

Classes polynomiales du problème CSP : entre théorie et pratique

Habilitation à diriger des recherches

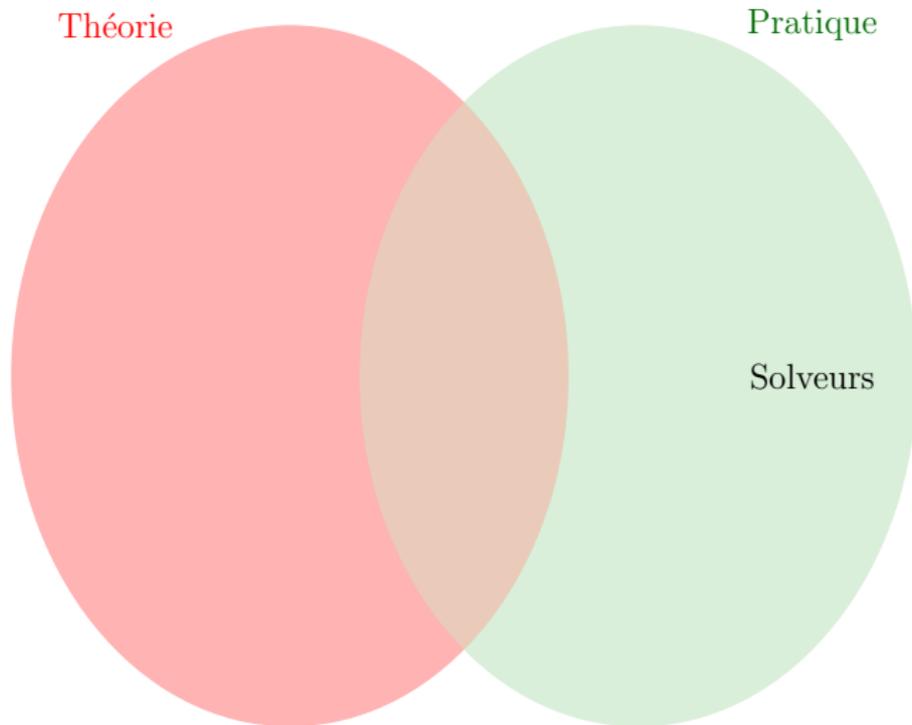
Cyril TERRIOUX

LSIS UMR CNRS 7296

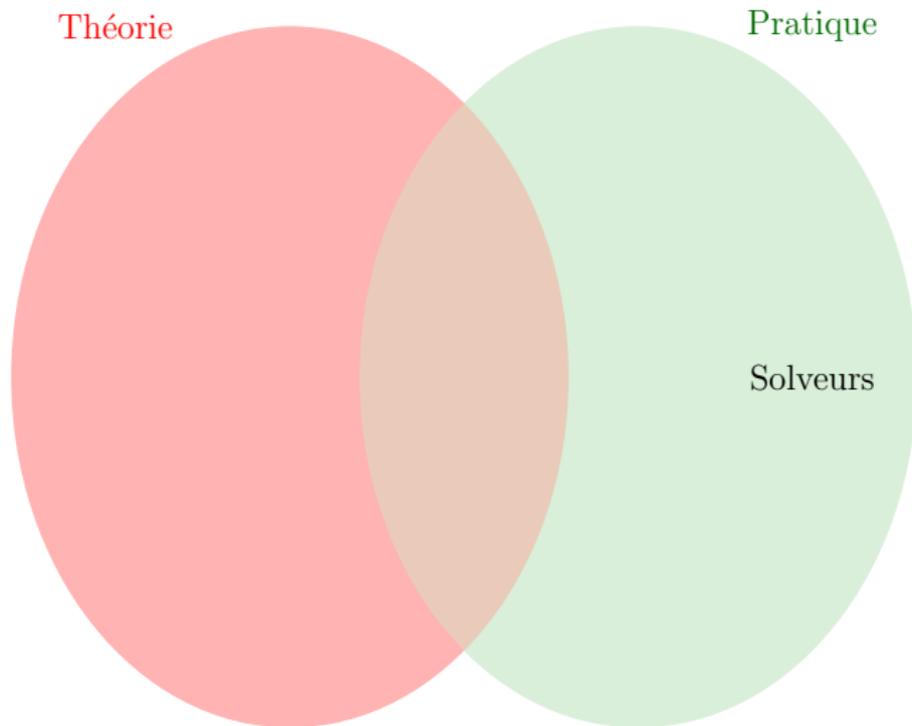


Programmation par contraintes

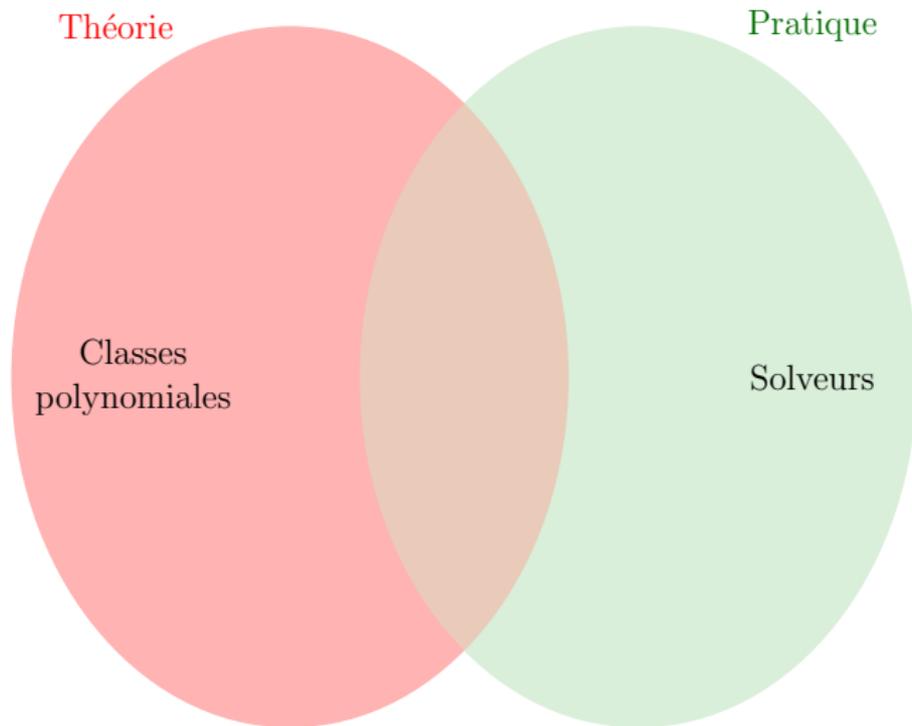
De nombreux solveurs efficaces



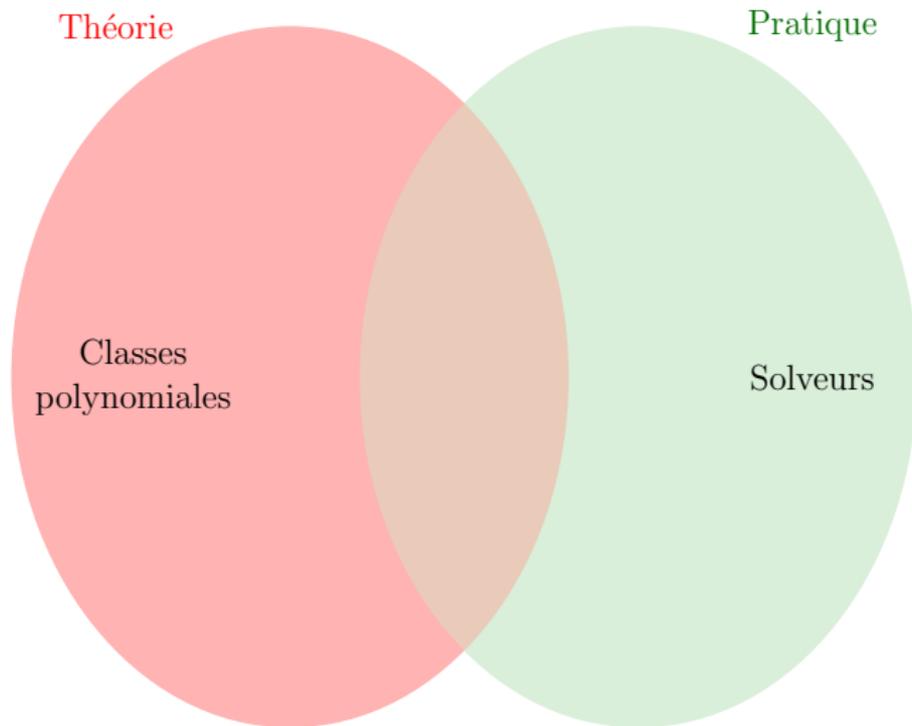
Efficacité sans fondement théorique connu



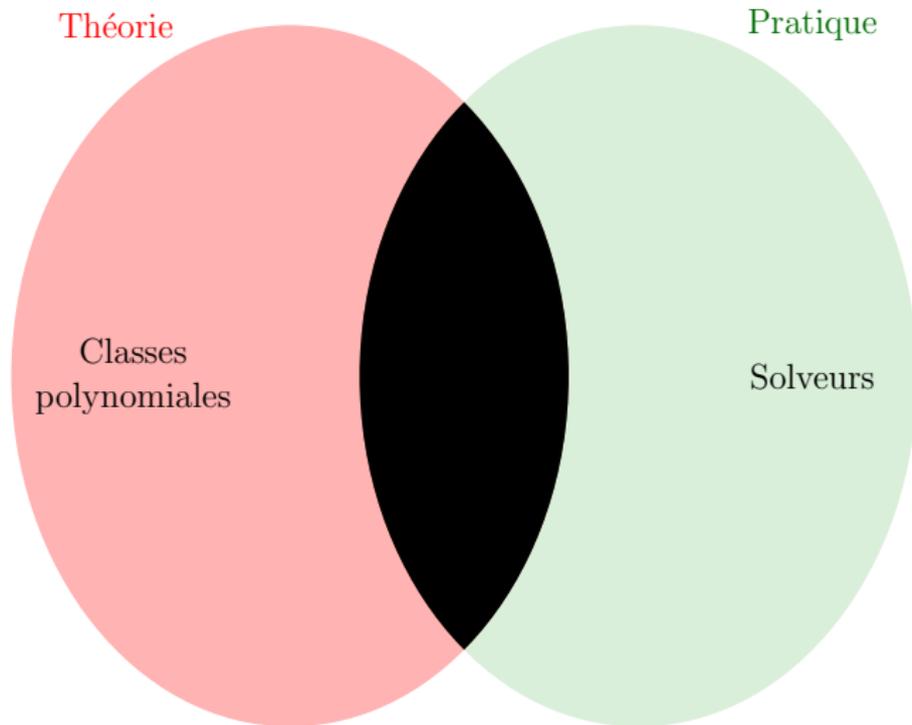
Pas d'exploitation des classes polynomiales



Sans retombée pratique



Mes contributions



- 1 Introduction
- 2 Classes polynomiales hybrides
- 3 Classes polynomiales structurelles
- 4 Conclusion et perspectives

Plan

- 1 Introduction
- 2 Classes polynomiales hybrides
- 3 Classes polynomiales structurelles
- 4 Conclusion et perspectives

Problème de satisfaction de contraintes (CSP)

Problème CSP

- Donnée : une instance CSP $P = (X, D, C)$ avec
 - $X = \{x_1, x_2, \dots, x_n\}$ un ensemble de n variables,
 - $D = \{d_{x_1}, \dots, d_{x_n}\}$ un ensemble de domaines finis,
 - $C = \{c_1, \dots, c_e\}$ un ensemble fini de e contraintes où $c_i = (S(c_i), R(c_i))$ avec :
 - $S(c_i) = \{x_{i_1}, \dots, x_{i_k}\}$ la portée de c_i ,
 - $R(c_i) \subseteq d_{x_{i_1}} \times \dots \times d_{x_{i_k}}$ sa relation de compatibilité.
- Question : P possède-t-elle une solution ?

Problème de satisfaction de contraintes (CSP)

Problème CSP

- Donnée : une instance CSP $P = (X, D, C)$ avec
 - $X = \{x_1, x_2, \dots, x_n\}$ un ensemble de n variables,
 - $D = \{d_{x_1}, \dots, d_{x_n}\}$ un ensemble de domaines finis,
 - $C = \{c_1, \dots, c_e\}$ un ensemble fini de e contraintes où $c_i = (S(c_i), R(c_i))$ avec :
 - $S(c_i) = \{x_{i_1}, \dots, x_{i_k}\}$ la portée de c_i ,
 - $R(c_i) \subseteq d_{x_{i_1}} \times \dots \times d_{x_{i_k}}$ sa relation de compatibilité.
- Question : P possède-t-elle une solution ?

CSP est NP-complet.

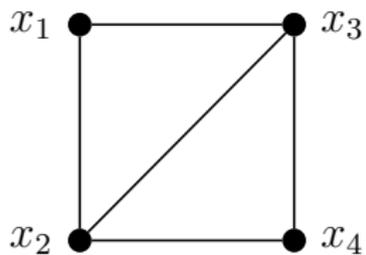
Exemple

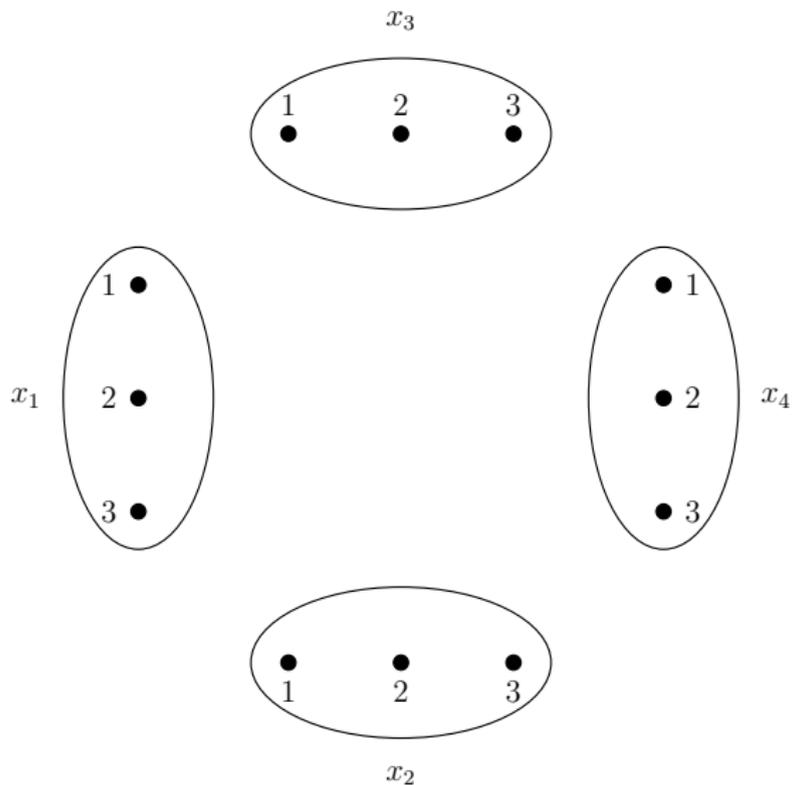
- $X = \{x_1, x_2, x_3, x_4\}$,
- $D = \{d_{x_1}, d_{x_2}, d_{x_3}, d_{x_4}\}$, avec $d_{x_1} = d_{x_2} = d_{x_3} = d_{x_4} = \{1, 2, 3\}$,
- $C = \{c_{12}, c_{13}, c_{23}, c_{24}, c_{34}\}$ avec :
 - $c_{13} = (S(c_{13}), R(c_{13}))$ où
 - $S(c_{13}) = \{x_1, x_3\}$
 - $R(c_{13}) = \{(v_1, v_3) \in d_{x_1} \times d_{x_3} \mid v_1 = v_3\}$

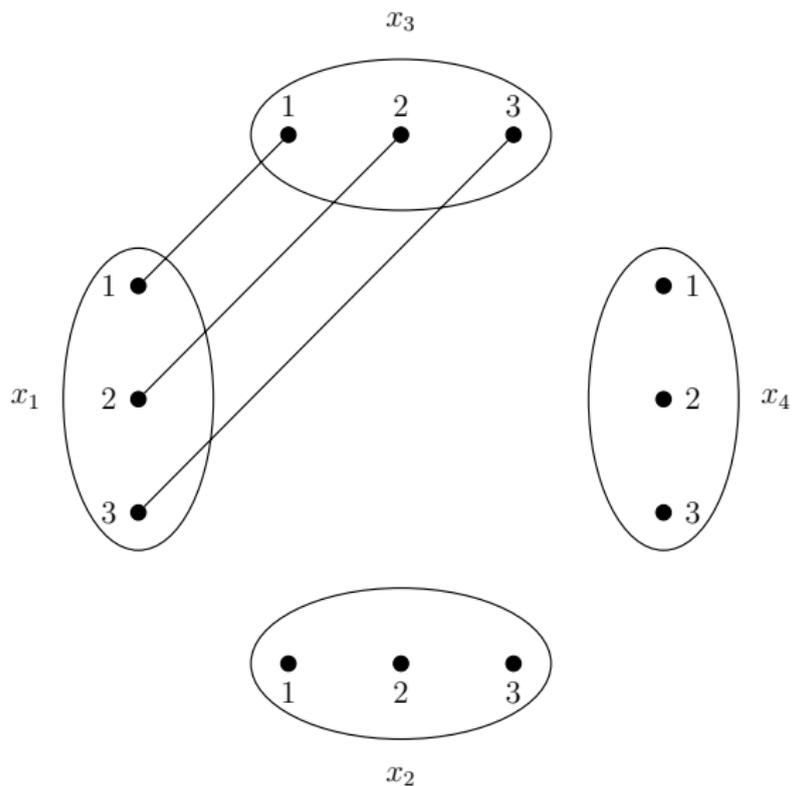
Exemple

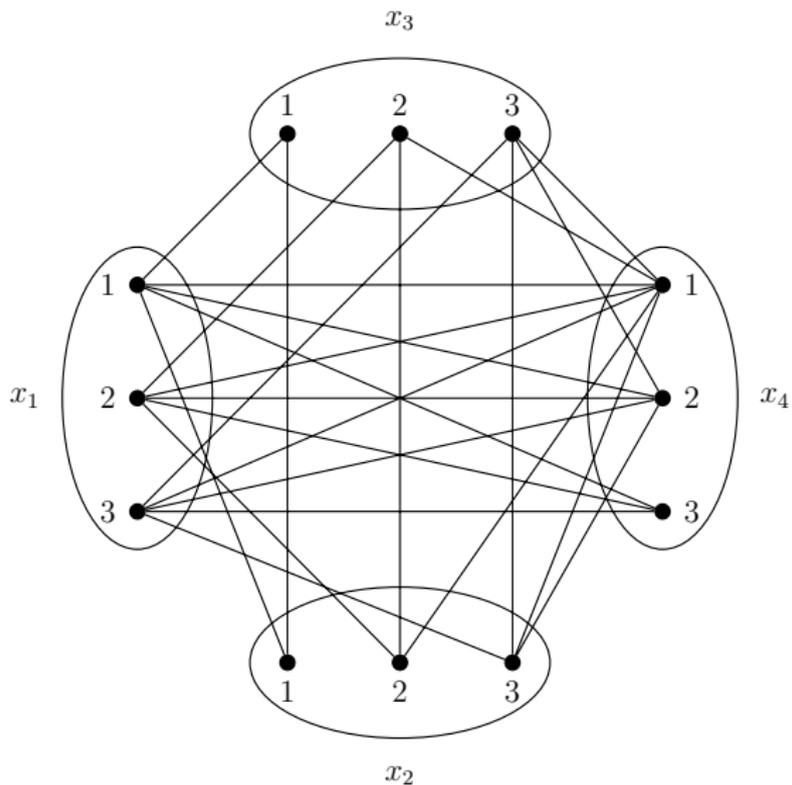
- $X = \{x_1, x_2, x_3, x_4\}$,
- $D = \{d_{x_1}, d_{x_2}, d_{x_3}, d_{x_4}\}$, avec $d_{x_1} = d_{x_2} = d_{x_3} = d_{x_4} = \{1, 2, 3\}$,
- $C = \{c_{12}, c_{13}, c_{23}, c_{24}, c_{34}\}$ avec :
 - $c_{13} = (S(c_{13}), R(c_{13}))$ où
 - $S(c_{13}) = \{x_1, x_3\}$
 - $R(c_{13}) = \{(v_1, v_3) \in d_{x_1} \times d_{x_3} \mid v_1 = v_3\}$
 - $c_{24} = (S(c_{24}), R(c_{24}))$ où
 - $S(c_{24}) = \{x_2, x_4\}$
 - $R(c_{24}) = \{(v_2, v_4) \in d_{x_2} \times d_{x_4} \mid v_2 > v_4\}$
 - ...

Structure : (hyper)graphe (X, C)



Microstructure : graphe (V, E) 

Microstructure : graphe (V, E) 

Microstructure : graphe (V, E) 

Algorithmes de résolution énumératifs

Exploration arborescente exhaustive de l'espace de recherche

Algorithmes de résolution énumératifs

Exploration arborescente exhaustive de l'espace de recherche

Exemples

- Forward-Checking (FC / nFC_i)
- Real Full Look-ahead (RFL)
- Maintaining Arc-Consistency (MAC)

Algorithmes de résolution énumératifs

Exploration arborescente exhaustive de l'espace de recherche

Exemples

- Forward-Checking (FC / nFC_i)
- Real Full Look-ahead (RFL)
- Maintaining Arc-Consistency (MAC)

Au cœur des solveurs

Algorithmes de résolution énumératifs

Exploration arborescente exhaustive de l'espace de recherche

Exemples

- Forward-Checking (FC / nFC_i)
- Real Full Look-ahead (RFL)
- Maintaining Arc-Consistency (MAC)

Au cœur des solveurs

Complexité : $O(S.d^n)$ avec S la taille de l'instance

Algorithmes de résolution énumératifs

Exploration arborescente exhaustive de l'espace de recherche

Exemples

- Forward-Checking (FC / nFC_i)
- Real Full Look-ahead (RFL)
- Maintaining Arc-Consistency (MAC)

Au cœur des solveurs

Complexité : $O(S.d^n)$ avec S la taille de l'instance

Une efficacité souvent remarquable en pratique

- cohérence et filtrage,
- apprentissage et retour en arrière intelligent,
- heuristiques,
- redémarrages,
- ...

Classes polynomiales

Classe polynomiale

Ensemble infini d'instances résolubles par un algorithme polynomial

Classes polynomiales

Classe polynomiale

Ensemble infini d'instances résolubles par un algorithme polynomial

Basées sur des restrictions

Classes polynomiales

Classe polynomiale

Ensemble infini d'instances résolubles par un algorithme polynomial

Basées sur des restrictions

Exemples

- classes structurelles : CSP de largeur arborescente bornée
- classes relationnelles : ZUT
- classes hybrides : BTP

Classes polynomiales

Classe polynomiale

Ensemble infini d'instances résolubles par un algorithme polynomial

Basées sur des restrictions

Exemples

- classes structurelles : CSP de largeur arborescente bornée
- classes relationnelles : ZUT
- classes hybrides : BTP

Quid de la reconnaissance (c.-à.-d. du méta-problème) ?

Classes polynomiales

Classe polynomiale

Ensemble infini d'instances résolubles par un algorithme polynomial

Basées sur des restrictions

Exemples

- classes structurelles : CSP de largeur arborescente bornée
- classes relationnelles : ZUT
- classes hybrides : BTP

Quid de la reconnaissance (c.-à.-d. du méta-problème) ?

Pas toujours nécessaire

Constat

Théorie

- Classes polynomiales : de nombreuses classes proposées

Constat

Théorie

- Classes polynomiales : de nombreuses classes proposées

Pratique

- Classes polynomiales : rarement utiles et utilisées

Constat

Théorie

- Classes polynomiales : de nombreuses classes proposées
- Méthodes énumératives : une complexité en temps exponentielle

Pratique

- Classes polynomiales : rarement utiles et utilisées

Constat

Théorie

- Classes polynomiales : de nombreuses classes proposées
- Méthodes énumératives : une complexité en temps exponentielle

Pratique

- Classes polynomiales : rarement utiles et utilisées
- Méthodes énumératives : une efficacité remarquable

Constat

Théorie

- Classes polynomiales : de nombreuses classes proposées
- Méthodes énumératives : une complexité en temps exponentielle

Fossé

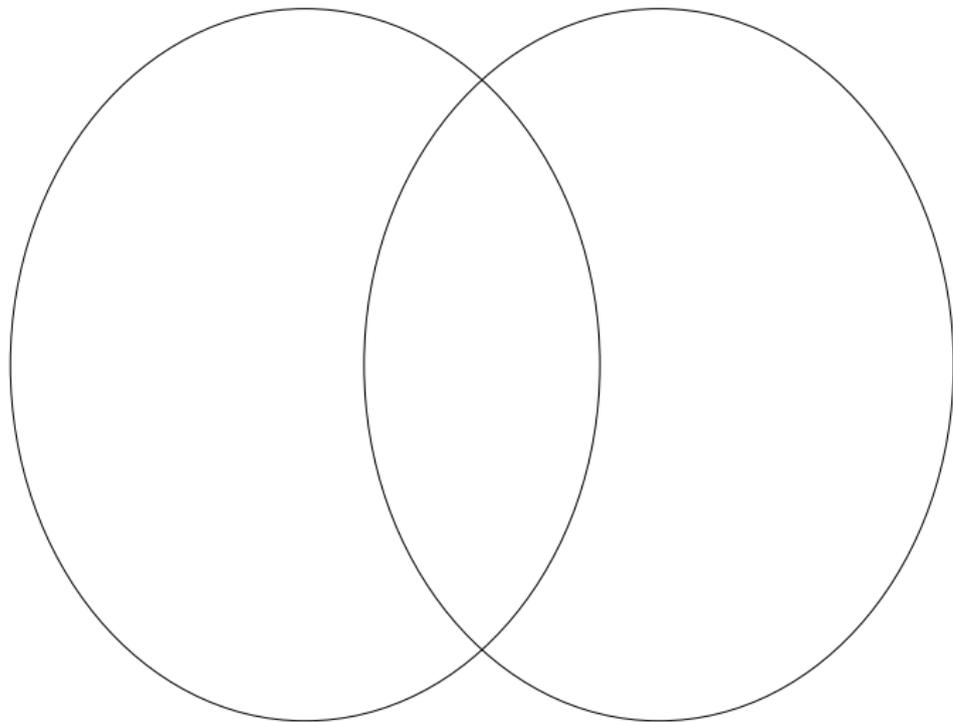
Pratique

- Classes polynomiales : rarement utiles et utilisées
- Méthodes énumératives : une efficacité remarquable

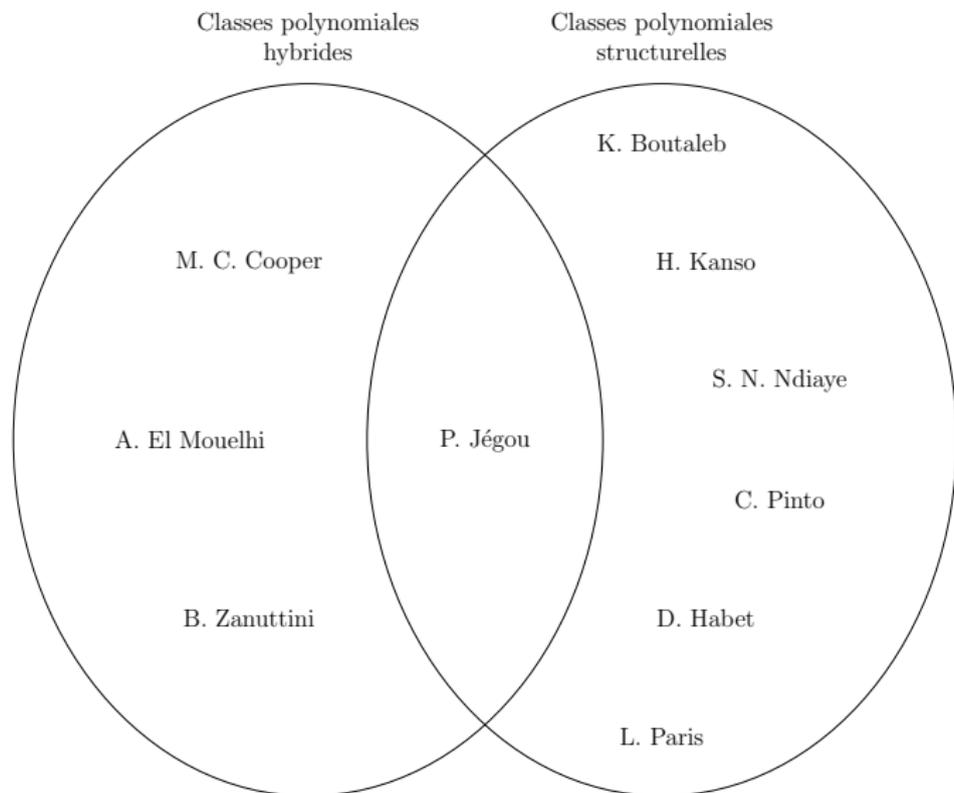
Contributions

Classes polynomiales
hybrides

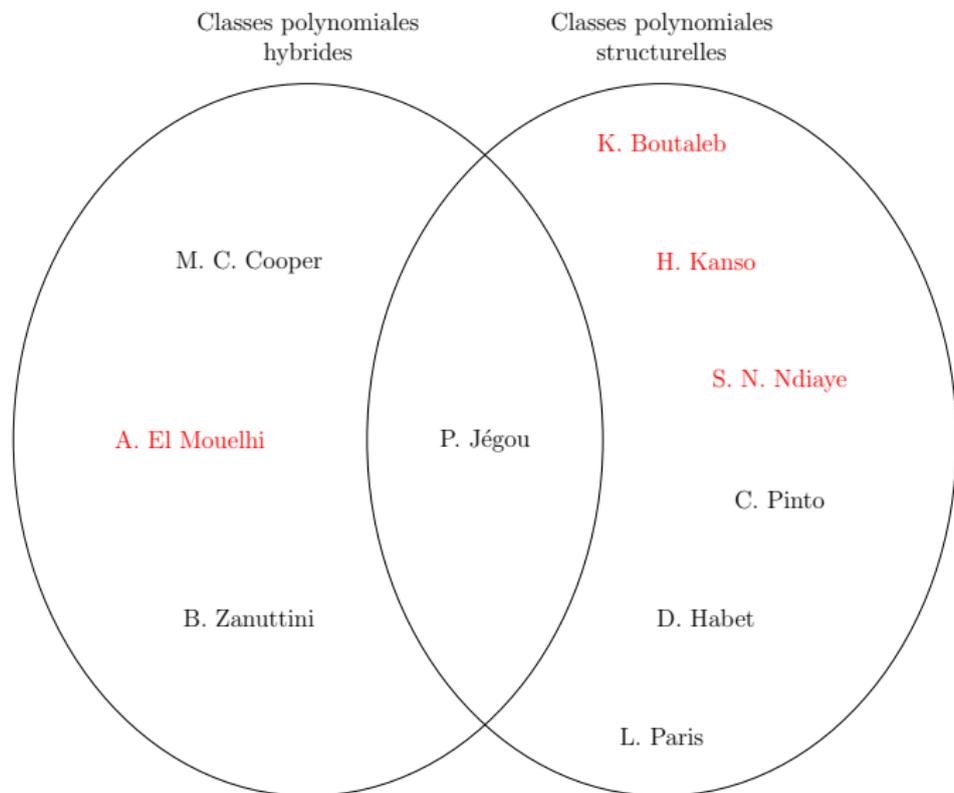
Classes polynomiales
structurelles



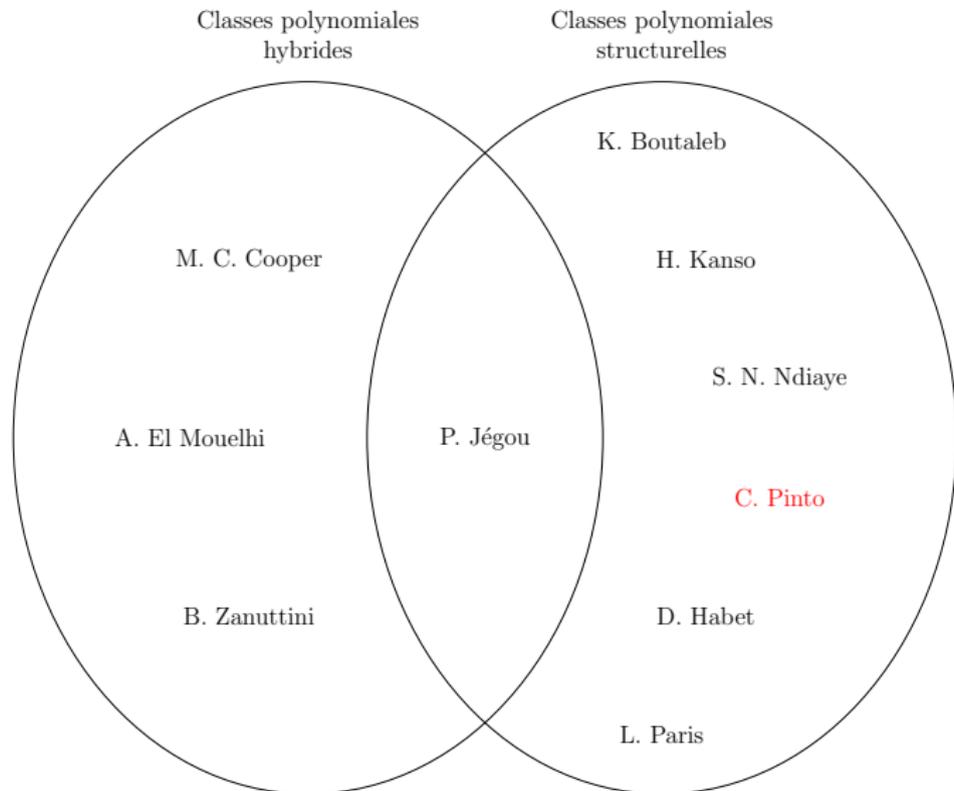
Contributions



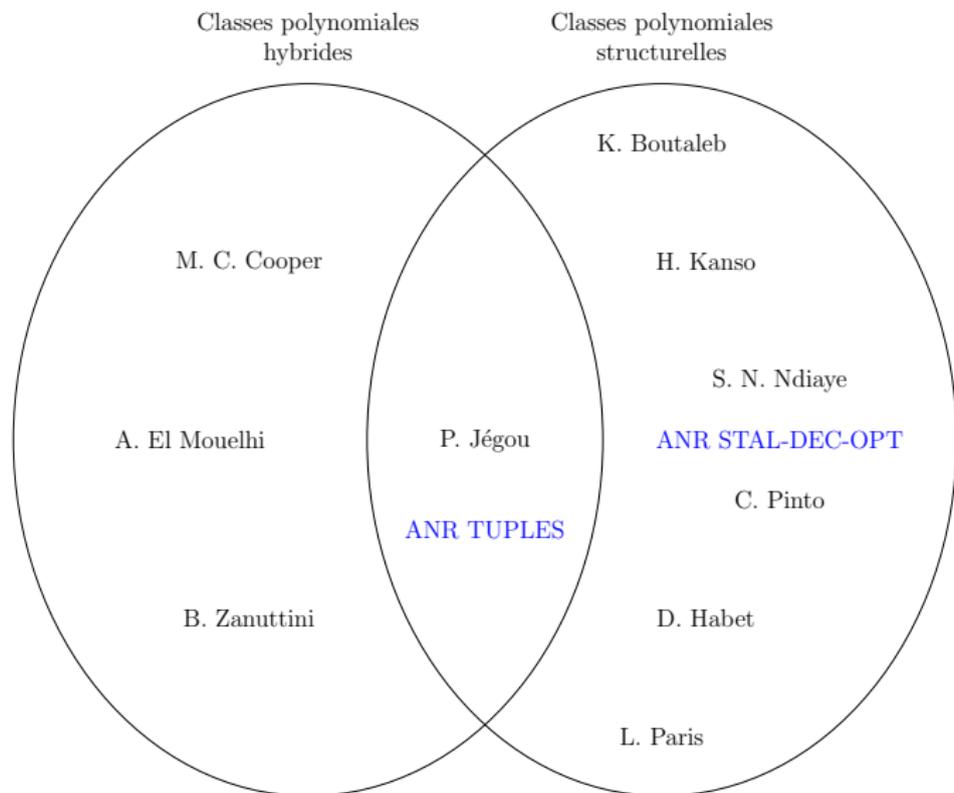
Contributions



Contributions



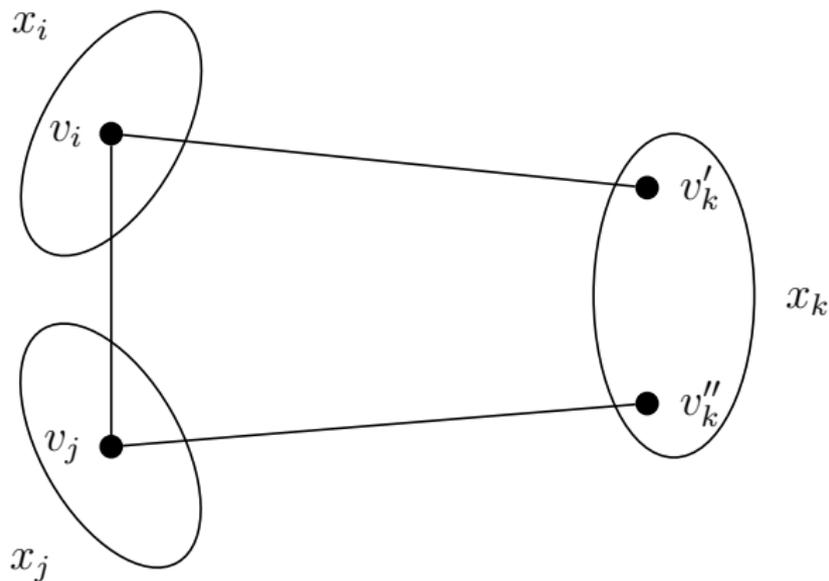
Contributions



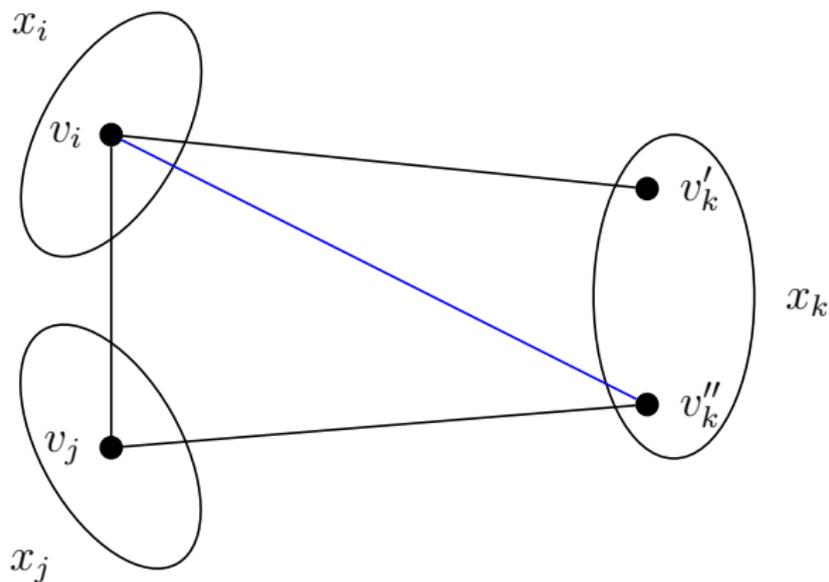
Plan

- 1 Introduction
- 2 Classes polynomiales hybrides**
- 3 Classes polynomiales structurelles
- 4 Conclusion et perspectives

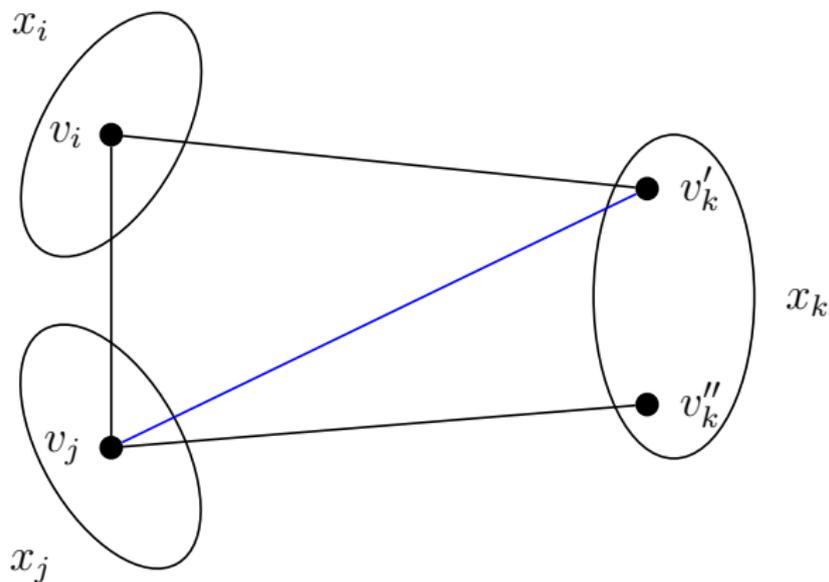
Broken Triangle Property [Cooper et al., 2008, 2010]



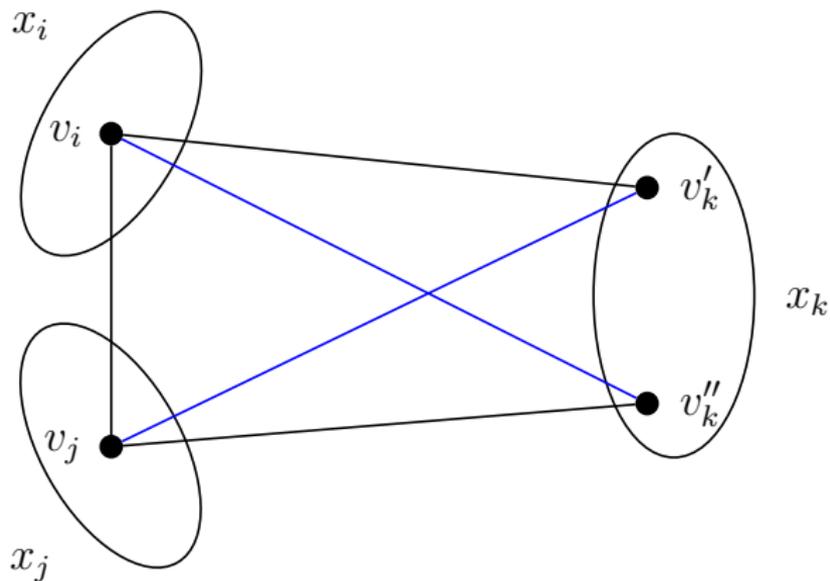
Broken Triangle Property [Cooper et al., 2008, 2010]



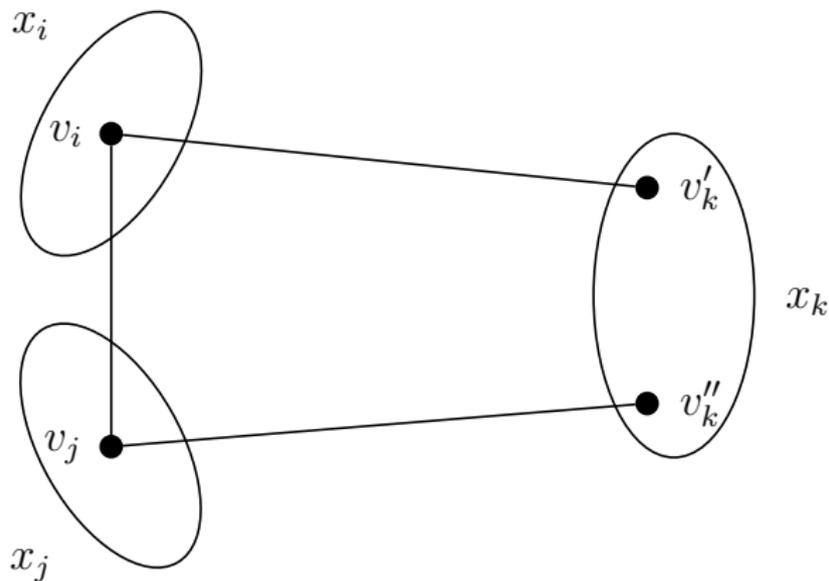
Broken Triangle Property [Cooper et al., 2008, 2010]



Broken Triangle Property [Cooper et al., 2008, 2010]

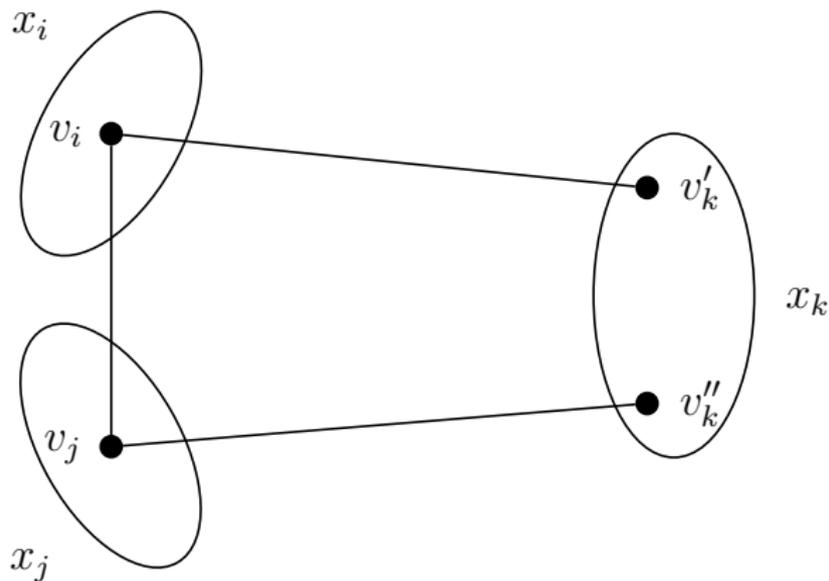


Broken Triangle Property [Cooper et al., 2008, 2010]



Triangle cassé sur x_k par rapport à x_i et x_j

Broken Triangle Property [Cooper et al., 2008, 2010]



Une cause potentielle de retour en arrière

Broken Triangle Property [Cooper et al., 2008, 2010]

BTP

P satisfait BTP par rapport à $<$ si, pour tout x_i, x_j, x_k t.q. $x_i < x_j < x_k$, il n'existe pas de triangle cassé sur x_k par rapport à x_i et x_j .

Broken Triangle Property [Cooper et al., 2008, 2010]

BTP

P satisfait BTP par rapport à $<$ si, pour tout x_i, x_j, x_k t.q. $x_i < x_j < x_k$, il n'existe pas de triangle cassé sur x_k par rapport à x_i et x_j .

Rôle central

Broken Triangle Property [Cooper et al., 2008, 2010]

BTP

P satisfait BTP par rapport à $<$ si, pour tout x_i, x_j, x_k t.q. $x_i < x_j < x_k$, il n'existe pas de triangle cassé sur x_k par rapport à x_i et x_j .

Rôle central

Avantages :

- Reconnaissance polynomiale
- Résolution grâce à AC
- Exploitation implicite par RFL ou MAC

Broken Triangle Property [Cooper et al., 2008, 2010]

BTP

P satisfait BTP par rapport à $<$ si, pour tout x_i, x_j, x_k t.q. $x_i < x_j < x_k$, il n'existe pas de triangle cassé sur x_k par rapport à x_i et x_j .

Rôle central

Avantages :

- Reconnaissance polynomiale
- Résolution grâce à AC
- Exploitation implicite par RFL ou MAC

Inconvénient :

- Condition trop restrictive

Contributions

- Définition de nouvelles classes polynomiales

Contributions

- Définition de nouvelles classes polynomiales :
 - Cliques maximales

Contributions

- Définition de nouvelles classes polynomiales :
 - Cliques maximales

Théorème

FC a une complexité en $O(n^2 d \cdot \omega_{\#}(\mu(P)))$

Contributions

- Définition de nouvelles classes polynomiales :
 - Cliques maximales : CSG^k , toroïdaux, plongeables

Théorème

FC a une complexité en $O(n^2 d \cdot \omega_{\#}(\mu(P)))$

Contributions

- Définition de nouvelles classes polynomiales :
 - Cliques maximales : CSG^k , toroïdaux, plongeables
 - Assouplissement de BTP : ETP, k -BTP, m -wBTP

Contributions

- Définition de nouvelles classes polynomiales :
 - Cliques maximales : CSG^k , toroïdaux, plongeables
 - Assouplissement de BTP : ETP, k -BTP, m -wBTP
- Extension de classes polynomiales existantes :
 - Microstructures pour les instances n -aires

Contributions

- Définition de nouvelles classes polynomiales :
 - Cliques maximales : CSG^k , toroïdaux, plongeables
 - Assouplissement de BTP : ETP, k -BTP, m -wBTP
- Extension de classes polynomiales existantes :
 - Microstructures pour les instances n -aires :
Graphes et transformations de CSP naires en CSP binaires

Contributions

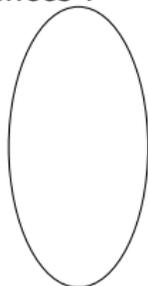
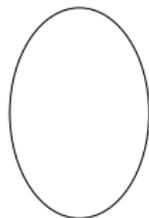
- Définition de nouvelles classes polynomiales :
 - Cliques maximales : CSG^k , toroïdaux, plongeables
 - Assouplissement de BTP : ETP, k -BTP, m -wBTP
- Extension de classes polynomiales existantes :
 - Microstructures pour les instances n -aires : DBTP

Contributions

- Définition de nouvelles classes polynomiales :
 - Cliques maximales : CSG^k , toroïdaux, plongeables
 - Assouplissement de BTP : ETP, k -BTP, m -wBTP
- Extension de classes polynomiales existantes :
 - Microstructures pour les instances n -aires : DBTP
 - Classes cachées

Contributions

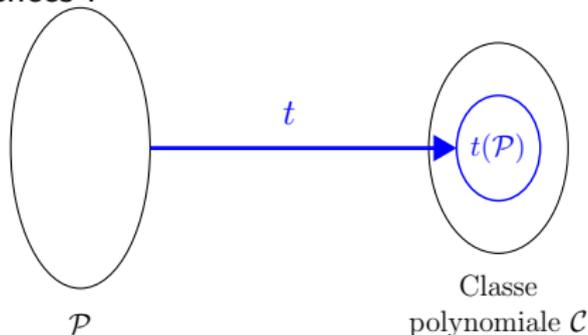
- Définition de nouvelles classes polynomiales :
 - Cliques maximales : CSG^k , toroïdaux, plongeables
 - Assouplissement de BTP : ETP, k -BTP, m -wBTP
- Extension de classes polynomiales existantes :
 - Microstructures pour les instances n -aires : DBTP
 - Classes cachées :

 \mathcal{P} 

Classe
polynomiale \mathcal{C}

Contributions

- Définition de nouvelles classes polynomiales :
 - Cliques maximales : CSG^k , toroïdaux, plongeables
 - Assouplissement de BTP : ETP, k -BTP, m -wBTP
- Extension de classes polynomiales existantes :
 - Microstructures pour les instances n -aires : DBTP
 - Classes cachées :



Contributions

- Définition de nouvelles classes polynomiales :
 - Cliques maximales : CSG^k , toroïdaux, plongeables
 - Assouplissement de BTP : ETP, k -BTP, m -wBTP
- Extension de classes polynomiales existantes :
 - Microstructures pour les instances n -aires : DBTP
 - Classes cachées : BTP^{AC} , $DBTP^{AC}$, ...

Contributions

- Définition de nouvelles classes polynomiales :
 - Cliques maximales : CSG^k , toroïdaux, plongeables
 - Assouplissement de BTP : ETP, k -BTP, m -wBTP
- Extension de classes polynomiales existantes :
 - Microstructures pour les instances n -aires : DBTP
 - Classes cachées : BTP^{AC} , $DBTP^{AC}$, ...
- Exploitations des classes polynomiales :
 - Explications de l'efficacité des solveurs

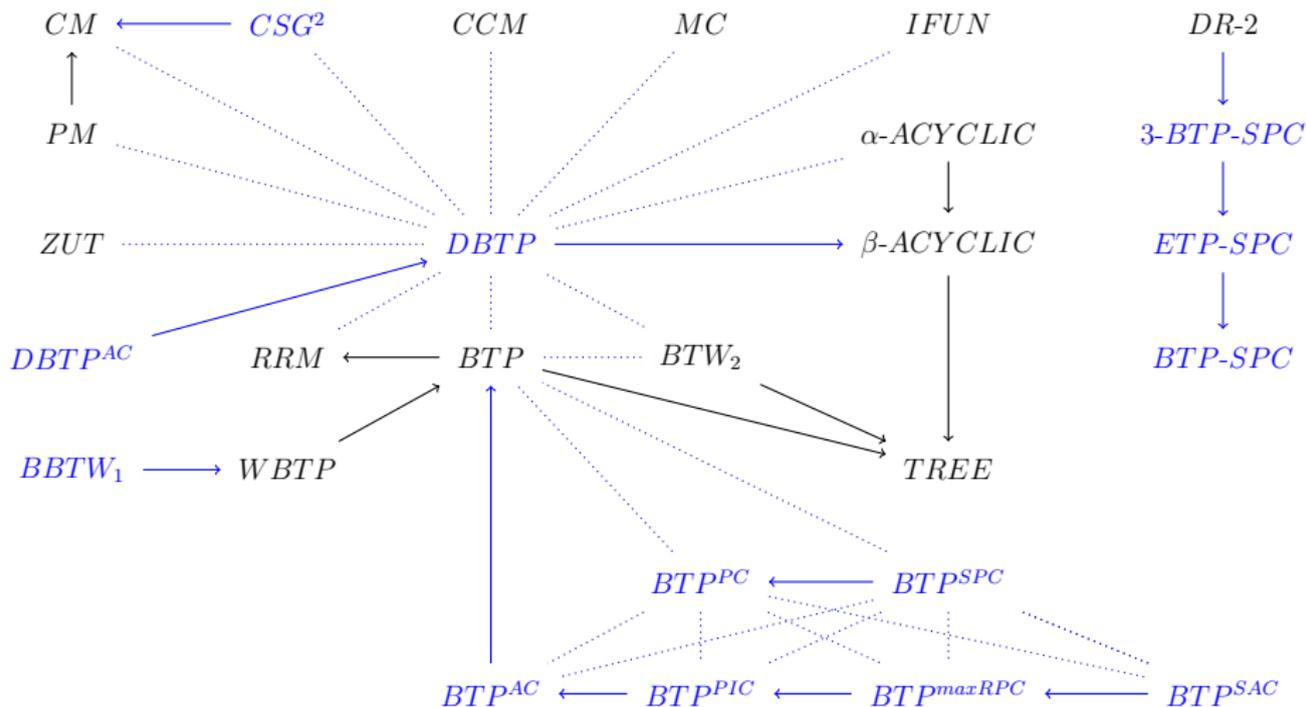
Contributions

- Définition de nouvelles classes polynomiales :
 - Cliques maximales : CSG^k , toroïdaux, plongeables
 - Assouplissement de BTP : ETP, k -BTP, m -wBTP
- Extension de classes polynomiales existantes :
 - Microstructures pour les instances n -aires : DBTP
 - Classes cachées : BTP^{AC} , $DBTP^{AC}$, ...
- Exploitations des classes polynomiales :
 - Explications de l'efficacité des solveurs
 - Fusion de valeurs

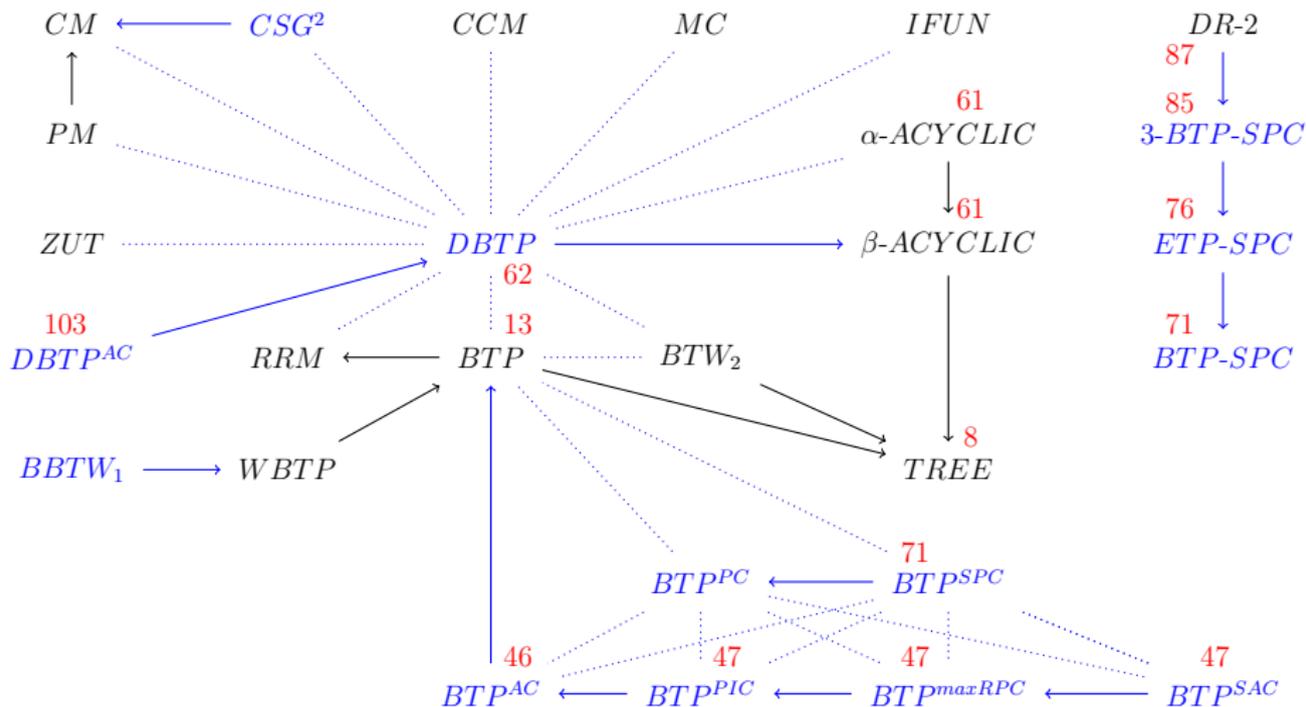
Contributions

- Définition de nouvelles classes polynomiales :
 - Cliques maximales : CSG^k , toroïdaux, plongeables
 - Assouplissement de BTP : ETP, k -BTP, m -wBTP
- Extension de classes polynomiales existantes :
 - Microstructures pour les instances n -aires : DBTP
 - Classes cachées : BTP^{AC} , $DBTP^{AC}$, ...
- Exploitations des classes polynomiales :
 - Explications de l'efficacité des solveurs
 - Fusion de valeurs

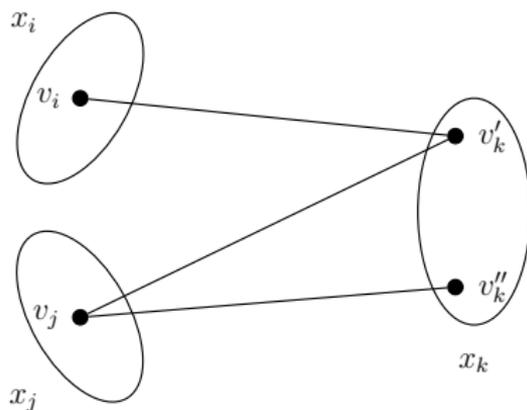
Entre théorie et pratique



Entre théorie et pratique



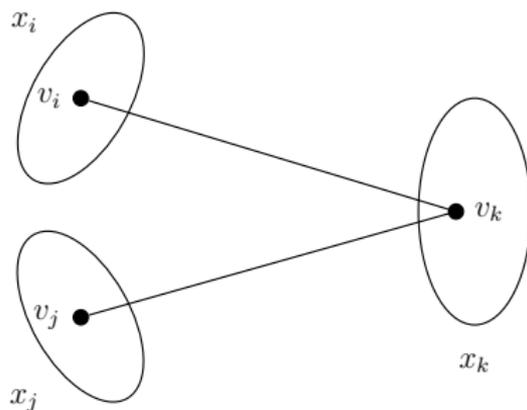
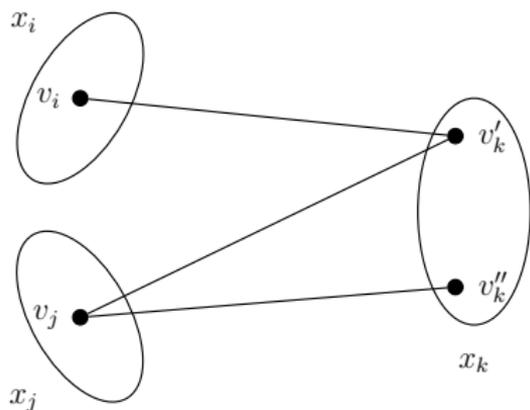
Fusion de valeurs



Définition (Fusion)

Fusionner $v'_k, v''_k \in d_{x_k}$ consiste à remplacer v'_k, v''_k par une nouvelle valeur v_k compatible avec toute valeur v compatible avec v'_k ou v''_k

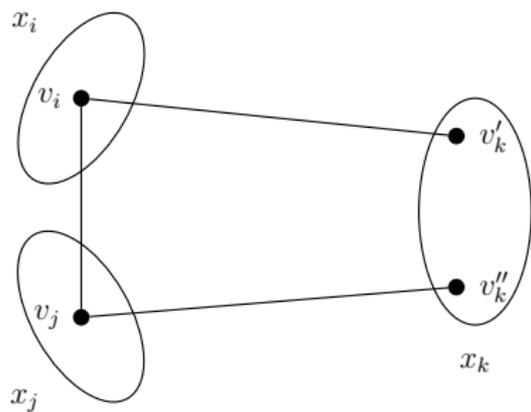
Fusion de valeurs



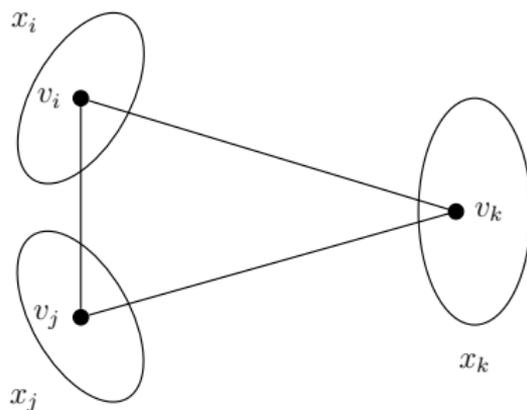
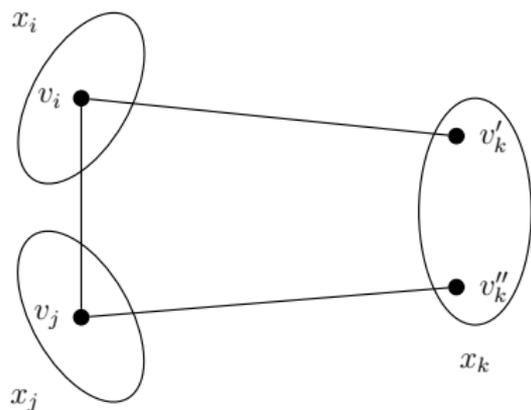
Définition (Fusion)

Fusionner $v'_k, v''_k \in d_{x_k}$ consiste à remplacer v'_k, v''_k par une nouvelle valeur v_k compatible avec toute valeur v compatible avec v'_k ou v''_k

Fusion de valeurs

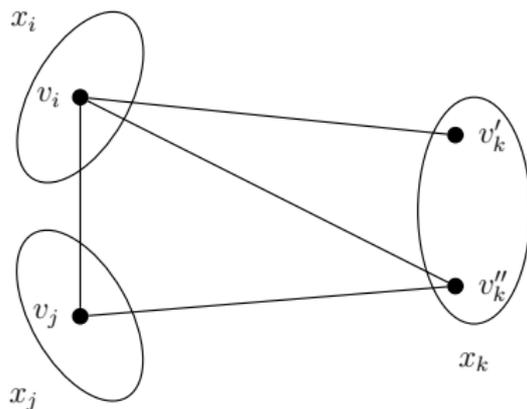


Fusion de valeurs



Problème : création d'une solution

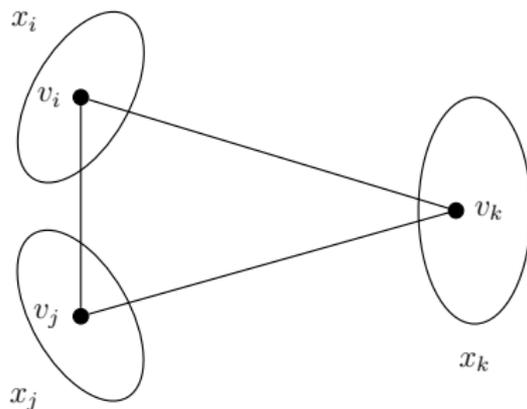
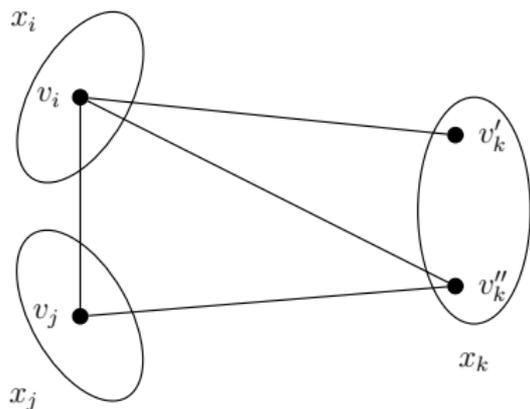
Fusion de valeurs et BTP



Théorème

Fusionner $v'_k, v''_k \in d_{x_k}$ en l'absence de triangle cassé sur v'_k, v''_k préserve la satisfiabilité.

Fusion de valeurs et BTP



Théorème

Fusionner $v'_k, v''_k \in d_{x_k}$ en l'absence de triangle cassé sur v'_k, v''_k préserve la satisfiabilité.

Fusion de valeurs et BTP

Propriété

La fusion basée sur BTP généralise :

- la substitution de voisinage [Freuder, 1991]
- l'interchangeabilité virtuelle [Likitvivatanavong and Yap, 2013]

Fusion de valeurs et BTP

Propriété

La fusion basée sur BTP généralise :

- la substitution de voisinage [Freuder, 1991]
- l'interchangeabilité virtuelle [Likitvivatanavong and Yap, 2013]

fusions

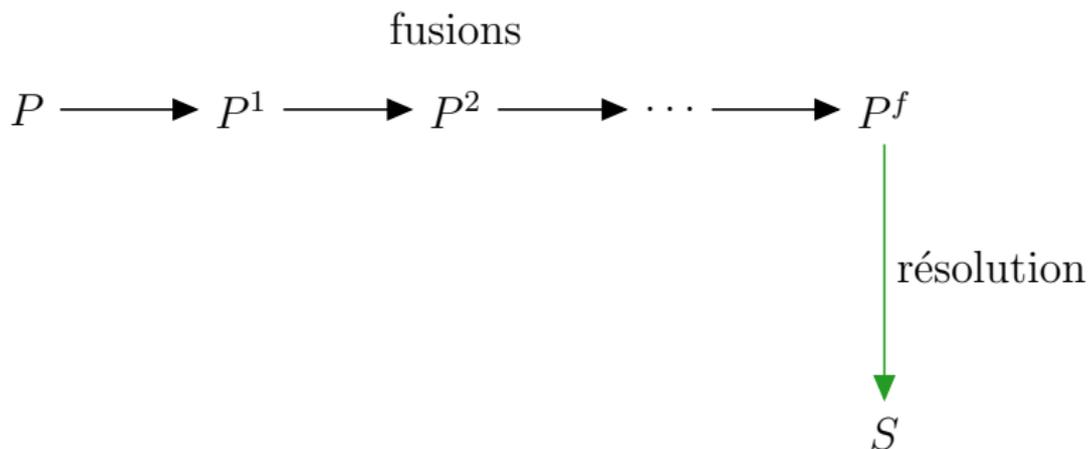
$$P \longrightarrow P^1 \longrightarrow P^2 \longrightarrow \dots \longrightarrow P^f$$

Fusion de valeurs et BTP

Propriété

La fusion basée sur BTP généralise :

- la substitution de voisinage [Freuder, 1991]
- l'interchangeabilité virtuelle [Likitvatanavong and Yap, 2013]

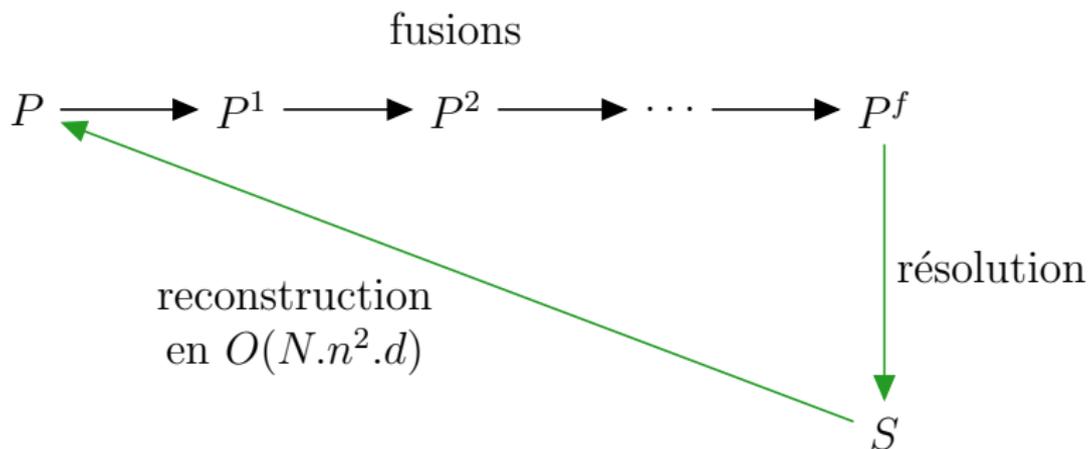


Fusion de valeurs et BTP

Propriété

La fusion basée sur BTP généralise :

- la substitution de voisinage [Freuder, 1991]
- l'interchangeabilité virtuelle [Likitvatanavong and Yap, 2013]



Fusion de valeurs et BTP

Importance de l'ordre

Fusion de valeurs et BTP

Importance de l'ordre

Pas d'unicité du point fixe

Fusion de valeurs et BTP

Importance de l'ordre

Pas d'unicité du point fixe

Maximiser le nombre de fusions est NP-difficile [Cooper et al., 2015]

Fusion de valeurs et BTP

Importance de l'ordre

Pas d'unicité du point fixe

Maximiser le nombre de fusions est NP-difficile [Cooper et al., 2015]

En pratique :

- Exploitation d'heuristiques
- Fusions jusqu'à l'obtention d'un point fixe

La fusion en pratique

Instances	n	#valeurs	NS	VI	BTP	1-wBTP
bqwh-15-106-22_ext	106	365	3	0	3	3
fpsol2-i-1-62	496	30 752	13 847	13 847	13 847	13 847
lei450-15b-13	450	5 850	0	24	24	24
haystacks-28	784	21 952	0	27	27	27
geom-40-2-ext	40	80	1	1	1	14
enddr1-10-by-5-8	50	5 720	422	18	422	426
os-taillard-4-95-8	16	3 034	2 696	0	2 704	2 764
BlackHole-4-7-h-5_ext	112	2 102	697	887	896	932
graph5	200	7 416	0	96	134	2 345
scen2	200	8 004	0	298	341	1 211
driverlogw-02c-sat_ext	301	1 161	8	2	8	74
scen06-sub4	44	1 856	0	0	78	639
ehi-85-297-22_ext	297	2 079	0	0	891	1 058
composed-25-1-2-8_ext	33	330	0	0	0	6
queenAttacking-9	82	6 576	0	0	0	110
qwh-10-57-8_ext	100	613	0	0	0	18
super-os-taillard-4-6	32	4 712	0	0	0	727
langford-4-17	68	4 624	0	0	0	1 258

La fusion en pratique

Instances	n	#valeurs	NS	VI	BTP	1-wBTP
bqwh-15-106-22_ext	106	365	3	0	3	3
fpsol2-i-1-62	496	30 752	13 847	13 847	13 847	13 847
lei450-15b-13	450	5 850	0	24	24	24
haystacks-28	784	21 952	0	27	27	27
geom-40-2-ext	40	80	1	1	1	14
enddr1-10-by-5-8	50	5 720	422	18	422	426
os-taillard-4-95-8	16	3 034	2 696	0	2 704	2 764
BlackHole-4-7-h-5_ext	112	2 102	697	887	896	932
graph5	200	7 416	0	96	134	2 345
scen2	200	8 004	0	298	341	1 211
driverlogw-02c-sat_ext	301	1 161	8	2	8	74
scen06-sub4	44	1 856	0	0	78	639
ehi-85-297-22_ext	297	2 079	0	0	891	1 058
composed-25-1-2-8_ext	33	330	0	0	0	6
queenAttacking-9	82	6 576	0	0	0	110
qwh-10-57-8_ext	100	613	0	0	0	18
super-os-taillard-4-6	32	4 712	0	0	0	727
langford-4-17	68	4 624	0	0	0	1 258

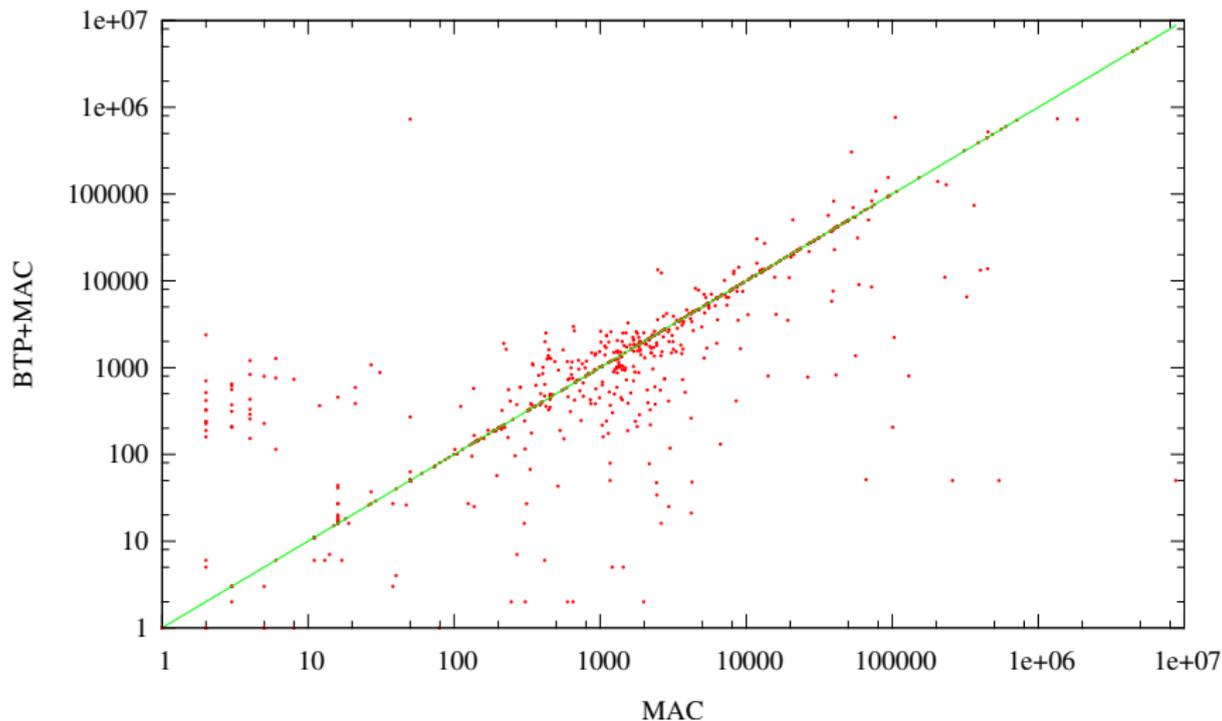
La fusion en pratique

Instances	n	#valeurs	NS	VI	BTP	1-wBTP
bqwh-15-106-22_ext	106	365	3	0	3	3
fpsol2-i-1-62	496	30 752	13 847	13 847	13 847	13 847
lei450-15b-13	450	5 850	0	24	24	24
haystacks-28	784	21 952	0	27	27	27
geom-40-2-ext	40	80	1	1	1	14
enddr1-10-by-5-8	50	5 720	422	18	422	426
os-taillard-4-95-8	16	3 034	2 696	0	2 704	2 764
BlackHole-4-7-h-5_ext	112	2 102	697	887	896	932
graph5	200	7 416	0	96	134	2 345
scen2	200	8 004	0	298	341	1 211
driverlogw-02c-sat_ext	301	1 161	8	2	8	74
scen06-sub4	44	1 856	0	0	78	639
ehi-85-297-22_ext	297	2 079	0	0	891	1 058
composed-25-1-2-8_ext	33	330	0	0	0	6
queenAttacking-9	82	6 576	0	0	0	110
qwh-10-57-8_ext	100	613	0	0	0	18
super-os-taillard-4-6	32	4 712	0	0	0	727
langford-4-17	68	4 624	0	0	0	1 258

La fusion en pratique

Instances	n	#valeurs	NS	VI	BTP	1-wBTP
bqwh-15-106-22_ext	106	365	3	0	3	3
fpsol2-i-1-62	496	30 752	13 847	13 847	13 847	13 847
lei450-15b-13	450	5 850	0	24	24	24
haystacks-28	784	21 952	0	27	27	27
geom-40-2-ext	40	80	1	1	1	14
enddr1-10-by-5-8	50	5 720	422	18	422	426
os-taillard-4-95-8	16	3 034	2 696	0	2 704	2 764
BlackHole-4-7-h-5_ext	112	2 102	697	887	896	932
graph5	200	7 416	0	96	134	2 345
scen2	200	8 004	0	298	341	1 211
driverlogw-02c-sat_ext	301	1 161	8	2	8	74
scen06-sub4	44	1 856	0	0	78	639
ehi-85-297-22_ext	297	2 079	0	0	891	1 058
composed-25-1-2-8_ext	33	330	0	0	0	6
queenAttacking-9	82	6 576	0	0	0	110
qwh-10-57-8_ext	100	613	0	0	0	18
super-os-taillard-4-6	32	4 712	0	0	0	727
langford-4-17	68	4 624	0	0	0	1 258

La fusion en pratique



Plan

- 1 Introduction
- 2 Classes polynomiales hybrides
- 3 Classes polynomiales structurelles**
- 4 Conclusion et perspectives

Classes polynomiales structurelles

Un intérêt pratique plus marqué

Classes polynomiales structurelles

Un intérêt pratique plus marqué

Instances possédant des propriétés structurelles plus fréquentes

Classes polynomiales structurelles

Un intérêt pratique plus marqué

Instances possédant des propriétés structurelles plus fréquentes

Programmation dynamique [Bertelè et Brioschi, 1972]

Diviser pour régner

Classes polynomiales structurelles

Un intérêt pratique plus marqué

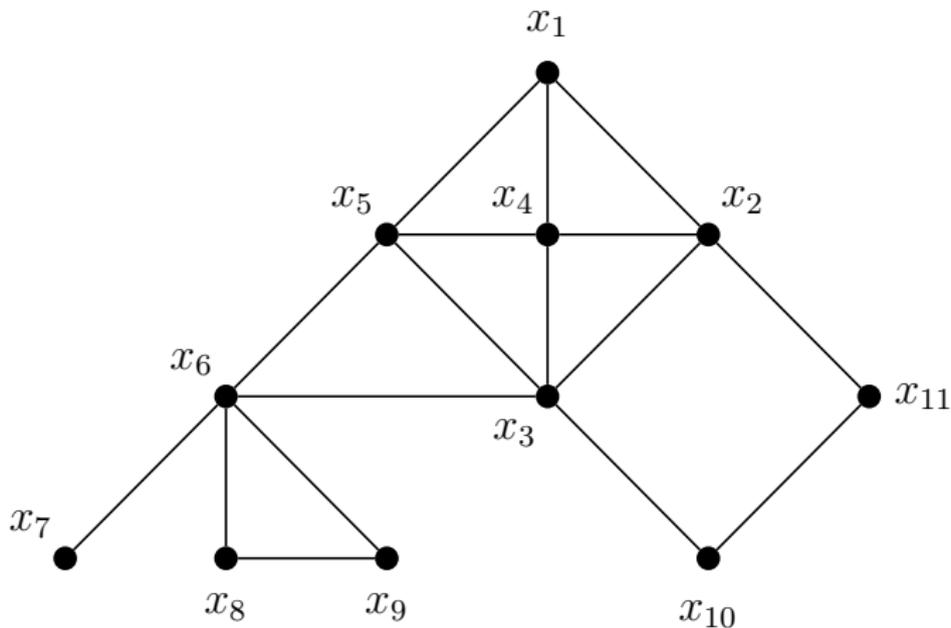
Instances possédant des propriétés structurelles plus fréquentes

Programmation dynamique [Bertelè et Brioschi, 1972]

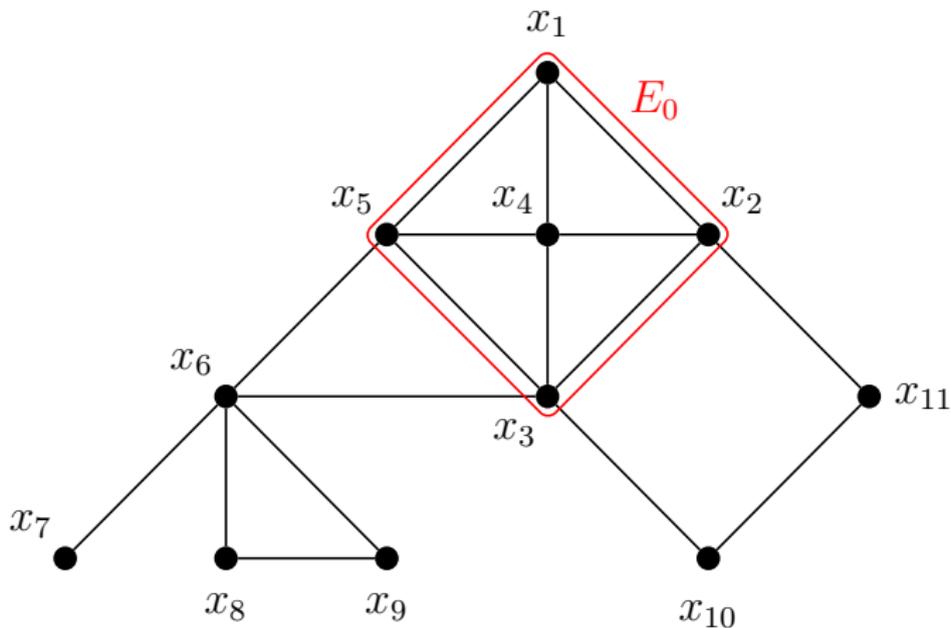
Diviser pour régner

Différentes notions de décompositions

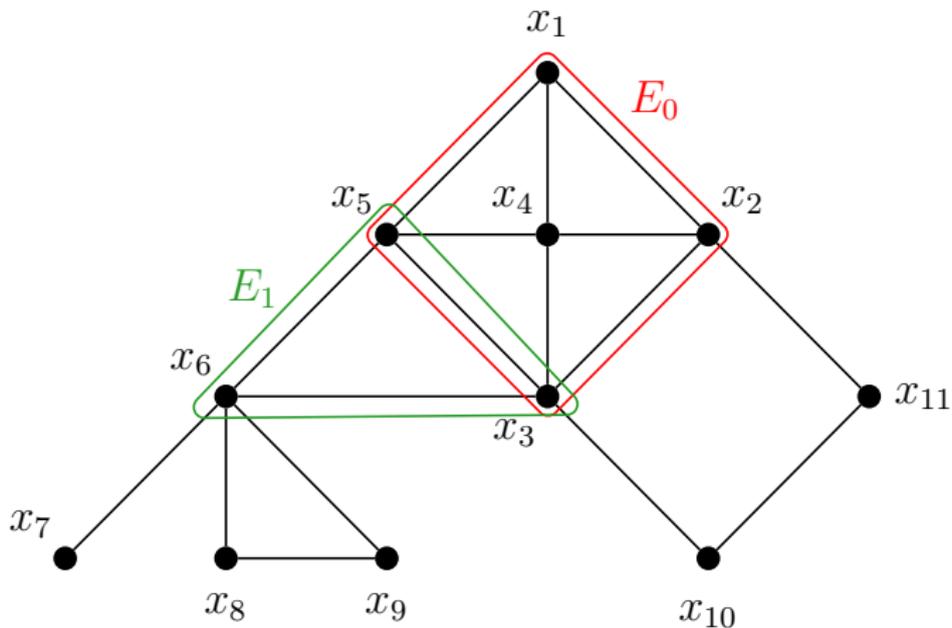
Décomposition arborescente [Robertson et Seymour, 1986]



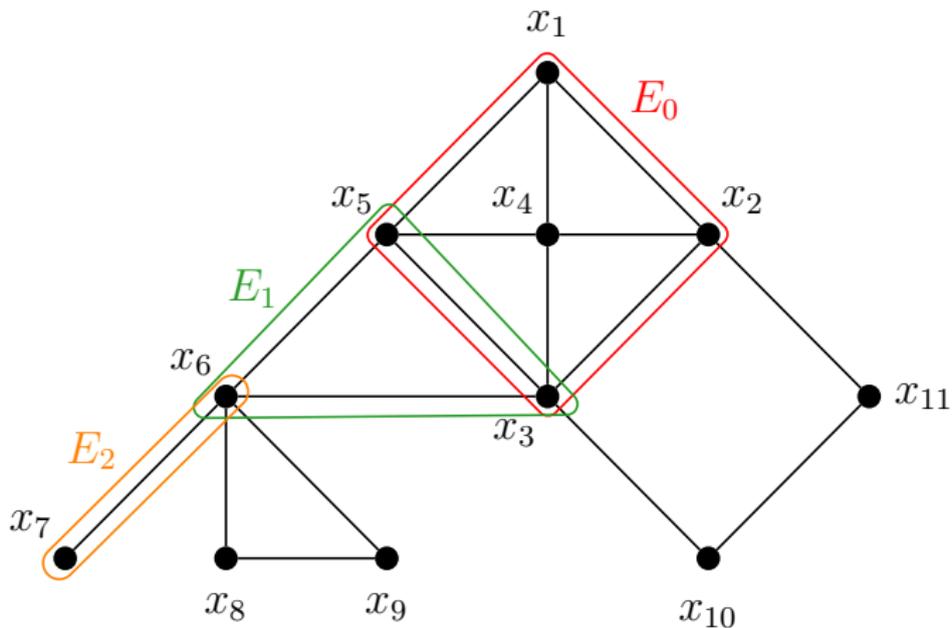
Décomposition arborescente [Robertson et Seymour, 1986]



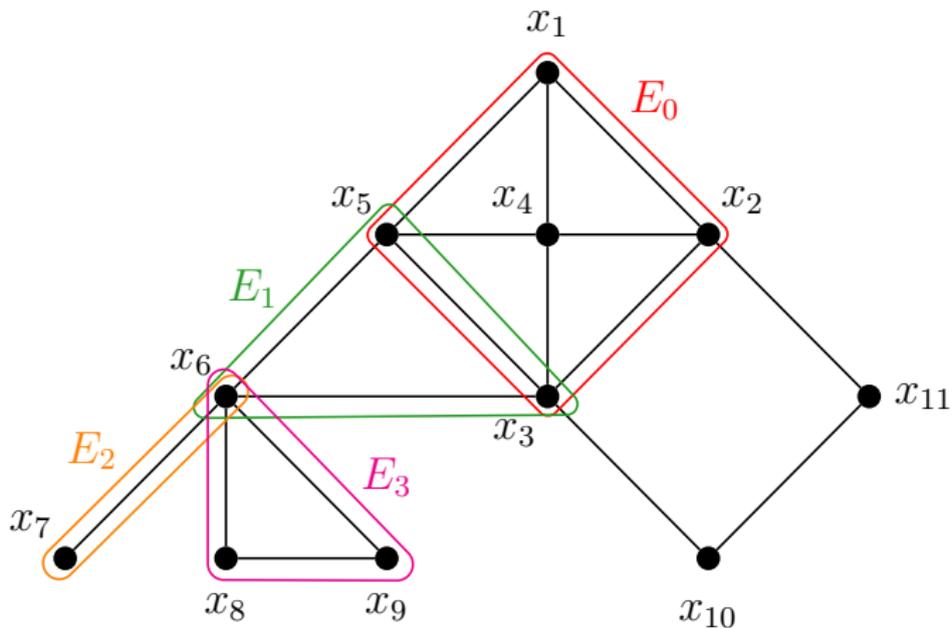
Décomposition arborescente [Robertson et Seymour, 1986]



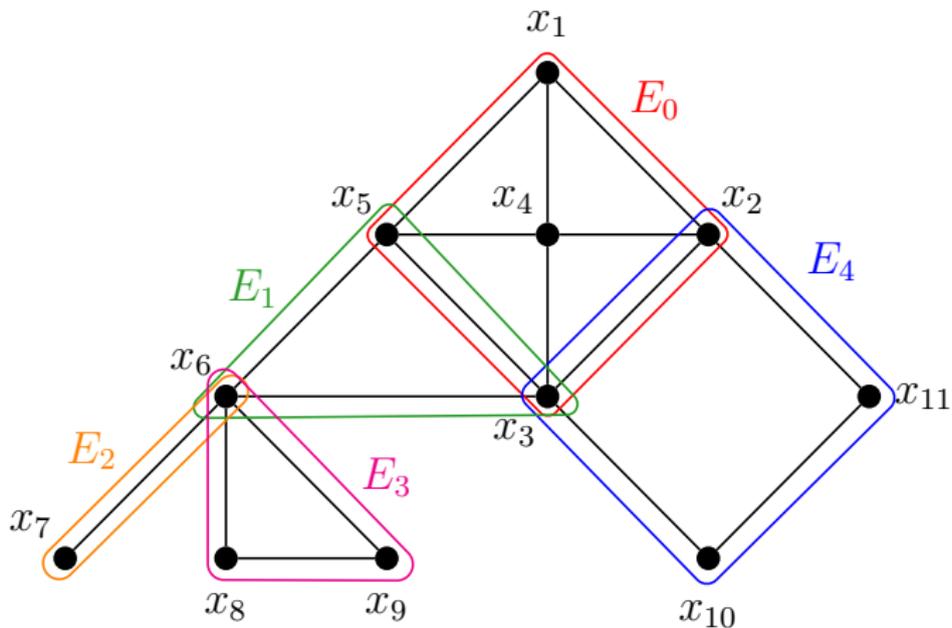
Décomposition arborescente [Robertson et Seymour, 1986]



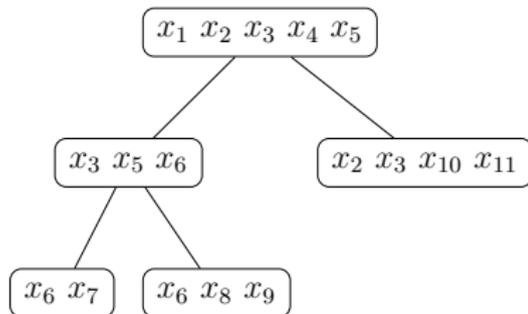
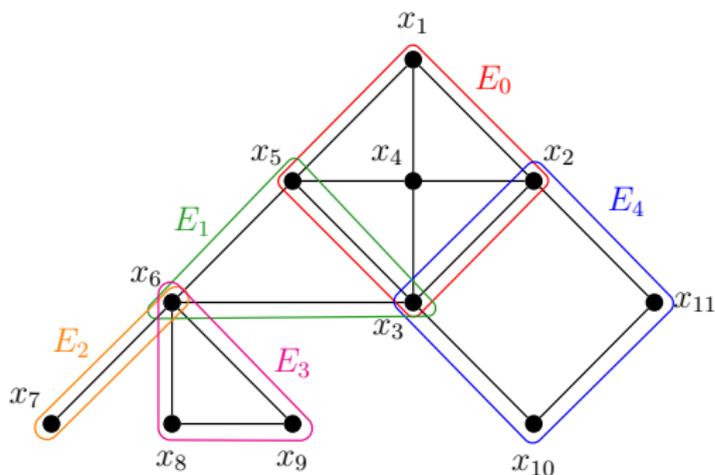
Décomposition arborescente [Robertson et Seymour, 1986]



Décomposition arborescente [Robertson et Seymour, 1986]

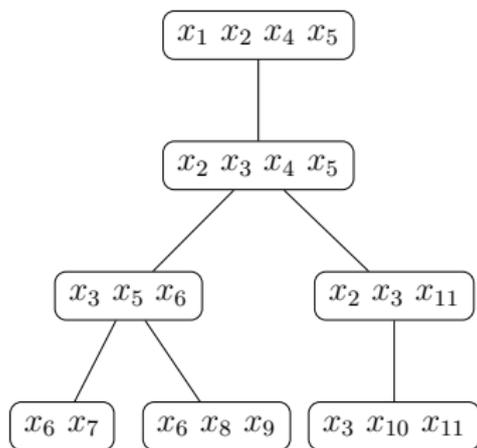
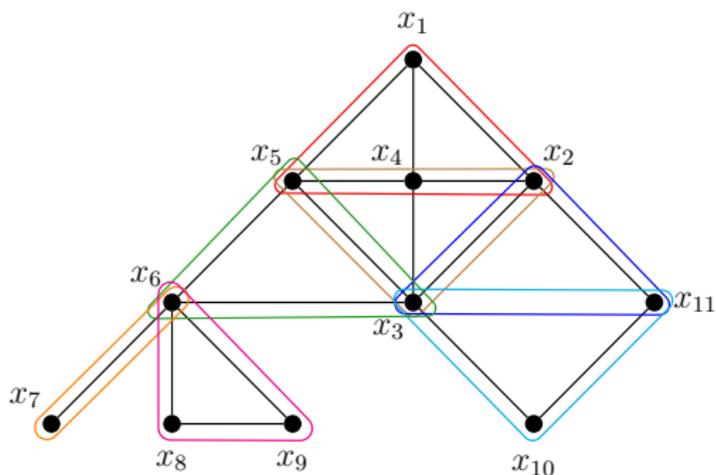


Décomposition arborescente [Robertson et Seymour, 1986]



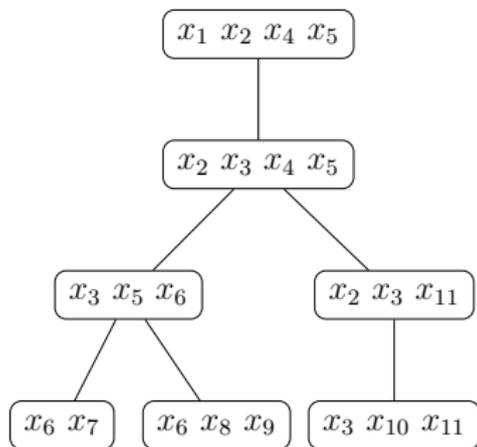
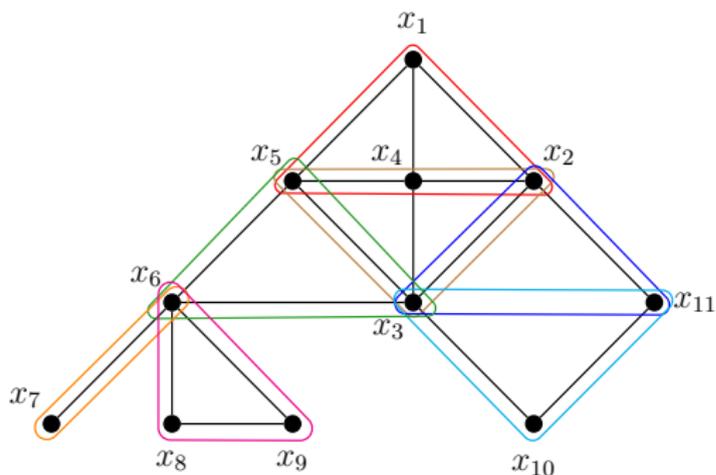
Largeur $w^+ = \text{taille du plus grand cluster} - 1$

Décomposition arborescente [Robertson et Seymour, 1986]



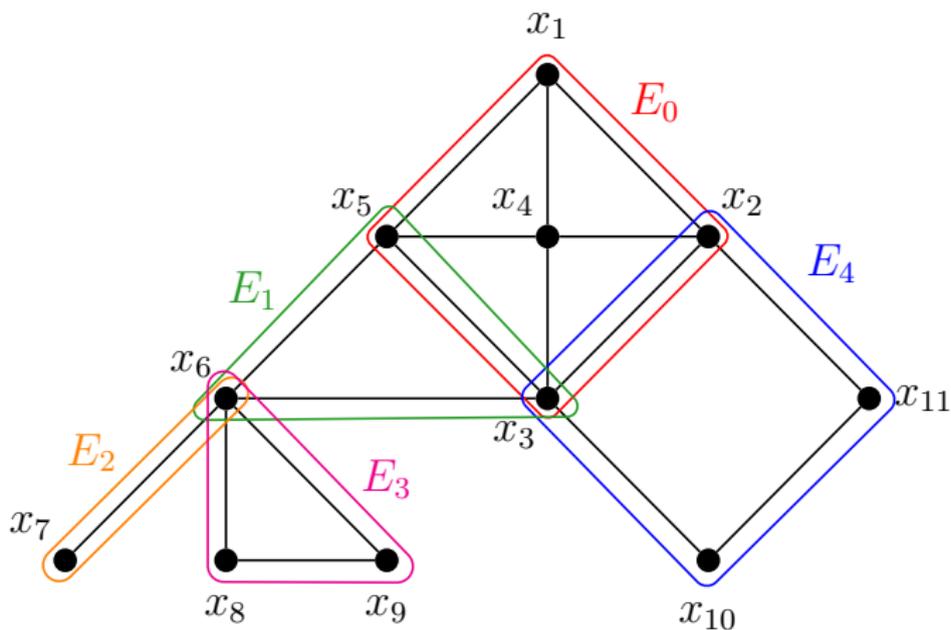
Largeur arborescente $w = \min w^+$

Décomposition arborescente [Robertson et Seymour, 1986]

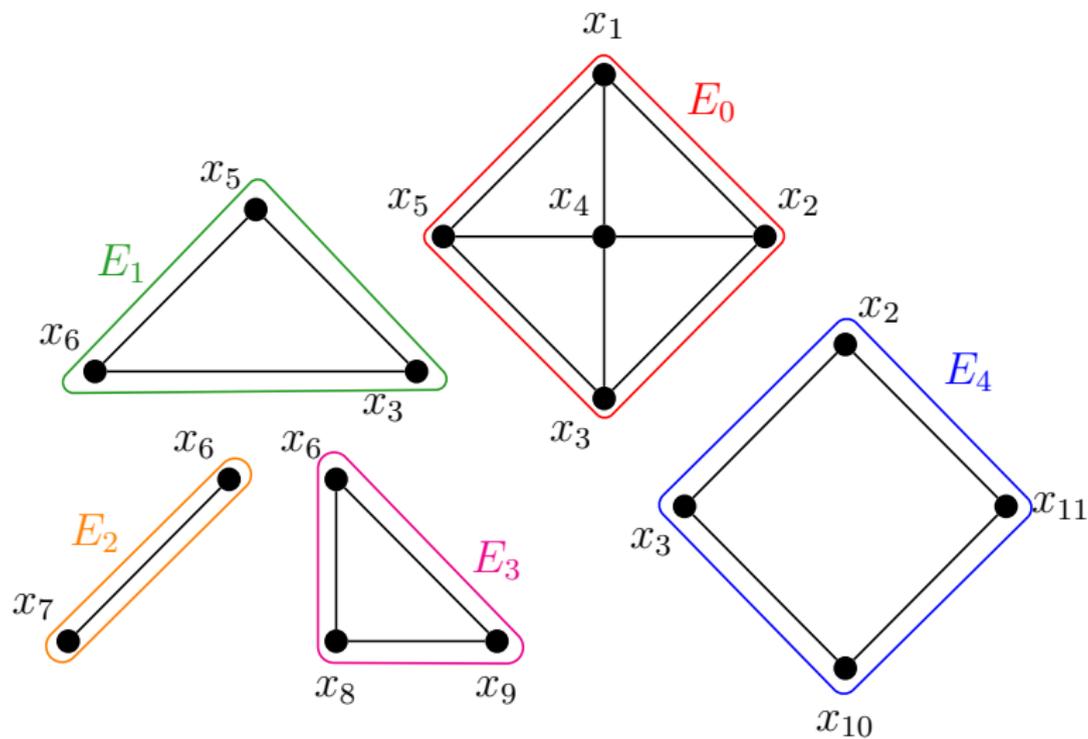


Classe polynomiale des CSP ayant une largeur bornée

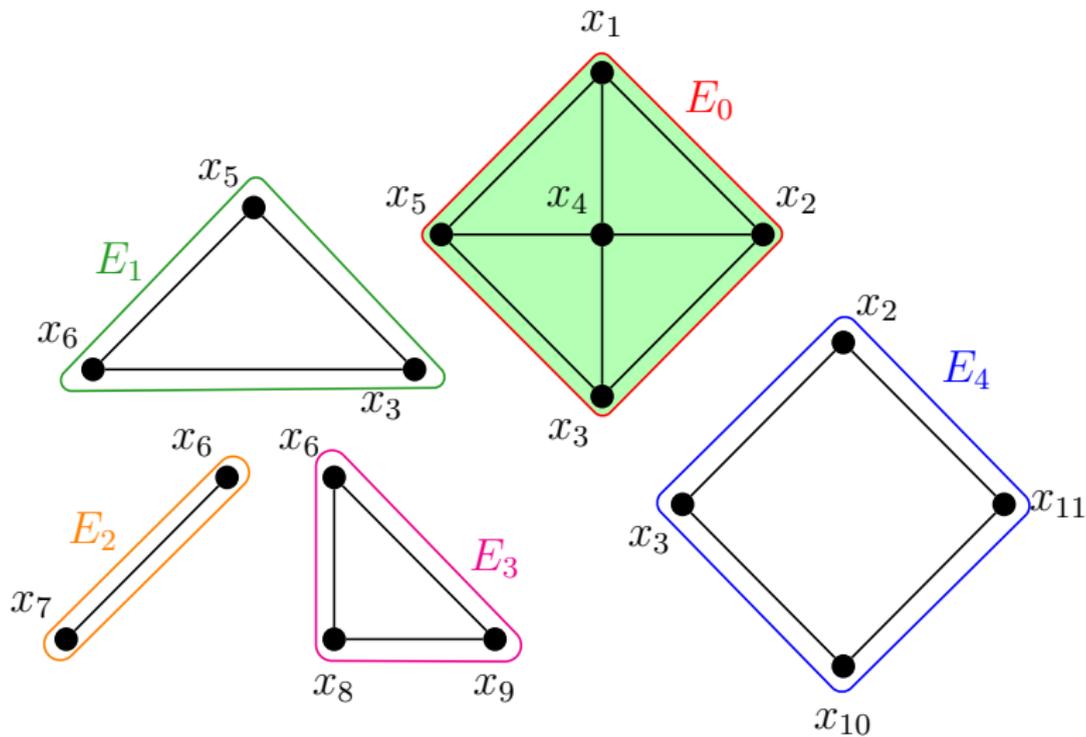
Tree-Clustering [Dechter et Pearl, 1989]



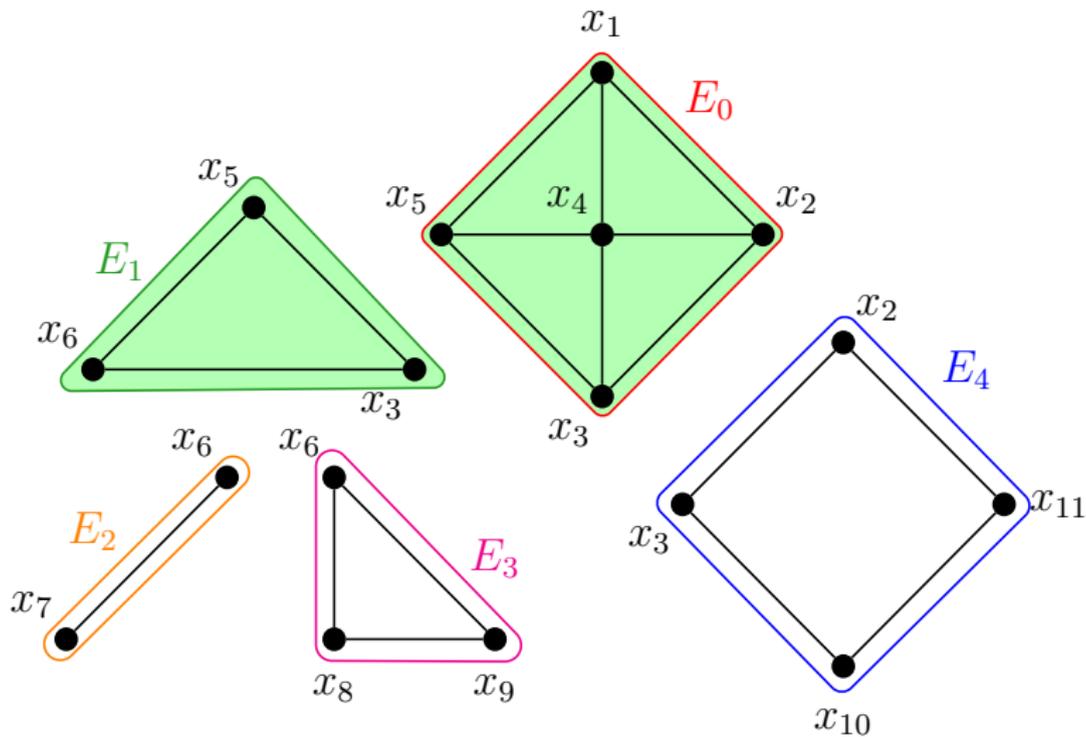
Tree-Clustering [Dechter et Pearl, 1989]



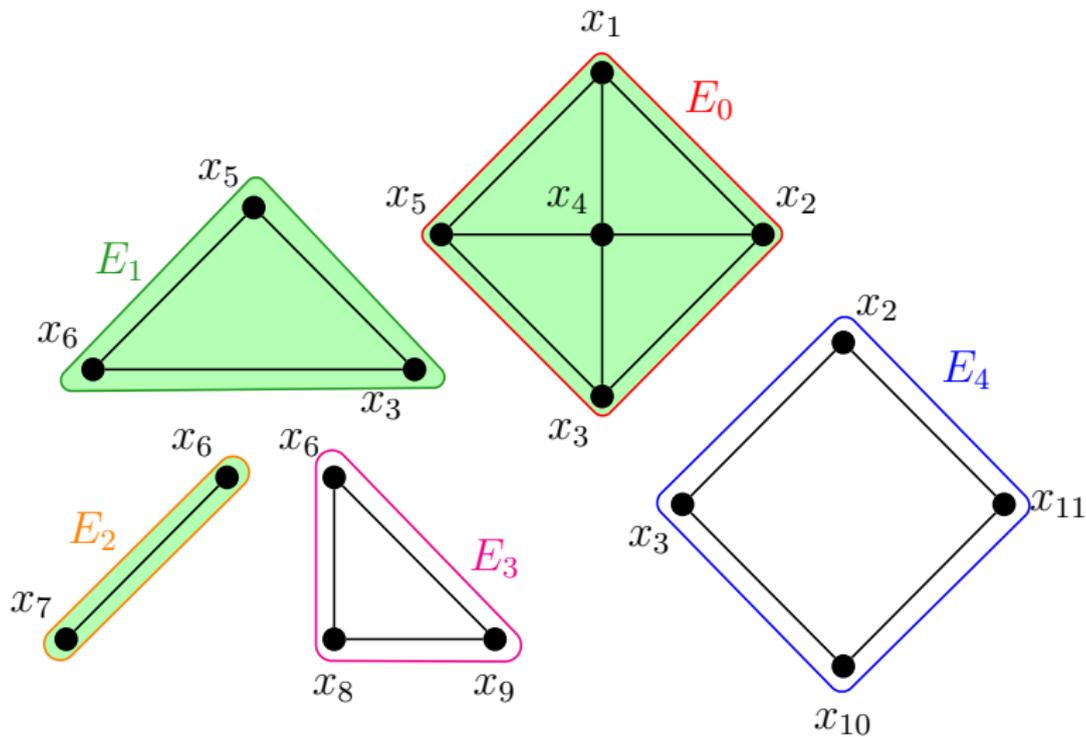
Tree-Clustering [Dechter et Pearl, 1989]



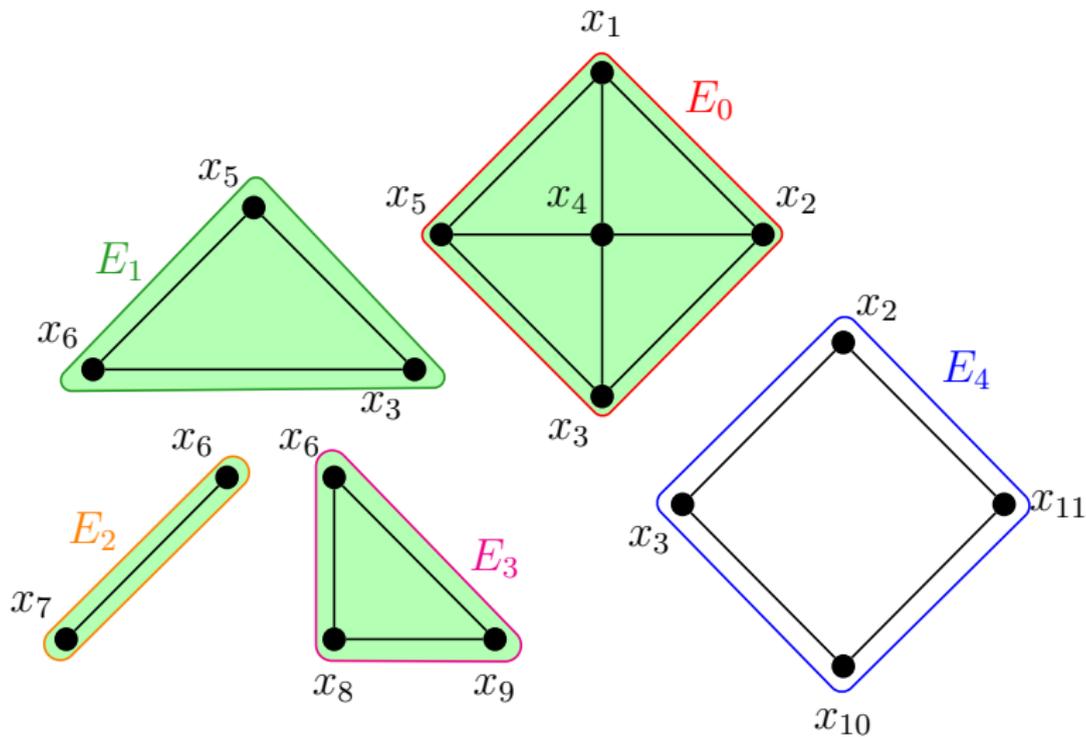
Tree-Clustering [Dechter et Pearl, 1989]



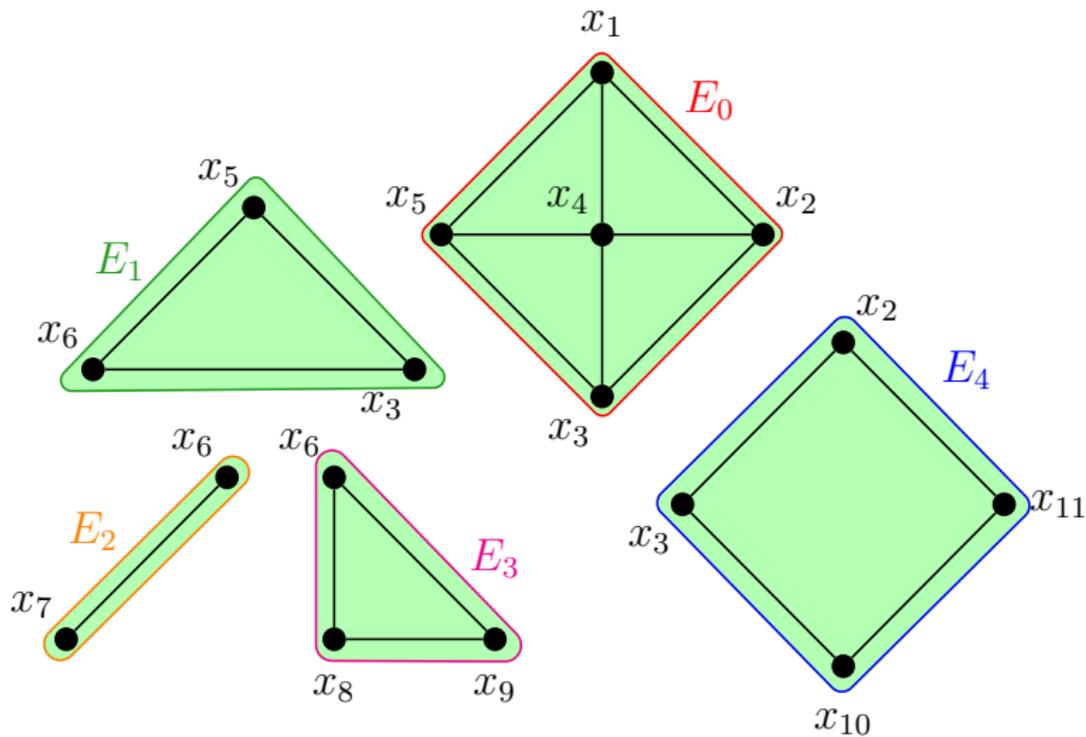
Tree-Clustering [Dechter et Pearl, 1989]



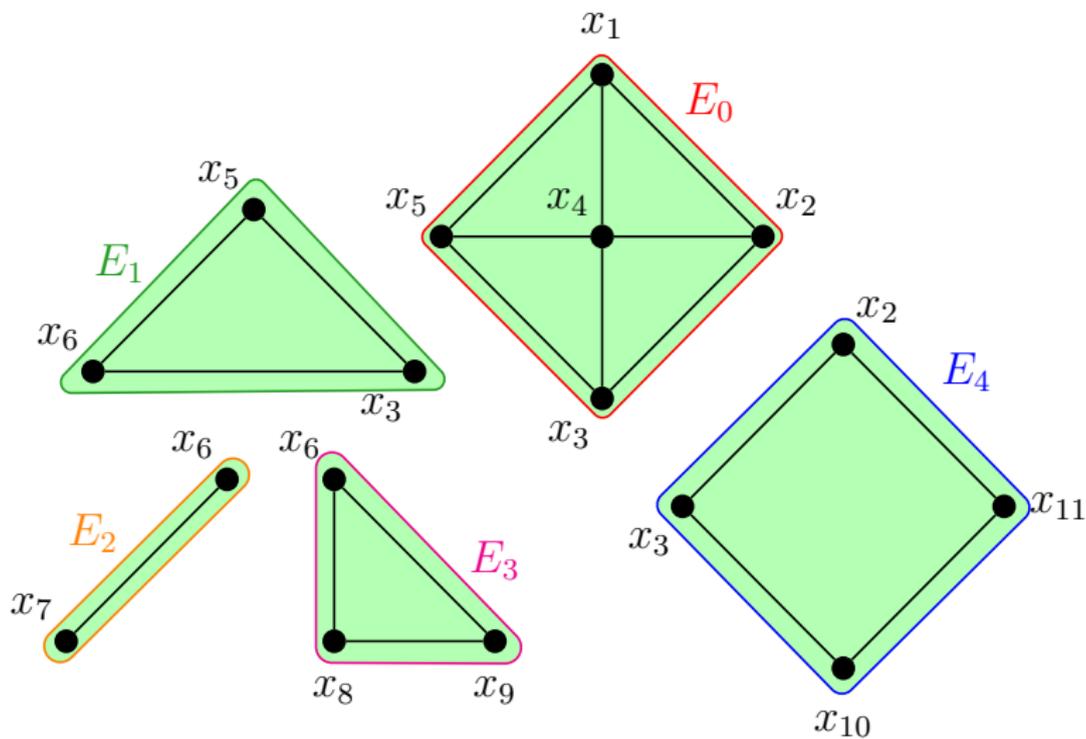
Tree-Clustering [Dechter et Pearl, 1989]



Tree-Clustering [Dechter et Pearl, 1989]



Tree-Clustering [Dechter et Pearl, 1989]



Complexité en temps et en espace : $O(S.d^{w^++1})$

Contributions

- Résolution et décompositions :
 - la méthode BTD
 - la méthode BDH
 - un cadre générique : SBBT

Contributions

- Résolution et décompositions :
 - la méthode BTD
 - la méthode BDH
 - un cadre générique : SBBT
- Calculs de décompositions arborescentes :
 - Calcul de décompositions arborescentes adaptées

Contributions

- Résolution et décompositions :
 - la méthode BTD
 - la méthode BDH
 - un cadre générique : SBBT
- Calculs de décompositions arborescentes :
 - Calcul de décompositions arborescentes adaptées
- Applications :
 - Cohérence structurelle
 - Adaptation à SAT
 - Extension aux CSP valués

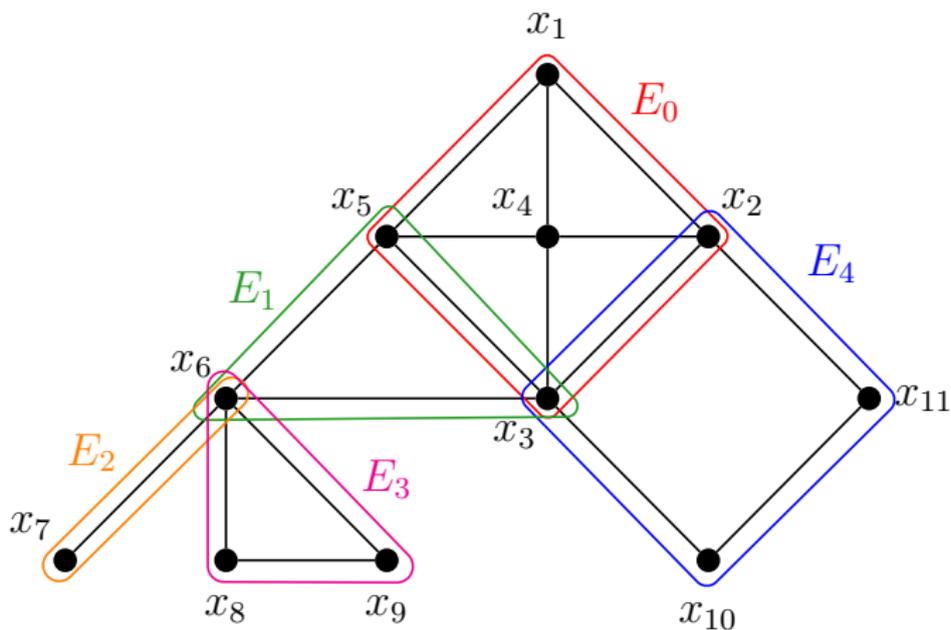
Contributions

- Résolution et décompositions :
 - la méthode **BTD**
 - la méthode BDH
 - un cadre générique : SBBT
- Calculs de décompositions arborescentes :
 - Calcul de décompositions arborescentes adaptées
- Applications :
 - Cohérence structurelle
 - Adaptation à SAT
 - Extension aux CSP valués

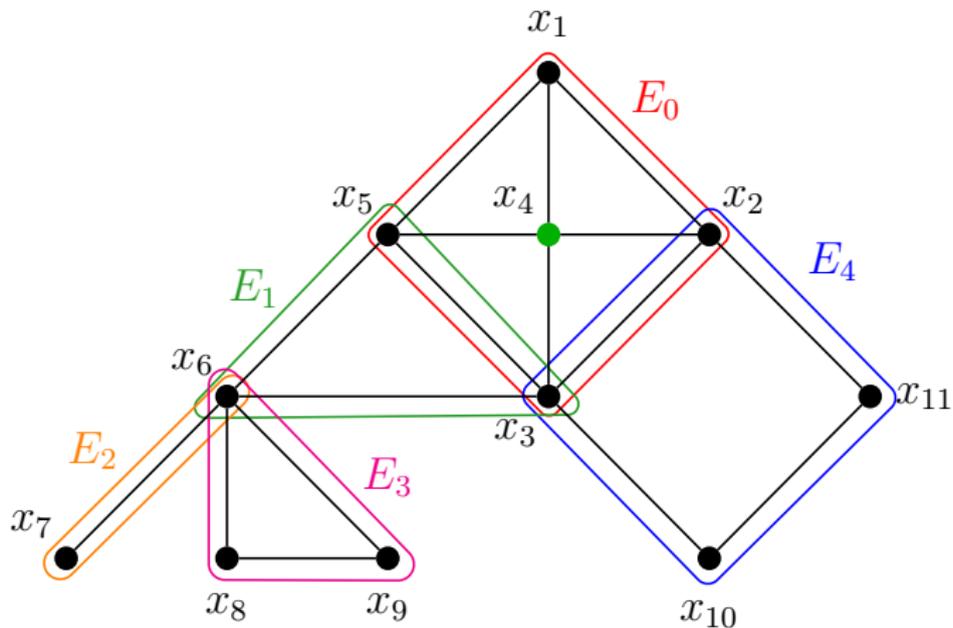
Contributions

- Résolution et décompositions :
 - la méthode **BTD**
 - la méthode BDH
 - un cadre générique : SBBT
- Calculs de décompositions arborescentes :
 - **Calcul de décompositions arborescentes adaptées**
- Applications :
 - Cohérence structurelle
 - Adaptation à SAT
 - Extension aux CSP valués

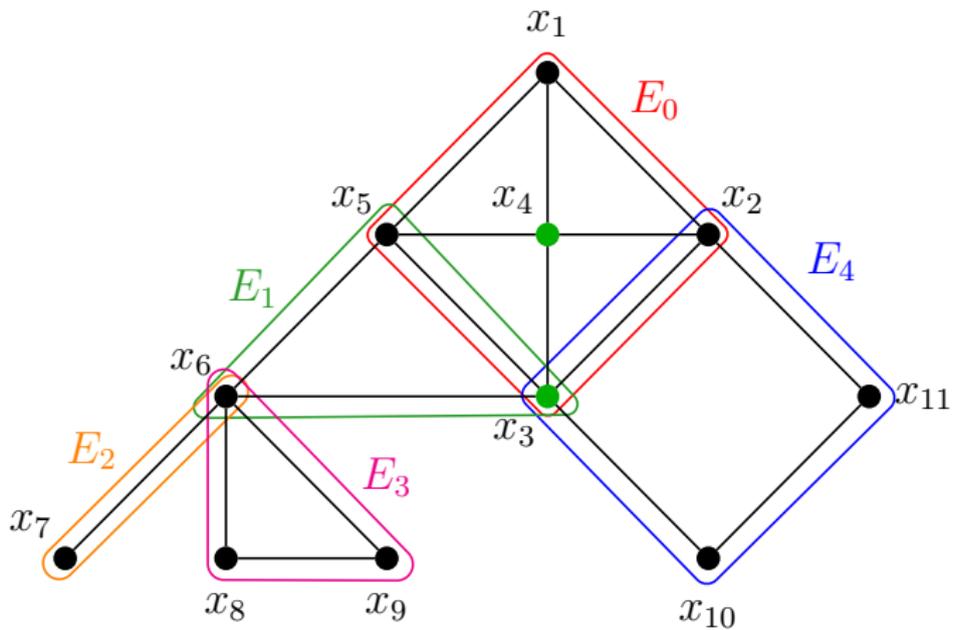
Backtracking on Tree-Decomposition

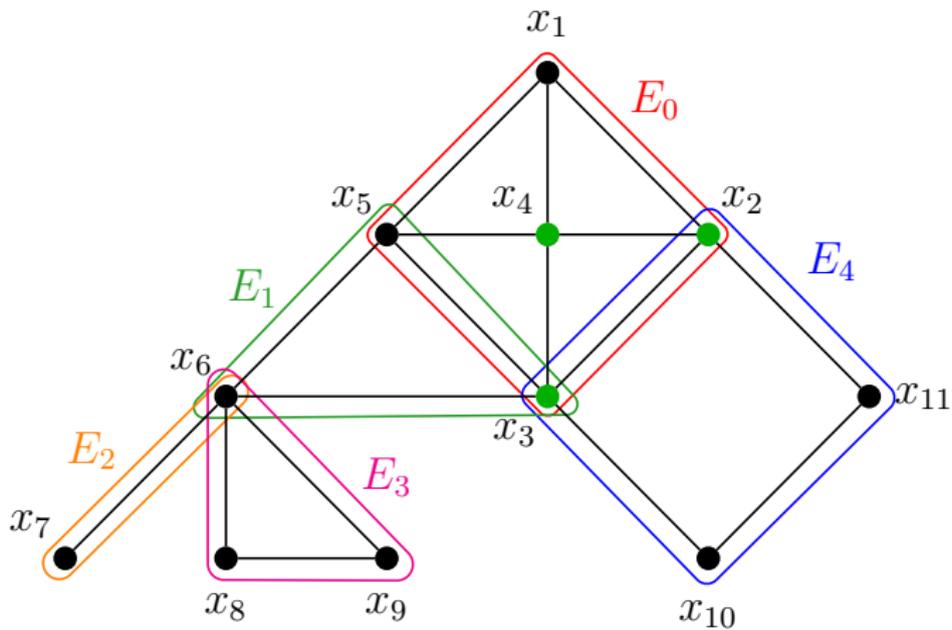


Compromis entre algorithme énumératif et programmation dynamique

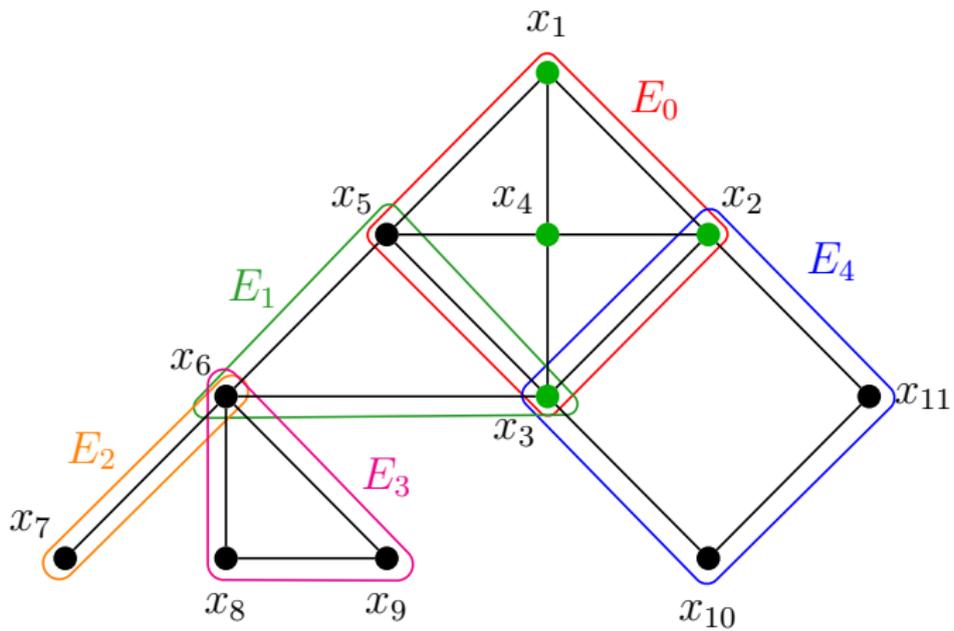


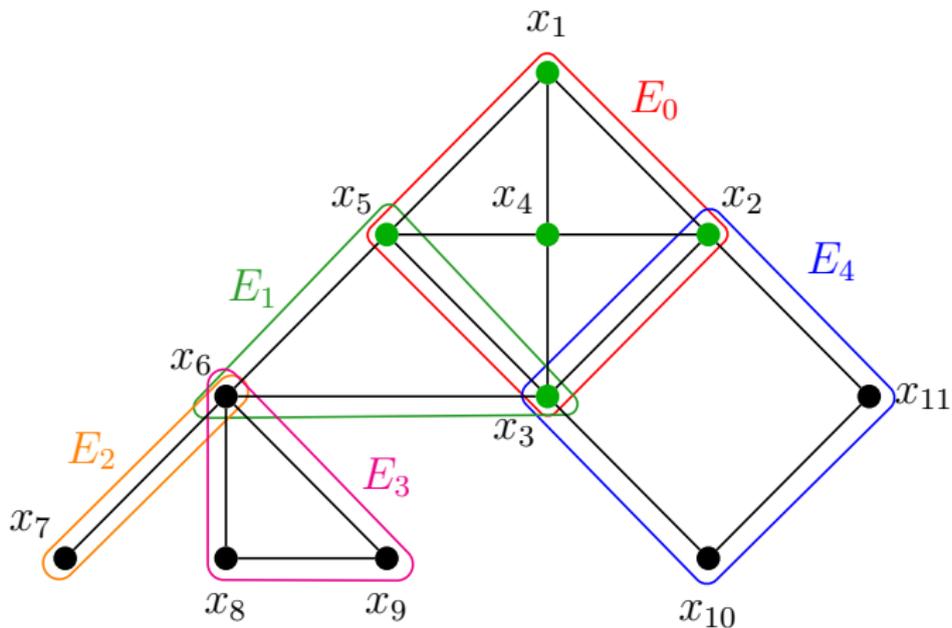
Compromis entre algorithme énumératif et programmation dynamique



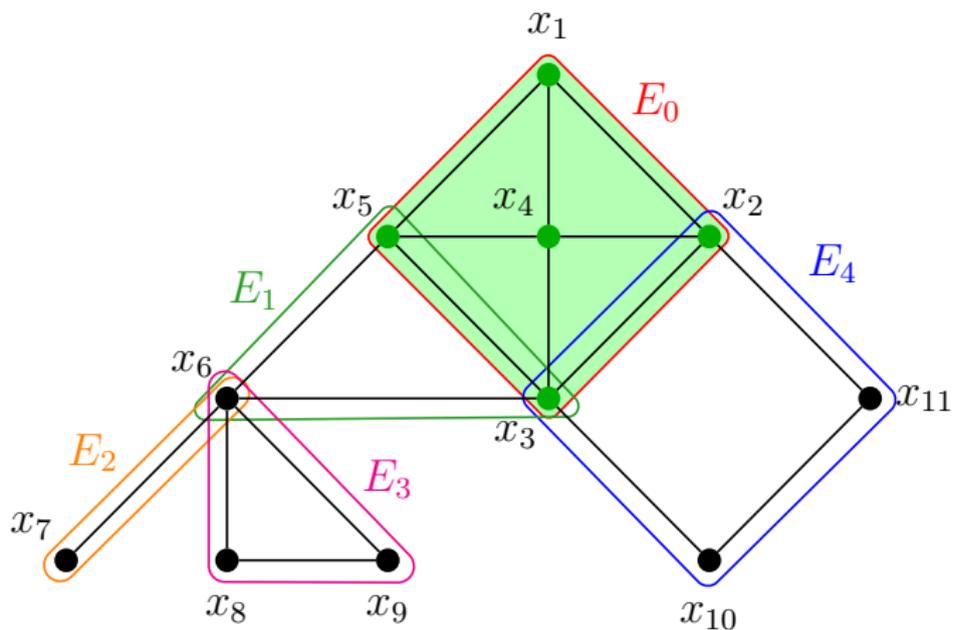


BTD

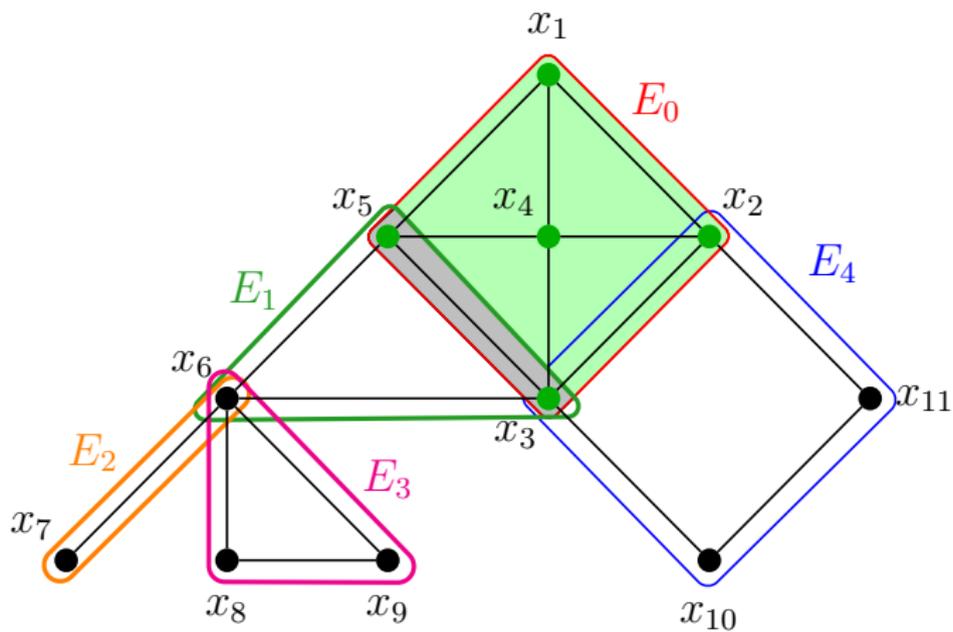


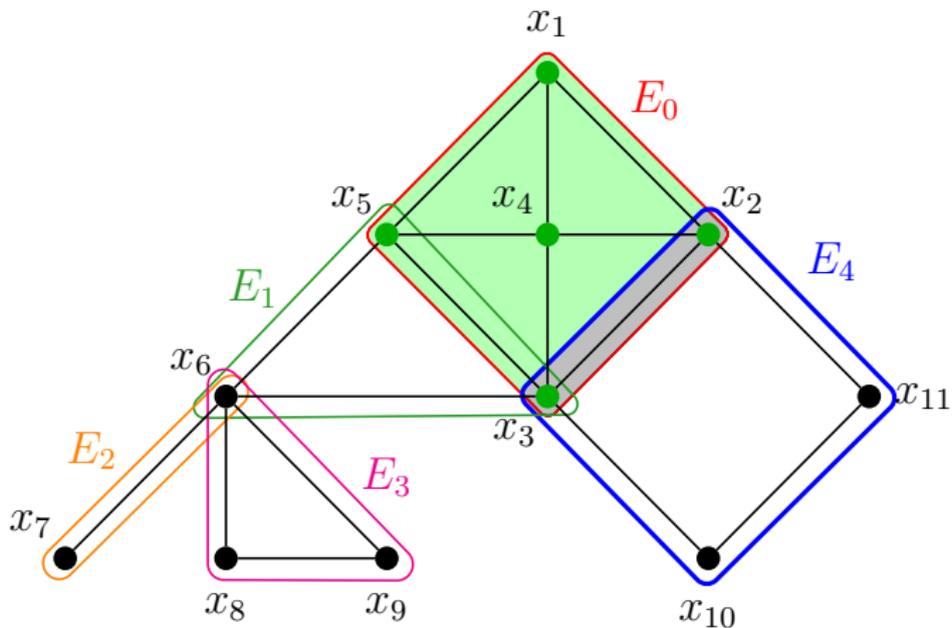


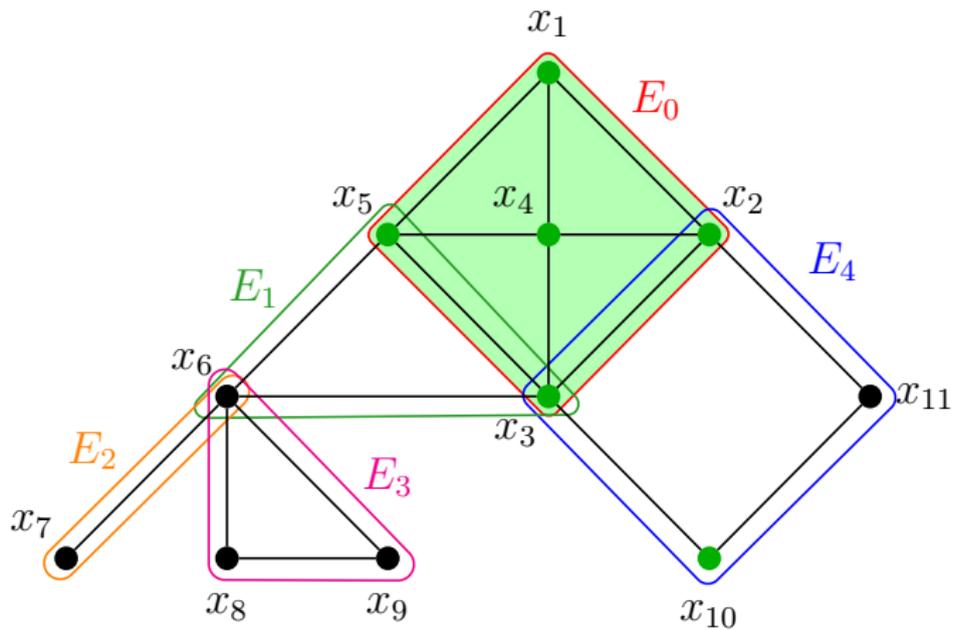
Résolution par BT, FC, RFL, MAC, ...

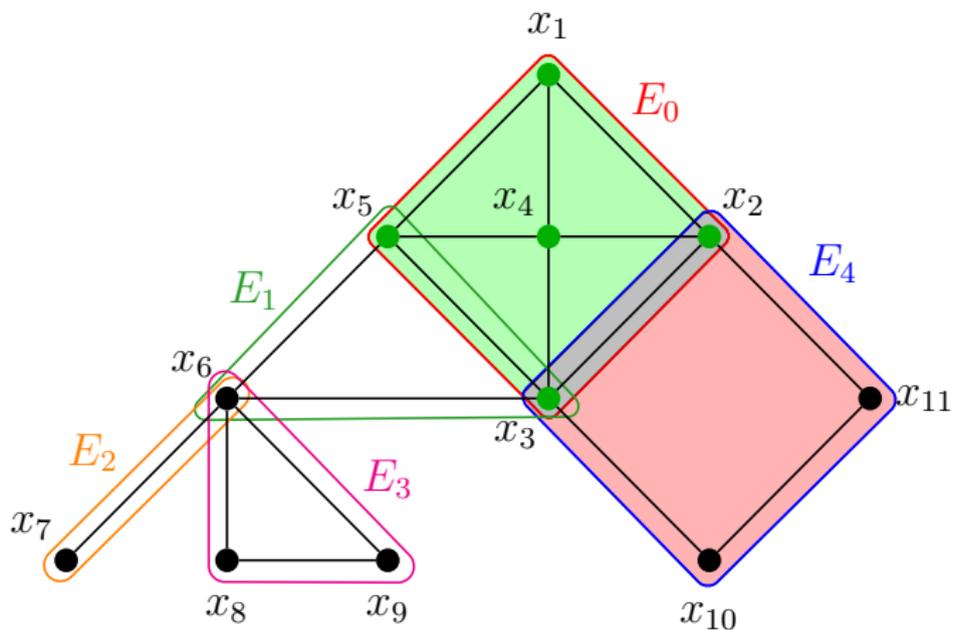


Résolution par BT, FC, RFL, MAC, ...



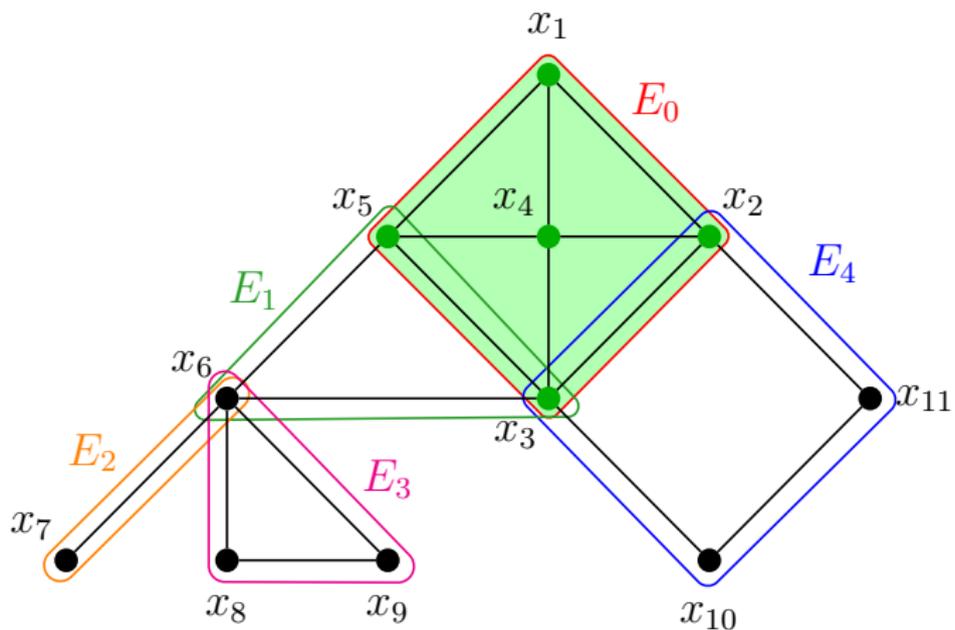


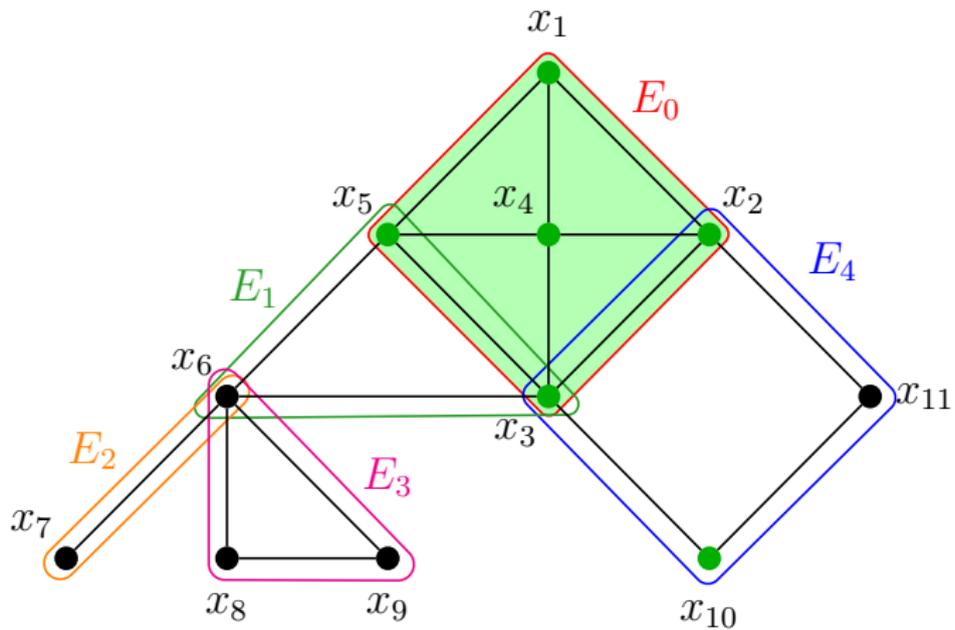


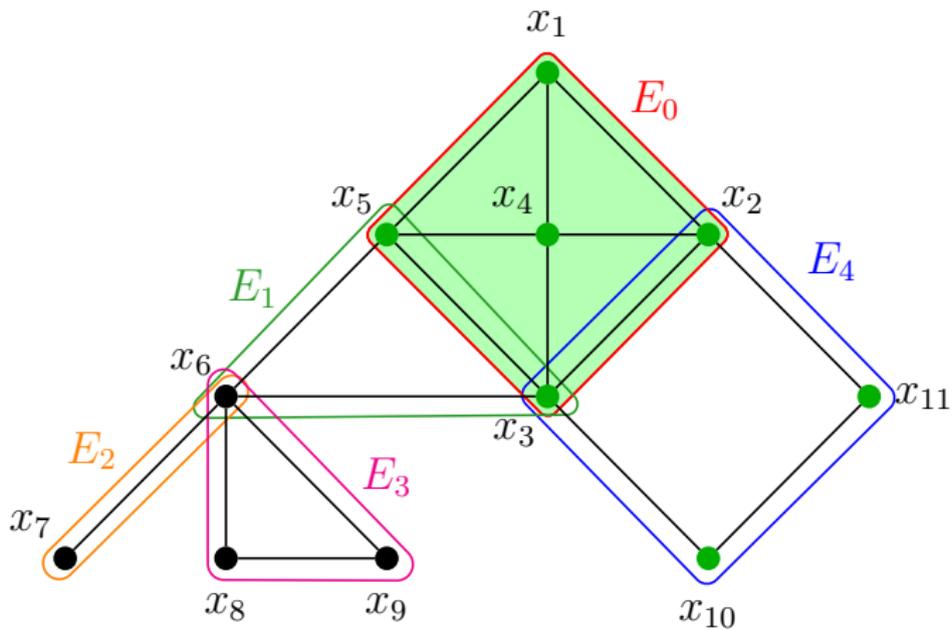


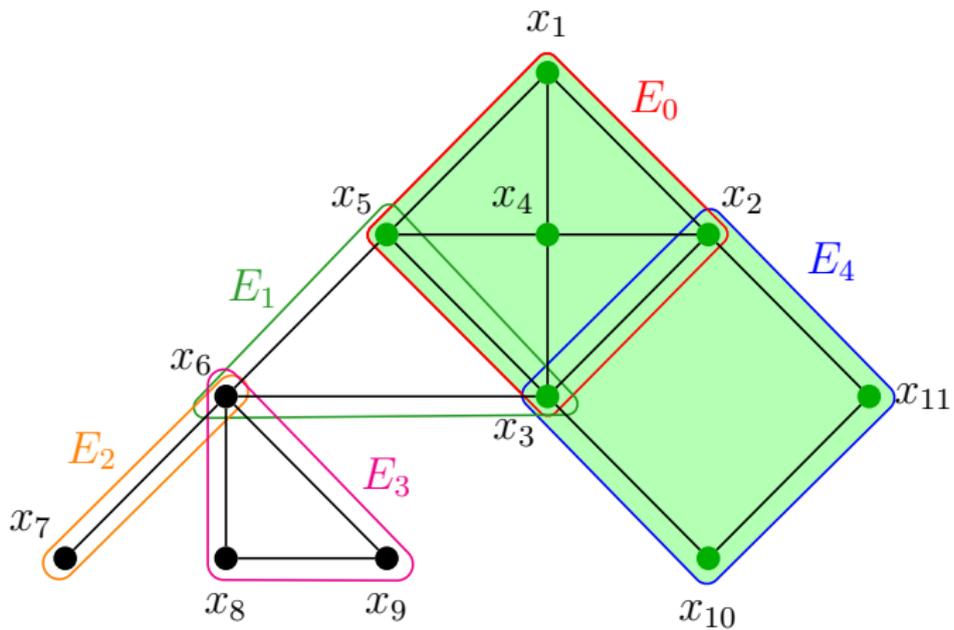
Enregistrement de nogoods structurels

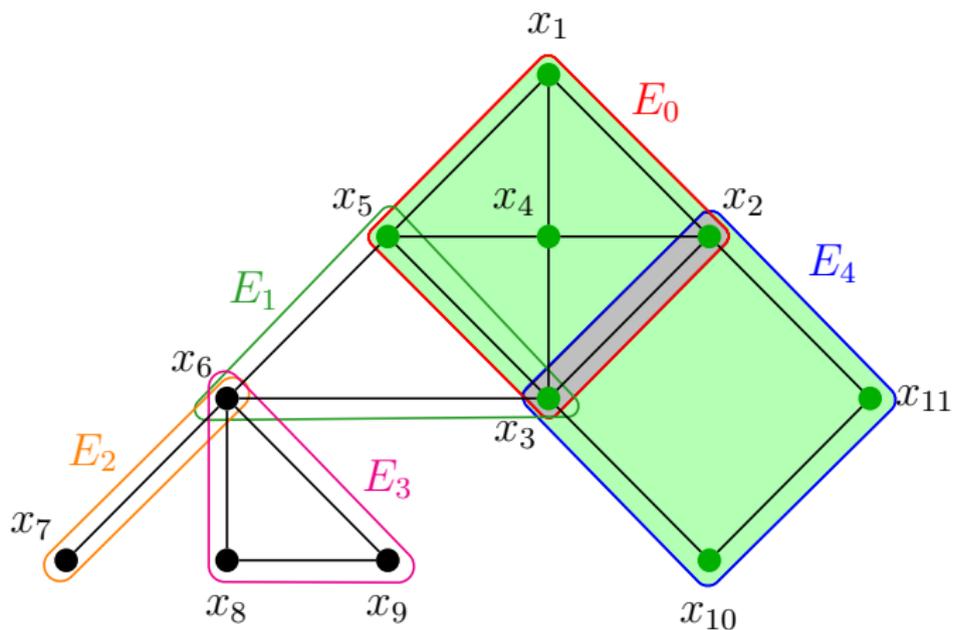
BTD



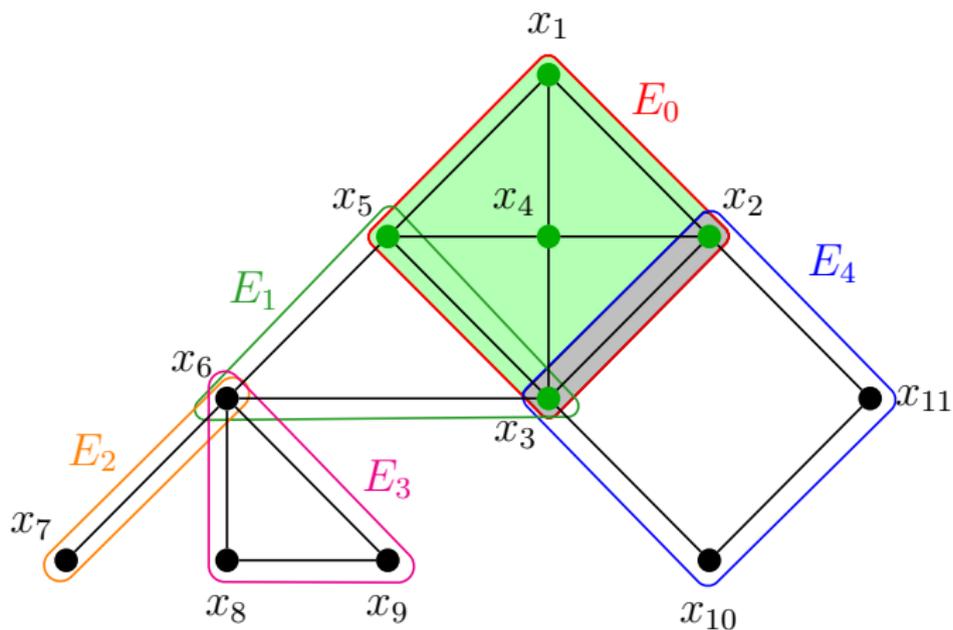








Enregistrement de goods structurels



Exploitation des (no)goods structurels

Complexités

- temps : $O(S.d^{w^++1})$ avec S la taille de l'instance
- espace : $O(n.s.d^s)$ avec $s = \max |E_i \cap E_j|$

Complexités

- temps : $O(S.d^{w^{++1}})$ avec S la taille de l'instance
- espace : $O(n.s.d^s)$ avec $s = \max |E_i \cap E_j|$

Résolution en temps polynomial d'instances ayant une largeur bornée

Complexités

- temps : $O(S.d^{w+1})$ avec S la taille de l'instance
- espace : $O(n.s.d^s)$ avec $s = \max |E_i \cap E_j|$

Résolution en temps polynomial d'instances ayant une largeur bornée

D'un point de vue pratique :

- Des redondances évitées

Complexités

- temps : $O(S.d^{w+1})$ avec S la taille de l'instance
- espace : $O(n.s.d^s)$ avec $s = \max |E_i \cap E_j|$

Résolution en temps polynomial d'instances ayant une largeur bornée

D'un point de vue pratique :

- Des redondances évitées
- Ordre sur les variables contraint

Complexités

- temps : $O(S.d^{w+1})$ avec S la taille de l'instance
- espace : $O(n.s.d^s)$ avec $s = \max |E_i \cap E_j|$

Résolution en temps polynomial d'instances ayant une largeur bornée

D'un point de vue pratique :

- Des redondances évitées
- Ordre sur les variables contraint :
 - Calcul de la décomposition
 - Choix d'un cluster racine
 - Heuristique de choix de variable

Complexités

- temps : $O(S.d^{w+1})$ avec S la taille de l'instance
- espace : $O(n.s.d^s)$ avec $s = \max |E_i \cap E_j|$

Résolution en temps polynomial d'instances ayant une largeur bornée

D'un point de vue pratique :

- Des redondances évitées
 - Ordre sur les variables contraint :
 - Calcul de la décomposition
 - Choix d'un cluster racine
 - Heuristique de choix de variable
- Fusion de clusters
-

Complexités

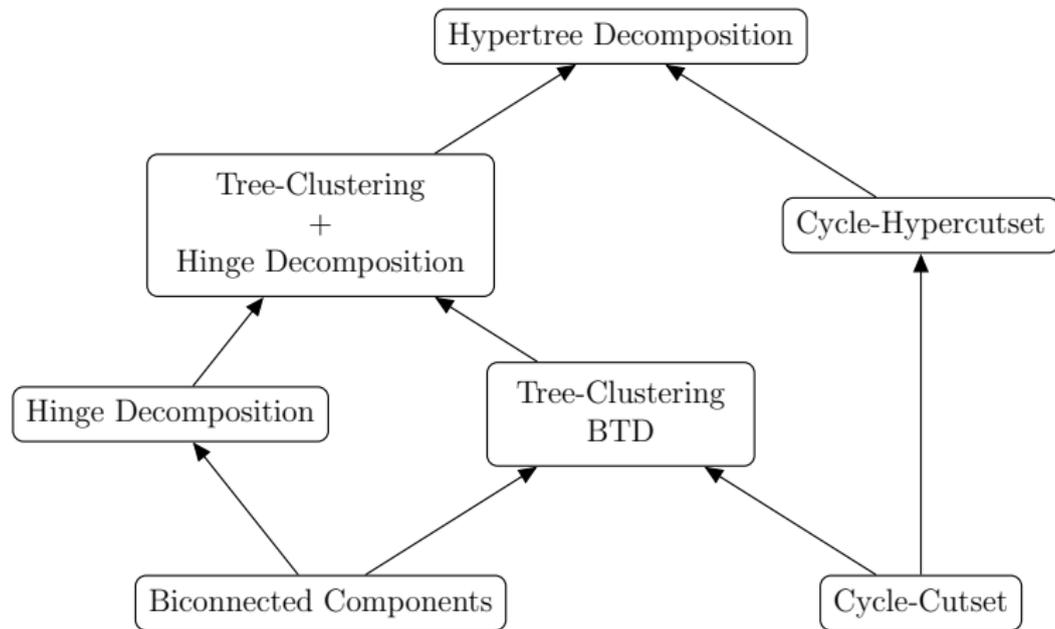
- temps : $O(S.d^{w+1})$ avec S la taille de l'instance
- espace : $O(n.s.d^s)$ avec $s = \max |E_i \cap E_j|$

Résolution en temps polynomial d'instances ayant une largeur bornée

D'un point de vue pratique :

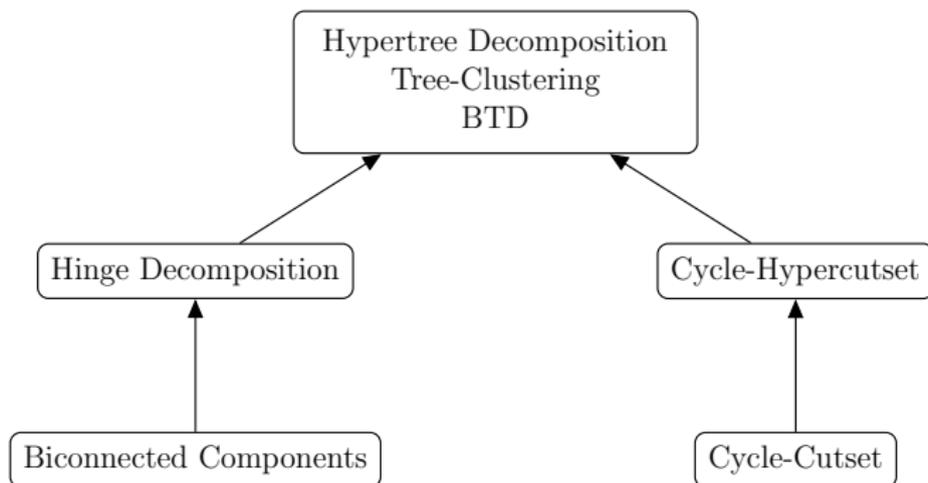
- Des redondances évitées
- Ordre sur les variables contraint :
 - Calcul de la décomposition
 - Choix d'un cluster racine  **Redémarrage**
 - Heuristique de choix de variable  **Redémarrage**

Hiérarchie des méthodes de décompositions structurées



[Gottlob et al., 2000]

Hierarchie des méthodes de décompositions structurelles



Calculs de décompositions arborescentes

Calculer une décomposition optimale est NP-difficile.

Calculs de décompositions arborescentes

Calculer une décomposition optimale est NP-difficile.

Différents types de méthodes de calcul :

- méthodes exactes

Calculs de décompositions arborescentes

Calculer une décomposition optimale est NP-difficile.

Différents types de méthodes de calcul :

- méthodes exactes
- méthodes approchées

Calculs de décompositions arborescentes

Calculer une décomposition optimale est NP-difficile.

Différents types de méthodes de calcul :

- méthodes exactes
- méthodes approchées

coût prohibitif

Calculs de décompositions arborescentes

Calculer une décomposition optimale est NP-difficile.

Différents types de méthodes de calcul :

- méthodes exactes
- méthodes approchées
- méthodes heuristiques :
 - MCS
 - Min-Fill

Calculs de décompositions arborescentes

Calculer une décomposition optimale est NP-difficile.

Différents types de méthodes de calcul :

- méthodes exactes
- méthodes approchées
- méthodes heuristiques :
 - MCS
 - Min-Fill

Souvent basées sur la notion de triangulation

Calculs de décompositions arborescentes et résolution

Décomposition optimale : coûteux et pas forcément pertinent

Calculs de décompositions arborescentes et résolution

Décomposition optimale : coûteux et pas forcément pertinent

Méthodes de décomposition heuristiques :

- MCS : détection des graphes triangulés
- Min-Fill : minimiser la largeur w^+

Calculs de décompositions arborescentes et résolution

Décomposition optimale : coûteux et pas forcément pertinent

Méthodes de décomposition heuristiques :

- MCS : détection des graphes triangulés
- Min-Fill : minimiser la largeur w^+

Exploitable mais pas idéales

Calculs de décompositions arborescentes et résolution

Décomposition optimale : coûteux et pas forcément pertinent

Méthodes de décomposition heuristiques :

- MCS : détection des graphes triangulés
- Min-Fill : minimiser la largeur w^+

Exploitable mais pas idéales

Différents paramètres à considérer :

- largeur w^+ ,
- taille des séparateurs entre clusters,
- connexité des clusters,
- ...

Heuristic Tree-Decomposition Without Triangulation

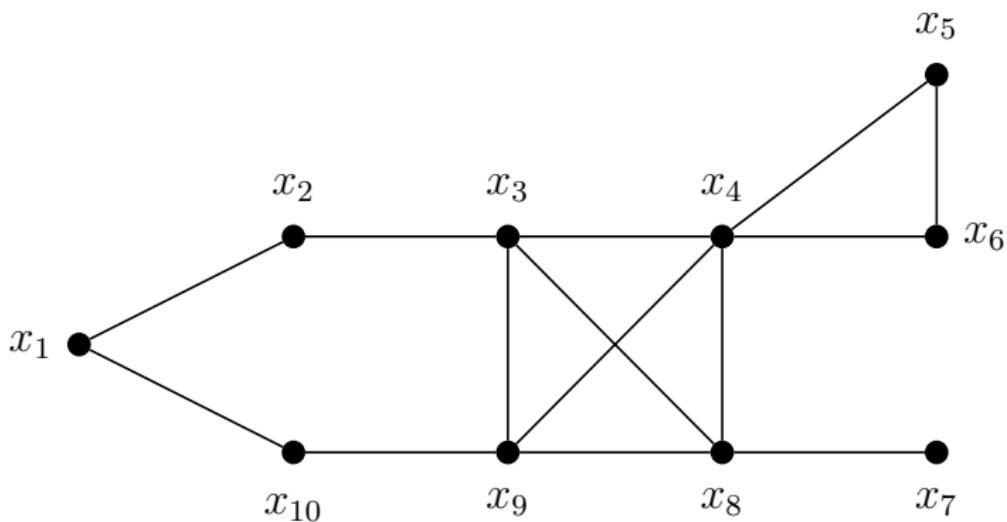
Un algorithme générique

Paramétrable selon l'objectif visé

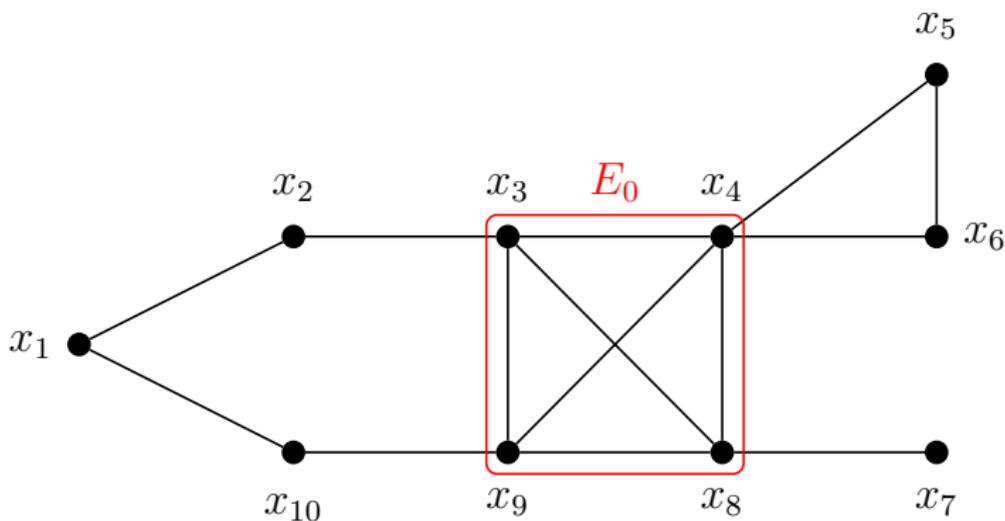
Sans triangulation

Exploitation de la topologie

H_2 -TD-WT (connexité des clusters)

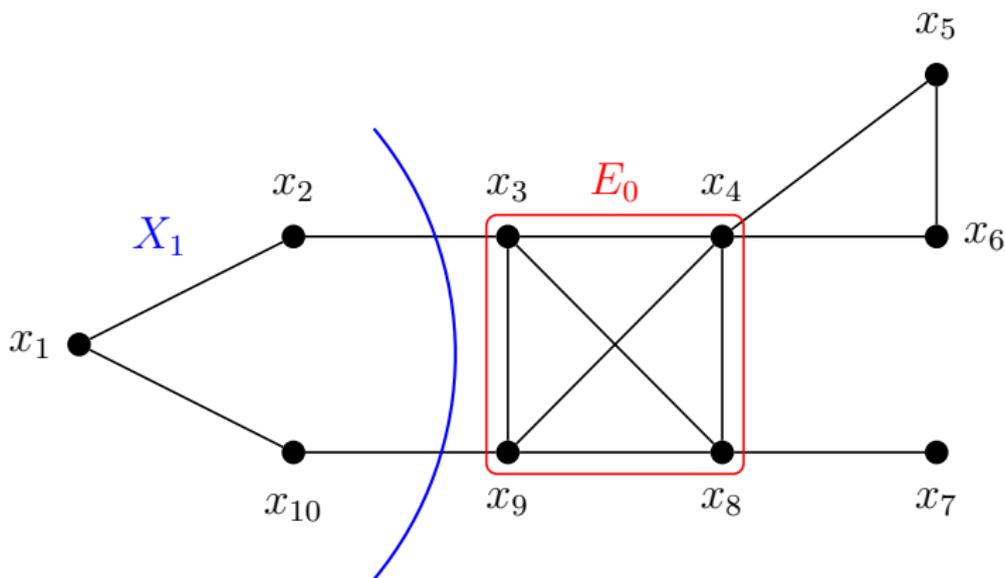


H_2 -TD-WT (connexité des clusters)



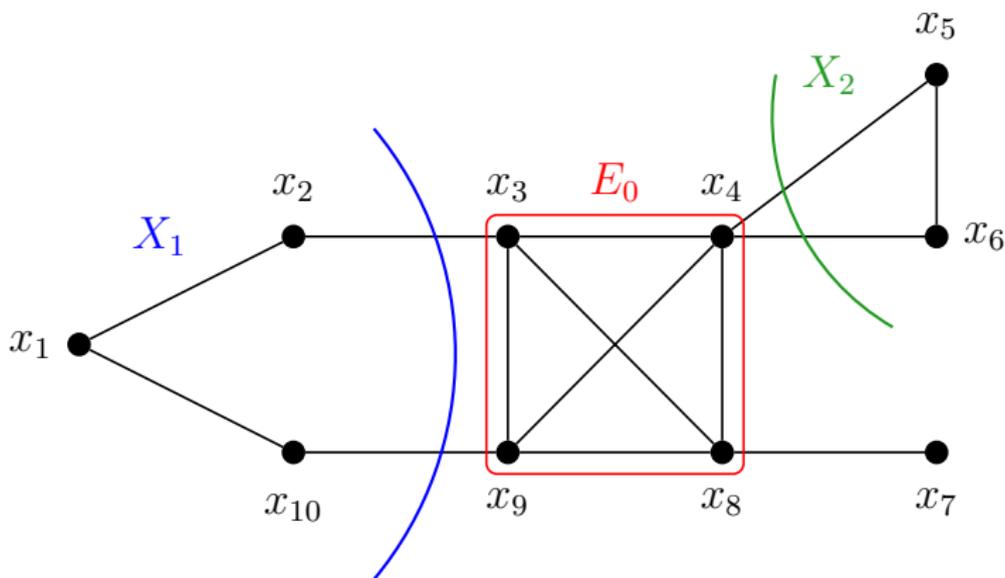
Choix d'un premier cluster E_0

H_2 -TD-WT (connexité des clusters)



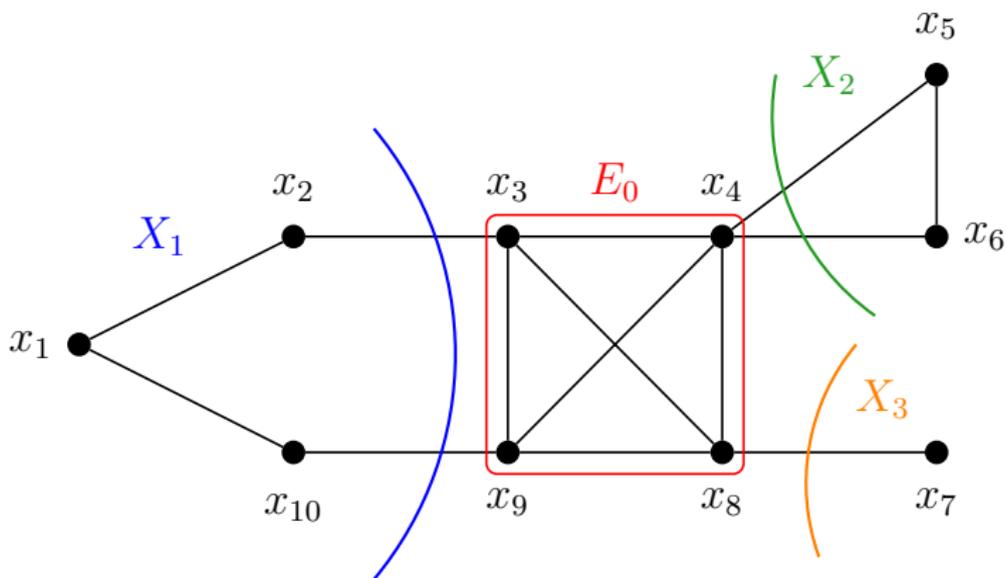
Calcul des composantes connexes

H_2 -TD-WT (connexité des clusters)



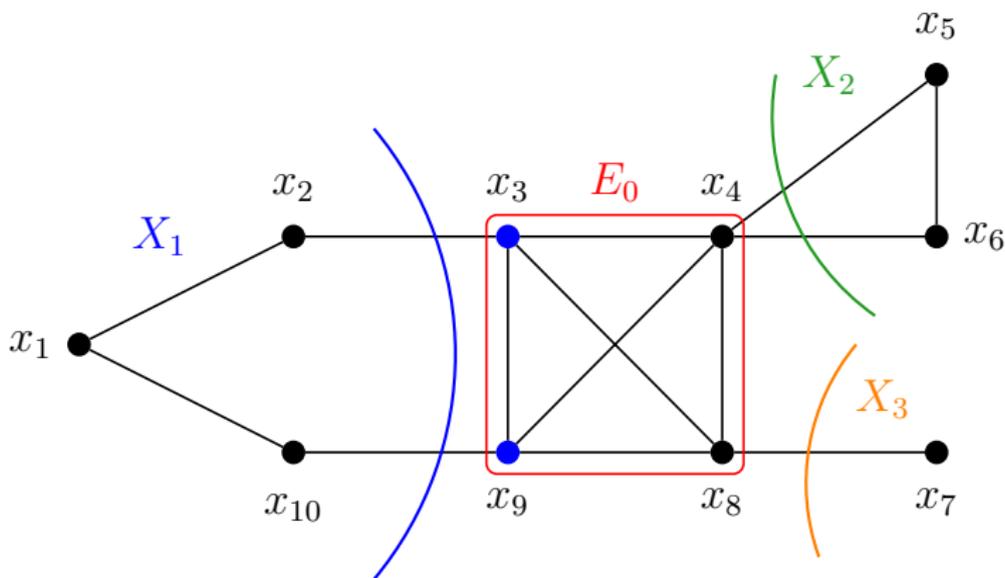
Calcul des composantes connexes

H_2 -TD-WT (connexité des clusters)



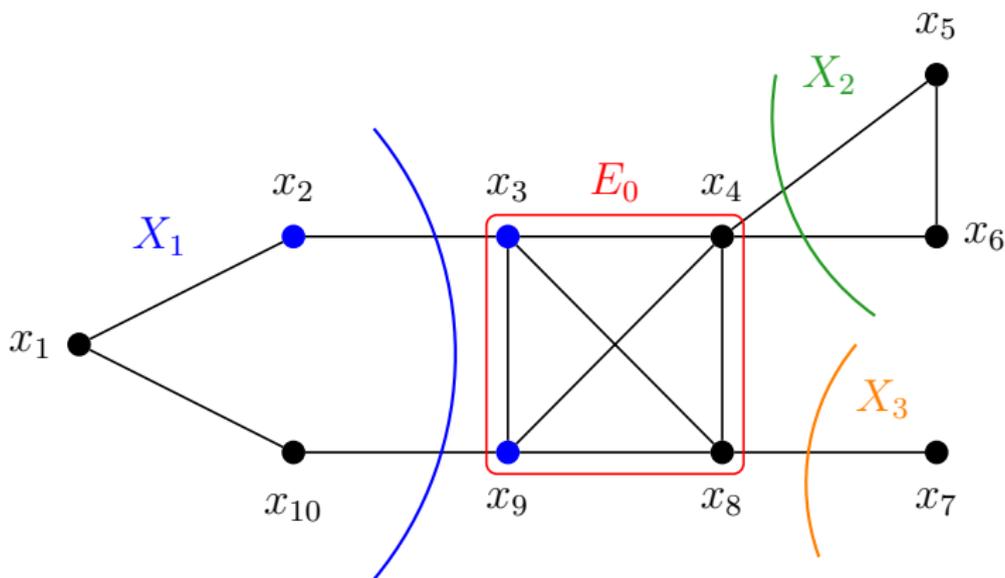
Calcul des composantes connexes

H_2 -TD-WT (connexité des clusters)



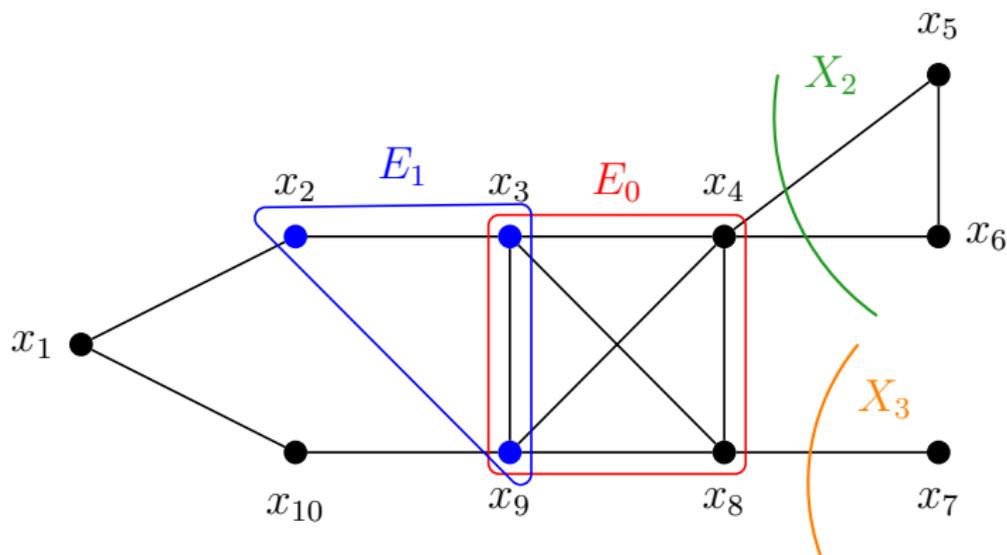
Choix d'une composante et identification du séparateur

H_2 -TD-WT (connexité des clusters)



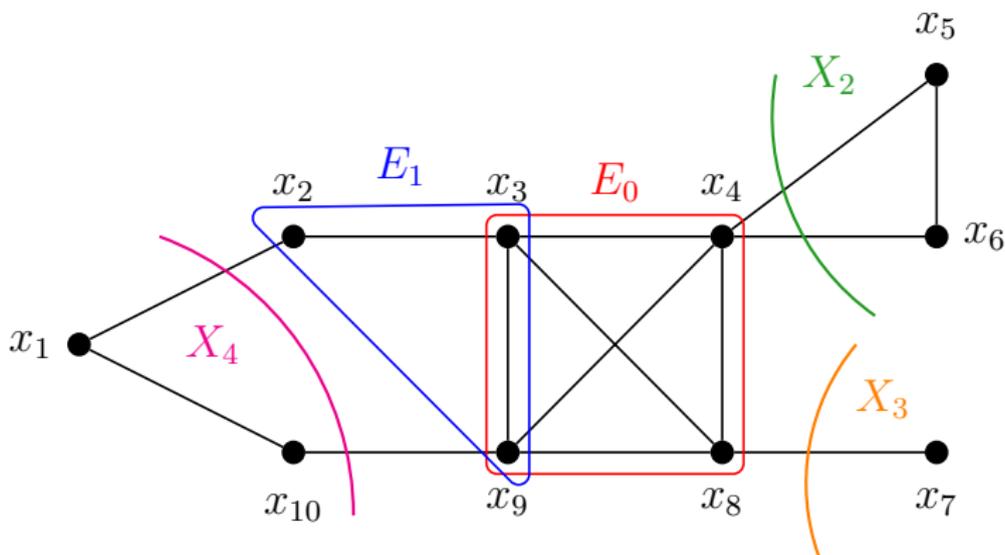
Ajout d'un sommet et test du critère désiré

H_2 -TD-WT (connexité des clusters)



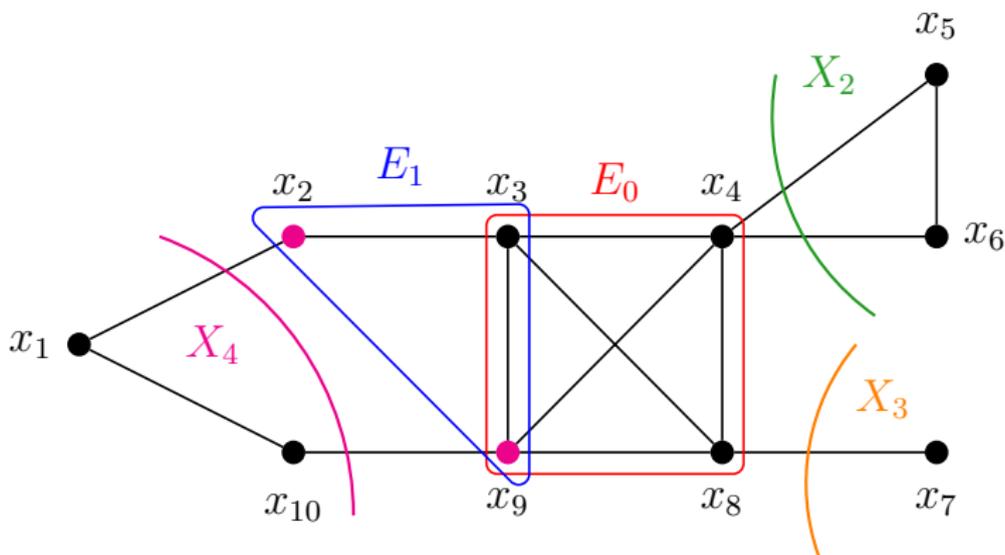
Création d'un nouveau cluster

H_2 -TD-WT (connexité des clusters)



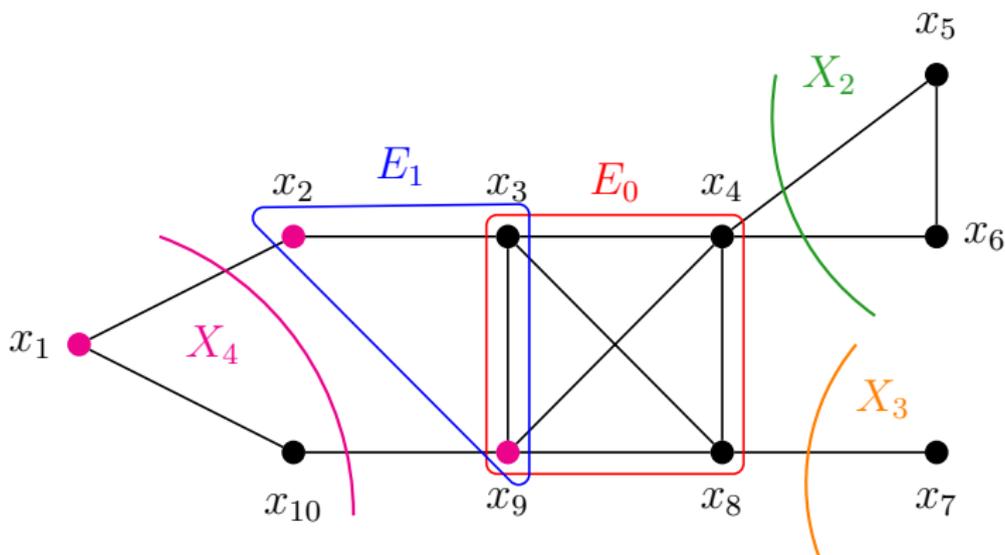
Calcul des nouvelles composantes connexes

H_2 -TD-WT (connexité des clusters)



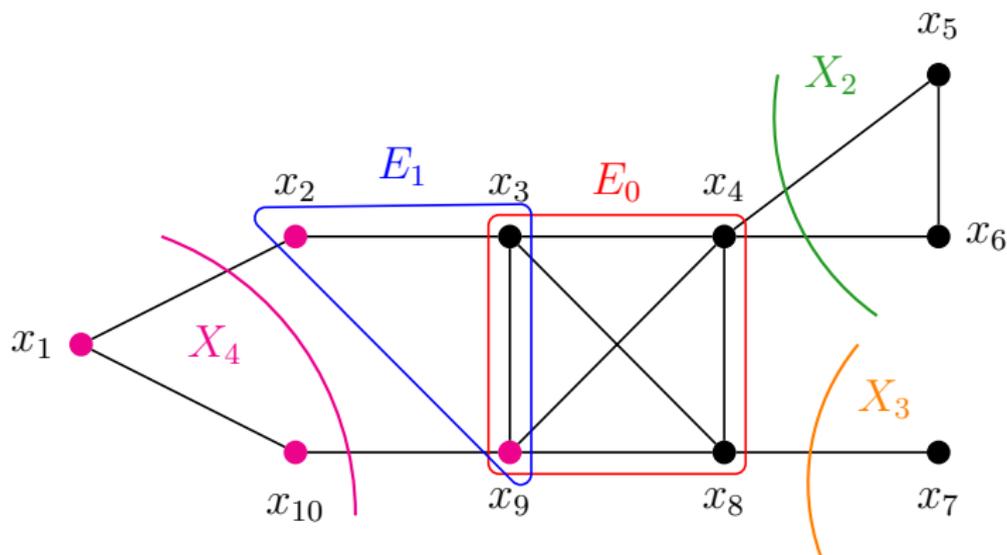
Choix d'une composante et identification du séparateur

H_2 -TD-WT (connexité des clusters)



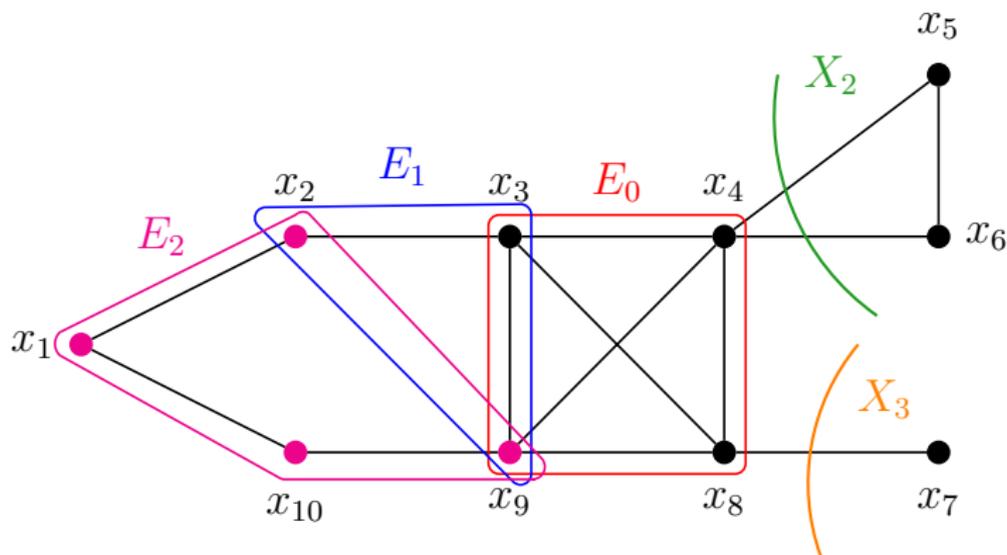
Ajout d'un sommet et test du critère désiré

H_2 -TD-WT (connexité des clusters)



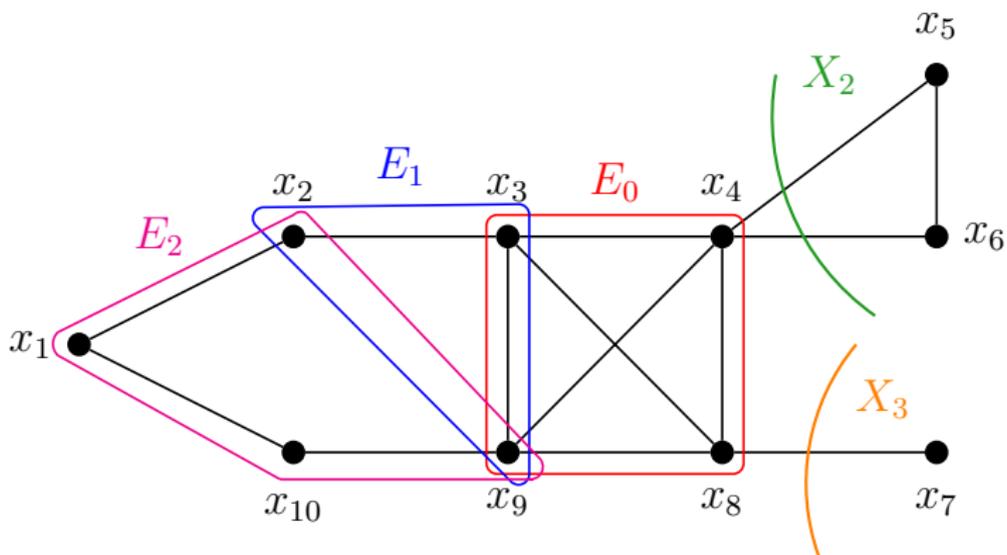
Ajout d'un sommet et test du critère désiré

H_2 -TD-WT (connexité des clusters)

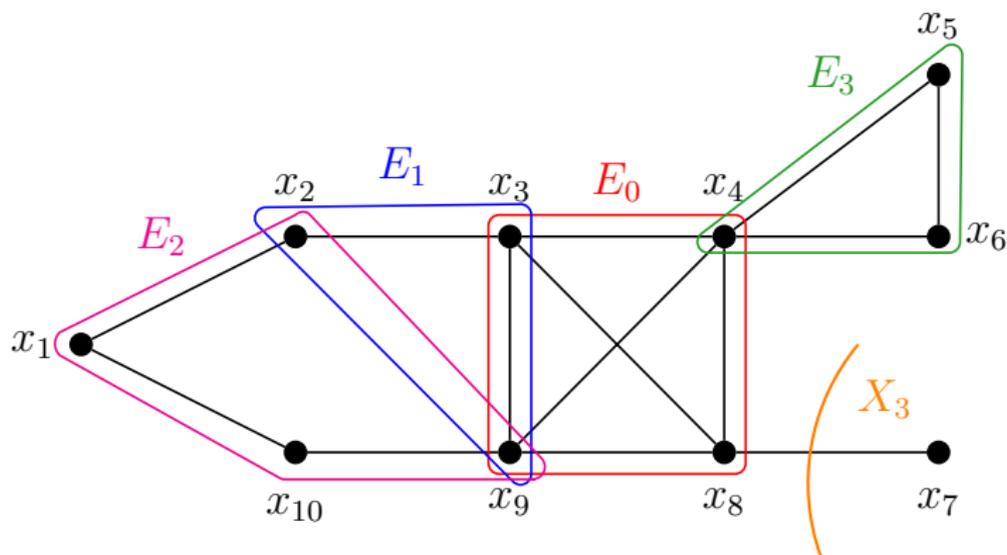


Création d'un nouveau cluster

H_2 -TD-WT (connexité des clusters)

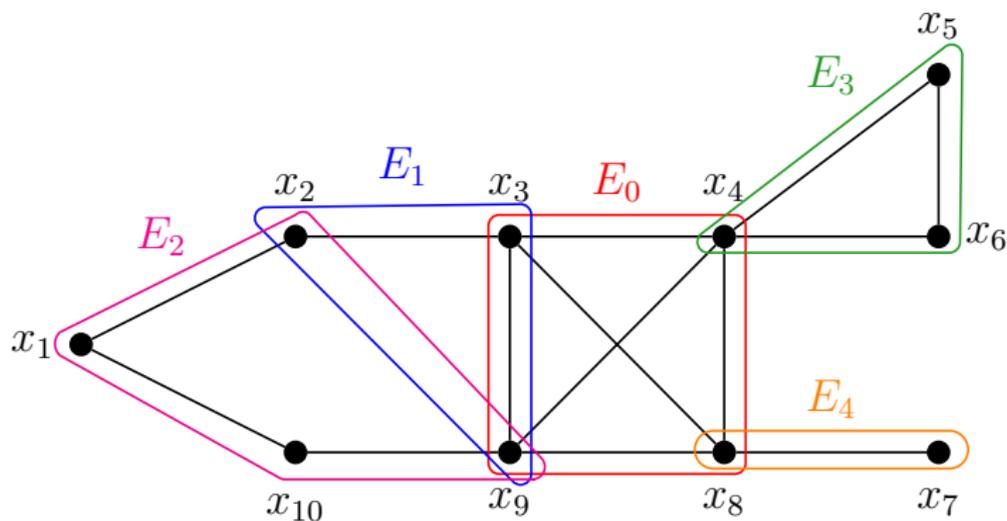


H_2 -TD-WT (connexité des clusters)



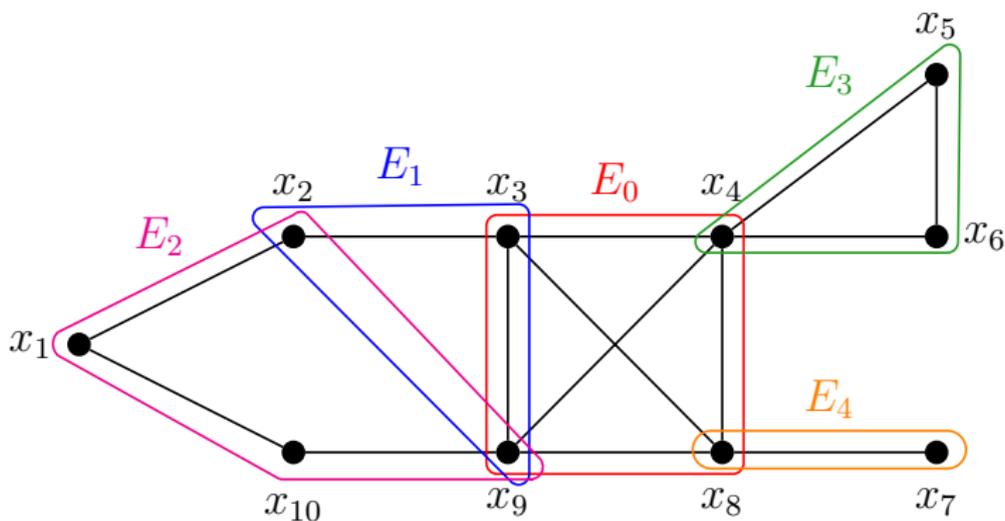
Création d'un nouveau cluster

H_2 -TD-WT (connexité des clusters)



Création d'un nouveau cluster

H_2 -TD-WT (connexité des clusters)



Obtention d'une décomposition ayant le critère désiré

H-TD-WT

Complexité :

- modulaire
- raisonnable en $O(n(n + e))$

H-TD-WT

Complexité :

- modulaire
- raisonnable en $O(n(n + e))$

En pratique :

- souvent plus rapide que Min-Fill
- taille des graphes accessibles plus importante

Min-Fill vs H_i

1 859 instances de la compétition CSP 2008

Algorithme	Min-Fill		H_2		H_5	
	#rés.	temps	#rés.	temps	#rés.	temps
BTD-MAC	1 344	43 272	1 405	31 429	1 469	33 564
BTD-MAC+RST	1 495	43 557	1 518	35 042	1 543	33 049
BTD-MAC+Fusion	1 481	42 505	1 518	37 440	1 534	34 048
BTD-MAC+RST+Fusion	1 544	41 622	1 547	32 547	1 567	34 432

- H_2 : connexité des clusters
- H_5 : maîtrise de la taille des séparateurs

Min-Fill vs H_i

1 859 instances de la compétition CSP 2008

Algorithme	Min-Fill		H_2		H_5	
	#rés.	temps	#rés.	temps	#rés.	temps
BTD-MAC	1 344	43 272	1 405	31 429	1 469	33 564
BTD-MAC+RST	1 495	43 557	1 518	35 042	1 543	33 049
BTD-MAC+Fusion	1 481	42 505	1 518	37 440	1 534	34 048
BTD-MAC+RST+Fusion	1 544	41 622	1 547	32 547	1 567	34 432

- H_2 : connexité des clusters
- H_5 : maîtrise de la taille des séparateurs

Min-Fill vs H_i

1 859 instances de la compétition CSP 2008

Algorithme	Min-Fill		H_2		H_5	
	#rés.	temps	#rés.	temps	#rés.	temps
BTD-MAC	1 344	43 272	1 405	31 429	1 469	33 564
BTD-MAC+RST	1 495	43 557	1 518	35 042	1 543	33 049
BTD-MAC+Fusion	1 481	42 505	1 518	37 440	1 534	34 048
BTD-MAC+RST+Fusion	1 544	41 622	1 547	32 547	1 567	34 432

- H_2 : connexité des clusters
- H_5 : maîtrise de la taille des séparateurs

BTD vs BTD

1 859 instances de la compétition CSP 2008

Algorithme	Min-Fill		H_2		H_5	
	#rés.	temps	#rés.	temps	#rés.	temps
BTD-MAC	1 344	43 272	1 405	31 429	1 469	33 564
BTD-MAC+RST	1 495	43 557	1 518	35 042	1 543	33 049
BTD-MAC+Fusion	1 481	42 505	1 518	37 440	1 534	34 048
BTD-MAC+RST+Fusion	1 544	41 622	1 547	32 547	1 567	34 432

- H_2 : connexité des clusters
- H_5 : maîtrise de la taille des séparateurs

BTD vs BTD

1 859 instances de la compétition CSP 2008

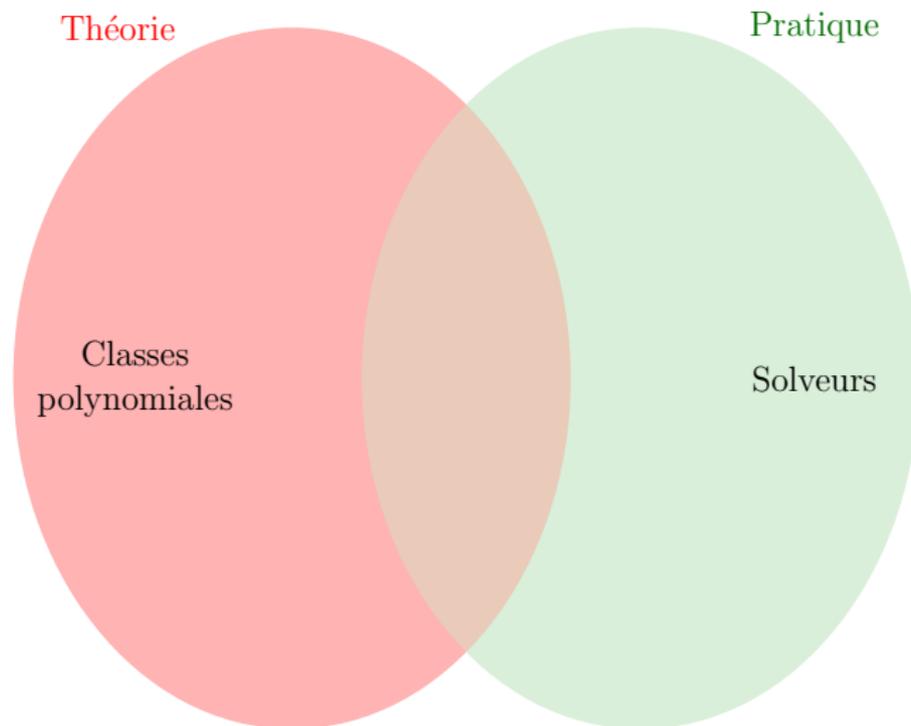
Algorithme	Min-Fill		H_2		H_5	
	#rés.	temps	#rés.	temps	#rés.	temps
BTD-MAC	1 344	43 272	1 405	31 429	1 469	33 564
BTD-MAC+RST	1 495	43 557	1 518	35 042	1 543	33 049
BTD-MAC+Fusion	1 481	42 505	1 518	37 440	1 534	34 048
BTD-MAC+RST+Fusion	1 544	41 622	1 547	32 547	1 567	34 432

- H_2 : connexité des clusters
- H_5 : maîtrise de la taille des séparateurs

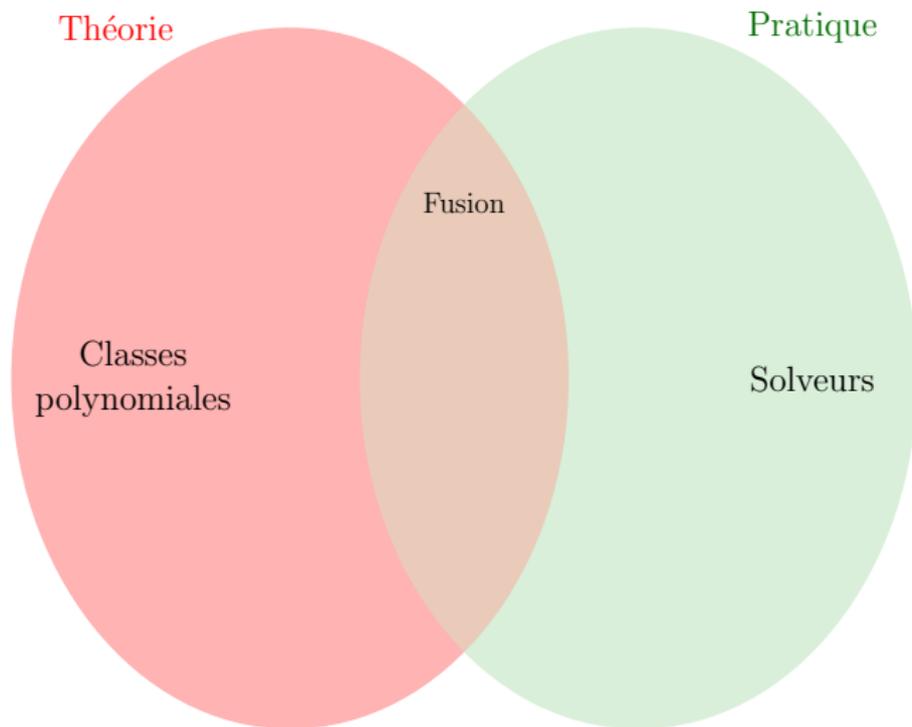
Plan

- 1 Introduction
- 2 Classes polynomiales hybrides
- 3 Classes polynomiales structurelles
- 4 Conclusion et perspectives

Conclusion

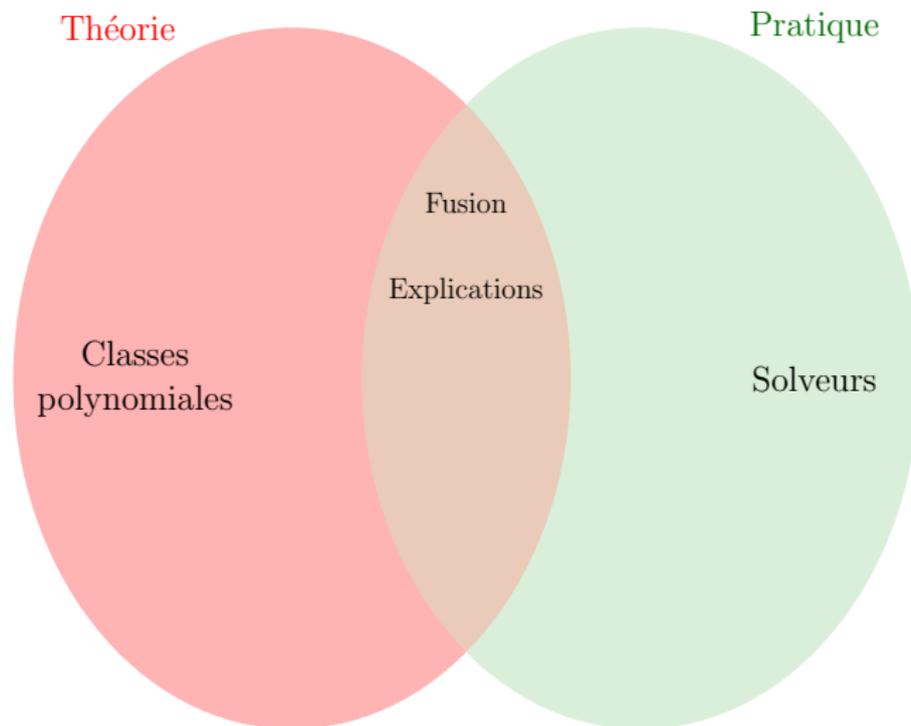


Conclusion

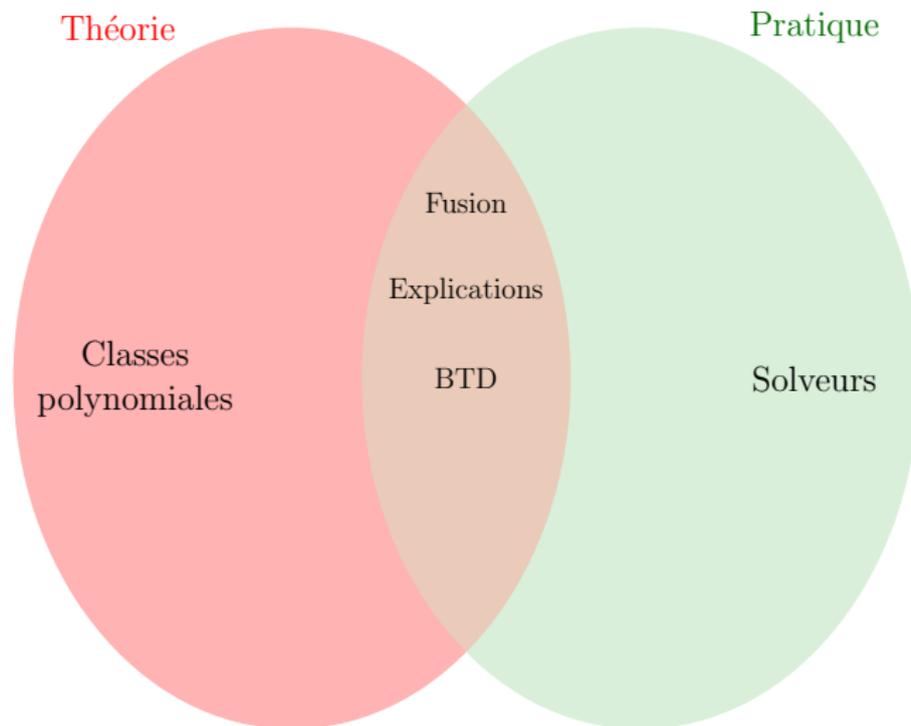


Award « Best Technical Track Paper », CP 2014

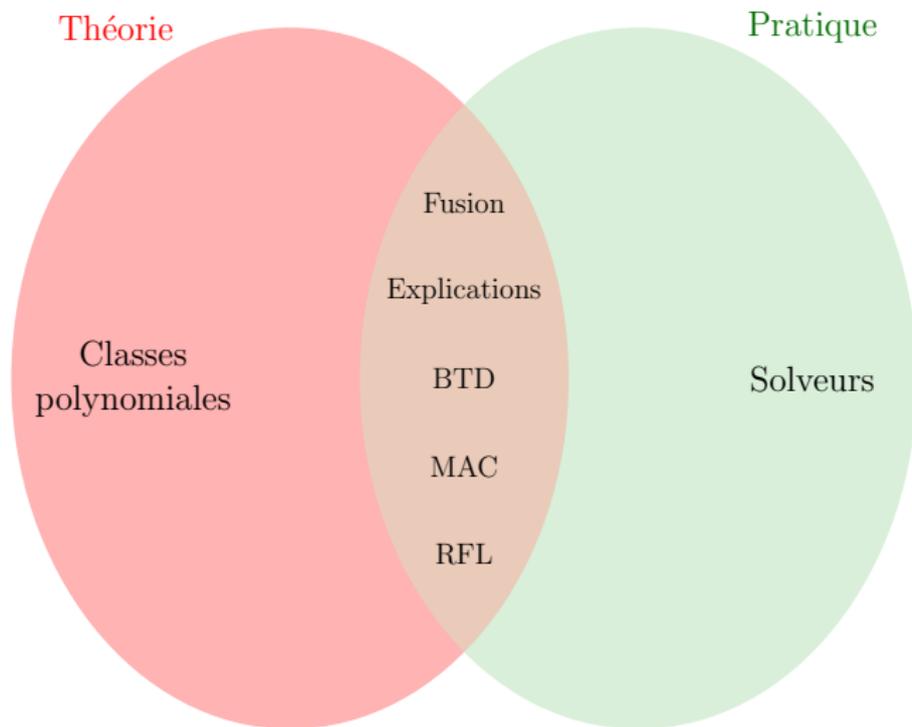
Conclusion



Conclusion

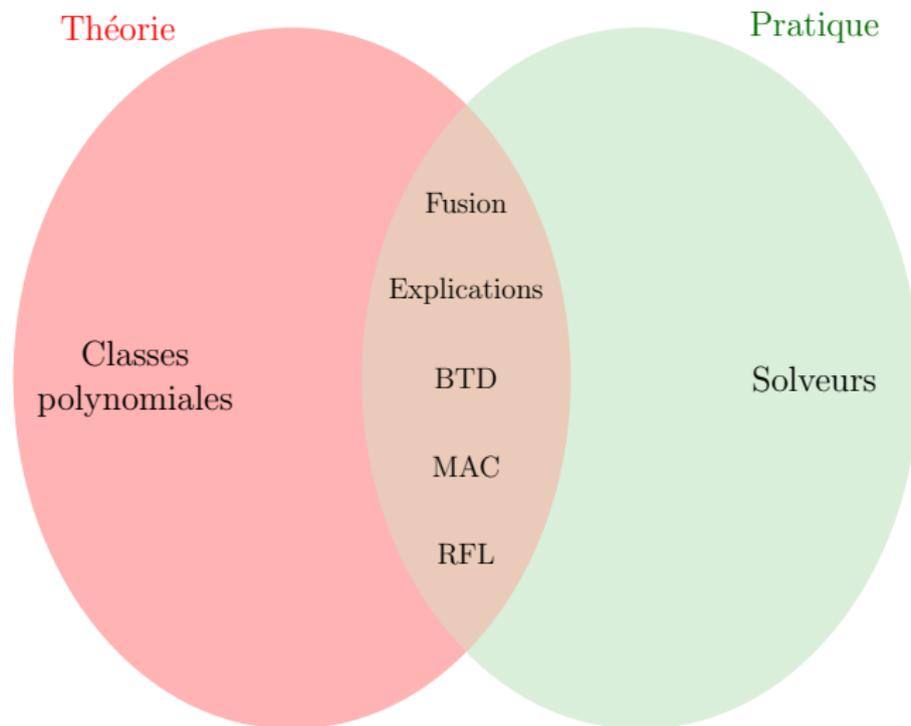


Conclusion



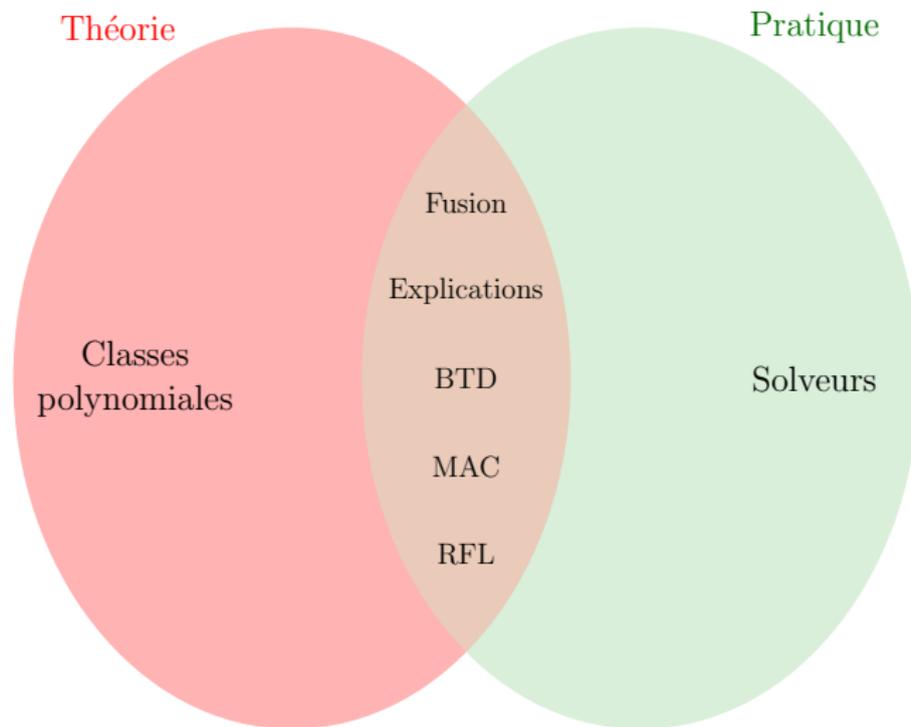
AIJ, Constraints, IJCAI, AAI, ECAI, CP, ...

Conclusion



Encadrements : 4 thèses, 8 M2R (21 publications internationales)

Conclusion



Participation à deux projets ANR

Perspectives

- Vers de nouvelles décompositions
 - de nombreuses décompositions proposées
 - mais pas d'algorithmes exploitables en pratique

Perspectives

- Vers de nouvelles décompositions
 - de nombreuses décompositions proposées
 - mais pas d'algorithmes exploitables en pratique
- Renforcer les liens entre classes polynomiales et solveurs :
 - Explications multi-critères
 - Définition de nouveaux algorithmes de résolution
 - Identification de nouvelles classes

Perspectives

- Vers de nouvelles décompositions
 - de nombreuses décompositions proposées
 - mais pas d'algorithmes exploitables en pratique
- Renforcer les liens entre classes polynomiales et solveurs :
 - Explications multi-critères
 - Définition de nouveaux algorithmes de résolution
 - Identification de nouvelles classes
- Explorer les relations entre classes hybrides et classes structurelles
 - $BTW_k \subsetneq (k+1)\text{-BTP}$

Perspectives

- Vers de nouvelles décompositions
 - de nombreuses décompositions proposées
 - mais pas d'algorithmes exploitables en pratique

- Renforcer les liens entre classes polynomiales et solveurs :
 - Explications multi-critères
 - Définition de nouveaux algorithmes de résolution
 - Identification de nouvelles classes

- Explorer les relations entre classes hybrides et classes structurelles
 - $BTW_k-S(k+1)C \not\subseteq (k+1)-BTP-S(k+1)C$

Perspectives

- Vers de nouvelles décompositions
 - de nombreuses décompositions proposées
 - mais pas d'algorithmes exploitables en pratique
- Renforcer les liens entre classes polynomiales et solveurs :
 - Explications multi-critères
 - Définition de nouveaux algorithmes de résolution
 - Identification de nouvelles classes
- Explorer les relations entre classes hybrides et classes structurelles
 - $BTW_k-S(k+1)C \not\subseteq (k+1)-BTP-S(k+1)C$
- Extensions à d'autres problèmes :
 - optimisation
 - comptage
 - compilation