Problèmes de Satisfaction de Contraintes Valués

Cyril Terrioux
cyril.terrioux@univ-amu.fr

Faculté des Sciences
Aix*Marseille Université

Plan

Plan

Contexte

Solution d'un CSP = instanciation qui satisfait **toutes** les contraintes

Contexte

Solution d'un CSP = instanciation qui satisfait **toutes** les contraintes

Solution d'un COP = instanciation qui satisfait **toutes** les contraintes et optimise la fonction objectif

Contexte

Solution d'un CSP = instanciation qui satisfait **toutes** les contraintes

Solution d'un COP = instanciation qui satisfait **toutes** les contraintes et optimise la fonction objectif

Que faire si l'instance est incohérente ?

Contexte

Solution d'un CSP = instanciation qui satisfait **toutes** les contraintes

Solution d'un COP = instanciation qui satisfait **toutes** les contraintes et optimise la fonction objectif

Que faire si l'instance est incohérente ?

Rechercher une affectation satisfaisant le plus possible les contraintes

Le pouvoir d'expression des contraintes

Diversité de la nature des contraintes :

- contraintes impliquant un nombre quelconque de variables,
- énumérations des affectations compatibles ou interdites,
- équations, inéquations,
- prédicats,
- contraintes globales,
- ...

Le pouvoir d'expression des contraintes

Un formalisme permettant de représenter toute sorte de problèmes :

- Enchères combinatoires
- Bio-informatique
- Logistique
- Jeux
- Routage
- Transport
- Cryptographie
- Planification
- Télécommunication : Problème d'allocation de fréquences
- Théorie des graphes
- Agriculture
- Chimie
- Vérification de circuits,
- ...

Le pouvoir d'expression des contraintes

Mais certaines notions posent problèmes :

- possibilité,
- probabilité,
- préférence,
- ...

Cas des instances du monde réel

Deux types de contraintes :

- contraintes « dures » :
  - contraintes matérielles,
  - contraintes physiques,
  - contraintes électroniques,
  - . . .

Cas des instances du monde réel

Deux types de contraintes :

- contraintes « dures » :
  - contraintes matérielles,
  - contraintes physiques,
  - contraintes électroniques,
  - ...
- contraintes « molles » :
  - préférence,
  - possibilité,
  - ...

Exemple 1 : Conception d'un emploi du temps

- contraintes physiques :
  - un enseignant ne peut faire qu'un seul cours à la fois,
  - exactement un enseignement par salle,
  - capacité des salles,
  - ...
- contraintes de préférence :
  - tel enseignement en premier,
  - minimiser les « trous »,
  - voeux des enseignants,
  - ...

Exemple 2 : Couverture d'une zone par des antennes 4G

- contraintes physiques :
  - portée limitée d'une antenne,
  - positionnement de l'antenne,
  - ...
- contraintes de préférence :
  - limiter les recouvrements multiples
  - minimiser les zones blanches
  - minimiser le nombre d'antennes
  - ...

Exemple 3 : Planification de prises de vue par satellite

- contraintes physiques :
  - nombre d'appareils limités,
  - temps de repositionnement,
  - couverture nuageuse,
  - ...
- contraintes de préférence :
  - prises de vue prioritaires,
  - ...

Solutions possibles

- ignorer ces contraintes
  - approximation du problème réel
  - grand nombre de solutions possibles
  - solutions inacceptables en pratique

Solutions possibles

- ignorer ces contraintes
  - approximation du problème réel
  - grand nombre de solutions possibles
  - solutions inacceptables en pratique
- coder sous forme de contraintes traditionnelles
  - problème souvent incohérent

Solutions possibles

- ignorer ces contraintes
  - approximation du problème réel
  - grand nombre de solutions possibles
  - solutions inacceptables en pratique
- coder sous forme de contraintes traditionnelles
  - problème souvent incohérent
- étendre le formalisme CSP

Plan

CSP possibilistes

Minimiser le maximum des priorités des contraintes violées

Priorité = réel compris entre 0 et 1

- Priorité 1 = contrainte à satisfaire absolument
- Priorité 0 = contrainte "sans importance"

CSP probabilistes

Minimiser la probabilité qu'une instanciation ne soit pas une solution

Probabilité de l'existence d'une contrainte = réel compris entre 0 et 1

- Probabilité 1 = la contrainte existe
- Probabilité 0 = la contrainte n'existe pas

Partial-CSP

On s'autorise la violation de certaines contraintes.

Minimiser les violations

Deux « sous-classes » :

- Max-CSP
- CSP pondérés

Max-CSP

Maximiser le nombre de contraintes satisfaites

c'est-à-dire minimiser le nombre de contraintes violées

Coût d'une contrainte violée $= 1$

CSP pondérés

Minimiser la somme pondérée des contraintes violées.

Coût d'une contrainte violée = un entier naturel

Remarques

- Plusieurs formalismes, plusieurs algorithmes similaires

  $\Rightarrow$ redondance des travaux

- Un même objectif : optimiser une fonction portant sur la satisfaction des contraintes

Plan

CSP valués (VCSP)

Instance $\mathcal{P} = (X, D, C, SV, \phi)$:

- un CSP classique (X, D, C),

CSP valués (VCSP)

Instance $\mathcal{P} = (X, D, C, SV, \phi)$:

- un CSP classique (X, D, C),
- une structure de valuation $SV = (E, \preceq, \oplus, \perp, \top)$,
    - E : ensemble de valuation
    - $\preceq$: relation d'ordre total
    - $\oplus$: loi de composition interne
    - $\perp$: élément minimum
    - $\top$: élément maximum

CSP valués (VCSP)

Instance $\mathcal{P} = (X, D, C, SV, \phi)$:

- un CSP classique (X, D, C),
- une structure de valuation $SV = (E, \preceq, \oplus, \perp, \top)$,
  - E : ensemble de valuation
  - $\preceq$: relation d'ordre total
  - $\oplus$: loi de composition interne
  - $\perp$: élément minimum
  - $\top$: élément maximum
- une application ϕ de C dans E.

CSP valués (VCSP)

Instance $\mathcal{P} = (X, D, C, SV, \phi)$:

- un CSP classique (X, D, C),
- une structure de valuation $SV = (E, \preceq, \oplus, \perp, \top)$,
  - E : ensemble de valuation
  - $\preceq$: relation d'ordre total
  - $\oplus$: loi de composition interne
  - $\perp$: élément minimum
  - $\top$: élément maximum
- une application ϕ de C dans E.

Valuation = graduation de la violation

- $\perp$ = absence de violation
- $\top$ = violation inacceptable

Structure de valuation

Propriétés minimales de $\oplus$:

- commutativité et associativité
- monotonie : $\forall a, b, c \in E$ tq $a \preceq b$ on a $a \oplus c \preceq b \oplus c$
- $\perp$ élément neutre : $\forall a \in E, a \oplus \perp = a$
- $\top$ élément absorbant : $\forall a \in E, a \oplus \top = \top$

Structure de valuation

Propriétés minimales de $\oplus$:

- commutativité et associativité
- monotonie : $\forall a, b, c \in E$ tq $a \preceq b$ on a $a \oplus c \preceq b \oplus c$
- $\perp$ élément neutre : $\forall a \in E, a \oplus \perp = a$
- $\top$ élément absorbant : $\forall a \in E, a \oplus \top = \top$

Propriétés supplémentaires éventuelles :

- idempotence : $\forall a \in E, a \oplus a = a$

Structure de valuation

Propriétés minimales de $\oplus$:

- commutativité et associativité
- monotonie : $\forall a, b, c \in E$ tq $a \preceq b$ on a $a \oplus c \preceq b \oplus c$
- $\perp$ élément neutre : $\forall a \in E, a \oplus \perp = a$
- $\top$ élément absorbant : $\forall a \in E, a \oplus \top = \top$

Propriétés supplémentaires éventuelles :

- idempotence : $\forall a \in E, a \oplus a = a$

  $\Rightarrow$ opérateur $\oplus$ = max

Structure de valuation

Propriétés minimales de $\oplus$:

- commutativité et associativité
- monotonie : $\forall a, b, c \in E$ tq $a \preceq b$ on a $a \oplus c \preceq b \oplus c$
- $\perp$ élément neutre : $\forall a \in E, a \oplus \perp = a$
- $\top$ élément absorbant : $\forall a \in E, a \oplus \top = \top$

Propriétés supplémentaires éventuelles :

- idempotence : $\forall a \in E, a \oplus a = a$

  $\Rightarrow$ opérateur $\oplus = \max$

- monotonie stricte :

  $\forall a, b, c \in E$ tq $a \prec b$ et $c \neq \top$ on a $a \oplus c \prec b \oplus c$

Illustrations

CSP	E	$\oplus$	$\perp$	$\top$	$\prec$
Classique	$\{t, f\}$	$\wedge = \max$	t	f	$t \prec f$
Possibiliste	$[0, 1]$	$\max$	0	1	$<$
Partial-CSP	$\mathbb{N} \cup \{+\infty\}$	$+$	0	$+\infty$	$<$
Probabiliste	$[0, 1]$	$x + y - xy$	0	1	$<$

Semiring CSP

Un autre formalisme général

Une structure de valuation différente

Possibilité d'utiliser un ordre partiel pour $\preceq$

Semiring CSP

Un autre formalisme général

Une structure de valuation différente

Possibilité d'utiliser un ordre partiel pour $\preceq$

Equivalent au formalisme VCSP si $\preceq$ est un ordre total

Fonctions de coût

Instance $\mathcal{P} = (X, D, W, SV)$:

- $X = \{x_1, \dots, x_n\}$ un ensemble de n variables,
- $D = \{d_{x_1}, \dots, d_{x_n}\}$ un ensemble de domaines finis de taille au plus d,
- W un ensemble de fonctions de coût $w : \prod_{x \in S(w)} d_x \mapsto E$
- une structure de valuation $SV = (E, \preceq, \oplus, \perp, \top)$,
  - E : ensemble de valuation
  - $\preceq$: relation d'ordre total
  - $\oplus$: loi de composition interne
  - $\perp$: élément minimum
  - $\top$: élément maximum

Fonctions de coût

Instance $\mathcal{P} = (X, D, W, SV)$:

- $X = \{x_1, \ldots, x_n\}$ un ensemble de n variables,
- $D = \{d_{x_1}, \ldots, d_{x_n}\}$ un ensemble de domaines finis de taille au plus d,
- W un ensemble de fonctions de coût $w : \prod_{x \in S(w)} d_x \mapsto E$
- une structure de valuation $SV = (E, \preceq, \oplus, \perp, \top)$,
  - E : ensemble de valuation
  - $\preceq$: relation d'ordre total
  - $\oplus$: loi de composition interne
  - $\perp$: élément minimum
  - $\top$: élément maximum

$W = \{w_\emptyset\} \cup \{w_{x_1}, \ldots w_{x_n}\} \cup \{\ldots\}$

Valuation d'affectations

Valuation :

Soit une instanciation $\mathcal{A}$ sur X.

$$\mathcal{V}(\mathcal{A}) = \bigoplus_{w \in W} w(\mathcal{A}[S(w)])$$

Valuation d'affectations

Valuation :

Soit une instanciation $\mathcal{A}$ sur X.

$$\mathcal{V}(\mathcal{A}) = \bigoplus_{w \in W} w(\mathcal{A}[S(w)])$$

Valuation locale :

Soit une instanciation $\mathcal{B}$ sur $Y \subseteq X$.

$$v(\mathcal{B}) = \bigoplus_{w \in W \ | \ S(w) \subseteq Y} w(\mathcal{B}[S(w)])$$

Valuation d'affectations

Valuation :

Soit une instanciation $\mathcal{A}$ sur X.

$$\mathcal{V}(\mathcal{A}) = \bigoplus_{w \in W} w(\mathcal{A}[S(w)])$$

Valuation locale :

Soit une instanciation $\mathcal{B}$ sur $Y \subseteq X$.

$$v(\mathcal{B}) = \bigoplus_{w \in W \mid S(w) \subseteq Y} w(\mathcal{B}[S(w)])$$

Propriété :

Si $\mathcal{B} \subseteq \mathcal{A}$, on a $v(\mathcal{B}) \preceq v(\mathcal{A}) = \mathcal{V}(\mathcal{A})$.

Coloration de graphes

Problème :

- Instance :
  - un graphe $G = (V, A)$,
  - un entier k
- Objectif : trouver une affectation se rapprochant le plus possible d'une k-coloration de G

Coloration de graphes

Problème :

- Instance :
  - un graphe $G = (V, A)$,
  - un entier k
- Objectif : trouver une affectation se rapprochant le plus possible d'une k-coloration de G

Modélisation VCSP :

- $X = \{x_1, \ldots, x_n\}$,
- $D = \{d_{x_1}, \ldots, d_{x_n}\}$ avec $d_{x_i} = \{1, \ldots, k\}$

Coloration de graphes

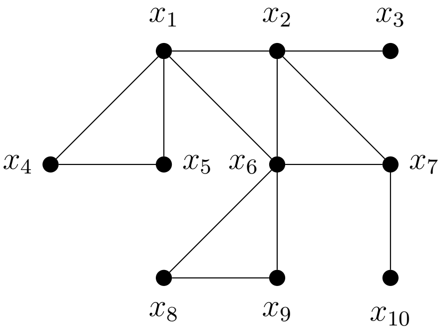
Problème :

- Instance :
  - un graphe $G = (V, A)$,
  - un entier k
- Objectif : trouver une affectation se rapprochant le plus possible d'une k-coloration de G

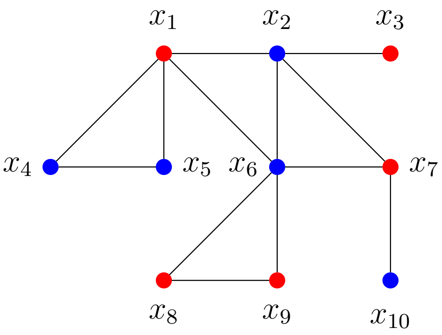
Modélisation VCSP :

- $X = \{x_1, \ldots, x_n\}$,
- $D = \{d_{x_1}, \ldots, d_{x_n}\}$ avec $d_{x_i} = \{1, \ldots, k\}$
- $W = \{w_{ij} | \{i, j\} \in A\}$ avec :
  - $S(w_{ij}) = \{x_i, x_j\}$
  - $w_{ij}(v_i, v_j) = 1$ si $v_i = v_j$, 0 sinon
- $SV = (\mathbb{N} \cup \{+\infty\}, \leq, +, 0, +\infty)$

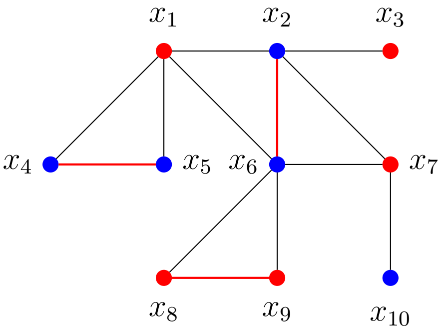
Exemple ($k = 2$)



Exemple ($k = 2$)



Exemple ($k = 2$)



$\Rightarrow \mathcal{V}(\mathcal{A}) = v(\mathcal{A}) = 3$

Le problème VCSP

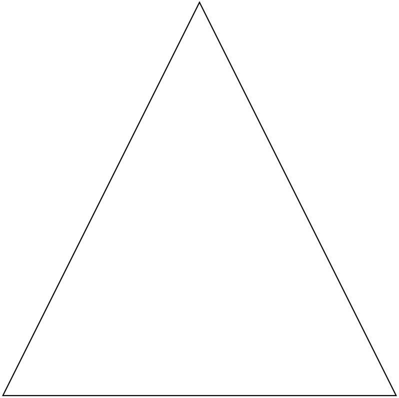
- *Instance* : $\mathcal{P} = (X, D, W, SV)$
- *Question* : Trouver une affectation sur X de valuation minimum.

Un problème NP-Difficile

Le problème VCSP

- *Instance* : $\mathcal{P} = (X, D, W, SV)$
- *Question* : Trouver une affectation sur X de valuation minimum.

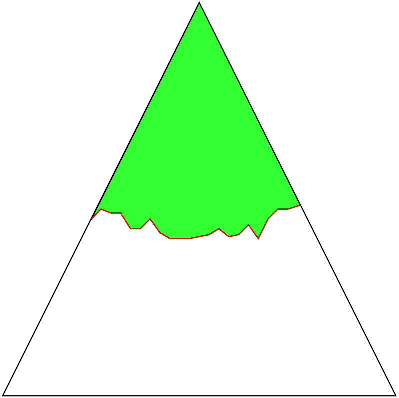
Un problème NP-Difficile



Le problème VCSP

- *Instance* : $\mathcal{P} = (X, D, W, SV)$
- *Question* : Trouver une affectation sur X de valuation minimum.

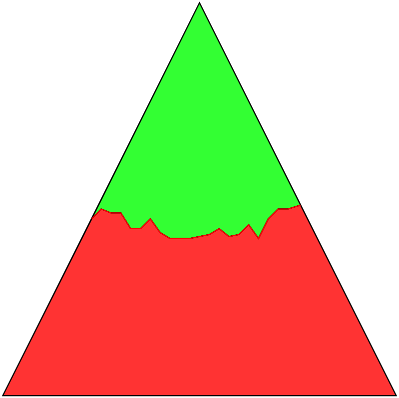
Un problème NP-Difficile



Le problème VCSP

- *Instance* : $\mathcal{P} = (X, D, W, SV)$
- *Question* : Trouver une affectation sur X de valuation minimum.

Un problème NP-Difficile



Plan

Résolution

Taille de l'espace de recherche : $O(d^n)$

Résolution

Taille de l'espace de recherche : $O(d^n)$

- Méthodes complètes :
  - Approches énumératives (Branch and Bound)
  - Approches par programmation dynamique
  - Approches des poupées russes
- Méthodes incomplètes

Branch and Bound (BnB)

Utilisation de deux bornes :

- lb : minorant du coût de l'affectation courante
- ub : majorant de la valeur optimale de f (coût de la meilleure solution connue)

Branch and Bound (BnB)

Utilisation de deux bornes :

- lb : minorant du coût de l'affectation courante
- ub : majorant de la valeur optimale de f (coût de la meilleure solution connue)

Principe :

- on étend progressivement l'affectation courante

Branch and Bound (BnB)

Utilisation de deux bornes :

- lb : minorant du coût de l'affectation courante
- ub : majorant de la valeur optimale de f (coût de la meilleure solution connue)

Principe :

- on étend progressivement l'affectation courante
- en cas d'échec (c.-à-d. si $lb \succeq ub$), on change la valeur de la variable courante

Branch and Bound (BnB)

Utilisation de deux bornes :

- lb : minorant du coût de l'affectation courante
- ub : majorant de la valeur optimale de f (coût de la meilleure solution connue)

Principe :

- on étend progressivement l'affectation courante
- en cas d'échec (c.-à-d. si $lb \succeq ub$), on change la valeur de la variable courante
- s'il n'y a plus de valeur, on revient sur la variable précédente

Branch and Bound (BnB)

Utilisation de deux bornes :

- lb : minorant du coût de l'affectation courante
- ub : majorant de la valeur optimale de f (coût de la meilleure solution connue)

Principe :

- on étend progressivement l'affectation courante
- en cas d'échec (c.-à-d. si $lb \succeq ub$), on change la valeur de la variable courante
- s'il n'y a plus de valeur, on revient sur la variable précédente
- on réitère le procédé tant qu'on n'a pas essayé toutes les possibilités et $lb \prec ub$

Trois types d'améliorations possibles

- choisir un bon ordre sur les variables
  - diminution de la taille de l'arbre
  - augmentation rapide du minorant

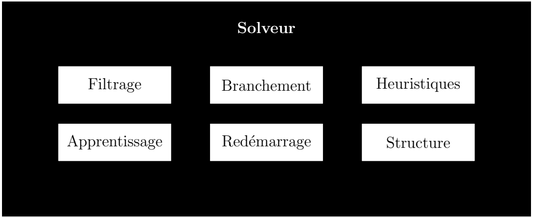
Trois types d'améliorations possibles

- choisir un bon ordre sur les variables
  - diminution de la taille de l'arbre
  - augmentation rapide du minorant
- choisir un bon ordre sur les valeurs
  - trouver de bonnes solutions
  - diminution rapide du majorant

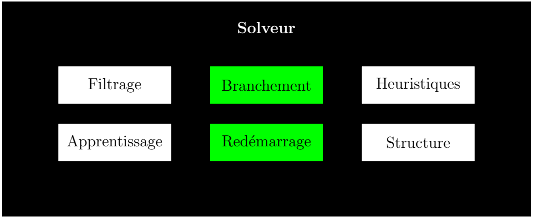
Trois types d'améliorations possibles

- choisir un bon ordre sur les variables
    - diminution de la taille de l'arbre
    - augmentation rapide du minorant
- choisir un bon ordre sur les valeurs
    - trouver de bonnes solutions
    - diminution rapide du majorant
- calculer un minorant de bonne qualité
    - diminution de la taille de l'arbre par élagage

Solveur


Solveur
Filtrage
Branchement
Heuristiques
Apprentissage
Redémarrage
Structure

Solveur


Solveur
Filtrage
Branchement
Heuristiques
Apprentissage
Redémarrage
Structure

Heuristiques

Heuristiques sur les variables : les mêmes que pour CSP et COP

Heuristiques sur les valeurs :

- Objectif : choisir une valeur menant à la solution la plus prometteuse possible

Heuristiques

Heuristiques sur les variables : les mêmes que pour CSP et COP

Heuristiques sur les valeurs :

- Objectif : choisir une valeur menant à la solution la plus prometteuse possible
- Exemple : choisir la valeur de d_x qui apporte la valuation la plus faible

  Information disponible grâce à w_x

Filtrage

Objectif : éviter de développer certaines branches inutiles

Filtrage

Objectif : éviter de développer certaines branches inutiles

Deux moyens :

- calcul d'un minorant de qualité
- suppression des valeurs inutiles

Filtrage

Objectif : éviter de développer certaines branches inutiles

Deux moyens :

- calcul d'un minorant de qualité
- suppression des valeurs inutiles

valeur inutile = valeur ne participant pas à une solution optimale

Suppression de la valeur v de x si $w_\emptyset \oplus w_x(v) \succeq ub$

Filtrage

Objectif : éviter de développer certaines branches inutiles

Deux moyens :

- calcul d'un minorant de qualité
- suppression des valeurs inutiles

valeur inutile = valeur ne participant pas à une solution optimale

Suppression de la valeur v de x si $w_\emptyset \oplus w_x(v) \succeq ub$

Un processus plus complexe quand $\oplus$ n'est pas idempotent

Cohérence d'arc

Hypothèse de travail : $\forall u, v \in E, u \preceq v, v \ominus u$ *existe*

Différence δ de v et de u : $u \oplus \delta = v$

$v \ominus u =$ la plus grande différence δ

Cohérence d'arc

Hypothèse de travail : $\forall u, v \in E, u \preceq v, v \ominus u$ *existe*

Différence δ de v et de u : $u \oplus \delta = v$

$v \ominus u =$ la plus grande différence δ

Principe

Déplacer les coûts tout en préservant l'équivalence

Cohérence d'arc

Hypothèse de travail : $\forall u, v \in E, u \preceq v, v \ominus u$ *existe*

Différence δ de v et de u : $u \oplus \delta = v$

$v \ominus u =$ la plus grande différence δ

Principe

Déplacer les coûts tout en préservant l'équivalence

Deux VCSP sont équivalents s'ils ont la même valuation optimale.

Cohérence d'arc

Hypothèse de travail : $\forall u, v \in E, u \preceq v, v \ominus u$ *existe*

Différence δ de v et de u : $u \oplus \delta = v$

$v \ominus u =$ la plus grande différence δ

Principe

Déplacer les coûts tout en préservant l'équivalence

Deux VCSP sont équivalents s'ils ont la même valuation optimale.

Deux opérations :

- projection
- extension

Projection

- on déplace les poids d'une contrainte binaire w_{x_i,x_j} vers une contrainte unaire w_{x_i} :
  - $\forall b \in d_{x_j}, w_{x_i,x_j}(a,b) \leftarrow w_{x_i,x_j}(a,b) \ominus v$
  - $w_{x_i}(a) \leftarrow w_{x_i}(a) \oplus v$

Projection

- on déplace les poids d'une contrainte binaire w_{x_i,x_j} vers une contrainte unaire w_{x_i} :
  - $\forall b \in d_{x_j}, w_{x_i,x_j}(a,b) \leftarrow w_{x_i,x_j}(a,b) \ominus v$
  - $w_{x_i}(a) \leftarrow w_{x_i}(a) \oplus v$
- on déplace les poids d'une contrainte unaire w_{x_i} vers $w_\emptyset$:
  - $\forall a \in d_{x_i}, w_{x_i}(a) \leftarrow w_{x_i}(a) \ominus v$
  - $w_\emptyset \leftarrow w_\emptyset \oplus v$

Extension

- on déplace les poids d'une contrainte unaire w_{x_i} vers une contrainte binaire w_{x_i,x_j} :
  - $w_{x_i}(a) \leftarrow w_{x_i}(a) \ominus v$
  - $\forall b \in d_{x_j}, w_{x_i,x_j}(a, b) \leftarrow w_{x_i,x_j}(a, b) \oplus v$

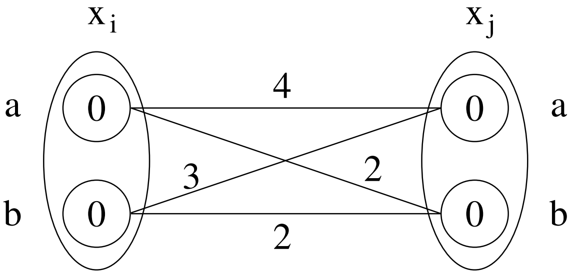
Extension

- on déplace les poids d'une contrainte unaire w_{x_i} vers une contrainte binaire w_{x_i,x_j} :
  - $w_{x_i}(a) \leftarrow w_{x_i}(a) \ominus v$
  - $\forall b \in d_{x_j}, w_{x_i,x_j}(a,b) \leftarrow w_{x_i,x_j}(a,b) \oplus v$
- on déplace les poids de $w_\emptyset$ vers une contrainte unaire w_{x_i} :
  - $w_\emptyset \leftarrow w_\emptyset \ominus v$
  - $\forall a \in d_{x_i}, w_{x_i}(a) \leftarrow w_{x_i}(a) \oplus v$

Cohérence d'arc (1)

$\top = 5$

Projection

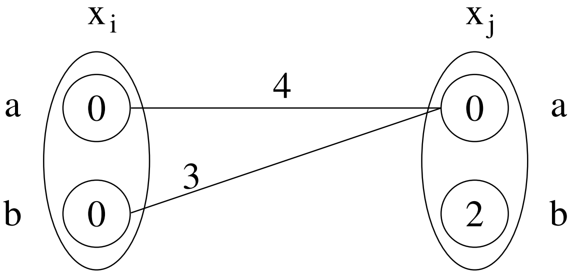


$w_\emptyset = 0$

Cohérence d'arc (1)

$\top = 5$

Projection

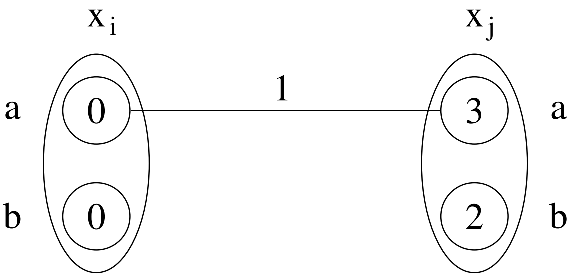


$w_\emptyset = 0$

Cohérence d'arc (1)

$\top = 5$

Projection

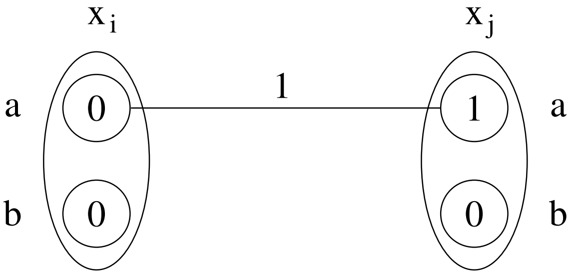


$w_\emptyset = 0$

Cohérence d'arc (1)

$\top = 5$

Projection

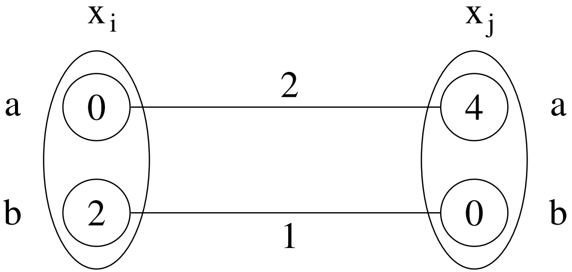


$w_\emptyset = 2$

Cohérence d'arc (2)

$\top = 5$

Extension

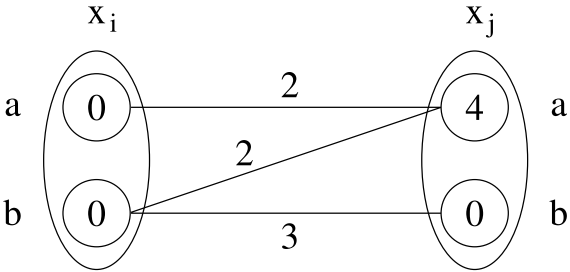


$w_\emptyset = 0$

Cohérence d'arc (2)

$\top = 5$

Extension

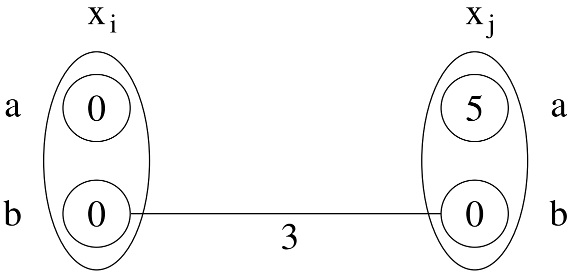


$w_\emptyset = 0$

Cohérence d'arc (2)

$\top = 5$

Projection

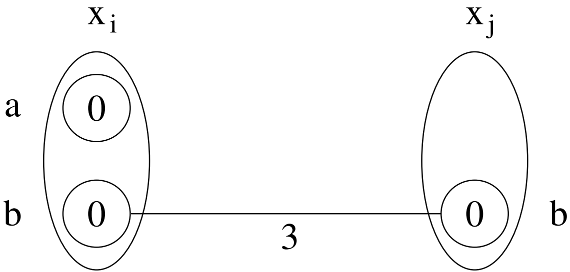


$w_\emptyset = 0$

Cohérence d'arc (2)

$\top = 5$

Suppression d'une valeur



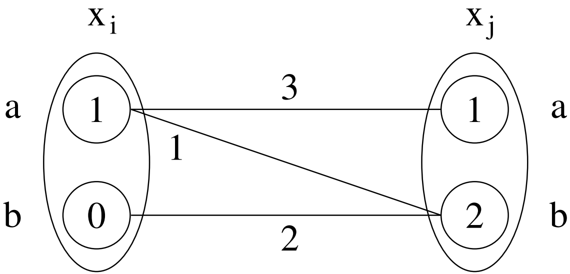
$w_\emptyset = 0$

Cohérence d'arc

Problèmes posés :

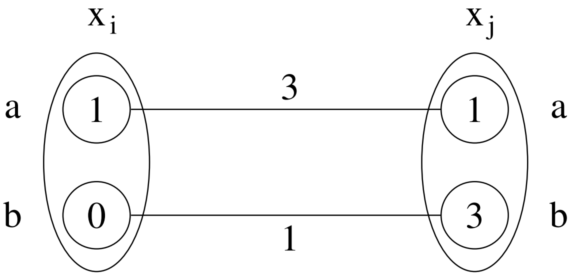
- Terminaison

Problème de terminaison



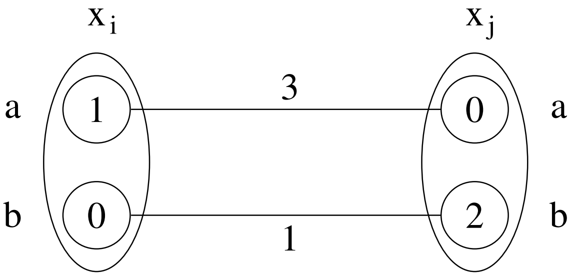
$w_\emptyset = 0$

Problème de terminaison



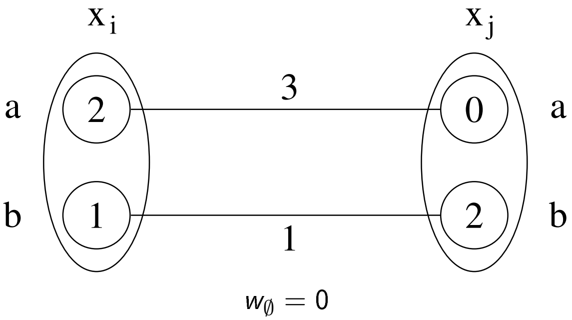
$w_\emptyset = 0$

Problème de terminaison

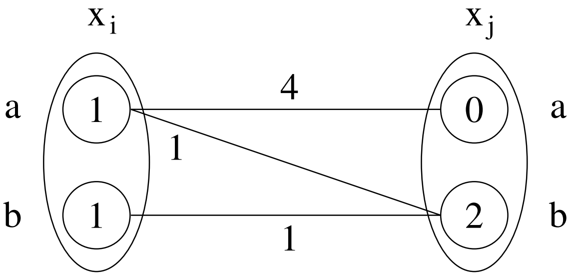


$w_{\emptyset} = 1$

Problème de terminaison

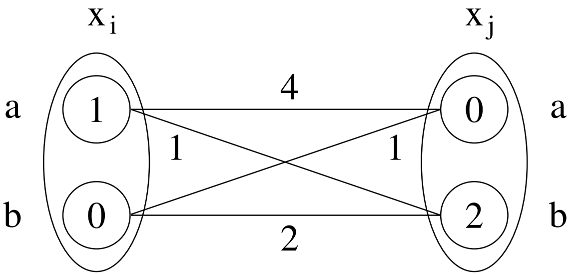


Problème de terminaison



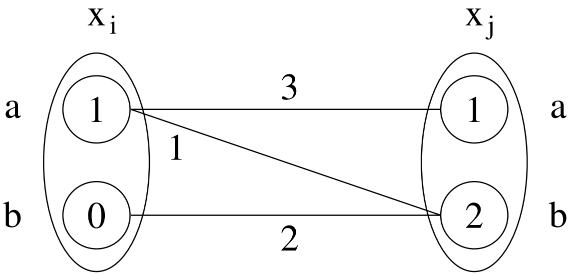
$w_\emptyset = 0$

Problème de terminaison



$w_\emptyset = 0$

Problème de terminaison



$w_\emptyset = 0$

Problème

Retour à la case départ

Cohérence d'arc

Problèmes posés :

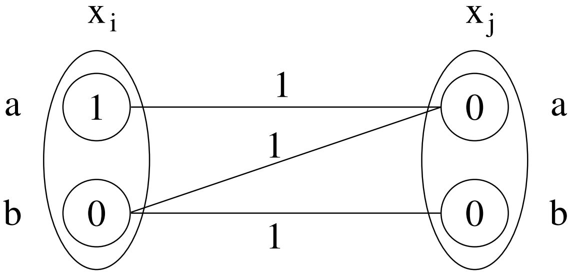
- Terminaison
  - Limiter les propagations
  - Orienter les contraintes

Cohérence d'arc

Problèmes posés :

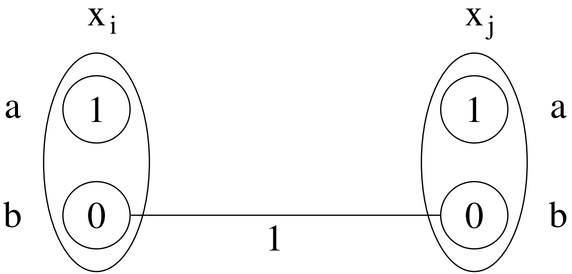
- Terminaison
  - Limiter les propagations
  - Orienter les contraintes
- Fermeture

Problème de fermeture



$$w_{\emptyset} = 0$$

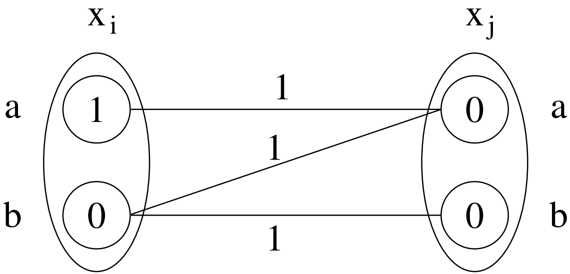
Problème de fermeture



$$w_\emptyset = 0$$

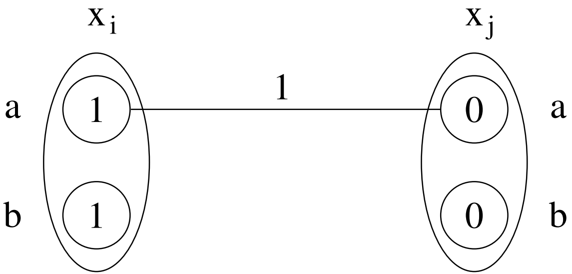
$$\Rightarrow \text{Minorant} = 0$$

Problème de fermeture



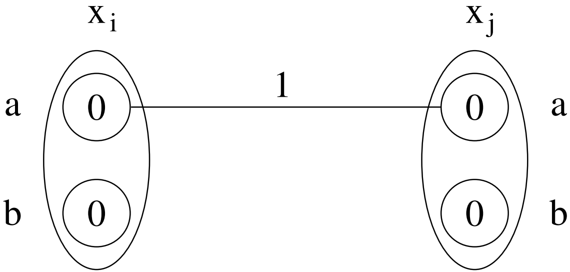
$$w_{\emptyset} = 0$$

Problème de fermeture



$$w_\emptyset = 0$$

Problème de fermeture



$$w_\emptyset = 1$$

$\Rightarrow$Minorant $= 1$

Cohérence d'arc

Problèmes posés :

- Terminaison
  - Limiter les propagations
  - Orienter les contraintes
- Fermeture
  - Pas d'unicité
  - Trouver la meilleure fermeture est un problème NP-difficile !

Cohérence d'arc

Plusieurs variantes :

- Arc-Consistency (AC*)
- Full Directional Arc-Consistency (FDAC*)
- Existential Directional Arc-Consistency (EDAC*)
- ...

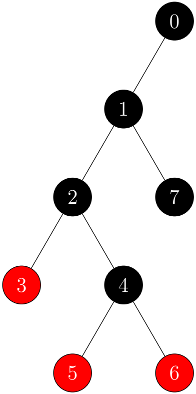
Hiérarchie

$$
\begin{tikzcd}
 & \text{NC*} \arrow[dl] \arrow[dr] & \\
\text{AC*} \arrow[dr] & & \text{DAC*} \arrow[dl] \\
 & \text{FDAC*} \arrow[d] & \\
 & \text{EDAC*} &
\end{tikzcd}
$$

Hiérarchie

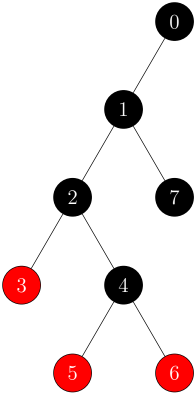
$$
\begin{tikzcd}
 & & \text{NC*} \quad {\color{red}O(nd)} \arrow[dl] \arrow[dr] & \\
{\color{red}O(n^2d^3)} & \text{AC*} \arrow[dr] & & \text{DAC*} \quad {\color{red}O(md^2)} \arrow[dl] \\
 & & \text{FDAC*} \quad {\color{red}O(nmd^3)} \arrow[d] & \\
 & & \text{EDAC*} \quad {\color{red}O(md^2.\max(nd, \top))} &
\end{tikzcd}
$$

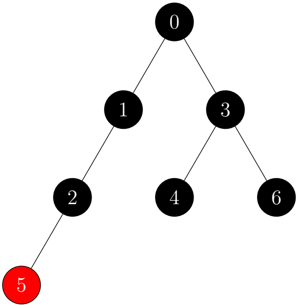
Profondeur d'abord vs le meilleur d'abord



- une borne supérieure anytime
- filtrage incrémental simple

Profondeur d'abord vs le meilleur d'abord

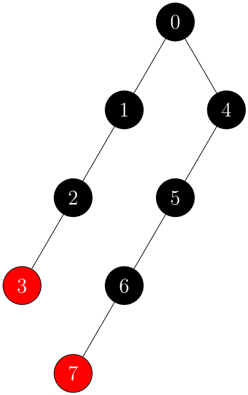




- une borne supérieure anytime
- filtrage incrémental simple

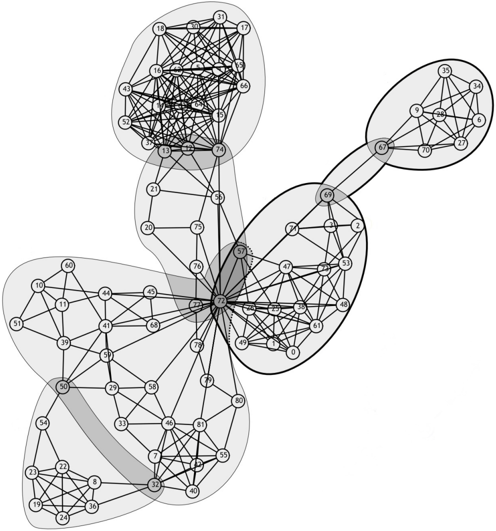
- une borne inférieure anytime
- filtrage incrémental complexe

Hybridation (HBFS)



- une borne inférieure anytime
- une borne supérieure anytime
- filtrage incrémental

Backtracking on Tree-Decomposition



Toulbar2

Ensemble d'outils :

- Filtrage : AC*, DAC*, EDAC*, ...
- Algorithmes de résolution : DFS, BTD, HBFS/DFS, HBFS/BTD, ...
- ...

Solveurs efficaces (HBFS/DFS et HBFS/BTD)

Utilisé pour résoudre différents types de problèmes :

- allocation de fréquence,
- satellite,
- génétique,
- ...

Plan

Localisation d'entrepôts

Problème :

- Instance :
  - un ensemble $E = \{e_1, \ldots, e_n\}$ d'entrepôts avec :
    - n_i : nombre maximum de magasins livrables par l'entrepôt i
    - cm_i : coût de maintenance
  - un ensemble $M = \{m_1, \ldots, m_k\}$ de magasins :
    - ca_{ij} : coût d'approvisionnement du magasin i par l'entrepôt j
- Objectif : trouver un sous-ensemble $E' \subseteq E$ t.q. chaque magasin soit approvisionné par exactement un entrepôt tout en minimisant les coûts

Localisation d'entrepôts

Modélisation COP :

- $X = \{m_1, \ldots, m_k\} \cup \{o_1, \ldots, o_n\} \cup \{c_1, \ldots, c_k\}$,
- $D = \{d_{m_1}, \ldots, d_{m_k}\} \cup \{d_{o_1}, \ldots, d_{o_n}\} \cup \{d_{c_1}, \ldots, d_{c_k}\}$ avec :
  - $\forall i, d_{m_i} = \{1, \ldots, n\}$
  - $\forall i, d_{o_i} = \{0, 1\}$
  - $\forall i, d_{c_i} = \{ca_{i1}, \ldots, ca_{in}\}$
- $C = \{\ldots\}$ avec :
  - $\forall i \in \{1, \ldots, n\},\ Count(\{m_1, \ldots, m_k\}, i, \leq, n_i)$
  - $\forall i \in \{1, \ldots, k\},\ Element(\{o_1, \ldots, o_n\}, m_i, 1)$
  - $\forall i \in \{1, \ldots, k\},\ Element(\{ca_{i1}, \ldots, ca_{in}\}, m_i, c_i)$
- Fonction objectif : $\min \sum_{i=1}^{k} c_i + \sum_{i=1}^{n} cm_i.o_i$

Localisation d'entrepôts

Modélisation VCSP :

- $X = \{m_1, \ldots, m_k\} \cup \{o_1, \ldots, o_n\}$,
- $D = \{d_{m_1}, \ldots, d_{m_k}\} \cup \{d_{o_1}, \ldots, d_{o_n}\}$ avec :
  - $\forall i, d_{m_i} = \{1, \ldots, n\}$
  - $\forall i, d_{o_i} = \{0, 1\}$

Localisation d'entrepôts

Modélisation VCSP :

- $X = \{m_1, \dots, m_k\} \cup \{o_1, \dots, o_n\}$,
- $D = \{d_{m_1}, \dots, d_{m_k}\} \cup \{d_{o_1}, \dots, d_{o_n}\}$ avec :
  - $\forall i, d_{m_i} = \{1, \dots, n\}$
  - $\forall i, d_{o_i} = \{0, 1\}$
- $C = \{\dots\}$ avec :
  - $\forall i \in \{1, \dots, n\},\ Count(\{m_1, \dots, m_k\}, i, \leq, n_i) \rightarrow 0, +\infty$
  - $\forall i \in \{1, \dots, k\},\ Element(\{o_1, \dots, o_n\}, m_i, 1) \rightarrow 0, +\infty$
  -

m_i	$w(m_i)$
1	ca_{i1}
2	ca_{i2}
...	...
k	ca_{ik}

o_i	$w(o_i)$
0	0
1	cm_i

- $SV = (\mathbb{N} \cup \{+\infty\}, \leq, +, 0, +\infty)$