Augmentation du filtrage de la contrainte AuPlusNValeur à l'aide d'altérations locales des multiplicateurs de Lagrange

Frédéric Berthiaume
Claude-Guy Quimper

UNIVERSITÉ LAVAL

Programmation par contraintes

En programmation par contraintes on modélise des problèmes à l'aide :

- d'objectifs;
- de variables;
- de contraintes;

Les variables

- Une variable X représente une **entité** dans le problème.
- Chaque X peut prendre certaines valeurs.
- L'ensemble des valeurs que peut prendre X s'appelle le **domaine** de X, noté $\mathrm{dom}(X)$.

Exemple de deux variables:

- X avec $\mathrm{dom}(X) = \{2, 3, 5\}$
- Y avec $\mathrm{dom}(Y) = \{1, 3, 4\}$

Les contraintes

Les contraintes **imposent** des relations sur les variables.

$$\mathrm{dom}(X) = \{2,3,5\}$$
$$\mathrm{dom}(Y) = \{1,3,4\}$$

Entrée : X Y | X Y, $X \leq Y$

Sortie $\langle X, Y \rangle$:

$$\begin{array}{ccc} \langle 2,1\rangle & \langle 2,3\rangle & \langle 2,4\rangle \\ \langle 3,1\rangle & \langle 3,3\rangle & \langle 3,4\rangle \\ \langle 5,1\rangle & \langle 5,3\rangle & \langle 5,4\rangle \end{array} \qquad \begin{array}{cc} \langle 2,3\rangle & \langle 2,4\rangle \\ \langle 3,3\rangle & \langle 3,4\rangle \end{array}$$

Les algorithmes de filtrage

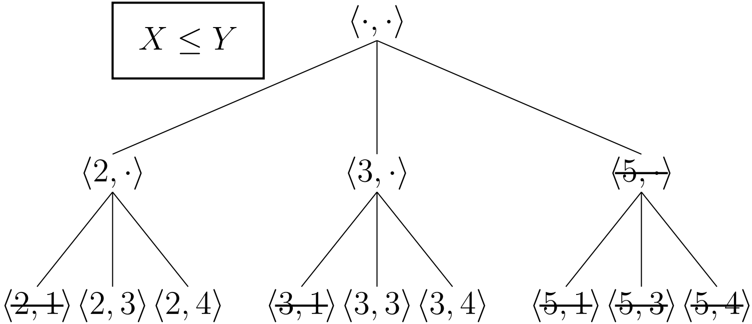
Chaque contrainte a un algorithme de filtrage qui retire les valeurs incohérentes du domaine des variables.

Avant	Algorithme de filtrage	Après
$\mathrm{dom}(X) = \{2, 3, 5\}$	$X \leq Y$	$\mathrm{dom}(X) = \{2, 3\}$
$\mathrm{dom}(Y) = \{1, 3, 4\}$		$\mathrm{dom}(Y) = \{3, 4\}$

Les valeurs $5 \notin \mathrm{dom}(X)$ et $1 \notin \mathrm{dom}(Y)$ après le filtrage.

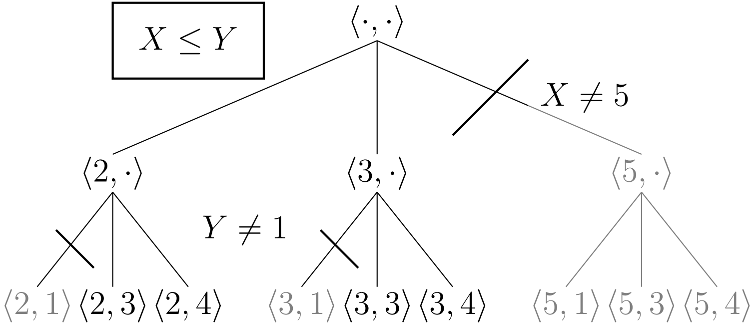
L'arbre de recherche

- **L'arbre de recherche** est un objet avec une structure d'arbre.
- Les feuilles de l'arbre sont toutes les combinaisons de valeurs que les variables peuvent prendre.



Algorithme de filtrage dans l'arbre de recherche

Les algorithmes de filtrage **élaguent** des branches de l'arbre.



La contrainte $\textsc{AuPlusNValeur}(\mathcal{X}, N)$

Les entrées :

- Un ensemble de variables $\mathcal{X} = \{X_1, X_2, \ldots, X_n\}$.
- Une variable de **cardinalité** N.

$$\textsc{AuPlusNValeur}(\mathcal{X}, N) \Leftrightarrow N \geq |\{X_1, X_2, \ldots, X_n\}|.$$

Exemple $\textsc{AuPlusNValeur}(\mathcal{X}, N)$

- $\mathrm{dom}(X_1) = \{1, 2\}$ et $\mathrm{dom}(X_2) = \{2, 3\}$,
- $\mathrm{dom}(N) = \{0, 1\}$

Avant	Algorithme de filtrage	Après
$\mathrm{dom}(X_1) = \{1, 2\}$ $\mathrm{dom}(X_2) = \{2, 3\}$ $\mathrm{dom}(N) = \{0, 1\}$	$\textsc{AuPlusNValeur}(\mathcal{X}, N)$	$\mathrm{dom}(X_1) = \{2\}$ $\mathrm{dom}(X_2) = \{2\}$ $\mathrm{dom}(N) = \{1\}$

Le programme linéaire de la contrainte AuPlusNValeur

La contrainte AuPlusNValeur s'encode avec ce programme linéaire.

$$\begin{aligned} \min_{\boldsymbol{y}} \quad & h(\boldsymbol{y}) = \mathbf{1}^\top \boldsymbol{y} \\ \text{s.t.} \quad & \mathrm{A}\boldsymbol{y} \geq \mathbf{1} \\ & \mathbf{0} \leq \boldsymbol{y} \leq \mathbf{1} \end{aligned}$$

Les $\boldsymbol{y}$ sont des variables *binaires* (relaxées).

$$\forall j \in \bigcup_{i=1}^{n} \mathrm{dom}(X_i), \quad j \longleftrightarrow y_j.$$

$y_j = 1$ signifie que la valeur j est utilisée pour couvrir des variables X_i.
$y_j = 0$ signifie que la valeur j n'est pas utilisée pour couvrir des variables X_i.

La fonction objectif $\min_{\boldsymbol{y}} \ h(\boldsymbol{y}) = \mathbf{1}^\top \boldsymbol{y}$

La contrainte AuPlusNValeur s'encode avec ce programme linéaire.

$$\begin{aligned} \min_{\boldsymbol{y}} \quad & h(\boldsymbol{y}) = \mathbf{1}^\top \boldsymbol{y} \\ \text{s.t.} \quad & \mathrm{A}\boldsymbol{y} \geq \mathbf{1} \\ & \mathbf{0} \leq \boldsymbol{y} \leq \mathbf{1} \end{aligned}$$

L'objectif est de minimiser le nombre de valeurs utilisées pour couvrir l'ensemble $\{X_1, \ldots, X_n\}$.

$$h(\boldsymbol{y}) = \mathbf{1}^\top \boldsymbol{y} = \sum_{j=1}^{m} y_j .$$

$h(\boldsymbol{y})$ est une borne inférieure sur N.

Les contraintes de couverture $\mathrm{A}\boldsymbol{y} \geq \mathbf{1}$

La matrice A contient de l'information sur les domaines des variables.

$$a_{ij} = \begin{cases} 1 & \text{si} \quad j \in \operatorname{dom}(X_i) \\ 0 & \text{si} \quad j \notin \operatorname{dom}(X_i) \end{cases}$$

Pour l'exemple $\operatorname{dom}(X_1) = \{1, 2\}$, $\operatorname{dom}(X_2) = \{2, 3\}$:

$$\mathrm{A} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}.$$

$$\mathrm{A}\boldsymbol{y} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} y_1 + y_2 \\ y_2 + y_3 \end{pmatrix} = \begin{pmatrix} \#\ \text{valeurs couvrant } X_1 \\ \#\ \text{valeurs couvrant } X_2 \end{pmatrix} \geq \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Le programme linéaire de la contrainte AuPlusNValeur

En résumé, voici le programme linéaire de la contrainte AuPlusNValeur est

$$\begin{aligned} \min_{\boldsymbol{y}} \quad & h(\boldsymbol{y}) = \mathbf{1}^\top \boldsymbol{y} \\ \text{s.t.} \quad & \mathrm{A}\boldsymbol{y} \geq \mathbf{1} \\ & \mathbf{0} \leq \boldsymbol{y} \leq \mathbf{1} \end{aligned}$$

Filtrage par coût-réduit

- Le programme linéaire a été résolu et on a trouvé une solution $\boldsymbol{y}$.
- Le coût-réduit d'une valeur j est écrit ρ_j.

$$\rho_j = h(\boldsymbol{y'}) - h(\boldsymbol{y})$$

  où $\boldsymbol{y'}$ est une solution où on a forcé j a être utilisée (si $y_j = 0$) ou a ne pas être utilisée (si $y_j = 1$).
- Si $h(\boldsymbol{y'}) > \max(\mathrm{dom}(N))$ la valeur j ne peut pas être forcé dans un autre état que celui de $\boldsymbol{y}$.

La relaxation Lagrangienne de la contrainte AuPlusNValeur

Cambazard et Fages ont fait cette relaxation

$$\begin{array}{rl} \min_{\boldsymbol{y}} & h(\boldsymbol{y}) = \mathbf{1}^\top \boldsymbol{y} \\ \text{s.t.} & \mathrm{A}\boldsymbol{y} \geq \mathbf{1} \\ & \mathbf{0} \leq \boldsymbol{y} \leq \mathbf{1} \end{array} \qquad \begin{array}{rl} \min_{\boldsymbol{y}} & h'(\boldsymbol{y}, \boldsymbol{\lambda}) = \mathbf{1}^\top \boldsymbol{y} + \boldsymbol{\lambda}^\top (\mathbf{1} - \mathrm{A}\boldsymbol{y}) \\ \text{s.t.} & \mathbf{0} \leq \boldsymbol{y} \leq \mathbf{1} \\ & \boldsymbol{\lambda} \geq \mathbf{0} \end{array}$$

$\boldsymbol{\lambda}$ est un vecteur de **multiplicateurs de Lagrange**.
Où $\boldsymbol{\lambda} \geq \mathbf{0}$ assure le respect des contraintes.

Le respect des contraintes et $\boldsymbol{\lambda} \geq \mathbf{0}$

$$\begin{array}{ll} \min_{\boldsymbol{y}} & h(\boldsymbol{y}) = \mathbf{1}^\top \boldsymbol{y} \\ \text{s.t.} & \mathrm{A}\boldsymbol{y} \geq \mathbf{1} \\ & \mathbf{0} \leq \boldsymbol{y} \leq \mathbf{1} \end{array} \qquad \begin{array}{ll} \min_{\boldsymbol{y}} & h'(\boldsymbol{y}, \boldsymbol{\lambda}) = \mathbf{1}^\top \boldsymbol{y} + \boldsymbol{\lambda}^\top (\mathbf{1} - \mathrm{A}\boldsymbol{y}) \\ \text{s.t.} & \mathbf{0} \leq \boldsymbol{y} \leq \mathbf{1} \\ & \boldsymbol{\lambda} \geq \mathbf{0} \end{array}$$

Regardons le terme ajouté pour la variable X_1 du problème avec $\mathrm{dom}(X_1) = \{1, 2\}$ et $\mathrm{dom}(X_1) = \{2, 3\}$:

$$\lambda_1 \left(1 - \begin{pmatrix} 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} y_1 & y_2 & y_3 \end{pmatrix}^\top \right) = \lambda_1 (1 - y_1 - y_2).$$

État de X_1	$\boldsymbol{y}$	$\lambda_1(1 - y_1 - y_2)$	$h'(\boldsymbol{y}, \boldsymbol{\lambda})$
non couverte	$(0, 0, y_3)^\top$	$\lambda_1(1 - 0 - 0) = \lambda_1$	augmente
couverte	$(1, 0, y_3)^\top$	$\lambda_1(1 - 1 - 0) = 0$	stagne
	$(0, 1, y_3)^\top$	$\lambda_1(1 - 0 - 1) = 0$	stagne
	$(1, 1, y_3)^\top$	$\lambda_1(1 - 1 - 1) = -\lambda_1$	diminue

Résoudre le problème relaxé

La relaxation présentée :

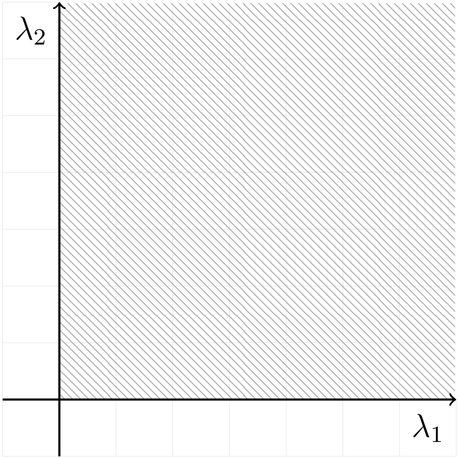
$$\begin{array}{cc}
\begin{aligned} \min_{\boldsymbol{y}} \;\; & h(\boldsymbol{y}) = \mathbf{1}^\top \boldsymbol{y} \\ \text{s.t.} \;\; & \mathrm{A}\boldsymbol{y} \geq \mathbf{1} \\ & \mathbf{0} \leq \boldsymbol{y} \leq \mathbf{1} \end{aligned}
&
\begin{aligned} \min_{\boldsymbol{y}} \;\; & h'(\boldsymbol{y}, \boldsymbol{\lambda}) = \mathbf{1}^\top \boldsymbol{y} + \boldsymbol{\lambda}^\top (\mathbf{1} - \mathrm{A}\boldsymbol{y}) \\ \text{s.t.} \;\; & \mathbf{0} \leq \boldsymbol{y} \leq \mathbf{1} \\ & \boldsymbol{\lambda} \geq \mathbf{0} \end{aligned}
\end{array}$$

Celle qu'on utilise pour résoudre le problème :

$$\begin{array}{cc}
\begin{aligned} \min_{\boldsymbol{y}} \;\; & h(\boldsymbol{y}) = \mathbf{1}^\top \boldsymbol{y} \\ \text{s.t.} \;\; & \mathrm{A}\boldsymbol{y} \geq \mathbf{1} \\ & \mathbf{0} \leq \boldsymbol{y} \leq \mathbf{1} \end{aligned}
&
\begin{aligned} \min_{\boldsymbol{y}} \;\; & h'(\boldsymbol{y}, \boldsymbol{\lambda}) = \left(\mathbf{1} - \mathrm{A}^\top \boldsymbol{\lambda}\right)^\top \boldsymbol{y} + \boldsymbol{\lambda}^\top \mathbf{1} \\ \text{s.t.} \;\; & \mathbf{0} \leq \boldsymbol{y} \leq \mathbf{1} \\ & \boldsymbol{\lambda} \geq \mathbf{0} \end{aligned}
\end{array}$$

Le vecteur de coût réduit est $\boldsymbol{q}(\boldsymbol{\lambda}) = \mathbf{1} - \mathrm{A}^\top \boldsymbol{\lambda}$.

L'espace des multiplicateurs de Lagrange

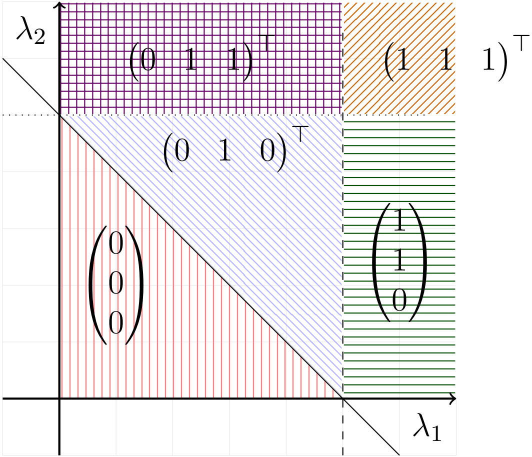


$\{X_1, X_2\}$

$\mathrm{dom}(X_1) = \{1, 2\}$

$\mathrm{dom}(X_2) = \{2, 3\}$

Différentes régions produisent différents $\boldsymbol{y}$

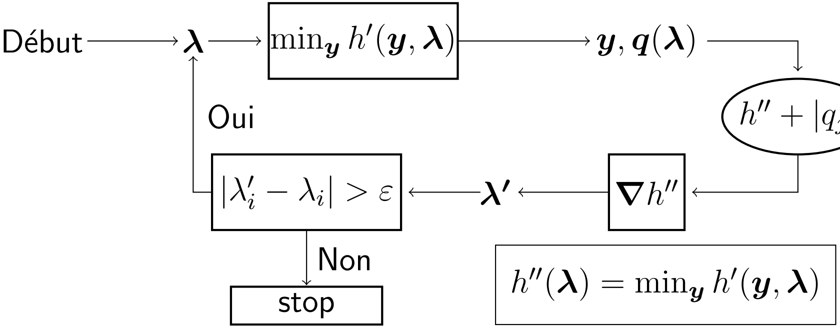


$$\{X_1, X_2\}$$

$$\mathrm{dom}(X_1) = \{1, 2\}$$

$$\mathrm{dom}(X_2) = \{2, 3\}$$

Relaxation Lagrangienne



L'optimisation Lagrangienne

La relaxation présentée :

$$\begin{array}{ll} \min_{\boldsymbol{y}} & h(\boldsymbol{y}) = \mathbf{1}^\top \boldsymbol{y} \\ \text{s.t.} & \mathrm{A}\boldsymbol{y} \geq \mathbf{1} \\ & \mathbf{0} \leq \boldsymbol{y} \leq \mathbf{1} \end{array} \qquad \begin{array}{ll} \min_{\boldsymbol{y}} & h'(\boldsymbol{y}, \boldsymbol{\lambda}) = \mathbf{1}^\top \boldsymbol{y} + \boldsymbol{\lambda}^\top (\mathbf{1} - \mathrm{A}\boldsymbol{y}) \\ \text{s.t.} & \mathbf{0} \leq \boldsymbol{y} \leq \mathbf{1} \\ & \boldsymbol{\lambda} \geq \mathbf{0} \end{array}$$

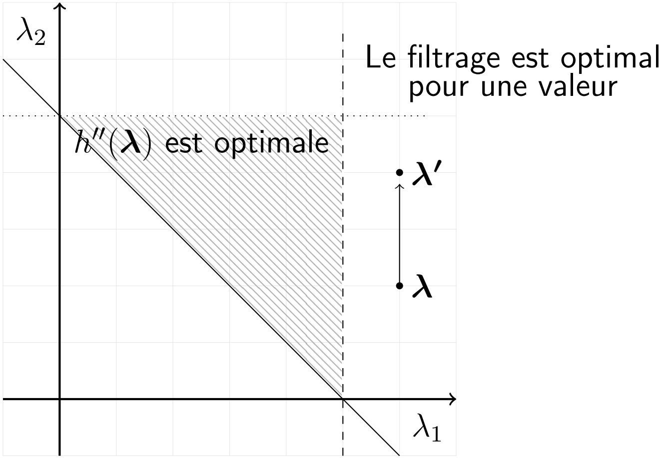
On regarde le respect des contraintes $\boldsymbol{\nabla} h''(\boldsymbol{\lambda}) = \mathbf{1} - \mathrm{A}\boldsymbol{y}$.
Plus il y a de reines, moins $h(\boldsymbol{y})$ est minimisée. Ainsi on optimise $\boldsymbol{\lambda}$ pour respecter le premier objectif

$$\boldsymbol{\lambda}^{(\boldsymbol{k}+\mathbf{1})} = \max(\mathbf{0}, \boldsymbol{\lambda}^{(\boldsymbol{k})} + \alpha_k \boldsymbol{\nabla} h''(\boldsymbol{\lambda}^{(\boldsymbol{k})}))$$

Ce type de méthode a la propriété de converger vers des maximums globaux parce que la fonction $h''(\boldsymbol{\lambda})$ est *concave*.

L'algorithme d'altérations locales

L'idée est de modifier les multiplicateurs de Lagrange pour augmenter le filtrage.



Quelle direction?

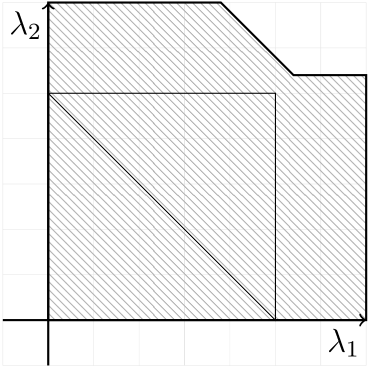
- On a définit une nouvelle fonction : la fonction de borne inférieure forcée

$$\rho_j(\boldsymbol{\lambda}) = h''(\boldsymbol{\lambda}) + |q_j(\boldsymbol{\lambda})|$$

- On utilise cette nouvelle fonction pour diriger l'optimisation des $\boldsymbol{\lambda}$.
- On recommence une optimisation Lagrangienne pour chaque (ou du moins presque) valeur j.

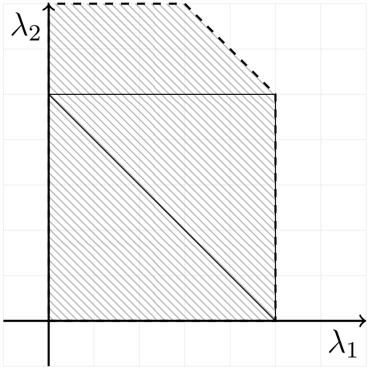
Un seuil

- Regarder toutes les valeurs possibles peut être très long.
- On ajoute une étape, donc il faut être efficace.
- On introduit un seuil pour choisir quelle valeur prendre.



Notre objectif

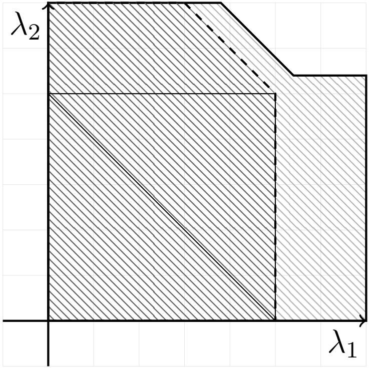
L'objectif est que l'optimisation de multiplicateur de Lagrange tombe dans la zone de filtrage:



Dans cette zone, on filtre la valeur testée.

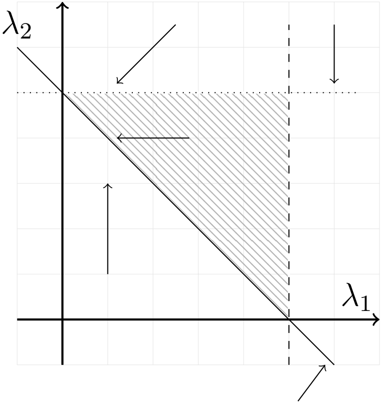
Le nombre de pas

On a aussi un critère de convergence sur le nombre de pas qu'on permet pour se rendre dans la zone de filtrage.



Un gros problème

La fonction $\rho_j(\boldsymbol{\lambda}) = h''(\boldsymbol{\lambda}) + |q_j(\boldsymbol{\lambda})|$ n'est pas **concave** (et ni convexe). Alors on perd la propriété de converger à un maximum global.

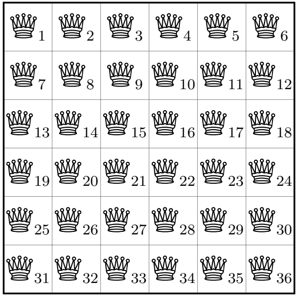


Expériences et résultats

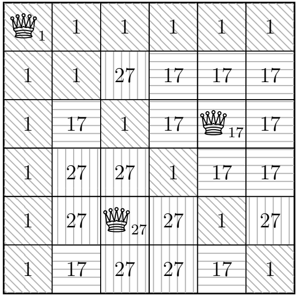
- Le problème des reines dominantes
- Le problème de disposition d'entrepôts

Le problème des reines dominantes

Le problème des reines dominante a vise à couvrir complètement un échiquier $n \times n$ avec le moins de reines possible.



$\Rightarrow$



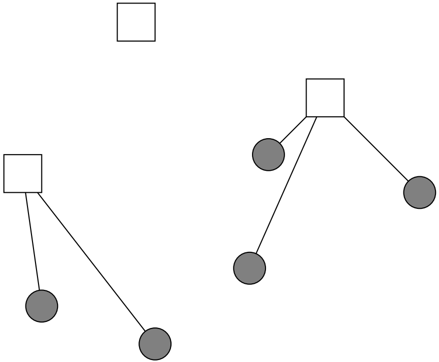
Résultats

		Harmonic				Geometric				Newton			
n/v F		Nodes	Fails	Iters	Time	Nodes	Fails	Iters	Time	Nodes	Fails	Iters	Time
7/4 Y	LR^0	142	112	170k	0.165	166	135	84k	0.089	176	145	19k	0.026
	LR^+	**31**	**4**	**19k**	**0.024**	**31**	**4**	**6k**	**0.011**	**31**	**4**	**2k**	**0.005**
8/5 Y	LR^0	267	225	329k	0.301	362	320	186k	0.197	283	241	33k	0.045
	LR^+	**41**	**6**	**26k**	**0.040**	**76**	**41**	**24k**	**0.028**	**41**	**6**	**3k**	**0.007**
8/4 N	LR^0	2k	2k	2.4M	3.497	3k	3k	1.6M	2.461	2k	2k	279k	0.765
	LR^+	**1.2k**	**1.2k**	**1.5M**	**2.066**	**1k**	**1k**	**698k**	**1.177**	**1k**	**1k**	**146k**	**0.362**
9/5 Y	LR^0	1k	953	1.1M	2.188	1k	969	544k	1.315	956	909	113k	0.443
	LR^+	**490**	**443**	**576k**	**0.880**	**561**	**514**	**304k**	**0.556**	**441**	**394**	**57k**	**0.167**
10/5 Y	LR^0	113k	113k	133M	300	195k	195k	107M	300	944k	944k	91M	300
	LR^+	**9.4k**	**9.4k**	**11M**	**26.9**	**761**	**725**	**471k**	**3.711**	**844**	**808**	**121k**	**2.59**
11/5 Y	LR^0	32k	32k	38M	102.2	54k	54k	30M	83.1	83k	83k	7.34M	37.4
	LR^+	**4k**	**4k**	**5.6M**	**28.7**	**3.4k**	**3.4k**	**2.1M**	**11.6**	**4k**	**4k**	**461k**	**9.61**

Table: Dominating queens instances on $n \times n$ chessboards with v queens. If the instance is *feasible* we wrote Y and if not N. The time is in seconds.

Le problème de disposition d'entrepôts

On a des clients et d'entrepôts et on minimise les coûts de transport pour amener les produits des entrepôts aux clients.



Comparaison de LR^0 et LR^+

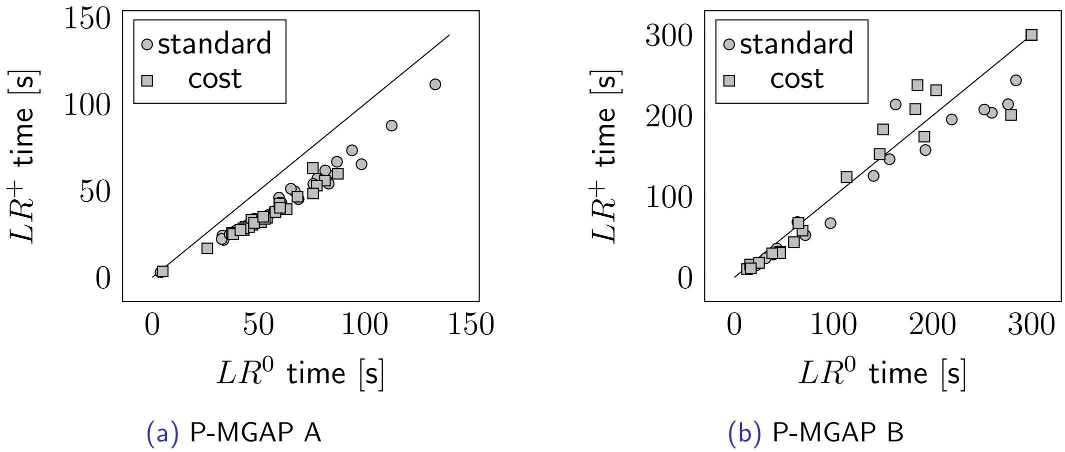


Figure: Time comparison of LR^0 and LR^+

Résultats sur les trois classes

Problem	Heuristics	Optimal		Proof optimal		# of better solutions		Av. time gain of LR^+ on LR^{0*}	Av. node reduction from LR^0 to LR^{+*}
		LR^0	LR^+	LR^0	LR^+	LR^0	LR^+	[%]	[%]
A	standard	27	27	27	27	0	**1**	28.4	71.5
	cost	29	**30**	27	27	0	**1**	33.2	72.0
B	standard	28	28	18	18	0	**2**	15.6	72.6
	cost	28	28	18	18	0	0	8.10	71.3
C	standard	5	**6**	0	0	0	**8**	–	–
	cost	8	**12**	0	0	0	**13**	–	–

Table 3: Results of the three classes of problem of the facility location problem. Each class has 30 instances and the solutions are known.
* *On the instances which we were able to prove the optimality.*

Conclusion

- Notre algorithme est plus rapide.
- Notre algorithme réduit significativement la taille de l'arbre de recherche.
- Nous allons nous pencher sur le problème du Sac à Dos.