Relaxation de Contraintes Globales : Mise en œuvre et Application

GREYC, CNRS UMR 6072
Université de Caen Basse-Normandie, France

Plan de la présentation

1. Contexte
   - CSPs et contraintes globales
   - Problèmes sur-contraints
2. Relaxation de contraintes globales
   - Relaxer une contrainte globale
3. Relaxation de `AllDifferent`
   - Représentation
   - Sémantique basée décomposition avec préférences
4. Relaxation de `Gcc`
   - La contrainte gcc

Problème de Satisfaction de Contraintes

Un *problème de satisfaction de contraintes* (CSP) est décrit par :

- $\mathcal{X}$: un ensemble fini de variables.
- $\mathcal{D}$: un ensemble de domaines qui contiennent l'ensemble fini de valeurs pouvant être prises par une variable.
- $\mathcal{C}$: un ensemble de contraintes restreignant les valeurs pouvant être prises par les variables.

Le but est de trouver une affectation pour chaque variable d'une valeur de son domaine de façon à ce que les contraintes soient toutes satisfaites.

Exemple de CSP

Variables	x_1, x_2, x_3
Domaines	$D(x_1) = \{1, 2\},\ D(x_2) = \{0, 1, 2, 3\},\ D(x_3) = \{2, 3\}$
Contraintes	$x_1 > x_2$
	$x_1 + x_2 = x_3$

CSP et Filtrage

Variables	x_1, x_2, x_3
Domaines	$D(x_1) = \{1, 2\},\ D(x_2) = \{0, 1\},\ D(x_3) = \{2, 3\}$
Contraintes	$x_1 > x_2$

CSP et Filtrage

Variables	x_1, x_2, x_3
Domaines	$D(x_1) = \{1, 2\},\ D(x_2) = \{0, 1\},\ D(x_3) = \{2, 3\}$
Contraintes	$x_1 > x_2$
	$x_1 + x_2 = x_3$

CSP et Filtrage

Variables	x_1, x_2, x_3
Domaines	$D(x_1) = \{1, 2\},\ D(x_2) = \{0, 1\},\ D(x_3) = \{2, 3\}$
Contraintes	$x_1 > x_2$
	$x_1 + x_2 = x_3$

Solution 1 : $x_1 = 2, x_2 = 0, x_3 = 2$

Solution 2 : $x_1 = 2, x_2 = 1, x_3 = 3$

Solution 3 : $x_1 = 1, x_2 = 1, x_3 = 2$

CSP et Filtrage

Variables	x_1, x_2, x_3
Domaines	$D(x_1) = \{2\},\ D(x_2) = \{0, 1\},\ D(x_3) = \{2, 3\}$
Contraintes	$x_1 > x_2$
	$x_1 + x_2 = x_3$
	$x_1 \neq x_2$

CSP et Filtrage

Variables	x_1, x_2, x_3
Domaines	$D(x_1) = \{2\},\ D(x_2) = \{0, 1\},\ D(x_3) = \{2, 3\}$
Contraintes	$x_1 > x_2$
	$x_1 + x_2 = x_3$
	$x_1 \neq x_2$

Solution 1 : $x_1 = 2, x_2 = 0, x_3 = 2$

Solution 2 : $x_1 = 2, x_2 = 1, x_3 = 3$

Contraintes globales

Contrainte globale

Une contrainte globale agglomère un ensemble de propriétés que doivent vérifier *n* variables.

- plus agréable de modéliser par une seule contrainte ;
- filtrage plus efficace.

Un niveau consistance au moins aussi élevé que celui que l'on pourrait maintenir sur la conjonction des contraintes élémentaires.

Test de consistance et filtrage mis en œuvre à l'aide d'algorithmes issues d'autres domaines.

Consistance globale

Théorème (Consistance globale en CSP)

Une contrainte globale C portant sur les variables $x_1, \ldots, x_n$ de domaines respectifs $D_1, \ldots, D_n$ est dite globalement consistante si pour chaque variable x_i et chaque valeur $d_i \in D_i$ $(i = 1, ...n)$ il existe une valeur $d_j \in D_j$ pour tout $i \neq j$ tel que $(d_i, \ldots, d_n)$ vérifie la contrainte C.

AllDifferent

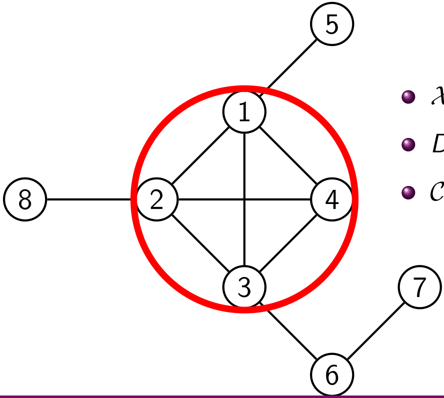
AllDifferent($\mathcal{X}$) [Régin, 1994]

La contrainte AllDifferent($\mathcal{X}$) admet une solution *ssi* il existe une instanciation complète telle que toutes les variables de $\mathcal{X}$ soient deux à deux différentes.

AllDifferent

`AllDifferent`($\mathcal{X}$) [Régin, 1994]

La contrainte `AllDifferent`($\mathcal{X}$) admet une solution *ssi* il existe une instanciation complète telle que toutes les variables de $\mathcal{X}$ soient deux à deux différentes.

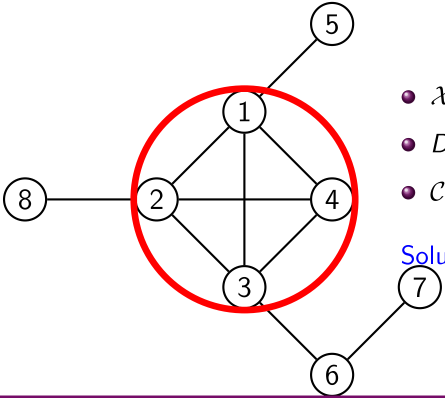


- $\mathcal{X}=\{X_1, X_2, X_3, X_4\}$
- $D_{X_1}=D_{X_2}=\{2,3\}, D_{X_3}=\{1,2,3\}, D_{X_4}=\{3,4\}$
- $\mathcal{C}=\{\texttt{AllDifferent}(\mathcal{X})\}$

AllDifferent

$\texttt{AllDifferent}(\mathcal{X})$ [Régin, 1994]

La contrainte $\texttt{AllDifferent}(\mathcal{X})$ admet une solution *ssi* il existe une instanciation complète telle que toutes les variables de $\mathcal{X}$ soient deux à deux différentes.



- $\mathcal{X}=\{X_1, X_2, X_3, X_4\}$
- $D_{X_1}=D_{X_2}=\{2,3\}, D_{X_3}=\{1,2,3\}, D_{X_4}=\{3,4\}$
- $\mathcal{C}=\{\texttt{AllDifferent}(\mathcal{X})\}$

Solution : $\{(X_1=2),(X_2,=3),(X_3=1),(X_4=4)\}$

Problèmes sur-contraints (1/2)

Beaucoup de problèmes sont sur-constraints, c'est-à-dire qu'il n'existe pas de solution satisfaisant toutes les contraintes.

La réponse d'un solveur CP : il n'existe aucune solution.

Comment trouver une solution satisfiable ?

- relaxer des contraintes du problème,
- chercher une affectation complète des variables du problème qui minimise les contraintes insatisfaites.

Problèmes sur-contraints (2/2)

Deux catégories de contraintes :

- **contraintes d'intégrité** : devant être satisfaites ;
- **contraintes de préférence** : pouvant être insatisfaites contre un coût.

solution ≡ instanciation complète satisfaisant les contraintes d'intégrité et **au mieux les contraintes de préférence**.

Au mieux ?

- minimiser l'agrégation des coûts des contraintes insatisfaites ;
- essayer de répartir la violation ;

Exemple

Soit le CSP sur-contraint ci-dessous :

$x_1 \in \{1, 2\}$, $x_2 \in \{2, 3\}$, $x_3 \in \{2, 3\}$

(i) $x_1 > x_2$

(ii) $x_1 + x_2 = x_3$

$(x_1, x_2, x_3) = (1, 2, 2)$ les deux contraintes sont violées

$(x_1, x_2, x_3) = (1, 2, 3)$ la contrainte (i) est violée $\leftarrow$ violation minimale

$(x_1, x_2, x_3) = (1, 3, 2)$ les deux contraintes sont violées

Weighted CSP [Larrosa & Schiex, 2003]

Un WCSP binaire est un CSP $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ muni d'une structure de valuation $\mathcal{S}(k)$, avec :

- $\mathcal{S}(k) = (E, +, >)$ avec $E = [0, \ldots, k]$ et $k \in [0, \ldots, \infty]$,
- À chaque contrainte c_{ij} est associé une fonction de coût f_{ij} définie sur $D_i \times D_j \rightarrow E$,
- Une contrainte d'arité nulle $c_{\emptyset}$ sert de minorant global.

Relaxation disjonctive [Petit *et al.*, 2000]

À chaque contrainte de préférence c_i :

- une variable de coût z_i est associée (l'insatisfaction de c_i). ;
- c_i est remplacée par :

$$[c_i \wedge z_i = 0] \vee [\overline{c_i} \wedge z_i > 0]$$

Définition (Sémantique de violation)

μ est une sémantique de violation pour une contrainte $c_i(X_1, .., X_n)$ ssi μ est une fonction de $D_{X_1} \times ... \times D_{X_n} \rightarrow \mathbb{R}^+$ telle que pour toute instanciation $\mathcal{A} \in D_{X_1} \times ... \times D_{X_n}$, $\mu(\mathcal{A}) = 0$ si c_i est satisfaite.

La valeur de z_i dépend de la sémantique de violation associée à c_i.

Relaxation disjonctive [Petit *et al.*, 2000]

À chaque contrainte de préférence c_i :

- une variable de coût z_i est associée (l'insatisfaction de c_i). ;
- c_i est remplacée par :

$$[c_i \wedge z_i = 0] \vee [\overline{c_i} \wedge z_i > 0]$$

Définition (Violation maximale)

Soit $\mathcal{S}$ l'ensemble de contraintes relaxables. Soit $c_i \in \mathcal{S}$ et soit z_i sa variable de coût.

- Soit $\mathcal{Z}$ la variable représentant la violation totale, alors $\mathcal{Z} = \sum_{c_i \in \mathcal{S}} z_i$
- Trouver une affectation complète $\mathcal{A}$ t.q $\mu(\mathcal{A}) \leq \max(D_Z)$

Relaxation disjonctive [Petit *et al.*, 2000]

Exemple de relaxation disjonctive

Soit le CSP $P{=}(\mathcal{X}, \mathcal{D}, \mathcal{C})$ tel que :

- $\mathcal{X} = \{X_1, X_2, X_3\}$,
- $\mathcal{D} = \{D_{X_1} = \{2,3\}, D_{X_2} = \{1,2\}, D_{X_3} = \{1,2,3\}\}$,
- $\mathcal{C} = \{c_1 = [X_1 > X_2], c_2 = [X_1 = X_3], c_3 = [X_3 > X_2]\}$

On suppose que :

- c_1 est une contrainte d'intégrité.
- c_2 et c_3 sont des contraintes de préférence.
- La sémantique de violation de c_2 est la distance entre les valeurs de ses deux variables.
- La sémantique de violation de c_3 est le carré de l'écart entre les valeurs de ses deux variables.

Relaxation disjonctive [Petit *et al.*, 2000]

Exemple de relaxation disjonctive

La relaxation disjonctive de P est le réseau de contrainte $P^{'}=(\mathcal{X}^{'},\mathcal{D}^{'},\mathcal{C}^{'})$ tel que :

- $\mathcal{X}^{'} = \mathcal{X} \cup \{z_1, z_2, Z\}$
- $\mathcal{D}^{'} = \mathcal{D} \cup \{D_{z_1}, D_{z_2}, D_Z\}$
- $\mathcal{C}^{'} = \{X_1 > X_2,$
  $[X_1 = X_3 \wedge z_1 = 0] \vee [X_1 \neq X_3 \wedge z_1 = |X_1 - X_3|],$
  $[X_3 > X_2 \wedge z_2 = 0] \vee [X_3 \leq X_2 \wedge z_2 = (X_3 - X_2)^2]$
  $Z = z_1 + z_2$

Remarque : Cette forme générale est difficilement exploitable en raison des disjonctions apparues.

Contraintes globales relaxées

Relaxer une contrainte globale

Pour relaxer la contrainte globale, il est nécessaire de tenir compte de la :

- Sémantique de la contrainte : que signifie la contrainte ?
- Sémantique de violation : comment quantifier la violation ?

Une même contrainte peut être relaxée suivant plusieurs sémantiques.

Sémantiques de violation (1/2)

Sémantique de violation basée variables [Petit *et al.*, 2001]

Mesure le nombre de variables à ré-instancier pour satisfaire la contrainte globale (μ_{var}).

Pour la contrainte `AllDifferent` :

$x_1 \in \{a, b\}$,$x_2 \in \{a, b\}$
$x_3 \in \{a, b\}$, $x_4 \in \{b, c\}$
AllDifferent(x_1,x_2,x_3,x_4)

x_1	x_2	x_3	x_4	μ_{var}
a	a	a	b	2
a	a	b	b	2
a	a	b	c	1
a	b	a	b	2
a	b	a	c	1
...	...	...	...	...

Sémantiques de violation (1/2)

Sémantique de violation basée variables [Petit *et al.*, 2001]

Mesure le nombre de variables à ré-instancier pour satisfaire la contrainte globale (μ_{var}).

Contraintes relaxées selon cette sémantique :

- SoftAllDifferent(μ_{var}) [Petit *et al.*,2001] [Hoeve, 2004] ;
- SoftGcc(μ_{var}) [Hoeve *et al.*, 2006] ;
- Σ-Gcc(μ_{var}) [Métivier *et al.*, 2009] ;
- SoftRegular(μ_{var}) [Hoeve *et al.*, 2006] ;
- Σ-Regular(μ_{var}) [Métivier *et al.*, 2009] ;

Sémantiques de violation (2/2)

Sémantique de violation basée décomposition [Petit *et al.*, 2001]

Mesure le nombre de contraintes élémentaires insatisfaites (issues d'une décomposition de la contrainte globale) (μ_{dec}).

Pour la contrainte `AllDifferent` :

$x_1 \in \{a, b\}, x_2 \in \{a, b\}$
$x_3 \in \{a, b\}, x_4 \in \{b, c\}$
AllDifferent(x_1, x_2, x_3, x_4)

$\Downarrow$

$x_1 \neq x_2, x_1 \neq x_3, x_1 \neq x_4$
$x_2 \neq x_3, x_2 \neq x_4, x_3 \neq x_4$

x_1	x_2	x_3	x_4	μ_{var}	μ_{dec}
a	a	a	b	2	3
a	a	b	b	2	2
a	a	b	c	1	1
a	b	a	b	2	2
a	b	a	c	1	1
...	...	...	...	...	...

Sémantiques de violation (2/2)

Sémantique de violation basée décomposition [Petit *et al.*, 2001]

Mesure le nombre de contraintes élémentaires insatisfaites (issues d'une décomposition de la contrainte globale) (μ_{dec}).

Contraintes relaxées selon cette sémantique :

- `SoftAllDifferent`(μ_{dec}) [Hoeve, 2004] ;
- Σ-`AllDifferent`(μ_{dec}) [Métivier *et al.*, 2009] ;
- `SoftGcc`(μ_{val}) [Hoeve *et al.*, 2006] ;

Relaxation de AllDifferent

Représentation sous la forme d'un réseau

La contrainte `AllDifferent` peut être représentée par un réseau $\mathcal{R} = \{\mathcal{V}, \mathcal{A}\}$, avec

$\mathcal{V} = \mathcal{X} \cup D_X \cup \{s, t\}$ où s et t sont les sommets source et puit.

$\mathcal{A} = \mathcal{A}_s \cup \mathcal{A}_X \cup \mathcal{A}_t$ où :

- $\mathcal{A}_s = \{(s, x_i) \mid x_i \in \mathcal{X}\}$,
- $\mathcal{A}_X = \{(x_i, d_j) \mid d_j \in D_i, i = 1, ..., n\}$,
- $\mathcal{A}_t = \{(d_j, t) \mid d_j \in D_X\}$.

Représentation sous la forme d'un réseau

La contrainte `AllDifferent` peut être représentée par un réseau $\mathcal{R} = \{\mathcal{V}, \mathcal{A}\}$, avec

$\mathcal{V} = \mathcal{X} \cup D_X \cup \{s, t\}$ où s et t sont les sommets source et puit.

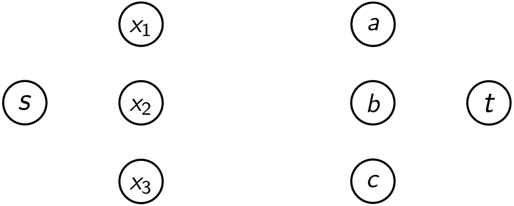
$\mathcal{A} = \mathcal{A}_s \cup \mathcal{A}_X \cup \mathcal{A}_t$ où :

- $\mathcal{A}_s = \{(s, x_i) \mid x_i \in \mathcal{X}\}$,
- $\mathcal{A}_X = \{(x_i, d_j) \mid d_j \in D_i, i = 1, ..., n\}$,
- $\mathcal{A}_t = \{(d_j, t) \mid d_j \in D_X\}$.

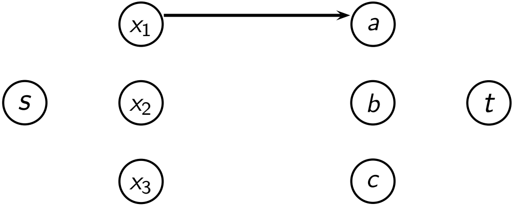
À chaque arc $a \in \mathcal{A}$ est associée une *capacité* $c(a) = 1$ et une *demande* $d(a)$ définie comme suit :

$$d(a) = \begin{cases} 1 & \text{si } a \in \mathcal{A}_s \\ 0 & \text{si } a \in \mathcal{A}_X \\ 0 & \text{si } a \in \mathcal{A}_t \end{cases}$$

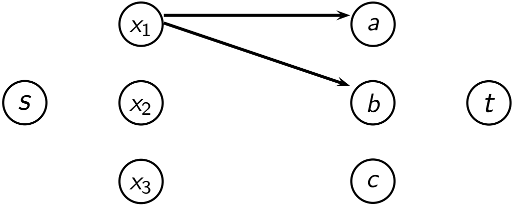
Représentation sous la forme d'un réseau



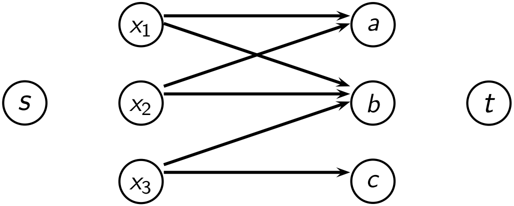
Représentation sous la forme d'un réseau



Représentation sous la forme d'un réseau

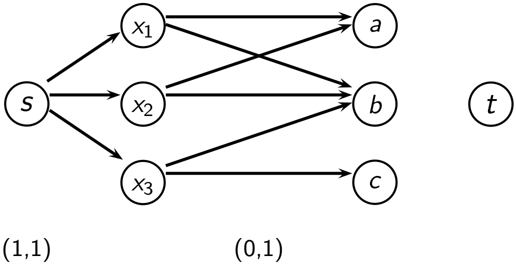


Représentation sous la forme d'un réseau

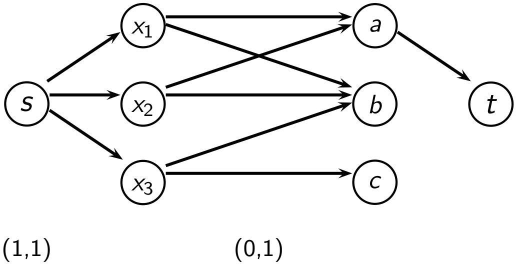


(0,1)

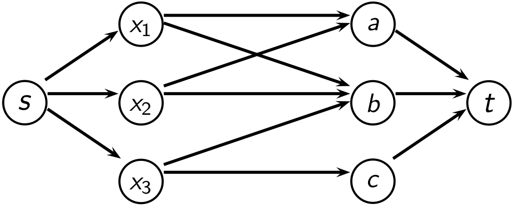
Représentation sous la forme d'un réseau



Représentation sous la forme d'un réseau

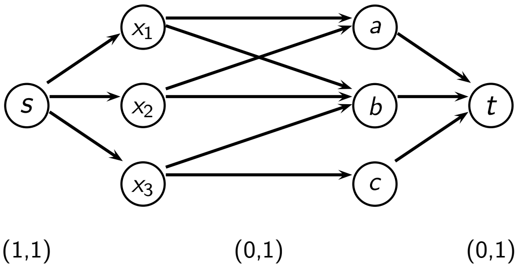


Représentation sous la forme d'un réseau



(1,1) (0,1) (0,1)

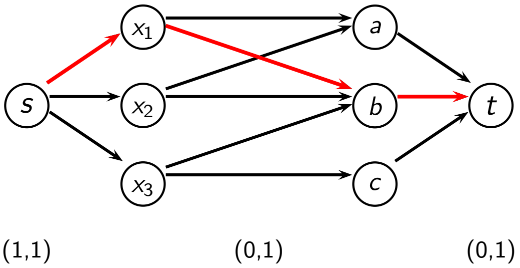
AllDifferent : Test de cohérence



Théorème (Cohérence de AllDifferent (Régin, 1996))

La contrainte AllDifferent *admet une solution* ssi *il existe, dans le réseau associé, un flot faisable f de valeur n.*

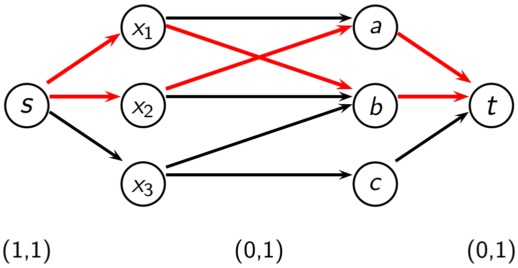
AllDifferent : Test de cohérence



Théorème (Cohérence de AllDifferent (Régin, 1996))

La contrainte AllDifferent *admet une solution* <u>*ssi*</u> *il existe, dans le réseau associé, un flot faisable f de valeur n.*

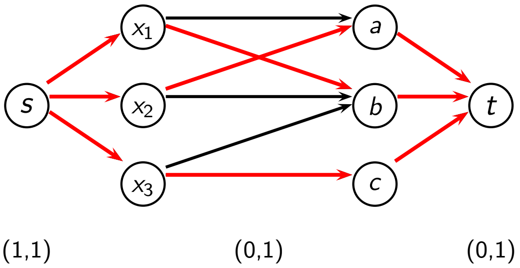
`AllDifferent` : Test de cohérence



Théorème (Cohérence de `AllDifferent` (Régin, 1996))

La contrainte `AllDifferent` *admet une solution* <u>*ssi*</u> *il existe, dans le réseau associé, un flot faisable f de valeur n.*

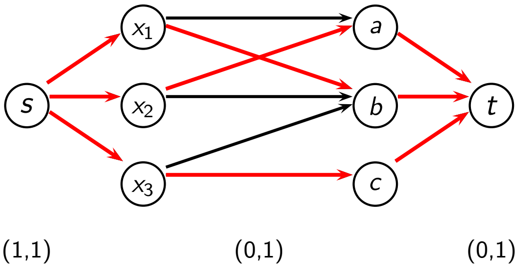
AllDifferent : Test de cohérence



Théorème (Cohérence de AllDifferent (Régin, 1996))

La contrainte AllDifferent *admet une solution* <u>*ssi*</u> *il existe, dans le réseau associé, un flot faisable f de valeur n.*

AllDifferent : Test de cohérence

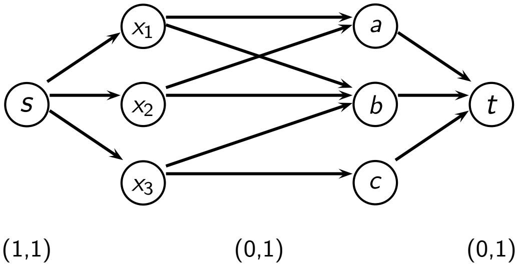


Théorème (Cohérence de AllDifferent (Régin, 1996))

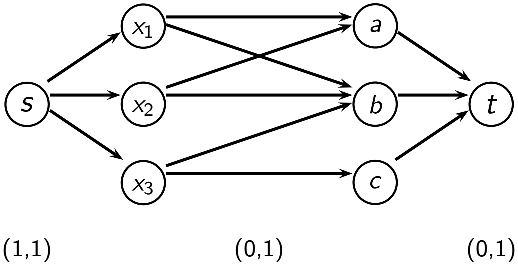
La contrainte `AllDifferent` *admet une solution* <u>*ssi*</u> *il existe, dans le réseau associé, un flot faisable f de valeur n.*

Le flot maximal (en rouge) correspond à la solution $\{x_1 = b, x_2 = a, x_3 = c\}$

AllDifferent : Filtrage



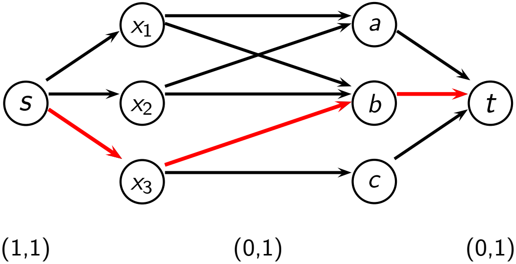
AllDifferent : Filtrage



Théorème (viabilité d'une valeur (Régin, 1996))

une valeur (x_j, d_j) est viable si et seulement si il existe un flot faisable f de valeur n passant par l'arc $(x_i \rightarrow d_j)$.

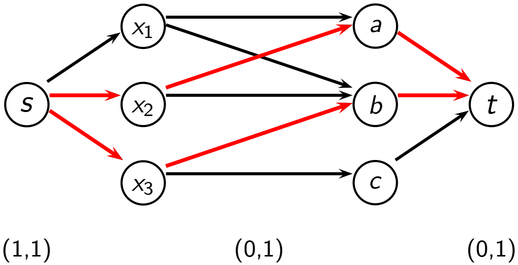
AllDifferent : Filtrage



Théorème (viabilité d'une valeur (Régin, 1996))

une valeur (x_j,d_j) est viable si et seulement si il existe un flot faisable f de valeur n passant par l'arc ($x_i \rightarrow d_j$).

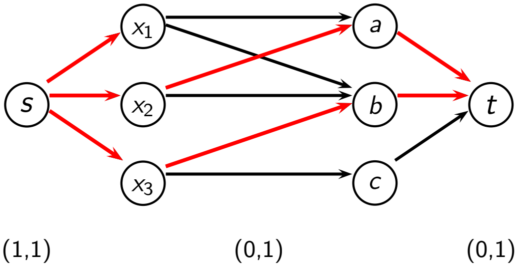
AllDifferent : Filtrage



Théorème (viabilité d'une valeur (Régin, 1996))

une valeur (x_j, d_j) est viable si et seulement si il existe un flot faisable f de valeur n passant par l'arc $(x_i \rightarrow d_j)$.

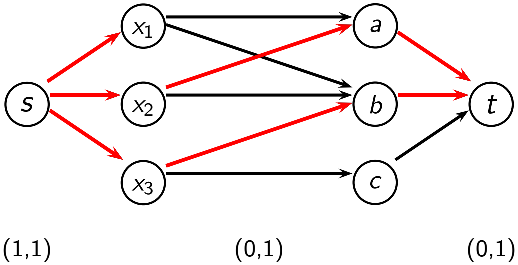
AllDifferent : Filtrage



Théorème (viabilité d'une valeur (Régin, 1996))

une valeur (x_j,d_j) est viable si et seulement si il existe un flot faisable f de valeur n passant par l'arc ($x_i \rightarrow d_j$).

AllDifferent : Filtrage



Théorème (viabilité d'une valeur (Régin, 1996))

une valeur (x_j,d_j) est viable si et seulement si il existe un flot faisable f de valeur n passant par l'arc ($x_i \rightarrow d_j$).

L'arête ($x_3 \rightarrow b$) ne peut apparaître que dans un flot de valeur 2 ; (x_3,b) est une valeur non viable.

Relaxation de AllDifferent

Exemple de CSP sur-contraint :

$x_1 \in \{a, b\}$, $x_2 \in \{a, b\}$

$x_3 \in \{a, b\}$, $x_4 \in \{b, c\}$

$\texttt{AllDifferent}(x_1, x_2, x_3, x_4)$

Autoriser certaines contraintes binaires à ne pas être satisfaites :

$x_1 \in \{a, b\}$, $x_2 \in \{a, b\}$

$x_3 \in \{a, b\}$, $x_4 \in \{b, c\}$

$z \in \{0, 1, 3, \ldots\}$

$\texttt{Soft-AllDifferent}(x_1, x_2, x_3, x_4, \mu, z)$

Sémantique basée variable

La mesure de violation basée variables μ_{var} détermine, pour une instanciation complète, le nombre de variables à ré-instancier pour satisfaire la contrainte `AllDifferent`($\mathcal{X}$).

Sémantique basée variable pour `AllDifferent`

$$\mu_{var}(\mathcal{X}) = \sum_{v_j \in D_X} max(|\{x_i \in \mathcal{X} \mid x_i = v_j\}| - 1, 0)$$

Sémantique basée décomposition

La mesure de violation basée décomposition μ_{dec} détermine, pour une instanciation complète, le nombre de contraintes binaires de différence insatisfaites.

Sémantique basée décomposition pour `AllDifferent`

$$\mu_{dec}(\mathcal{X}) = \mid \{(x_i, x_j) \mid x_i \in \mathcal{X}, x_j \in \mathcal{X} \text{ t.q. } i < j, x_i = x_j\} \mid$$

Arcs de violation

Arcs de violation [Hoeve, 05]

Pour les contraintes globales représentées par un réseau, il est possible *de mesurer et de quantifier la violation* par l'ajout d'*arcs de violation*.

$$\texttt{AllDifferent} \rightarrow \begin{cases} \text{représentation} = \text{réseau} \\ \text{solution} \equiv \text{flot faisable} \end{cases}$$

Au réseau de `AllDifferent`, on ajoute :

- des arcs de violation ;
- dépend de la sémantique utilisée.

Arcs de violation pour la mesure μ_{var}

Arcs de violation pour la mesure μ_{var} [Hoeve, 05]

$$\tilde{A}_t = \{(d_j \rightarrow t) \mid d_j \in D_X\}$$

avec

$$\forall a = (d_j \rightarrow t) \in \tilde{A}_t,\ d(a) = 0,\ c(a) = \delta^+(d_j) - 1 \text{ et } w(a) = 1$$

La quantité de flot circulant par un arc de violation $(d_j \rightarrow t)$ est égale au nombre de variables excédentaires ayant pour valeur d_j dans une instanciation. Comme le poids de cet arc est de 1, le poids du flot passant par cet arc reflète exactement le coût de violation engendré par les variables excédentaires.

Arcs de violation pour la mesure μ_{dec}

Arcs de violation pour la mesure μ_{dec} [Hoeve, 05]

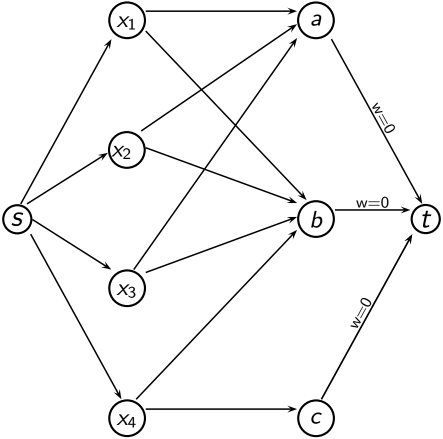
$$\tilde{A}_t = \{(d_j \rightarrow t)_i \mid d_j \in D_X, i = 1, ..., n\}$$

avec

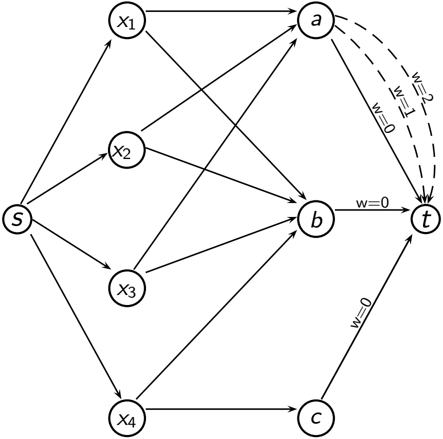
$$\forall a = (d_j \rightarrow t)_i \in \tilde{A}_t,\ d(a) = 0,\ c(a) = 1 \text{ et } w(a) = i$$

À chaque valeur $d_j \in D_X$ est associé $\delta^+(d_j) - 1$ arcs de violation. Chaque arc $a{=}(d_j \rightarrow t)_i$ (avec $1 \leq i \leq \delta^+(d_j) - 1$) possède un poids i. La quantité de violation associée à $d_j \in D_X$ correspond à la somme des poids des arcs entre d_j et t utilisés par f.

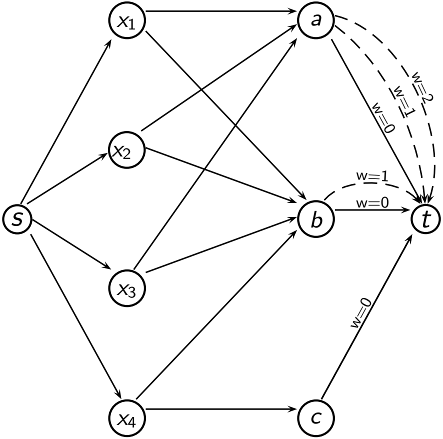
Arcs de violation pour la mesure μ_{dec}



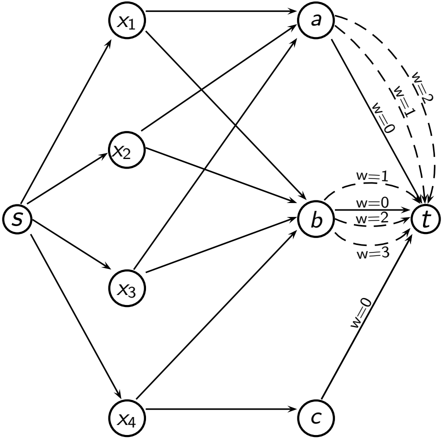
Arcs de violation pour la mesure μ_{dec}



Arcs de violation pour la mesure μ_{dec}



Arcs de violation pour la mesure μ_{dec}



Test de cohérence

Réutiliser le réseau de flots :

Test de cohérence

Réutiliser le réseau de flots :

Théorème (cohérence de Soft-AllDifferent(X, μ_{dec}, z))

La contrainte Soft-AllDifferent(X, μ_{dec}, z) admet une solution ssi il existe, dans le réseau associé, un flot faisable f de valeur n et de poids inférieur ou égal à $max(D_z)$.

Test de cohérence

Réutiliser le réseau de flots :

Théorème (cohérence de Soft-AllDifferent(X, μ_{dec}, z))

La contrainte Soft-AllDifferent(X, μ_{dec}, z) admet une solution ssi il existe, dans le réseau associé, un flot faisable f de valeur n et de poids inférieur ou égal à $max(D_z)$.

Exemple :

$x_1 \in \{a, b\}$, $x_2 \in \{a, b\}$, $x_3 \in \{a, b\}$, $x_4 \in \{b, c\}$, $z \in \{0, 1\}$

Soft-AllDifferent(x_1,x_2,x_3,x_4,z,μ_{dec})

Test de cohérence

Réutiliser le réseau de flots :

Théorème (cohérence de Soft-AllDifferent(X, μ_{dec}, z))

La contrainte Soft-AllDifferent(X, μ_{dec}, z) admet une solution ssi il existe, dans le réseau associé, un flot faisable f de valeur n et de poids inférieur ou égal à $max(D_z)$.

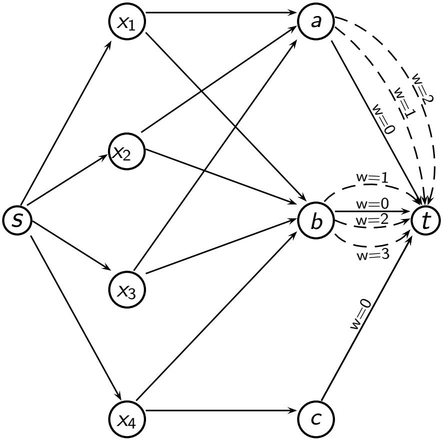
Exemple :

$x_1 \in \{a, b\}$, $x_2 \in \{a, b\}$, $x_3 \in \{a, b\}$, $x_4 \in \{b, c\}$, $z \in \{0, 1\}$

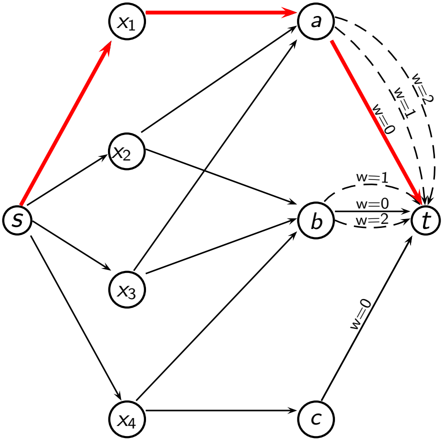
Soft-AllDifferent(x_1,x_2,x_3,x_4,z,μ_{dec})

- Pour l'instanciation $\{x_1 = a, x_2 = b, x_3 = b, x_4 = c\}$, il existe un (s, t)-flot de valeur 4, et de poids 1 ; elle donc solution.

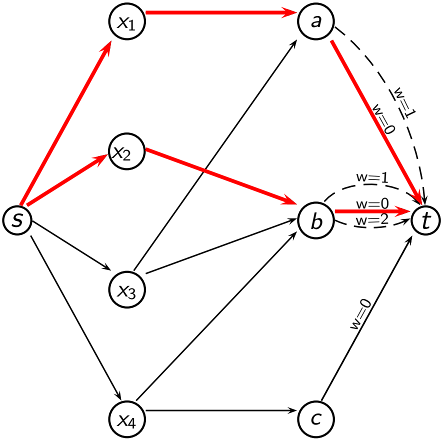
Test de cohérence



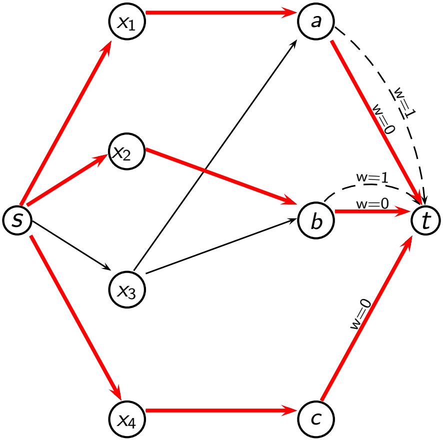
Test de cohérence



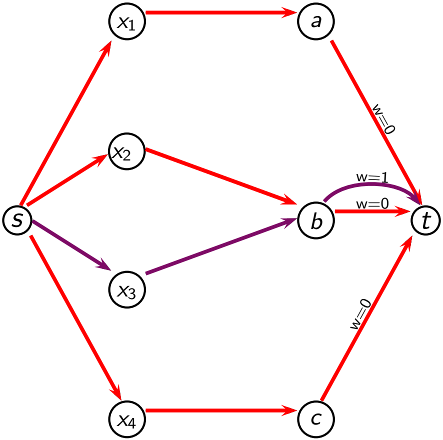
Test de cohérence



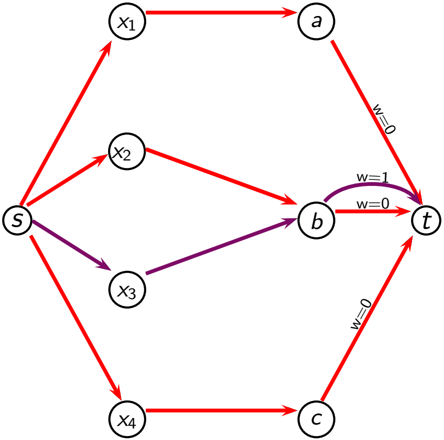
Test de cohérence



Test de cohérence



Test de cohérence



$\max(D_z) = 1$, $weight(f) = 1$

La contrainte est cohérente.

Filtrage

On peut aussi caractériser les valeurs viables grâce à l'utilisation des flots :

Filtrage

On peut aussi caractériser les valeurs viables grâce à l'utilisation des flots :

Théorème (Viabilité de (x_i, v_j) pour `Soft-AllDifferent`(X, μ_{dec}, z))

Une valeur (x_i, v_j) est viable ssi il existe un flot faisable f utilisant l'arc $(x_i \rightarrow v_j)$ et de poids $weight(f) \leq max(D_z)$.

Filtrage

On peut aussi caractériser les valeurs viables grâce à l'utilisation des flots :

Théorème (Viabilité de (x_i, v_j) pour `Soft-AllDifferent`(X, μ_{dec}, z))

Une valeur (x_i, v_j) est viable ssi il existe un flot faisable f utilisant l'arc $(x_i \rightarrow v_j)$ et de poids $weight(f) \leq max(D_z)$.

Dans l'exemple précédent,

- L'arête (x_4, b) ne peut apparaître que dans des flots de poids 2.
- Comme $max(D_z) = 1$, (x_4, b) est donc une valeur non viable.

Prise en compte des préférences

Dans beaucoup de problèmes réels sur-contraints, on aimerait pourvoir exprimer des préférences formulées par l'utilisateur :

- pour μ_{var}, prendre en compte les préférences sur les variables à réinstancier ;
- pour μ_{dec}, prendre en compte les préférences sur le choix des contraintes binaires à relaxer.

Comment prendre en compte ces préférences ?

Sémantique μ_{dec} avec préférences (Métivier, Boizumault, Loudni, 2008)

$$\texttt{AllDifferent}(\mathcal{X}) \equiv \bigwedge_{X_i \in \mathcal{X}, X_j \in \mathcal{X} \text{ t.q. } i<j} (X_i \neq X_j)$$

À chaque contrainte $X_i \neq X_j$ est associé un poids φ_{ij}.

Sémantique basée décomposition pour AllDifferent

$$\mu_{dec}^{\Sigma}(\mathcal{X}) = \sum_{X_i \in \mathcal{X}, X_j \in \mathcal{X} \text{ t.q. } i<j,\ X_i = X_j} \varphi_{ij}$$

Applications de cette sémantique :

- Coloration de graphes pondérés
- Allocation de fréquences

Test de cohérence

Classe de complexité du test de cohérence

Tester la cohérence de la relaxation de `AllDifferent` selon μ^{Σ}_{dec} est un problème NP-Complet.

$$k\text{-coloration} \leq_P \Sigma\text{-}\mathtt{AllDifferent}(\mu^{\Sigma}_{dec})$$

Idée de la Preuve :

- soit un graphe $G{=}(\mathcal{X}, E)$,
- $\forall X_i \in \mathcal{X},\ D_{X_i} = \{1..k\}$,
- si l'arête $(X_i, X_j) \in E$ alors $\varphi_{ij} = 1$ sinon $\varphi_{ij} = 0$,
- $D_z{=}\{0\}$.

Sémantique μ_{var} avec préférences (Métivier, Boizumault, Loudni, 2008)

μ_{var} représente le nombre de variables à réinstancier pour satisfaire la contrainte.

$$\mu_{var}(x_1, \ldots, x_n) = \sum_{d \in D_X} \max(\mid \{i, \mid x_i = d\} \mid -1, 0)$$

Sémantique μ_{var} avec préférences (Métivier, Boizumault, Loudni, 2008)

μ_{var} représente le nombre de variables à réinstancier pour satisfaire la contrainte.

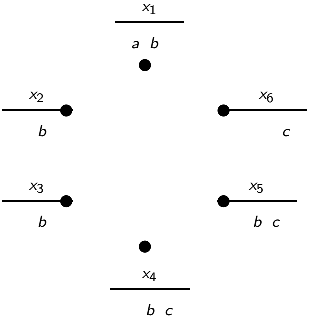
$$\mu_{var}(x_1, \ldots, x_n) = \sum_{d \in D_X} \max(\mid \{i, \mid x_i = d\} \mid -1, 0)$$

Pour la version avec préférences μ_{var}^{Σ} :

- à chaque variable X_i est associé un poids φ_i
- la violation d'une instanciation est la somme des poids des variables à réinstancier pour satisfaire la contrainte :

$$\mu_{var}^{\Sigma}(x_1, \ldots, x_n) = \sum_{d \in D_X} (\sum_{x_i = d, x_i \in X} \varphi_i - max_{x_i = d,\, x_i \in X}(\varphi_i))$$

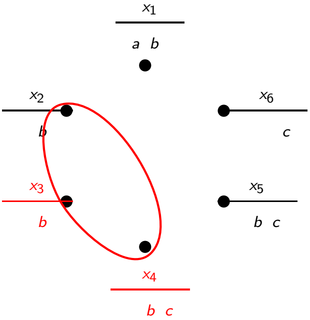
Sémantique basée variable μ_{var}^{Σ}



x_i	x_1	x_2	x_3	x_4	x_5	x_6
φ_i	1	9	5	3	8	4

Instanciation	À réinstancier	μ_{var}	μ_{var}^{Σ}
a b b b c c		3	12
a b b b b c	$x_3, x_{4,5}$	3	15
b b b b c c	x_1, x_3, x_4, x_6	4	13

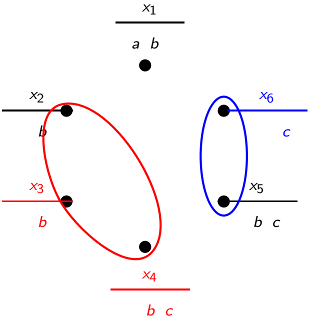
Sémantique basée variable μ_{var}^{Σ}



x_i	x_1	x_2	x_3	x_4	x_5	x_6
φ_i	1	9	5	3	8	4

Instanciation	À réinstancier	μ_{var}	μ_{var}^{Σ}
$a\ b\ b\ b\ c\ c$	x_3,x_4,	3	12
$a\ b\ b\ b\ b\ c$	x_3,x_4,5	3	15
$b\ b\ b\ b\ c\ c$	x_1,x_3,x_4,x_6	4	13

Sémantique basée variable μ_{var}^{Σ}



x_i	x_1	x_2	x_3	x_4	x_5	x_6
φ_i	1	9	5	3	8	4

Instanciation	À réinstancier	μ_{var}	μ_{var}^{Σ}
a b b b c c	x_3, x_4, x_6	3	12
a b b b b c	$x_3, x_{4,5}$	3	15
b b b b c c	x_1, x_3, x_4, x_6	4	13

Relaxation de la contrainte globale de cardinalité

La contrainte globale de cardinalité

Soit un ensemble de variables X. A chaque valeur $d_j \in D_X$ sont associées une borne inférieure ℓ_j et une borne supérieure u_j. La contrainte globale de cardinalité (Gcc) impose que le nombre d'occurrences de (certaines) valeurs d_j soit compris entre une borne inférieure ℓ_j et une borne supérieure u_j.

Définition (Régin, 1996)

Soit $X = \{x_1, ..., x_n\}$ un ensemble de variables, soit $occ(d_j, \tau)$ le nombre d'occurrences de la valeur d_j dans un tuple τ. Alors,

$$\mathrm{gcc}(X, l, u) = \{\tau = (d_1, ..., d_n) \mid \forall j \in [1..n], \ell_j \leq occ(d_j, \tau) \leq u_j\}$$

Exemple : Gestion de personnels

Employés	M	P	N
Alice	X	X	
Bob	X	X	
Charlie	X		
Daniel		X	
Eve		X	X

Activité	min	max
M	1	2
P	1	3
N	0	1

Exemple : Gestion de personnels

Employés	M	P	N
Alice	X	X	
Bob	X	X	
Charlie	X		
Daniel		X	
Eve		X	X

Activité	min	max
M	1	2
P	1	3
N	0	1

- Planifier les activités de 5 personnes, pour effectuer 3 tâches $\mathcal{X}=\{X_A, X_B, X_C, X_D, X_E\}$
- $D_{X_A}=D_{X_B}=\{\mathbf{P}, \mathbf{M}\}, D_{X_E}=\{\mathbf{N}, \mathbf{P}\}, D_{X_C}=\{\mathbf{M}\}, D_{X_D}=\{\mathbf{P}\}$
- $\mathcal{C}=\{\texttt{Gcc}(\mathcal{X},[\mathbf{1},\mathbf{1},\mathbf{0}],[\mathbf{2},\mathbf{3},\mathbf{1}])\}$

Exemple : Gestion de personnels

Employés	M	P	N
Alice	X	X	
Bob	X	X	
Charlie	X		
Daniel		X	
Eve		X	X

Activité	min	max
M	1	2
P	1	3
N	0	1

- Planifier les activités de 5 personnes, pour effectuer 3 tâches $\mathcal{X}=\{X_A, X_B, X_C, X_D, X_E\}$
- $D_{X_A}=D_{X_B}=\{\mathbf{P}, \mathbf{M}\}, D_{X_E}=\{\mathbf{N}, \mathbf{P}\}, D_{X_C}=\{\mathbf{M}\}, D_{X_D}=\{\mathbf{P}\}$
- $\mathcal{C}=\{\texttt{Gcc}(\mathcal{X},[\mathbf{1},\mathbf{1},\mathbf{0}],[\mathbf{2},\mathbf{3},\mathbf{1}])\}$
- Solution : $\{(X_A=\mathbf{M}),(X_B=\mathbf{P}),(X_C=\mathbf{M}),(X_D=\mathbf{P}),(X_E=\mathbf{N})\}$

Représentation sous la forme d'un réseau

Comme pour `AllDifferent`, la contrainte $\texttt{Gcc}(X, \ell, u)$ peut être peut être représentée par un réseau $\mathcal{R} = \{\mathcal{V}, \mathcal{A}\}$, avec

Représentation sous la forme d'un réseau

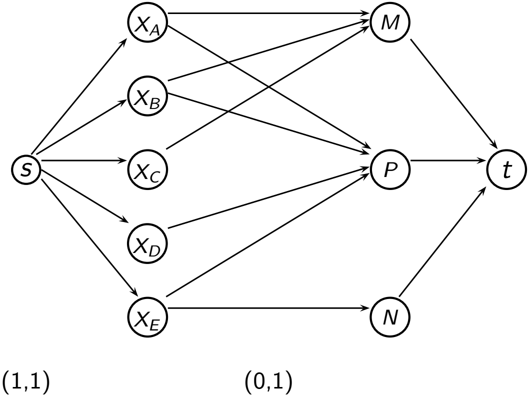
Comme pour `AllDifferent`, la contrainte $\texttt{Gcc}(X, \ell, u)$ peut être peut être représentée par un réseau $\mathcal{R} = \{\mathcal{V}, \mathcal{A}\}$, avec

$$\begin{array}{rcl} V & = & \{s\} \cup X \cup D_X \cup \{t\}, A = A_s \cup A_X \cup A_t \\ A_s & = & \{(s \to x_i) \mid i \in [1 \ldots n]\} \\ A_X & = & \{(x_i \to d_j) \mid d_j \in D_i,\ i \in [1 \ldots n]\} \\ A_t & = & \{(d_j \to t) \mid d_j \in D_X\} \end{array}$$

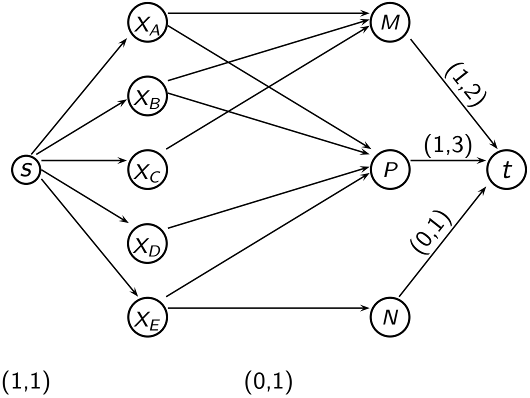
avec pour demandes et capacités :

$$\begin{array}{ll} \text{si } a \in A_s, & d(a) = 1 \text{ et } c(a) = 1 \\ \text{si } a \in A_X, & d(a) = 0 \text{ et } c(a) = 1 \\ \text{si } a = (d_j \to t) \in A_t, & d(a) = l_j \text{ et } c(a) = u_j \end{array}$$

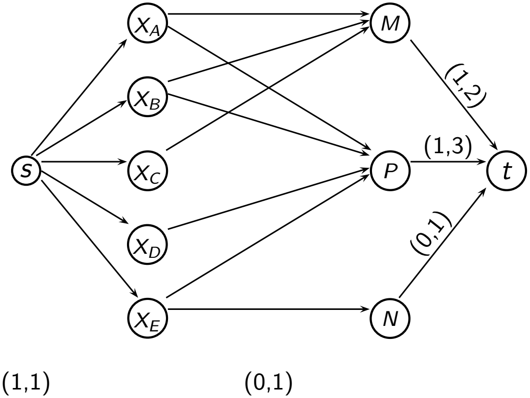
Représentation sous la forme d'un réseau



Représentation sous la forme d'un réseau



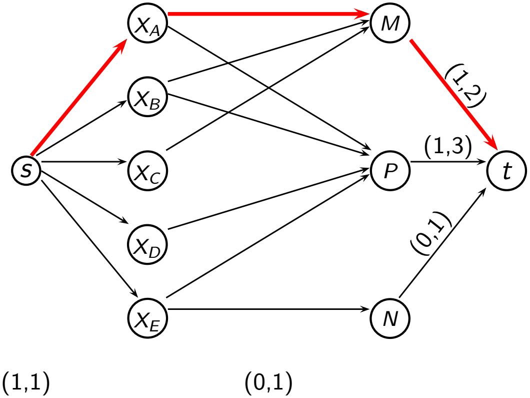
Gcc : Test de cohérence



Théorème (Cohérence de Gcc(X, ℓ, u) (Régin, 1996))

La contrainte Gcc(X, ℓ, u) admet une solution <u>*ssi*</u> *il existe, dans le réseau associé, un flot faisable f de valeur n.*

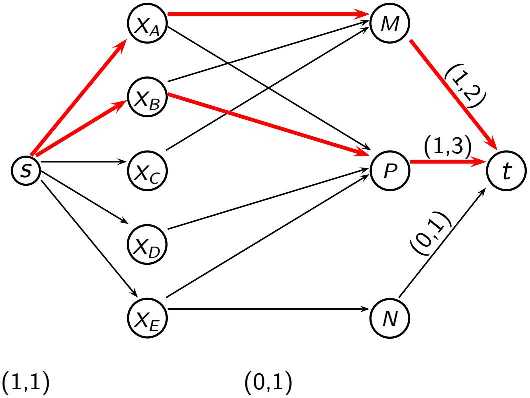
Gcc : Test de cohérence



Théorème (Cohérence de Gcc(X, ℓ, u) (Régin, 1996))

La contrainte Gcc(X, ℓ, u) admet une solution ssi il existe, dans le réseau associé, un flot faisable f de valeur n.

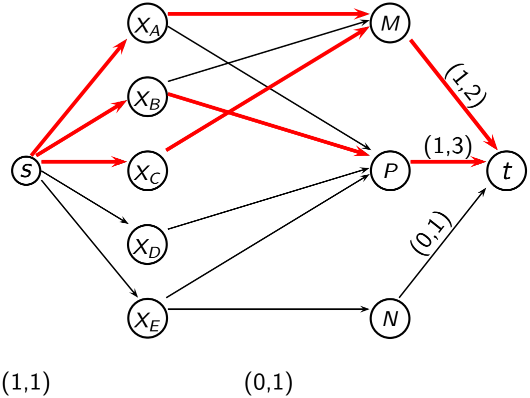
Gcc : Test de cohérence



Théorème (Cohérence de Gcc(X, ℓ, u) (Régin, 1996))

La contrainte Gcc(X, ℓ, u) admet une solution ssi il existe, dans le réseau associé, un flot faisable f de valeur n.

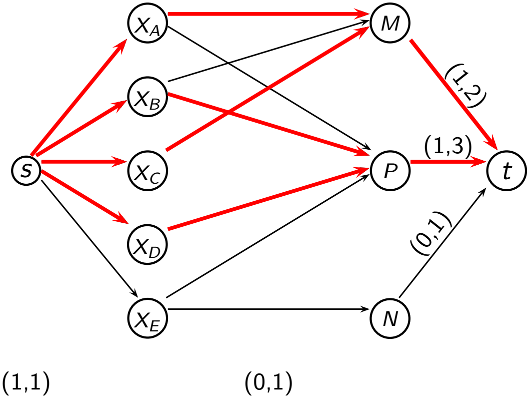
Gcc : Test de cohérence



Théorème (Cohérence de $\texttt{Gcc}(X, \ell, u)$ (Régin, 1996))

La contrainte $\texttt{Gcc}(X, \ell, u)$ admet une solution ssi il existe, dans le réseau associé, un flot faisable f de valeur n.

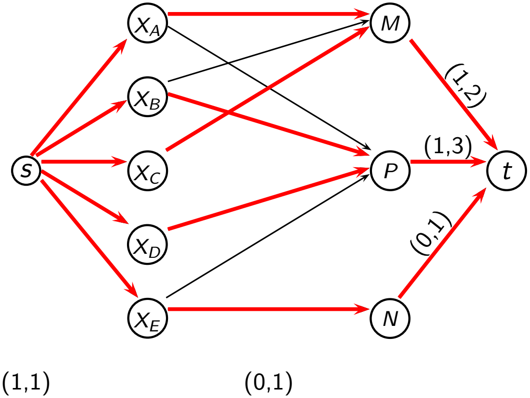
Gcc : Test de cohérence



Théorème (Cohérence de Gcc(X, ℓ, u) (Régin, 1996))

La contrainte Gcc(X, ℓ, u) admet une solution ssi *il existe, dans le réseau associé, un flot faisable f de valeur n.*

`Gcc` : Test de cohérence



Théorème (Cohérence de $\texttt{Gcc}(X, \ell, u)$ (Régin, 1996))

La contrainte $\texttt{Gcc}(X, \ell, u)$ admet une solution <u>ssi</u> il existe, dans le réseau associé, un flot faisable f de valeur n.

Cas sur-contraint

Employés	M	P	N
Alice		X	X
Bob			X
Charlie	X	X	
Daniel	X	X	X
Eve		X	X

Activité	min	max
M	3	4
P	1	1
N	1	1

- Il n'existe pas solution ; car il n'est pas possible de satisfaire la borne inférieure de l'équipe de Matin.

Cas sur-contraint

Employés	**M**	**P**	**N**
Alice		X	X
Bob			X
Charlie	X	X	
Daniel	X	X	X
Eve		X	X

Activité	min	max
M	3	4
P	1	1
N	1	1

- Il n'existe pas solution ; car il n'est pas possible de satisfaire la borne inférieure de l'équipe de Matin.
- Autoriser les bornes à ne pas être respectées → Autoriser les embauches ou les sur-encadrements.

Cas sur-contraint

Employés	M	P	N
Alice		X	X
Bob			X
Charlie	X	X	
Daniel	X	X	X
Eve		X	X

Activité	min	max
M	3	4
P	1	1
N	1	1

- Il n'existe pas solution ; car il n'est pas possible de satisfaire la borne inférieure de l'équipe de Matin.
- Autoriser les bornes à ne pas être respectées → Autoriser les embauches ou les sur-encadrements.
- Sémantique sans préférence → compter le nombre de moniteurs à embaucher et ou en sur-encadrements.

Cas sur-contraint

Employés	M	P	N
Alice		X	X
Bob			X
Charlie	X	X	
Daniel	X	X	X
Eve		X	X

Activité	min	max
M	3	4
P	1	1
N	1	1

- Il n'existe pas solution ; car il n'est pas possible de satisfaire la borne inférieure de l'équipe de Matin.
- Autoriser les bornes à ne pas être respectées → Autoriser les embauches ou les sur-encadrements.
- Sémantique sans préférence → compter le nombre de moniteurs à embaucher et ou en sur-encadrements.
- Il est nécessaire **d'exprimer des préférences** qui vont refléter les coûts d'embauches ou de sur-encadrements.

Sémantique basée décomposition

$$\texttt{gcc}(\mathcal{X}, l, u) \equiv \bigwedge_{d_j \in Doms} (\texttt{atleast}(\mathcal{X}, d_j, l_j) \wedge \texttt{atmost}(\mathcal{X}, d_j, u_j))$$

Mesurer l'écart aux bornes [Hoeve, 2006]

$$\begin{aligned} s(\mathcal{X}, d_j) &= \max(0, l_j - |\{X_i \mid X_i \in \mathcal{X}, X_i = d_j\}|) \\ e(\mathcal{X}, d_j) &= \max(0, |\{X_i \mid X_i \in \mathcal{X}, X_i = d_j\}| - u_j) \end{aligned}$$

Sémantique basée décomposition sans préférence pour Gcc

$$\mu_{dec}(\mathcal{X}) = \sum_{v_j \in Doms} (s(\mathcal{X}, v_j) + e(\mathcal{X}, v_j))$$

Sémantique μ_{dec} avec préférences (Métivier, Boizumault, Loudni, 2009)

$$\texttt{gcc}(\mathcal{X}, l, u) \equiv \bigwedge_{d_j \in Doms} (\texttt{atleast}(\mathcal{X}, d_j, l_j) \wedge \texttt{atmost}(\mathcal{X}, d_j, u_j))$$

Différencier les bornes

$$\begin{array}{ccc} \texttt{atleast}(\mathcal{X}, v_j, l_j) & \rightarrow & \varphi_j^{atleast} \\ \texttt{atmost}(\mathcal{X}, v_j, l_j) & \rightarrow & \varphi_j^{atmost} \end{array}$$

Sémantique basée décomposition avec préférences pour Gcc

$$\mu_{dec}^{\Sigma}(\mathcal{X}) = \sum_{d_j \in Doms} \left(s(\mathcal{X}, d_j) \times \varphi_j^{atleast} + e(\mathcal{X}, d_j) \times \varphi_j^{atmost} \right)$$

Retour sur la gestion de Personnels

Activité	min	max
M	3	4
P	1	1
N	1	1

coût	embauche	sur-encadrement
M	25	10
P	12	7
N	5	1

Employés	M	P	N
Alice		X	X
Bob			X
Charlie	X	X	
Daniel	X	X	X
Eve		X	X

Retour sur la gestion de Personnels

Activité	min	max
M	3	4
P	1	1
N	1	1

coût	embauche	sur-encadrement
M	25	10
P	12	7
N	5	1

Employés	**M**	**P**	**N**
Alice		X	X
Bob			X
Charlie	X	X	
Daniel	X	X	X
Eve		X	X

- $\mathcal{X}=\{X_A, X_B, X_C, X_D, X_E\}$
- $D_{X_A}=D_{X_E}=\{\mathbf{P}, \mathbf{N}\}, D_{X_B}=\{\mathbf{N}\}, D_{X_C}=\{\mathbf{M}, \mathbf{P}\}, D_{X_D}=\{\mathbf{M}, \mathbf{P}, \mathbf{N}\}$
- $\mathcal{C}=\{\Sigma\text{-}\mathtt{Gcc}(\mathcal{X}, \qquad , \qquad , \qquad , z)\}$

Retour sur la gestion de Personnels

Activité	min	max
M	3	4
P	1	1
N	1	1

Employés	**M**	**P**	**N**
Alice		X	X
Bob			X
Charlie	X	X	
Daniel	X	X	X
Eve		X	X

coût	embauche	sur-encadrement
M	25	10
P	12	7
N	5	1

- $\mathcal{X}=\{X_A, X_B, X_C, X_D, X_E\}$
- $D_{X_A}=D_{X_E}=\{\mathbf{P}, \mathbf{N}\}, D_{X_B}=\{\mathbf{N}\}, D_{X_C}=\{\mathbf{M}, \mathbf{P}\}, D_{X_D}=\{\mathbf{M}, \mathbf{P}, \mathbf{N}\}$
- $\mathcal{C}=\{\Sigma\text{-}\mathtt{Gcc}(\mathcal{X},[\mathbf{3},\mathbf{1},\mathbf{1}],[\mathbf{4},\mathbf{1},\mathbf{1}], \quad , \quad ,z)\}$

Retour sur la gestion de Personnels

Activité	min	max
M	3	4
P	1	1
N	1	1

coût	embauche	sur-encadrement
M	25	10
P	12	7
N	5	1

Employés	**M**	**P**	**N**
Alice		X	X
Bob			X
Charlie	X	X	
Daniel	X	X	X
Eve		X	X

- $\mathcal{X}=\{X_A, X_B, X_C, X_D, X_E\}$
- $D_{X_A}=D_{X_E}=\{\mathbf{P}, \mathbf{N}\}, D_{X_B}=\{\mathbf{N}\}, D_{X_C}=\{\mathbf{M}, \mathbf{P}\}, D_{X_D}=\{\mathbf{M}, \mathbf{P}, \mathbf{N}\}$
- $\mathcal{C}=\{\Sigma\text{-}\mathtt{Gcc}(\mathcal{X}, [\mathbf{3}, \mathbf{1}, \mathbf{1}], [\mathbf{4}, \mathbf{1}, \mathbf{1}], [\mathbf{25}, \mathbf{12}, \mathbf{5}], \qquad , z)\}$

Retour sur la gestion de Personnels

Activité	min	max
M	3	4
P	1	1
N	1	1

coût	embauche	sur-encadrement
M	25	10
P	12	7
N	5	1

Employés	**M**	**P**	**N**
Alice		X	X
Bob			X
Charlie	X	X	
Daniel	X	X	X
Eve		X	X

- $\mathcal{X}=\{X_A, X_B, X_C, X_D, X_E\}$
- $D_{X_A}=D_{X_E}=\{\mathbf{P}, \mathbf{N}\}, D_{X_B}=\{\mathbf{N}\}, D_{X_C}=\{\mathbf{M}, \mathbf{P}\}, D_{X_D}=\{\mathbf{M}, \mathbf{P}, \mathbf{N}\}$
- $\mathcal{C}=\{\Sigma\text{-}\mathtt{Gcc}(\mathcal{X}, [\mathbf{3},\mathbf{1},\mathbf{1}], [\mathbf{4},\mathbf{1},\mathbf{1}], [\mathbf{25},\mathbf{12},\mathbf{5}], [\mathbf{10},\mathbf{7},\mathbf{1}], z)\}$