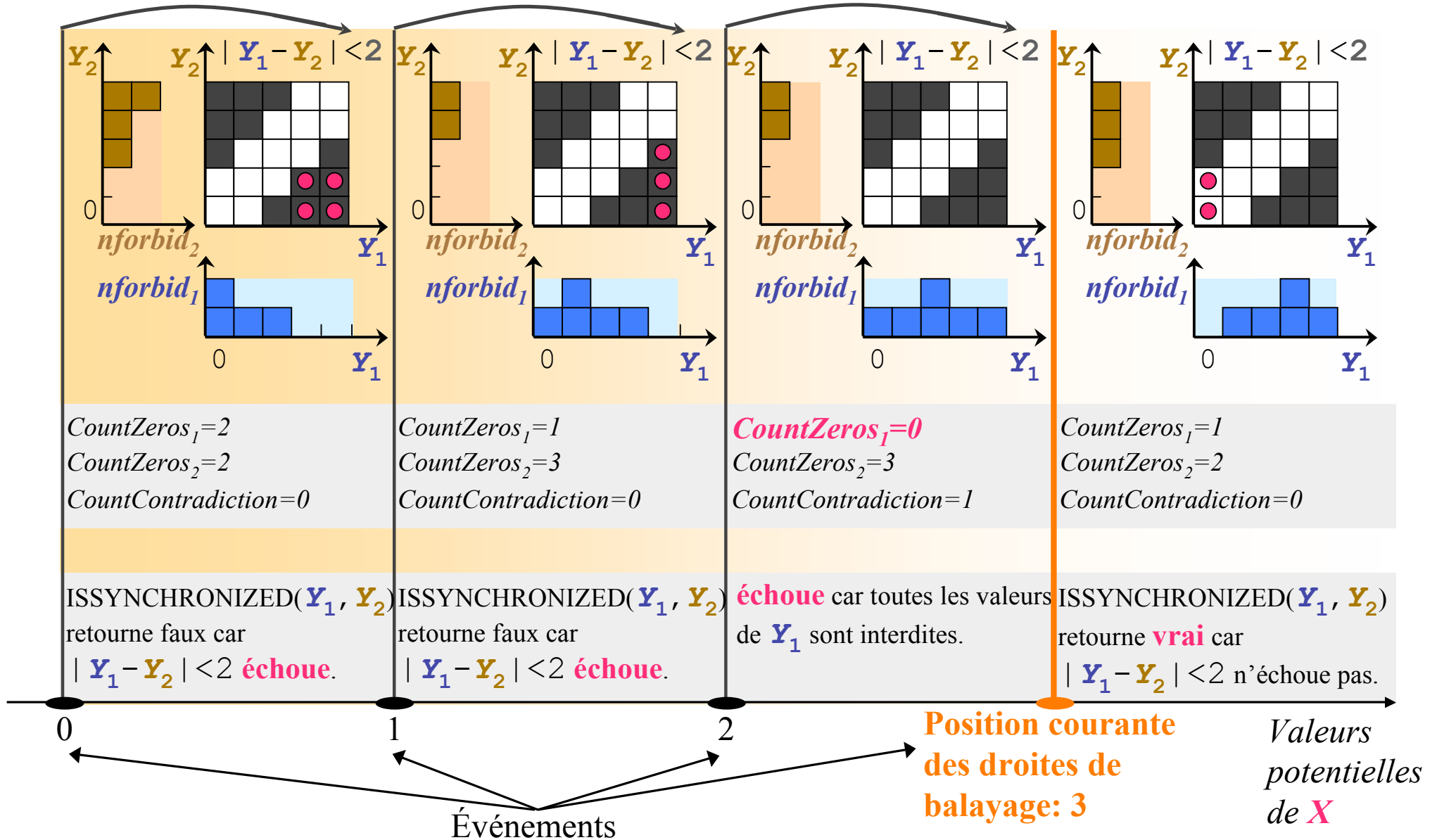
$$Y2 + 1 \leq U \vee U + 4 \leq Y2$$

$$|Y1 - Y2| < 2$$



Droites de balayage pas "synchronisées"

Synchronisation de Droites de Balayage: Exemple



↳ **ORDONNANCEMENT AVEC CONTRAINTES DE CAPACITÉ**

↳ **GÉOMETRIE**

• **Utilisation de la méthode de balayage pour différents patrons**

– Une conjonction de contraintes ayant deux variables en commun

– Synchronisation de plusieurs conjonctions de contraintes

– **L'opérateur de cardinalité**

• **Contrainte de non-recoupement**

↳ **OPTIMISATION**

↳ **AUTOMATES POUR LA CONCEPTION D'ALGORITHMES DE FILTRAGE**

Le Cas de l'Opérateur de *cardinalité*

L'**opérateur de *cardinalité***, P. VAN HENTENRYCK, Y. DEVILLE, (ICLP 1991) :

- **Définition:** $C = \sum_{j=1}^n \#CTR_j(V_1, \dots, V_{m_j})$
- **Élagage** : basé sur $\left\{ \begin{array}{l} \text{comptage} \\ \text{«entailment»} \end{array} \right.$

Un cas **structuré** de l'opérateur de *cardinalité* : [BELDICEANU & CARLSSON 01c]

- **Restriction:** 2 variables **X** et **Y** apparaissent dans chaque CTR_j
- **Élagage** : $\left\{ \begin{array}{l} \text{considère l'interaction entre les contraintes 😊} \\ \text{(à travers leurs variables partagées)} \end{array} \right.$

APPLICATION: **relaxation** de la contrainte de non-recoupement de rectangles

Dualité Régions **Interdites** et Régions **Sures**

DEFINITION région **interdite** par rapport à une contrainte **Ctr** et 2 variables **X, Y** de Ctr :

Deux intervalles $\text{inf}_x..sup_x$ et $\text{inf}_y..sup_y$ tels que:

$$\forall x \in \text{inf}_x..sup_x,$$

$\forall y \in \text{inf}_y..sup_y$: **Ctr** avec l'affectation **X=x, Y=y** est **fausse**.

DEFINITION région **sure** par rapport à une contrainte **Ctr** et 2 variables **X, Y** de Ctr :

Deux intervalles $\text{inf}_x..sup_x$ et $\text{inf}_y..sup_y$ tels que:

$$\forall x \in \text{inf}_x..sup_x,$$

$\forall y \in \text{inf}_y..sup_y$: **Ctr** avec l'affectation **X=x, Y=y** est **vraie**.

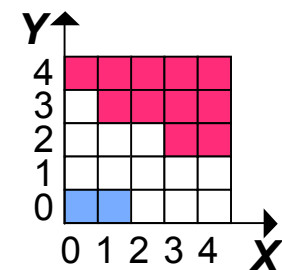
EXEMPLE

$$0 \leq X \leq 4$$

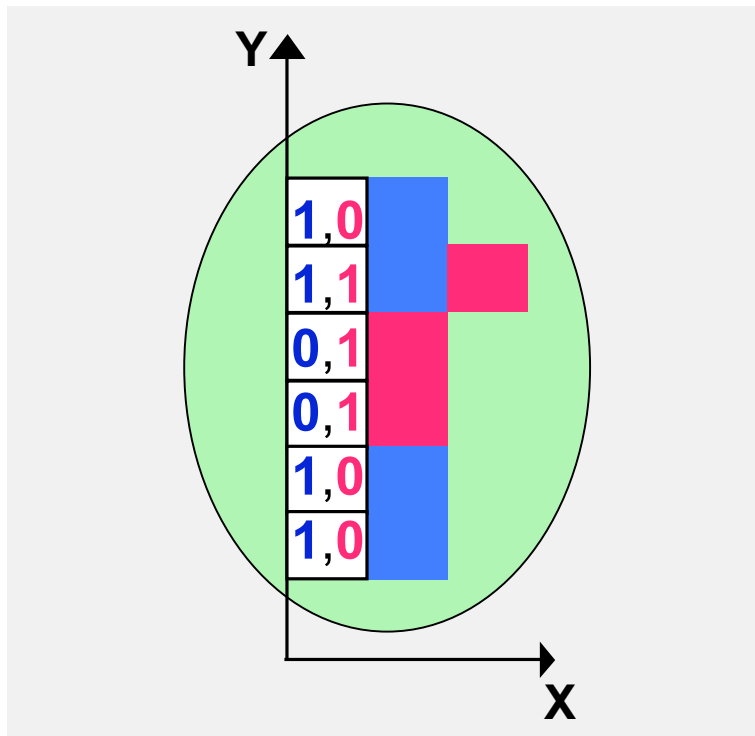
$$0 \leq Y \leq 4$$

$$1 \leq S \leq 6$$

$$X + 2Y \leq S$$

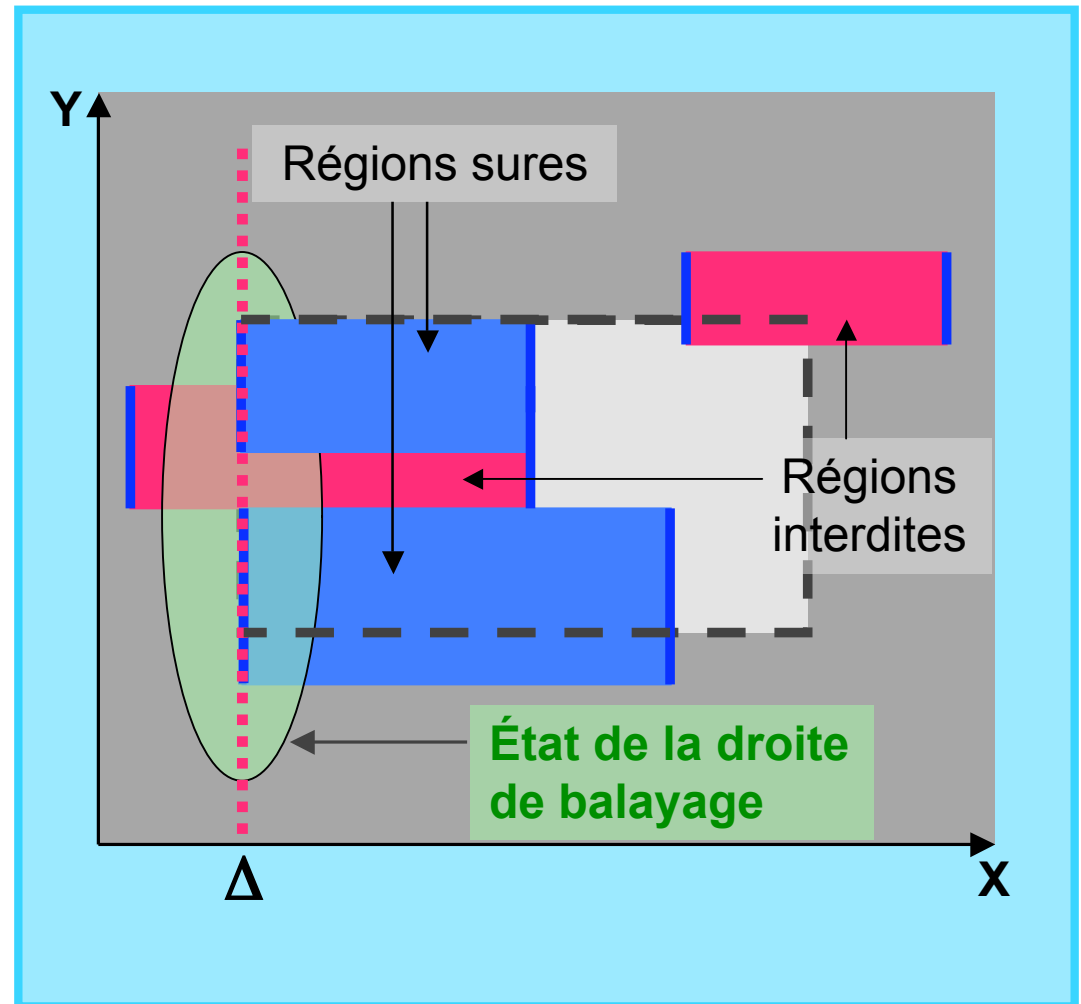


Status de la Droite de Balayage



Pour tout $y \in \text{dom}(Y)$:

- **nombre** de régions **sure**s
 - **nombre** de régions **interdites**
- contenant le point (Δ, y)



Enlève une valeur $\Delta \in \text{dom}(X)$ si
pour tout $y \in \text{dom}(Y)$:

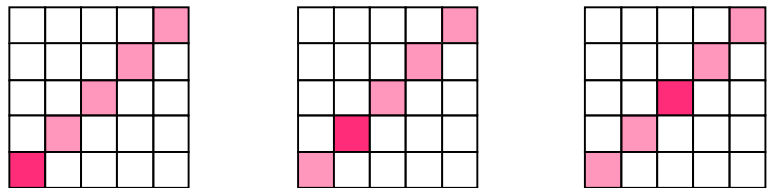
$$n_{\text{safe}}[y]..n_{\text{ctr}} - n_{\text{forbidden}}[y] \cap C = \emptyset$$

Exemple

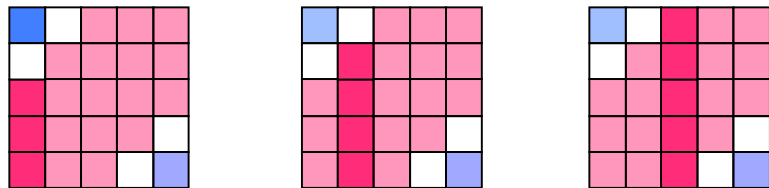
PROBLÈME:

Ajuste minimum de **X**
par rapport à **Y** et au fait
que **4 ou 5 constraints**
doivent être **satisfaites**:

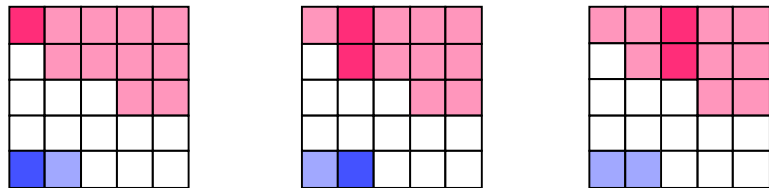
- $0 \leq X \leq 4$ $0 \leq Y \leq 4$ $2 \leq Z \leq 3$
- $1 \leq S \leq 6$ $0 \leq T \leq 0$ $1 \leq U \leq 2$
- alldifferent({**X**,**Y**,**R**})
- $|X - Y| > Z$
- $X + 2Y \leq S$
- $X + 3 \leq T \vee T + 1 \leq X \vee$
- $Y + 3 \leq U \vee U + 4 \leq Y$
- $X + Y \equiv 0 \pmod{2}$



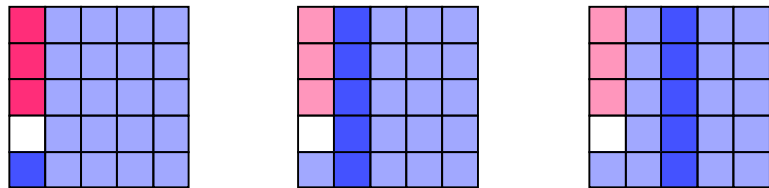
alldifferent({**X**,**Y**,**R**})



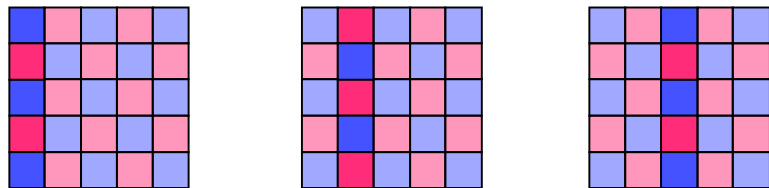
$|X - Y| > Z$



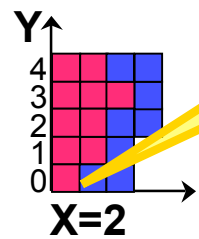
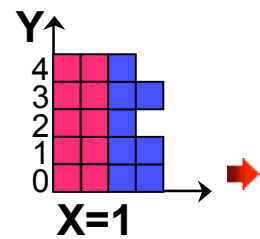
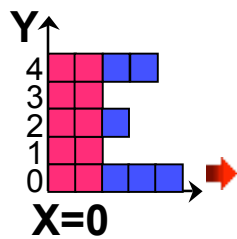
$X + 2Y \leq S$



$X + 1 \leq T \vee T + 1 \leq X \vee$
 $Y + 1 \leq U \vee U + 4 \leq Y$



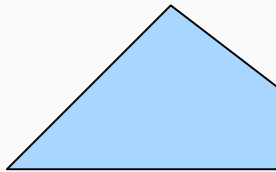
$X + Y \equiv 0 \pmod{2}$



Déduction:
X > 1

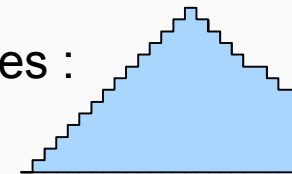
- ↳ ORDONNANCEMENT AVEC CONTRAINTES DE CAPACITÉ
- ↳ **GÉOMETRIE**
 - Utilisation de la méthode de balayage pour différents patrons
 - ↳ **Contrainte de non-recoupement**
- ↳ OPTIMISATION
- ↳ AUTOMATES POUR LA CONCEPTION D'ALGORITHMES DE FILTRAGE

Motivation initiale: [CHAMARD, DECÈS & FISCHLER 94], fabrication d'ailes d'avions

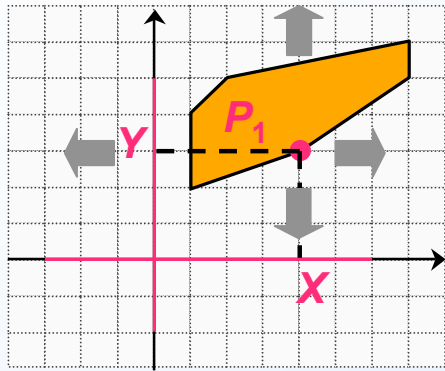


La discrétisation en termes de rectangles :

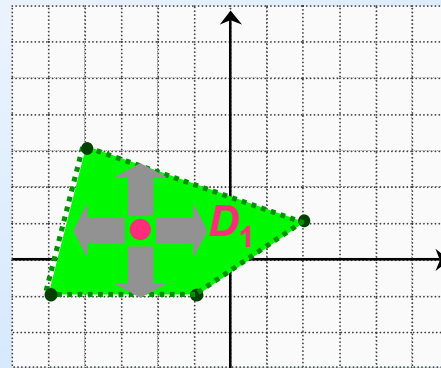
- **augmente** la mémoire requise
- **casse** la déduction



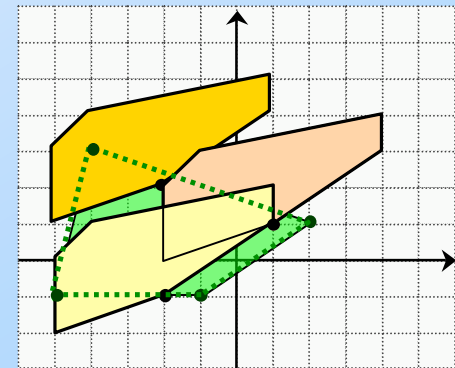
Famille de Polygones



Polygone P_1 avec une origine de coordonnées X, Y

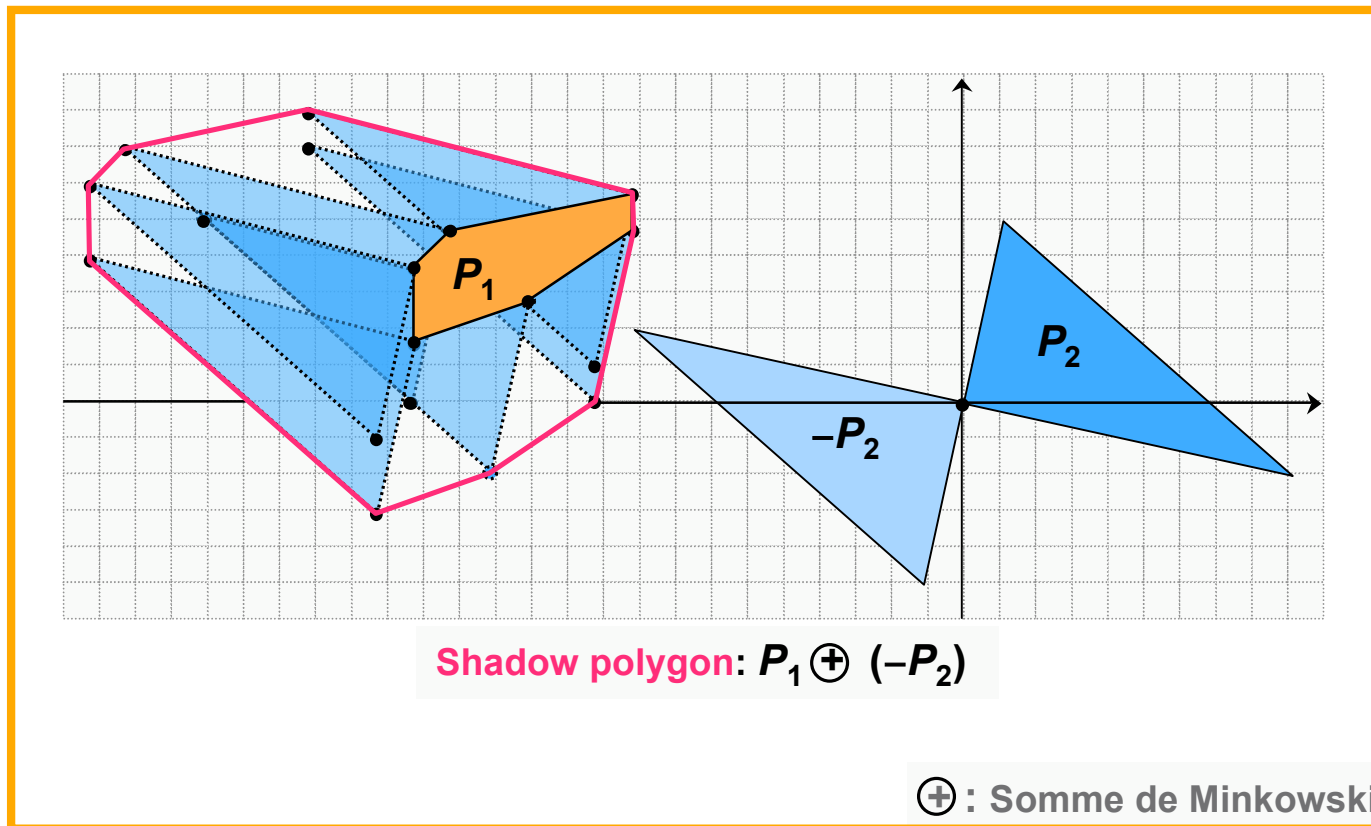


Polygone D_1 domaine de l'origine de P_1



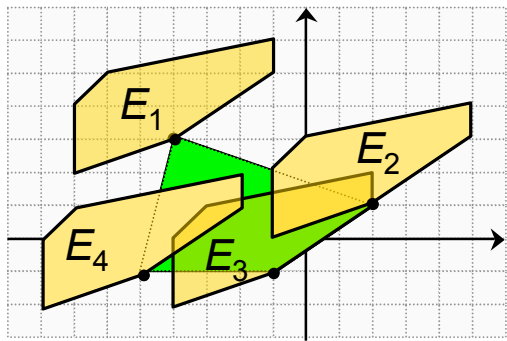
Translations de P_1 avec son origine dans D_1

Shadow Polygon

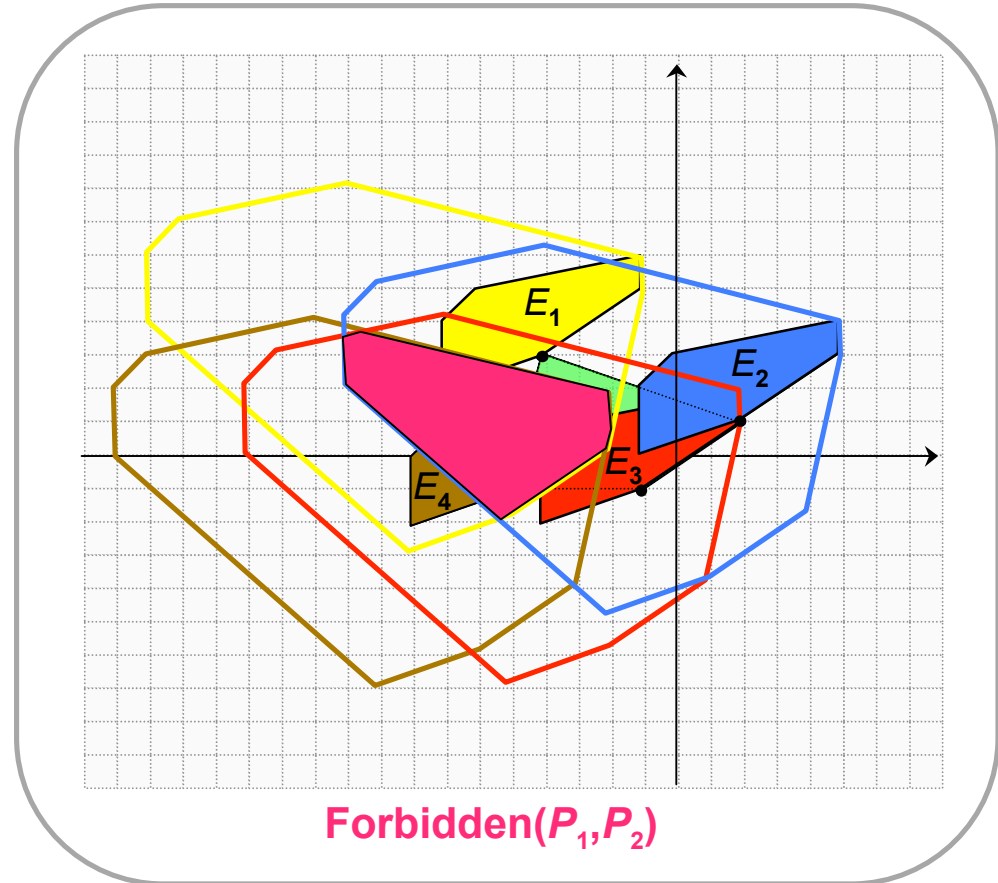


Si l'origine de P_2 est située **dans** le shadow polygon alors P_2 coupe P_1
 Si l'origine de P_2 est située **en dehors** du shadow polygon alors P_2 ne coupe pas P_1

Polygone Interdit



Polygones **extremum** de P_1
par rapport au domaine D_1



Si l'origine de P_2 est située **dans** le polygone interdit alors P_2 coupe toute translation de P_1

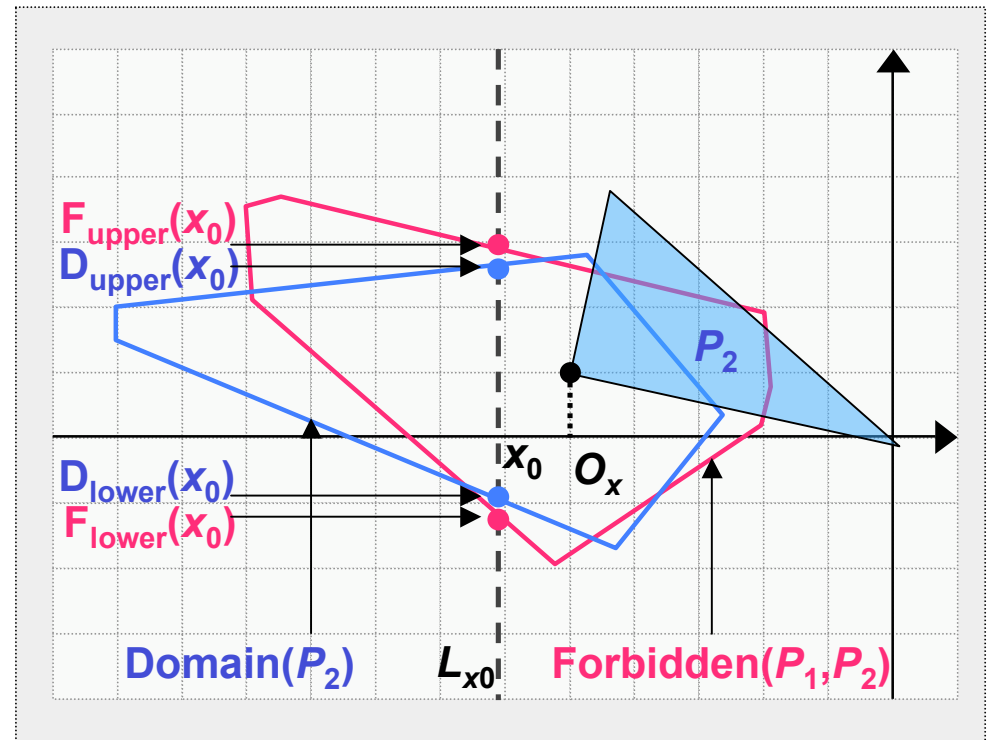
Élage les coordonnées de l'origine de P_2
avec un algorithme de **balayage**

Condition d'Élagage

BUT: élague la coordonnée O_x sur l'axe des x de l'origine de P_2 de manière à empêcher l'origine d'être localisée dans $\text{Forbidden}(P_1, P_2)$

CONDITION NÉCESSAIRE pour enlever x_0 de O_x :

$(\text{Domain}(P_2) \setminus \text{Forbidden}(P_1, P_2)) \cap L_{x_0}$
ne contient aucun point de
coordonnées entières.



↳ ORDONNANCEMENT AVEC CONTRAINTES DE CAPACITÉ

↳ GÉOMETRIE

 OPTIMISATION

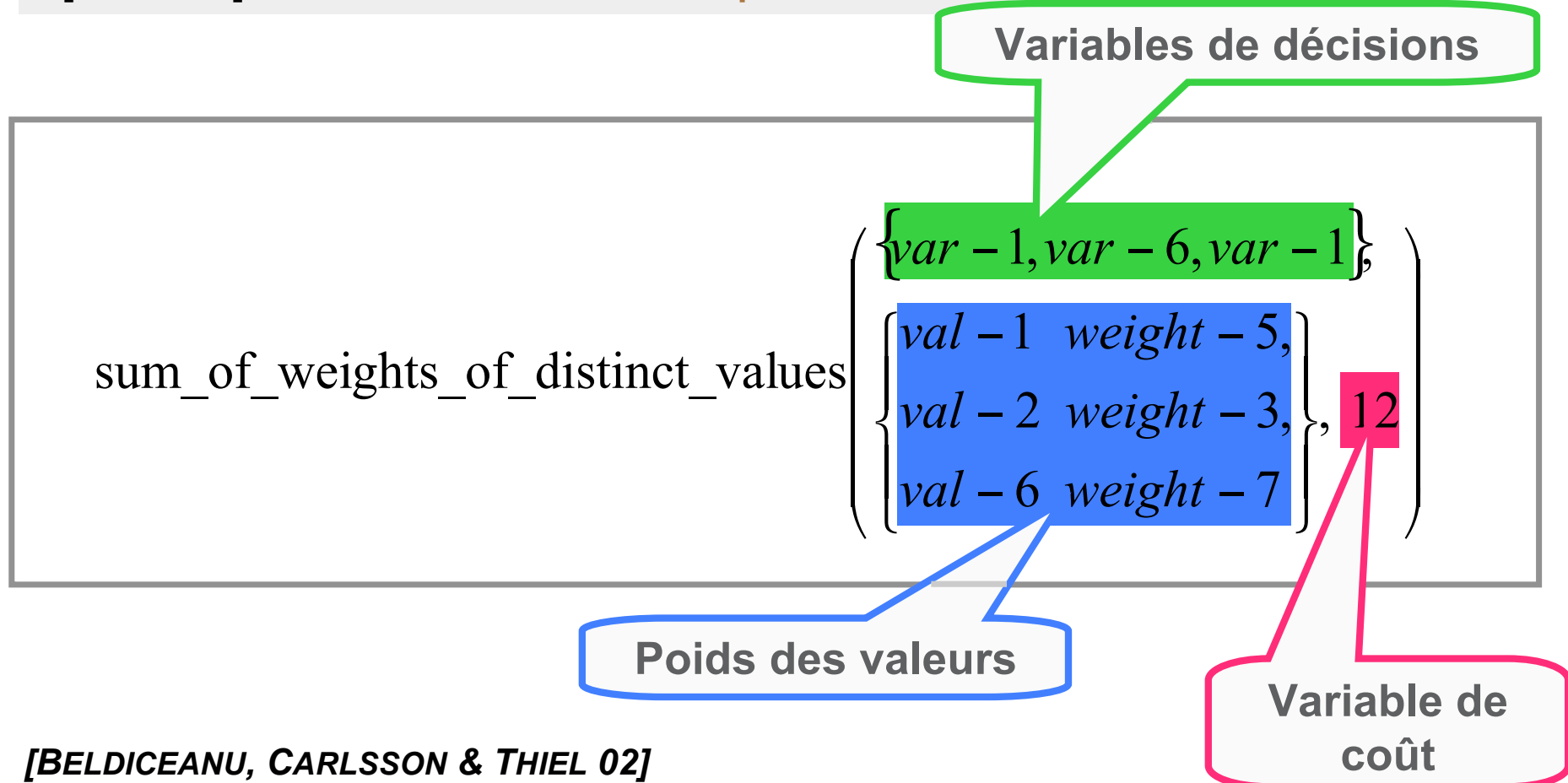
• *«somme des poids associés aux valeurs distinctes»*

↳ AUTOMATES POUR LA CONCEPTION D'ALGORITHMES DE FILTRAGE

Une Nouvelle Contrainte d'Optimisation

- [CASEAU & LABURTHE 97a]
- [FOCCACI, LODI & MILANO 99]
- [BAPTISTE, LE PAPE & PERIDY 98]
- [FAHLE 02]

voyageur de commerce
 problème d'affectation pondéré
 relaxation de contraintes disjonctives
 clique maximum



[BELDICEANU, CARLSSON & THIEL 02]

Exemple d'Utilisation

PROBLÈME

Contrôle de l'échiquier par des reines

29	23	35	46	29	35	23	46
35	29	35	12	29	46	46	29
46	35	29	35	29	46	29	46
35	35	35	29	29	29	35	35
29	29	29	29	29	29	29	29
35	23	35	29	29	29	23	23
12	46	29	12	29	35	29	23
46	29	35	12	29	46	35	29

MODÈLE:

- Numéroté les cases de 1 à 64
- Créer une variable pour chaque case C (son domaine est constitué par les numéros des cases pouvant être attaquées à partir de C)
- Exprime **tout** le problème par **une seule** contrainte **swdv**

swdv([46,29,35,12,29,46,35,29,...,46],
[1-1,...,64-1],
5)

Autres utilisations: problèmes de **domination**,
relaxation pondérée de la contrainte *alldifferent*,
problèmes d'**affectation**.

Les Différents Algorithmes

swdv(**V**ariables, **V**alues, **C**ost)

ALGORITHMES DE FILTRAGE

Trouver les
BORNES de **Cost**

Borne inf. **LB**:
domination

Borne sup. **UB**:
couplage

*borne réalisable calculable
en $O(n \cdot \log n + m)$
(si pas de trous)*

PROPAGE des bornes
de **Cost** vers les **V**ariables

Enlève **val** de **var** ssi:
LB+**lower_regret**(var, val) > max(**Cost**)

Enlève **val** de **var** ssi:
UB-**upper_regret**(var, val) < min(**Cost**)

Propage à partir de
min(**Cost**) et max(**Cost**)

- ↳ ORDONNANCEMENT AVEC CONTRAINTES DE CAPACITÉ
- ↳ GÉOMETRIE
- ↳ OPTIMISATION
- ➔ **AUTOMATES POUR LA CONCEPTION D'ALGORITHMES DE FILTRAGE**
 - Contraintes d'ordre lexicographique

BUT: systématiser la construction d'algorithmes de filtrage corrects. 😊

ctr (

X11	X12	...	X1n
X21	X22	...	X2n
...
Xm1	Xm2	...	Xmn

)

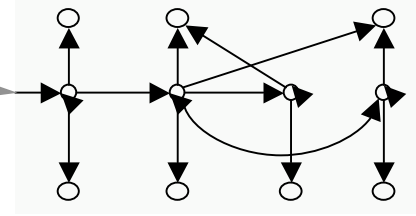
Signature de **ctr** par rapport à l'état courant des variables

S1 , S2 , ... , Sn , \$

Mots correspondant à des solutions de **ctr**

Mots correspondant à des non-solutions de **ctr**

Automate reconnaissant les solutions et non-solutions de **ctr**



Motivation et Définition de la Contrainte **Between**

$$\vec{x} = \langle x_0, \dots, x_{n-1} \rangle \leq_{lex} \vec{y} = \langle y_0, \dots, y_{n-1} \rangle :$$

$$\mathbf{V} \begin{cases} n = 0 \\ x_0 < y_0 \\ x_0 = y_0 \wedge \langle x_1, \dots, x_{n-1} \rangle \leq_{lex} \langle y_1, \dots, y_{n-1} \rangle \end{cases}$$

$$\langle 1, 2, 6, 9 \rangle \leq_{lex} \langle 1, 2, 7, 1 \rangle$$

Soit une suite de contraintes d'ordre lexicographique $\vec{x}_1 \leq_{lex} \vec{x}_2 \leq_{lex} \dots \leq_{lex} \vec{x}_m$
 Calcule une borne inférieure \vec{a}_i et supérieure \vec{b}_i **réalisables** pour chaque vecteur \vec{x}_i
 Se ramène à la résolution de contraintes **between**

Temps polynomial pour
 filtrage complet. 😊

$$\mathbf{between}(\vec{a}, \vec{x}, \vec{b}) :$$

$$\mathbf{V} \begin{cases} n = 0 \\ a_0 = x_0 = b_0 \wedge \vec{a}_{[1,n]} \leq_{lex} \vec{x}_{[1,n]} \leq_{lex} \vec{b}_{[1,n]} \\ a_0 = x_0 < b_0 \wedge \vec{a}_{[1,n]} \leq_{lex} \vec{x}_{[1,n]} \\ a_0 < x_0 = b_0 \wedge \vec{x}_{[1,n]} \leq_{lex} \vec{b}_{[1,n]} \\ a_0 < x_0 < b_0 \end{cases}$$

$$\mathbf{between}(\langle 4, 2, 6, 7 \rangle, \langle 4, 2, 6, 9 \rangle, \langle 4, 2, 7, 1 \rangle)$$

Between: Élagage

PROBLÉMATIQUE:

Élager X_0, X_1, X_2

en fonction de la

conjonction de

contraintes suivante :

$$C_1 : X_0 \in [0, 5]$$

$$C_2 : X_1 \in [0, 5]$$

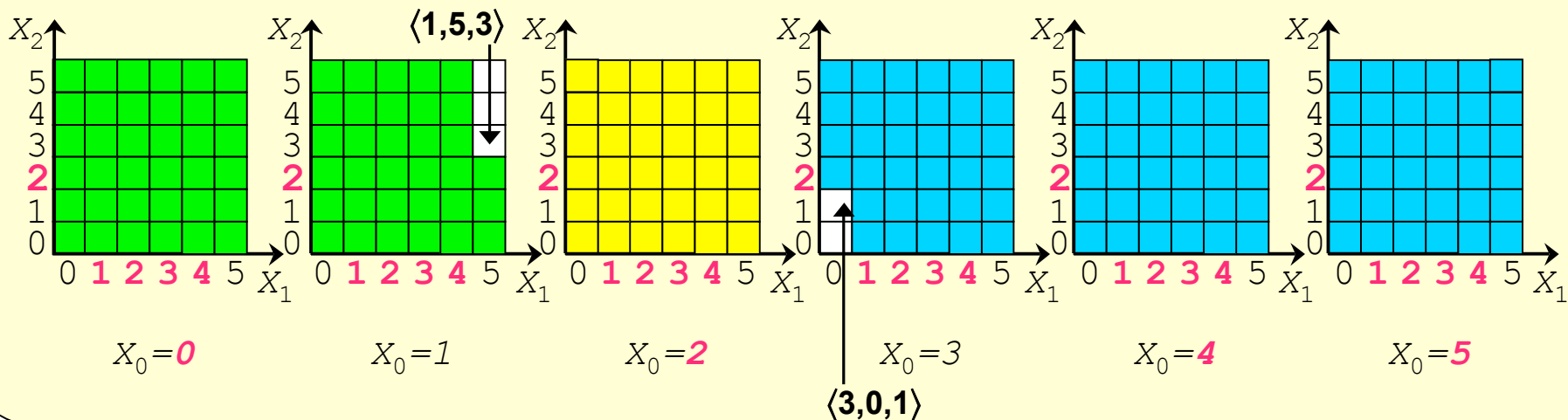
$$C_3 : X_2 \in [0, 5]$$

$$C_4 : \langle 1, 5, 3 \rangle \leq_{\text{lex}} \langle X_0, X_1, X_2 \rangle$$

$$C_5 : X_0 \neq 2$$

$$C_6 : \langle X_0, X_1, X_2 \rangle \leq_{\text{lex}} \langle 3, 0, 1 \rangle$$

DES TRIPLETS DE VALEURS INTERDITS VERS LES VALEURS INTERDITES



Between: signature

Soit A l'**alphabet** $\{ \boxed{<} , \boxed{<^{\wedge}} , \boxed{=} , \boxed{=^{\wedge}} , \boxed{>} , \boxed{>^{\wedge}} , \boxed{\$} \}$

La **signature** de $\text{between}(\vec{a}, \vec{x}, \vec{b})$ versus le "constraint store" est le **mot** S sur A (de longueur $n+1$), tel que:

$S_n = \boxed{\$}$, et pour $0 \leq i < n$:

si $a_i < b_i$: $S_i = \boxed{<}$ si $\Gamma \Rightarrow (x_i \leq a_i \vee x_i \geq b_i)$; $S_i = \boxed{<^{\wedge}}$ autrement.

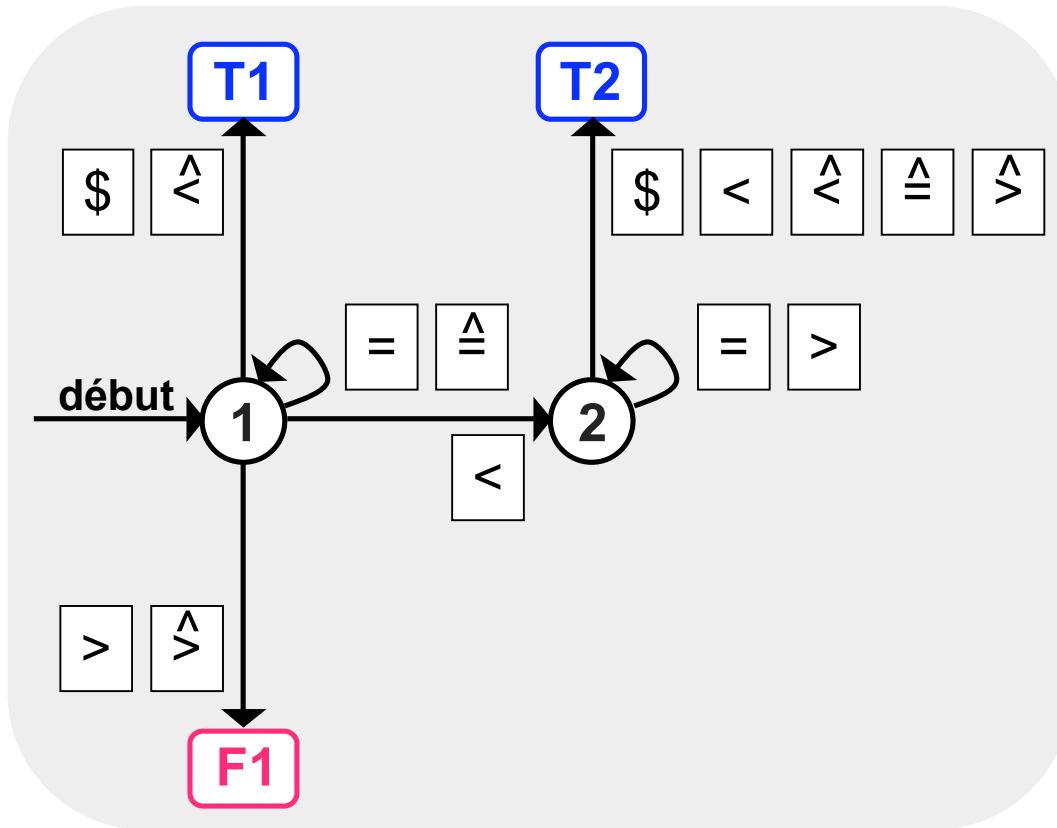
si $a_i = b_i$: $S_i = \boxed{=}$ si $\Gamma \Rightarrow (a_i = x_i = b_i)$; $S_i = \boxed{=^{\wedge}}$ autrement.

si $a_i > b_i$: $S_i = \boxed{>}$ si $\Gamma \Rightarrow (b_i \leq x_i \leq a_i)$; $S_i = \boxed{>^{\wedge}}$ autrement.

x_i ne coupe pas $[a_{i+1}, b_{i-1}]$

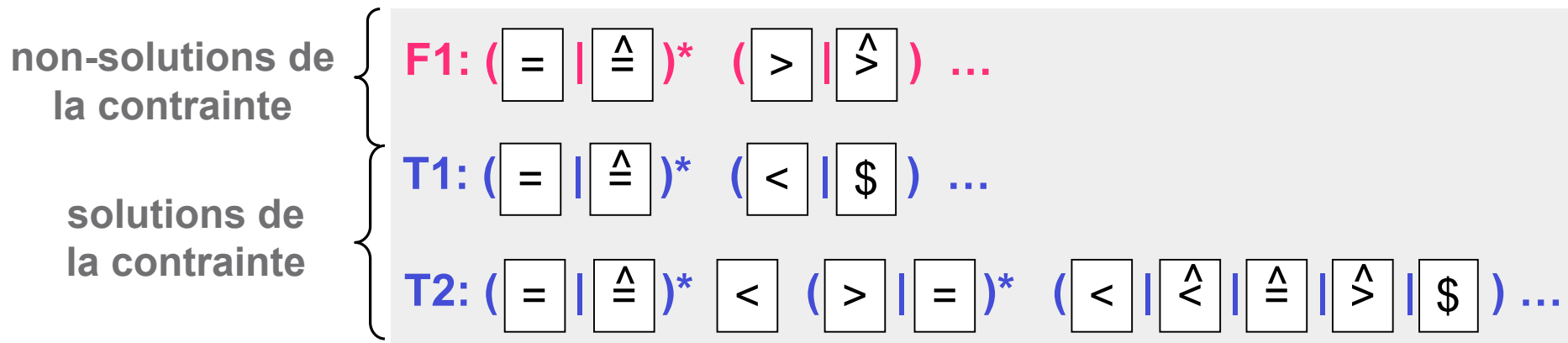
x_i entre b_i et a_i

**A
u
t
o
m
a
t
e**



Validité assurée car :

- Chaque état terminal ne reconnaît **que** des mots correspondant à des solutions (ou des non-solutions),
- Pour chaque état **toutes** les transitions sont envisagées.



IV. CONTRIBUTIONS SUR LES ASPECTS DÉCLARATIFS

- **CLASSIFICATION DES CONTRAINTES GLOBALES**
 - Les différents aspects des contraintes globales
 - Principe de classification
 - Un catalogue de contraintes globales

Des Catalogues de Problèmes vers un Catalogue de Contraintes Globales

- Les recueils traditionnels de problèmes ont les inconvénients suivants :
 - ◆ utilisent le langage naturel : **inexploitable** par un programme,
 - ◆ niveau de granularité trop important : pas de **décomposition**,
 - ◆ conduisent à des algorithmes efficaces mais **monolithiques**.
- Pour qu'un recueil de contraintes globales soit vraiment utile il est nécessaire d'avoir une **description explicite** du sens des contraintes globales.

BUT: avoir des programmes exploitant cette description.

Les **Différents Aspects** Associés aux Contraintes Globales

- **Filtrage** pour la résolution de contraintes,
- Génération d'**explications**,
- **Visualisation** de l'état de contraintes,
- Génération de **coupes** pour la programmation linéaire,
- Définition de **voisinages** pour la recherche locale.

Motivations d'une Classification des Contraintes Globales

- **Trouver** les **constituants** élémentaires des contraintes,
- **Classifier** les **propriétés** de chaque constituant,
- **Comprendre** comment ces propriétés **interagissent**.

Idée Clef de la Classification

Contrainte Globale vue comme :

Caractéristique de Graphe

sur **un Réseau Structuré**

de **Contraintes Élémentaires**

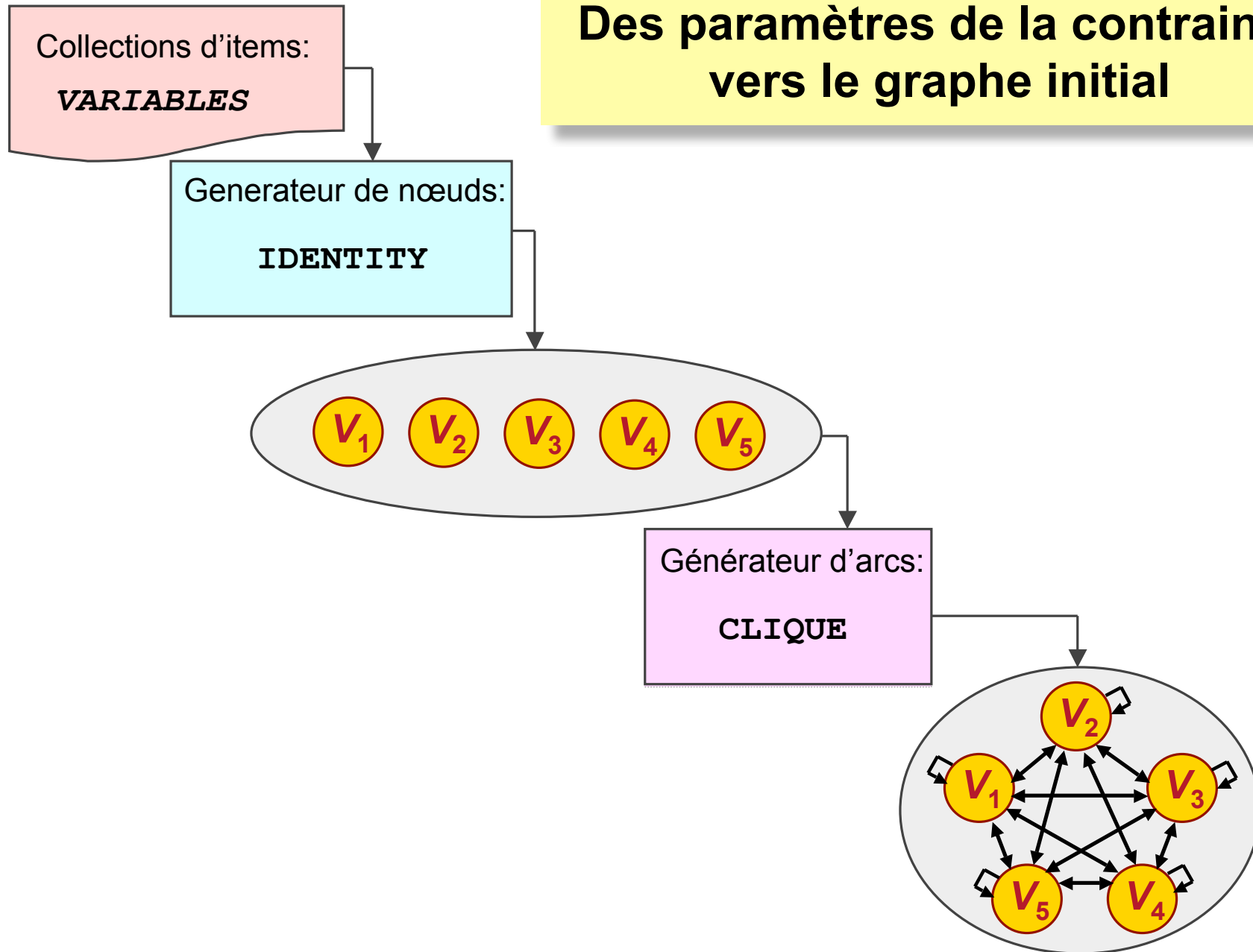
de **Même Type**

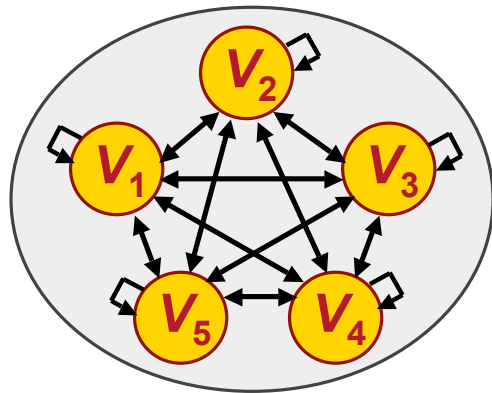
Carte d'identité de la contrainte `nvalue(NVAL, VARIABLES)`

• ARGUMENT	:	<i>NVAL</i>	:	<i>dvar</i>
				<i>VARIABLES</i> : <i>collection(var-dvar)</i>
• RESTRICTION(S)	:	<i>NVAL</i> ≥ 0		
		<i>NVAL</i> ≤ <i>VARIABLES</i>		
		<i>required(VARIABLES.var)</i>		
• VERTEX INPUT	:	<i>VARIABLES</i>		
• VERTEX GENERATOR	:	IDENTITY		
• EDGE INPUT	:	<i>VARIABLES</i>		
• EDGE GENERATOR	:	CLIQUE		
• EDGE ARITY	:	2		
• EDGE CONSTRAINT	:	<i>VARIABLES1.var = VARIABLES2.var</i>		
• GRAPH PROPERTY	:	<i>NSCC = NVAL</i>		

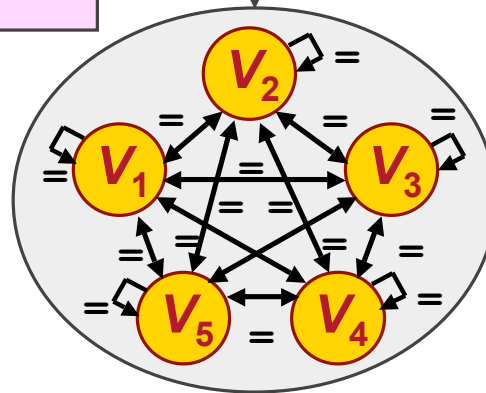
`nvalue(4, { var-3, var-1, var-7, var-1, var-6 })`

Des paramètres de la contrainte vers le graphe initial



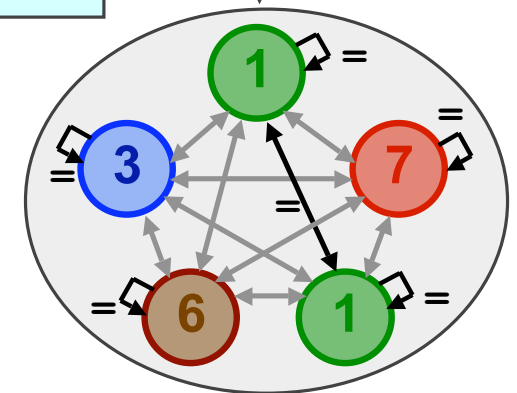


Contraintes sur
les arcs: =



Caractéristique sur
le graphe:

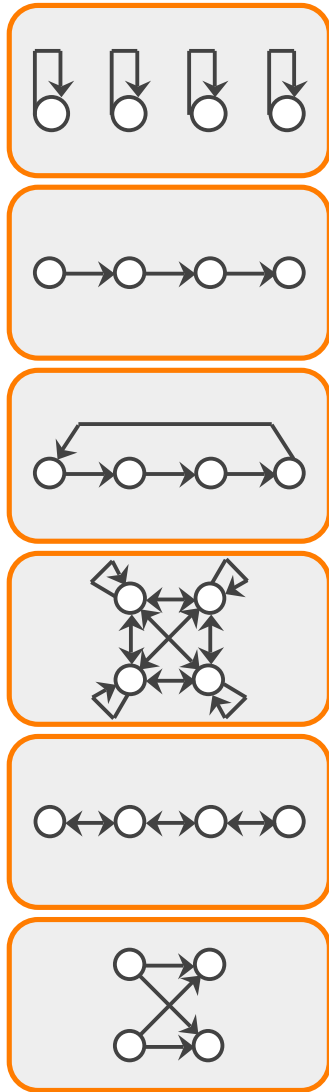
$NSCC=NVAL$



$nvalue(4, \{ var-3, var-1, var-7, var-1, var-6 \})$

**Caractéristique imposée
au graphe final**

Structure du graphe initial



loop
path
circuit
clique
chain
product

equivalence

transitive

reflexive

symetric

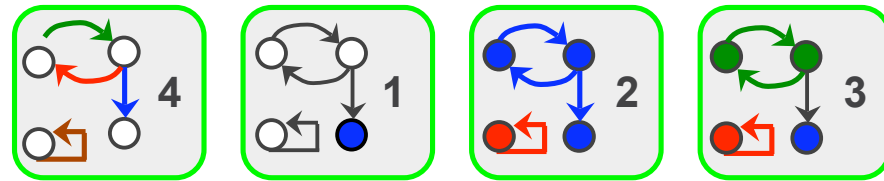
nvalue

nombre d'arcs
nombre de puits
nombre de
composantes

nombre de
composantes
connexes
nombre de
composantes
fortement
connexes

Propriétés des contraintes élémentaires

L'espace des Contraintes



Caractéristiques de graphes

Catalogue de Contraintes Globales

- Générateurs d'arcs
(une dizaine)
- Contraintes sur un arc
(contraintes arithmétiques et logiques de SICStus et contraintes globales)
- Caractéristiques de graphe
(une vingtaine)

😊 Décrit de **manière explicite** le sens de 160 contraintes globales

(*alldifferent*, *cumulative*, *cutset*, *cycle*, *diffn*, *element*, *global_cardinality*,
 [FAGES] [VAN HENTENRYCK] [LECONTE]

nvalue, *sort*, *stretch*, *strongly_connected*, *symmetric_alldifferent*, *symmetric_gcc*, ...)
 [OLDER] [PESANT] [ALTHAUS] [RÉGIN] [KOCJAN]

V. PERSPECTIVES (algorithmes spécifiques)

- **Algorithmes ad-hoc pour des contraintes basées sur:**
 - ◆ Structure de données efficaces
 - ◆ Géométrie algorithmique *(prise en compte de rotations)*
(interactions entre plus de 2 dimensions)
 - ◆ Algorithmique sur les mots *(contraintes sur les séquences)*
 - ◆ Algorithmes de graphes *(partitionnement de graphe)*
 - ◆ Automates *(contraintes d'ordre)*

Propager efficacement et encore propager !

V. PERSPECTIVES (classification de contraintes)

- Version **en ligne** du catalogue de contraintes (diffusion)
- **Méta évaluateur** de contraintes pour **tout** le catalogue (identification)
- **Méta filtrage** pour **toutes** les contraintes (utilisation concrète)
 - ◆ **Composition** de composants associés aux constituants élémentaires
 - ◆ **Invariants** de graphes
- Catalogue d'instances de **petits graphes** révélant des défauts de propagation (mise en évidence de cas limites intéressants)

*Comprendre les contraintes globales en termes des **propriétés** de leur **constituants***