Introduction à la programmation linéaire à travers le "Production Routing Problem"

Anne-Elisabeth FALQ

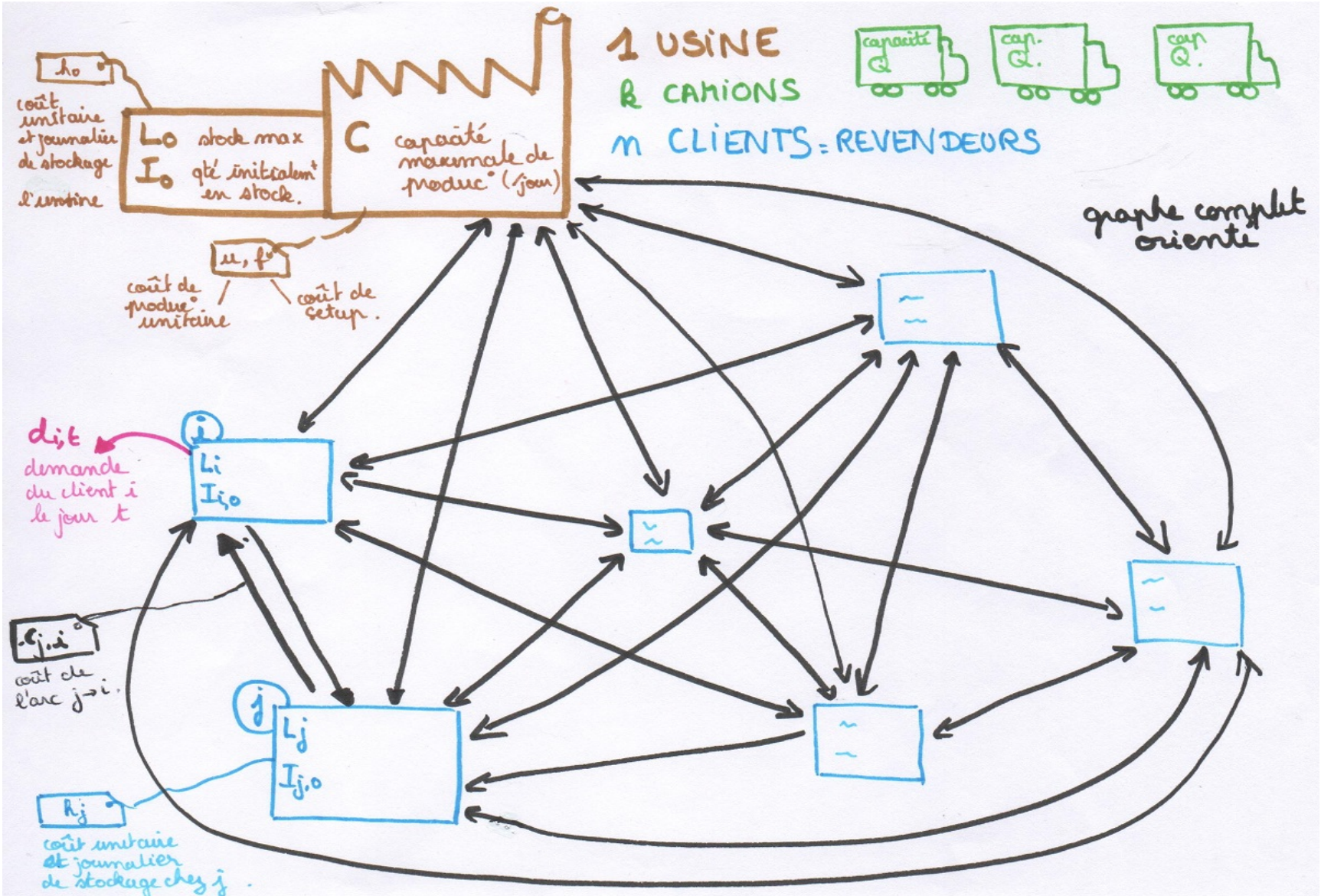
de l'UPMC à l'ENS Rennes

3 Février 2017

Plan

1. Le problème concret
2. Modélisation par un PL mixte
3. Comment résoudre une formulation compacte ?
4. Une reformulation non compacte
5. Conclusion

Le problème concret en image



Quelques hypothèses

- Chaque client ne peut être livré que par un seul camion
- La flotte de camion est homogène
- Lorsque les camions partent en tournée la production du jour est disponible

1. Le problème concret

2. Modélisation par un PL mixte
   - Aparté qu'est ce qu'un Programme Linéaire (PL)
   - Un PL pour le PRP
   - Zoom sur la fonction objectif
   - Zoom sur les contraintes (11) et (12)

3. Comment résoudre une formulation compacte ?

4. Une reformulation non compacte

5. Conclusion

Qu'est ce qu'un PL

$$\text{un PL} = \begin{cases} \text{Un nombre fini de variables, disons } n & \\ \text{Une fonction linéaire de } \mathbb{R}^n \text{ dans } \mathbb{R} & \leftarrow \text{ la fonction objectif} \\ \text{Des inégalités linéaires} & \leftarrow \text{ les contraintes} \end{cases}$$

Qu'est ce qu'un PL

$$\text{un PL} = \begin{cases} \text{Un nombre fini de variables, disons } n & \\ \text{Une fonction linéaire de } \mathbb{R}^n \text{ dans } \mathbb{R} & \leftarrow \text{ la fonction objectif} \\ \text{Des inégalités linéaires} & \leftarrow \text{ les contraintes} \end{cases}$$

$$\text{exemple 1 : } \begin{cases} \max\ (1,5x_1 + 5x_2) & \leftarrow \text{ la fonction objectif} \\ \left.\begin{array}{l} 2x_1 + 3x_2 \leqslant 7 \\ x_1 \geqslant 0 \\ x_2 \geqslant 0 \end{array}\right\} & \text{les contraintes} \end{cases}$$

Qu'est ce qu'un PL

$$\text{un PL} = \begin{cases} \text{Un nombre fini de variables, disons } n & \\ \text{Une fonction linéaire de } \mathbb{R}^n \text{ dans } \mathbb{R} & \leftarrow \text{ la fonction objectif} \\ \text{Des inégalités linéaires} & \leftarrow \text{ les contraintes} \end{cases}$$

$$\text{un PLNE} = \begin{cases} \text{Un nombre fini de variables, disons } n & \\ \text{Une fonction linéaire de } \mathbb{R}^n \text{ dans } \mathbb{R} & \leftarrow \text{ la fonction objectif} \\ \text{Des inégalités linéaires} & \left.\begin{array}{l} \\ \\ \end{array}\right\} \text{ les contraintes} \\ \text{Des contraintes d'intégrité} & \end{cases}$$

Qu'est ce qu'un PL

$$\text{un PLNE} = \begin{cases} \text{Un nombre fini de variables, disons } n \\ \text{Une fonction linéaire de } \mathbb{R}^n \text{ dans } \mathbb{R} & \leftarrow \text{ la fonction objectif} \\ \text{Des inégalités linéaires} \\ \text{Des contraintes d'intégrité} \end{cases} \Big\} \text{ les contraintes}$$

$$\text{exemple 2 : } \begin{cases} \max\ (1,5x_1 + 5x_2) & \leftarrow \text{ la fonction objectif} \\ 2x_1 + 3x_2 \leqslant 7 \\ x_1 \geqslant 0 \\ x_2 \geqslant 0 \\ x_1 \in \mathbb{Z} \\ x_2 \in \mathbb{Z} \end{cases} \Big\} \text{ les contraintes}$$

Image à avoir en tête pour un PL

exemple 1 :

$$\begin{cases} \max\ (1,5x_1 + 5x_2) \\ \\ 2x_1 + 3x_2 \leqslant 7 \\ x_1 \geqslant 0 \\ x_2 \geqslant 0 \end{cases}$$

Image à avoir en tête pour un PL

exemple 1 :

$$\begin{cases} \max\ (1,5x_1 + 5x_2) \\ \\ 2x_1 + 3x_2 \leqslant 7 \\ x_1 \geqslant 0 \\ x_2 \geqslant 0 \end{cases}$$

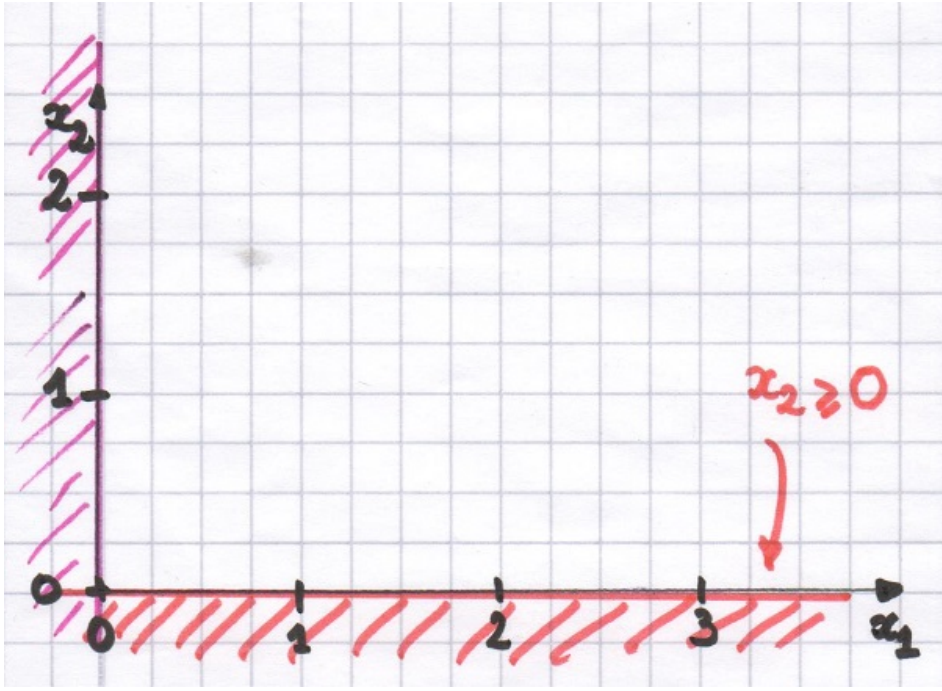


Image à avoir en tête pour un PL

exemple 1 :

$$\begin{cases} \max\ (1,5x_1 + 5x_2) \\ \\ 2x_1 + 3x_2 \leqslant 7 \\ x_1 \geqslant 0 \\ x_2 \geqslant 0 \end{cases}$$

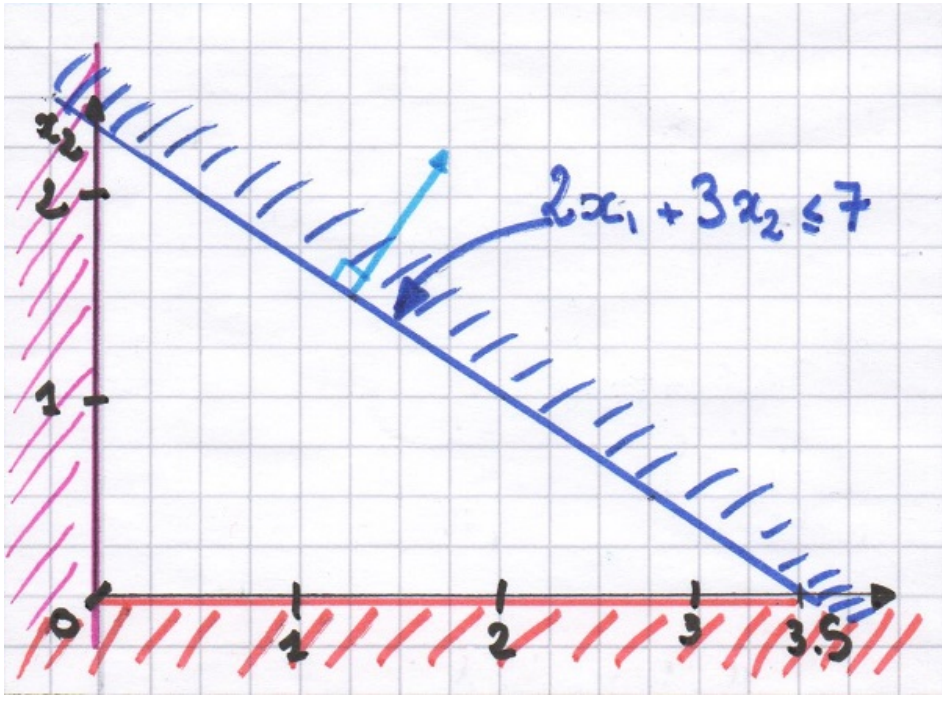


Image à avoir en tête pour un PL

exemple 1 :

$$\begin{cases} \max\ (1,5x_1 + 5x_2) \\ \\ 2x_1 + 3x_2 \leqslant 7 \\ x_1 \geqslant 0 \\ x_2 \geqslant 0 \end{cases}$$

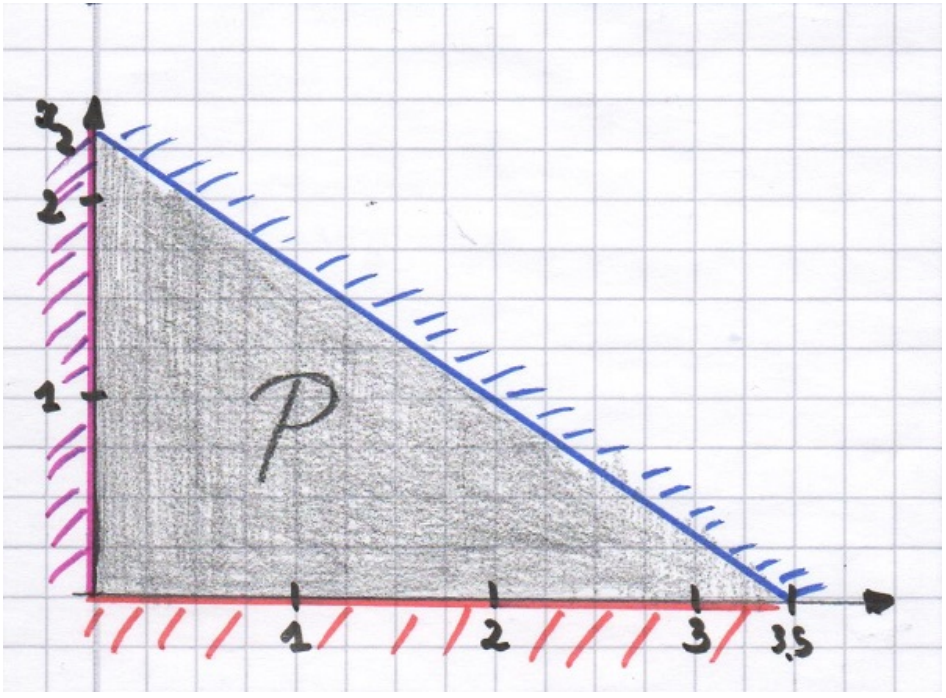


Image à avoir en tête pour un PL

exemple 1 :

$$\begin{cases} \max\ (1,5x_1 + 5x_2) \\ \\ 2x_1 + 3x_2 \leqslant 7 \\ x_1 \geqslant 0 \\ x_2 \geqslant 0 \end{cases}$$

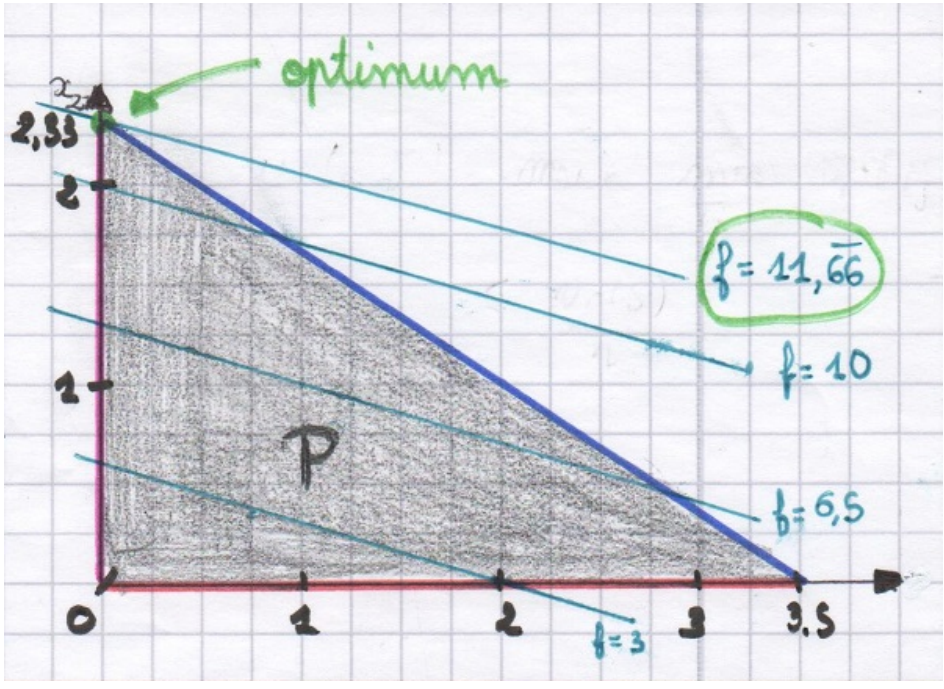


Image à avoir en tête pour un PLNE

exemple 2 :

$$\begin{cases} \max\ (1,5x_1 + 5x_2) \\ \\ 2x_1 + 3x_2 \leqslant 7 \\ x_1 \geqslant 0 \\ x_2 \geqslant 0 \\ x_1 \in \mathbb{Z} \\ x_2 \in \mathbb{Z} \end{cases}$$

Image à avoir en tête pour un PLNE

exemple 2 :

$$\begin{cases} \max\ (1,5x_1 + 5x_2) \\ \\ 2x_1 + 3x_2 \leqslant 7 \\ x_1 \geqslant 0 \\ x_2 \geqslant 0 \\ x_1 \in \mathbb{Z} \\ x_2 \in \mathbb{Z} \end{cases}$$

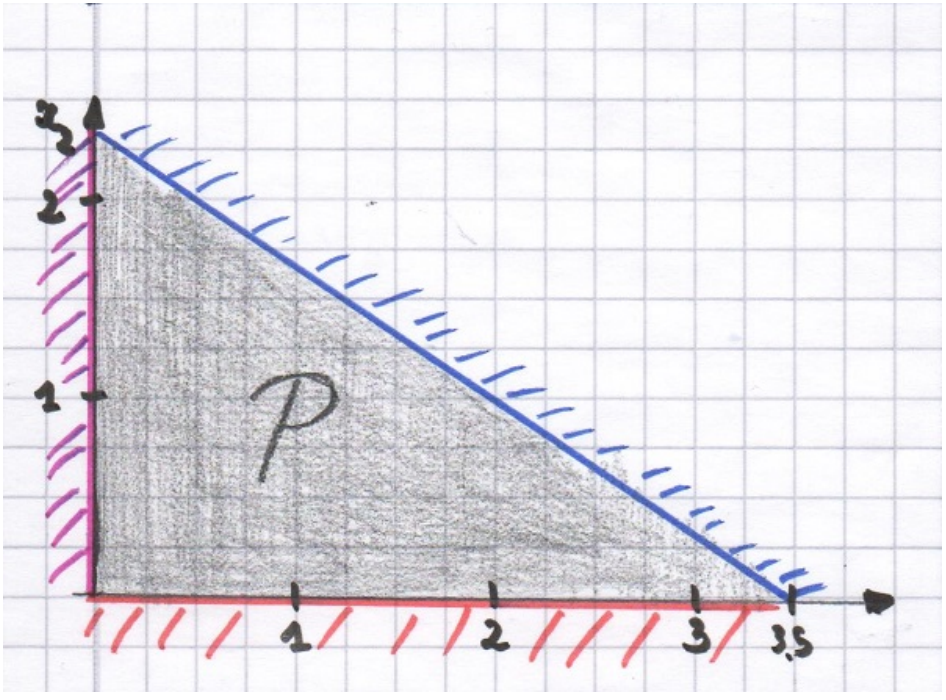


Image à avoir en tête pour un PLNE

exemple 2 :

$$\begin{cases} \max\ (1,5x_1 + 5x_2) \\ \\ 2x_1 + 3x_2 \leqslant 7 \\ x_1 \geqslant 0 \\ x_2 \geqslant 0 \\ x_1 \in \mathbb{Z} \\ x_2 \in \mathbb{Z} \end{cases}$$

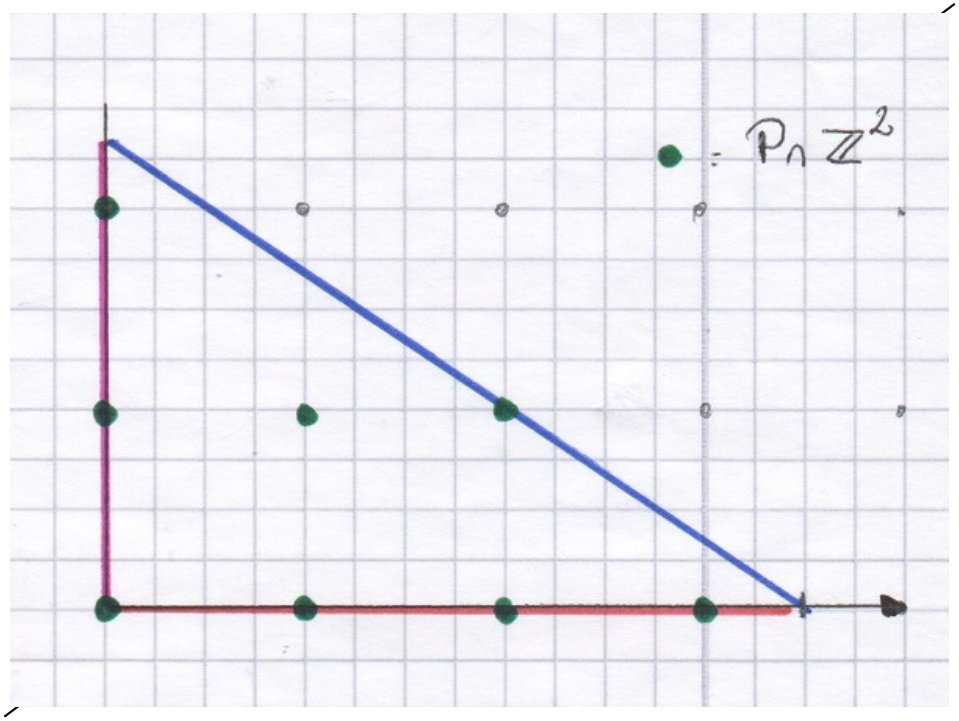
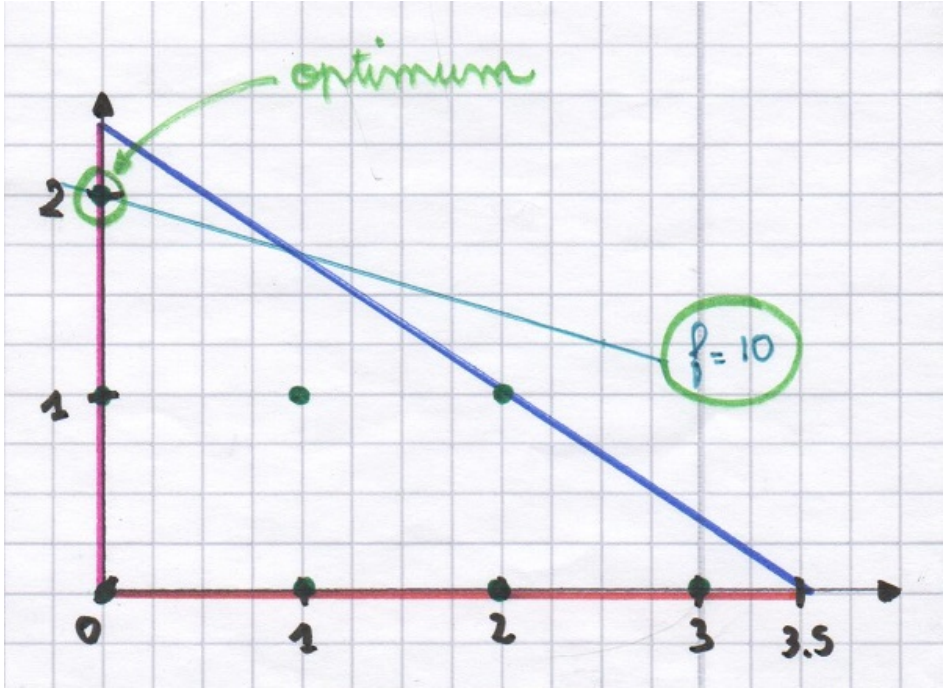


Image à avoir en tête pour un PLNE

exemple 2 :

$$\begin{cases} \max\ (1,5x_1 + 5x_2) \\ \\ 2x_1 + 3x_2 \leqslant 7 \\ x_1 \geqslant 0 \\ x_2 \geqslant 0 \\ x_1 \in \mathbb{Z} \\ x_2 \in \mathbb{Z} \end{cases}$$



PL ou PLNE, quelle différence ?

Mauvaise nouvelle : Résoudre un PLNE est difficile en général

PL ou PLNE, quelle différence ?

Mauvaise nouvelle : Résoudre un PLNE est difficile en général

Bonne nouvelle : Résoudre un PL se fait en temps polynomial

algorithme des points intérieurs

PL ou PLNE, quelle différence ?

Mauvaise nouvelle :	Résoudre un PLNE est difficile en général
Bonne nouvelle :	Résoudre un PL se fait en temps polynomial
	algorithme des points intérieurs
Meilleure nouvelle :	Résoudre un PL se fait efficacement
	algorithme du simplexe

PL ou PLNE, quelle différence ?

Mauvaise nouvelle :	Résoudre un PLNE est difficile en général
Bonne nouvelle :	Résoudre un PL se fait en temps polynomial
	algorithme des points intérieurs
Meilleure nouvelle :	Résoudre un PL se fait efficacement
	algorithme du simplexe

En maximisation, la résolution du PL fournit une borne sup' du PLNE.
En minimisation, la résolution du PL fournit une borne inf' du PLNE.

$$
\text{PRP}_1 = \begin{cases}
(1) & \min \sum_{t\in T} up_t + fy_t + \sum_{t\in T}\sum_{i\in N} h_i I_{i,t} + \sum_{t\in T}\sum_{(i,j)\in A} c_{i,j}x_{i,j,t} \\
(2) & \forall t \in T,\ \ I_{0,t-1} + p_t = \sum_{i\in N_c} q_{i,t} + I_{0,t} \\
(3) & \forall t \in T, \forall i \in N_c\ \ I_{i,t-1} + q_{i,t} = d_{i,t} + I_{i,t} \\
(4) & \forall t \in T,\ \ p_t \leqslant M_t y_t \\
(5) & \forall t \in T,\ \ I_{0,t} \leqslant L_0 \\
(6) & \forall t \in T, \forall i \in N_c,\ \ I_{i,t-1} + q_{i,t} \leqslant L_i \\
(7) & \forall t \in T, \forall i \in N_c,\ \ q_{i,t} \geqslant \tilde{M}_{i,t} z_{i,t} \\
(8) & \forall t \in T, \forall i \in N_c,\ \ z_{i,t} = \sum_{j\in N} x_{i,j,t} \\
(9) & \forall t \in T,\ \ 2z_{i,t} = \sum_{j\in N} x_{i,j,t} + \sum_{j\in N} x_{j,i,t} \\
(10) & \forall t \in T,\ \ z_{0,t} \leqslant k \\
(11) & \forall t \in T, \forall i \in N_c, \forall j \in N,\ \ w_{i,t} - w_{j,t} \geqslant q_{i,t} - Q(1 - x_{i,j,t}) \\
(12) & \forall t \in T, \forall i \in N_c,\ \ 0 \leqslant w_{i,t} \leqslant Q z_{i,t} \\
(13) & \forall t \in T, \forall i \in N\ \ p_t, I_{i,t}, q_{i,t} \geqslant 0 \\
(14) & \forall t \in T, \forall (i,j) \in N \times N\ \ y_t, x_{i,j,t} \in 0,1 \\
(15) & \forall t \in T, \forall i \in N_c\ \ z_{i,t} \in 0,1 \\
(16) & \forall t \in T,\ \ z_{0,t} \in \mathbb{N}
\end{cases}
$$

La fonction objectif

$$\min \sum_{t \in T} u\boldsymbol{p_t} + f\boldsymbol{y_t} + \sum_{t \in T} \sum_{i \in N} h_i \boldsymbol{l_{i,t}} + \sum_{t \in T} \sum_{(i,j) \in A} c_{i,j} \boldsymbol{x_{i,j,t}}$$

Choix des variables

Des variables continues...

$$\min \sum_{t \in T} u \quad \underbrace{\boldsymbol{p_t}}_{\substack{\text{production} \\ \text{du jour t}}} + f\,\boldsymbol{y_t} + \sum_{t \in T} \sum_{i \in N} h_i \quad \underbrace{\boldsymbol{l_{i,t}}}_{\substack{\text{stock chez} \\ \text{le client i} \\ \text{au soir t}}} + \sum_{t \in T} \sum_{(i,j) \in A} c_{i,j} \boldsymbol{x_{i,j,t}}$$

Choix des variables

... et d'autres booléennes donc a fortiori entières.

$$\min \sum_{t \in T} u\boldsymbol{p_t} + f \quad \underset{\substack{\uparrow \\ \text{ouverture} \\ \text{le jour t}}}{\boldsymbol{y_t}} \quad + \sum_{t \in T} \sum_{i \in N} h_i \boldsymbol{l_{i,t}} + \sum_{t \in T} \sum_{(i,j) \in A} c_{i,j} \quad \underset{\substack{\uparrow \\ \text{un camion} \\ \text{emprunte} \\ \text{la route} \\ \text{de } i \text{ à } j \\ \text{le jour t}}}{\boldsymbol{x_{i,j,t}}}$$

NB : D'autres variables apparaissent dans les contraintes

Interprétation de la fonction objectif

$$\min \sum_{t \in T} \overbrace{u\boldsymbol{p_t} + f\boldsymbol{y_t}}^{\text{coûts de production}} + \overbrace{\sum_{t \in T} \sum_{i \in N} h_i \boldsymbol{I_{i,t}}}^{\text{coûts de stockage}} + \overbrace{\sum_{t \in T} \sum_{(i,j) \in A} c_{i,j} \boldsymbol{x_{i,j,t}}}^{\text{coûts de distribution}}$$

La contrainte (11)

rappel

$w_{i,t}$	désigne le chargement du camion qui livre le client i le jour t
$q_{i,t}$	désigne la quantité livrée au client i le jour t
$x_{i,j,t}$	exprime si un camion va de i à j le jour t

$$(11)\ \forall t \in T, \forall i \in N_c, \forall j \in N,\ \ \boldsymbol{w_{i,t}} - \boldsymbol{w_{j,t}} \geqslant \boldsymbol{q_{i,t}} - Q(1 - \boldsymbol{x_{i,j,t}})$$

Cette inégalité permet d'assurer

- que l'on ne va pas faire des sous-tours
- que le chargement du camion décroît bien au fur et à mesure des livraisons.

La contrainte (12)

rappel

$w_{i,t}$ désigne le chargement du camion qui livre le client i le jour t

$z_{i,t}$ exprime si le client i est visité le jour t

$$(12)\ \forall t \in T, \forall i \in N_c,\ \ 0 \leqslant \boldsymbol{w_{i,t}} \leqslant Q\boldsymbol{z_{i,t}}$$

Cette inégalité permet d'assurer

- que l'on ne livre pas un client sans le visiter
- que l'on ne livre pas plus que ce qu'un camion peut contenir

1. Le problème concret

2. Modélisation par un PL mixte

3. Comment résoudre une formulation compacte ?
   Aparté sur le "Branch-and-Bound"
   Résultats avec Cplex

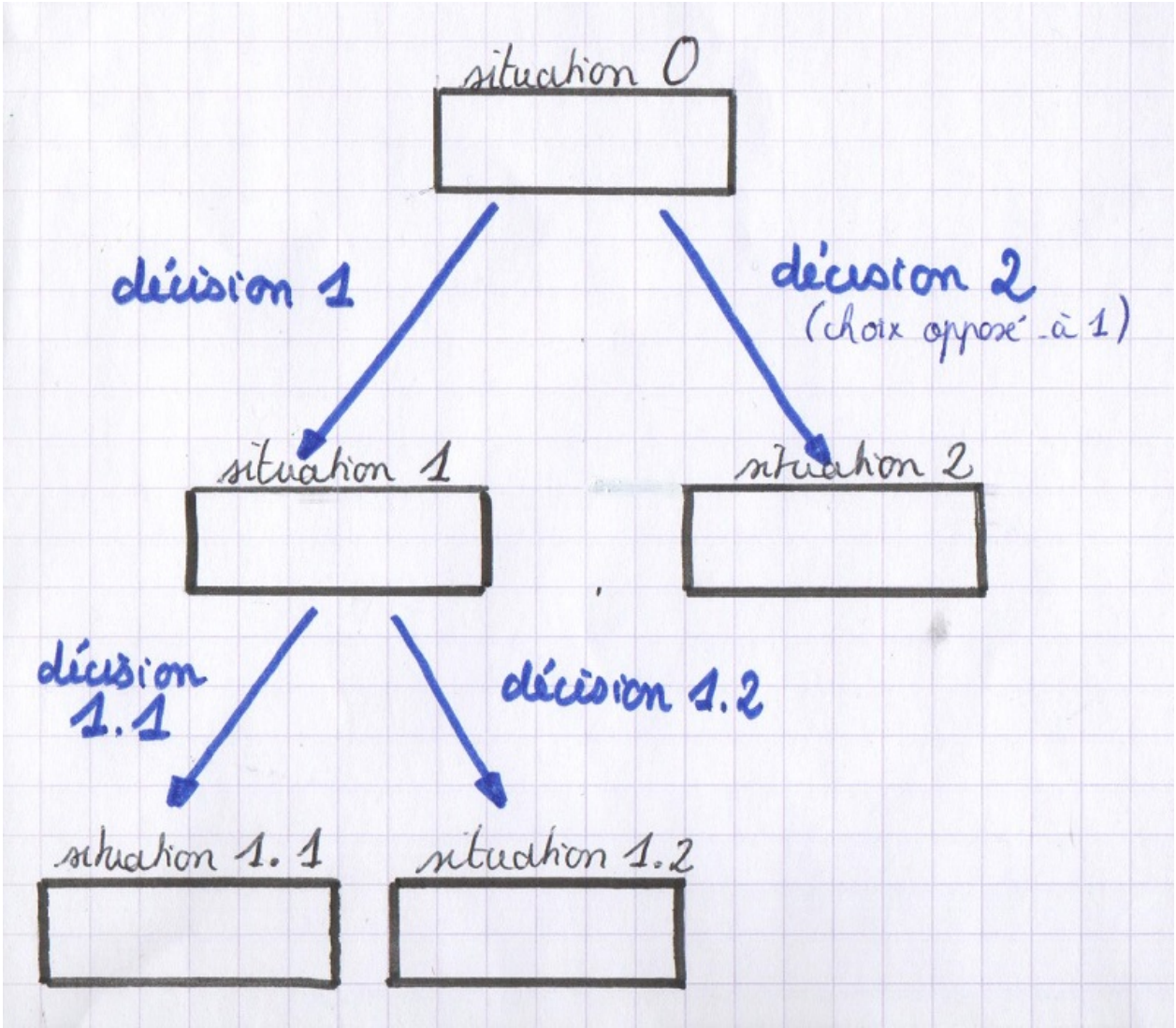
4. Une reformulation non compacte

5. Conclusion

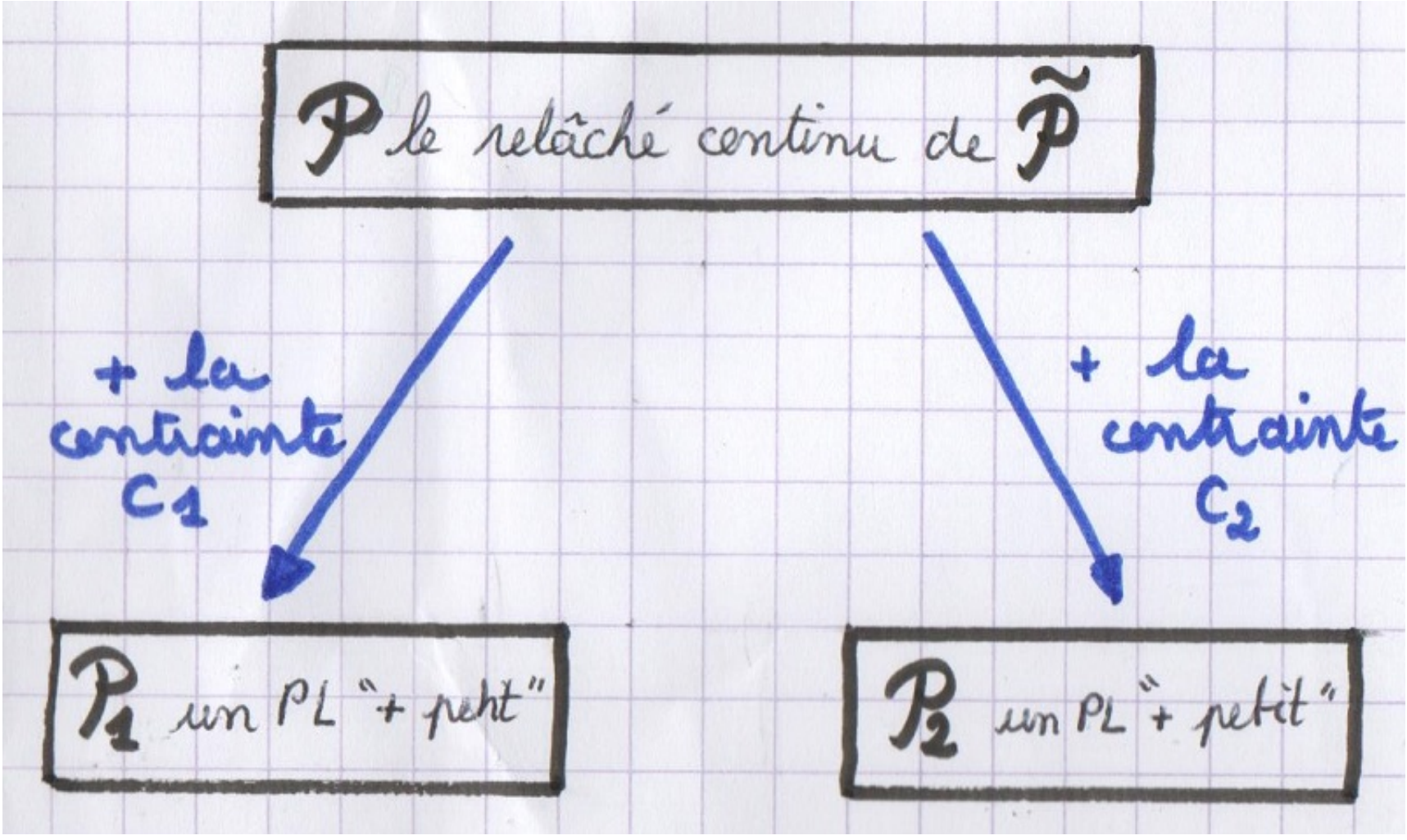
Qu'est ce que le "Branch-and-Bound"

Il s'agit de résoudre **un PLNE** $\tilde{\mathcal{P}}$ en résolvant **des PL** bien choisis, par disjonction de cas.

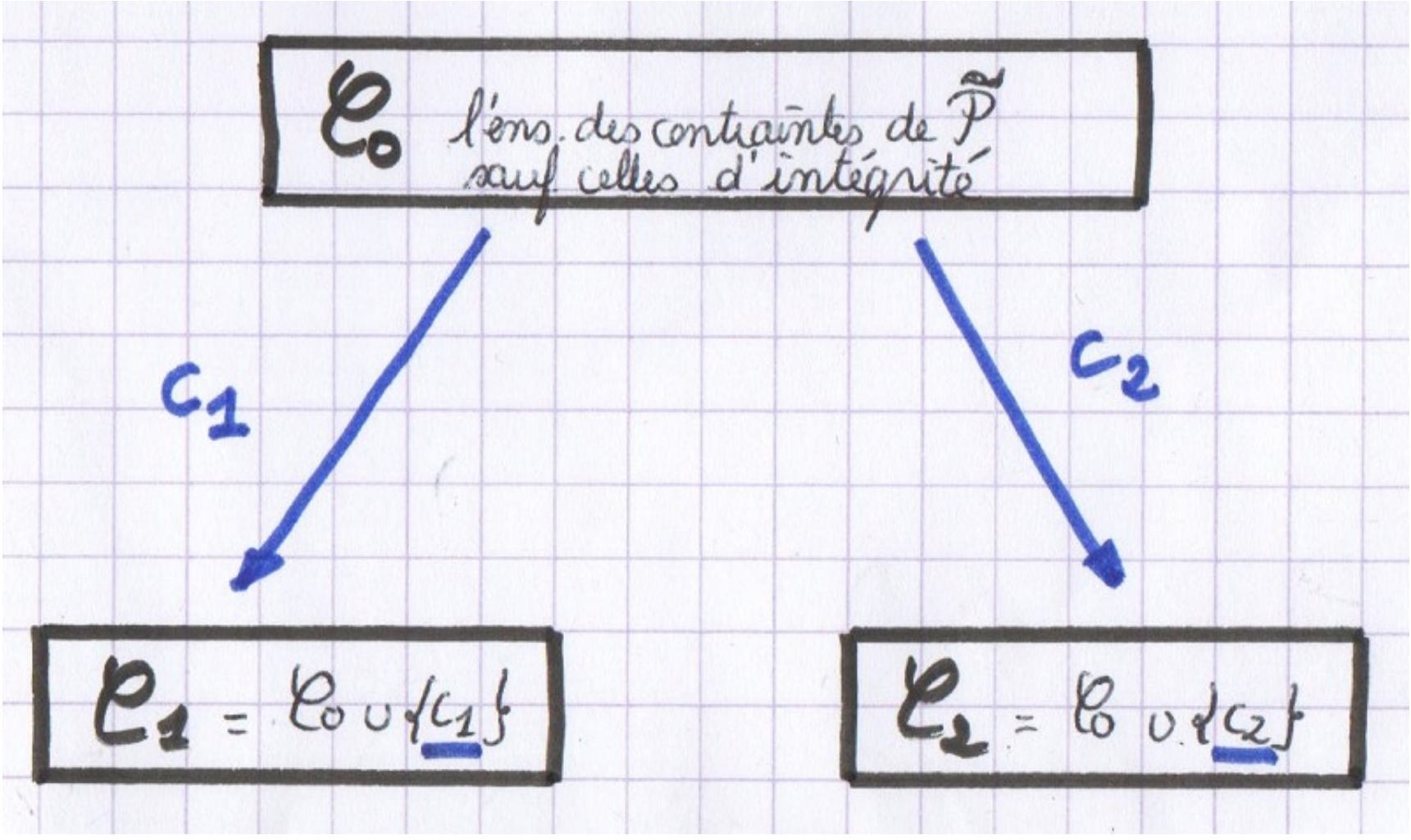
De loin : le côté "branchement"



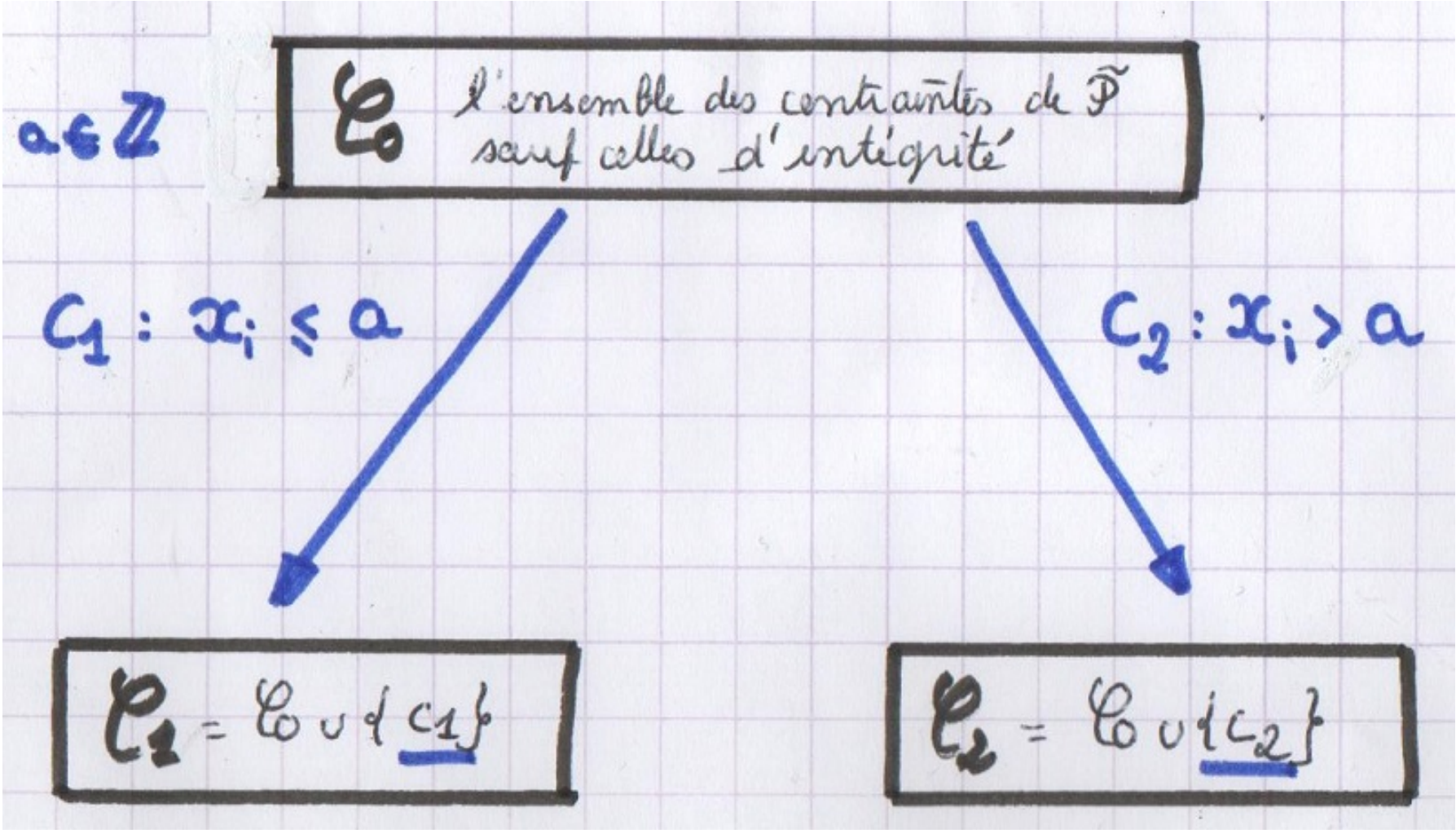
Disjonction des cas par ajout de contrainte



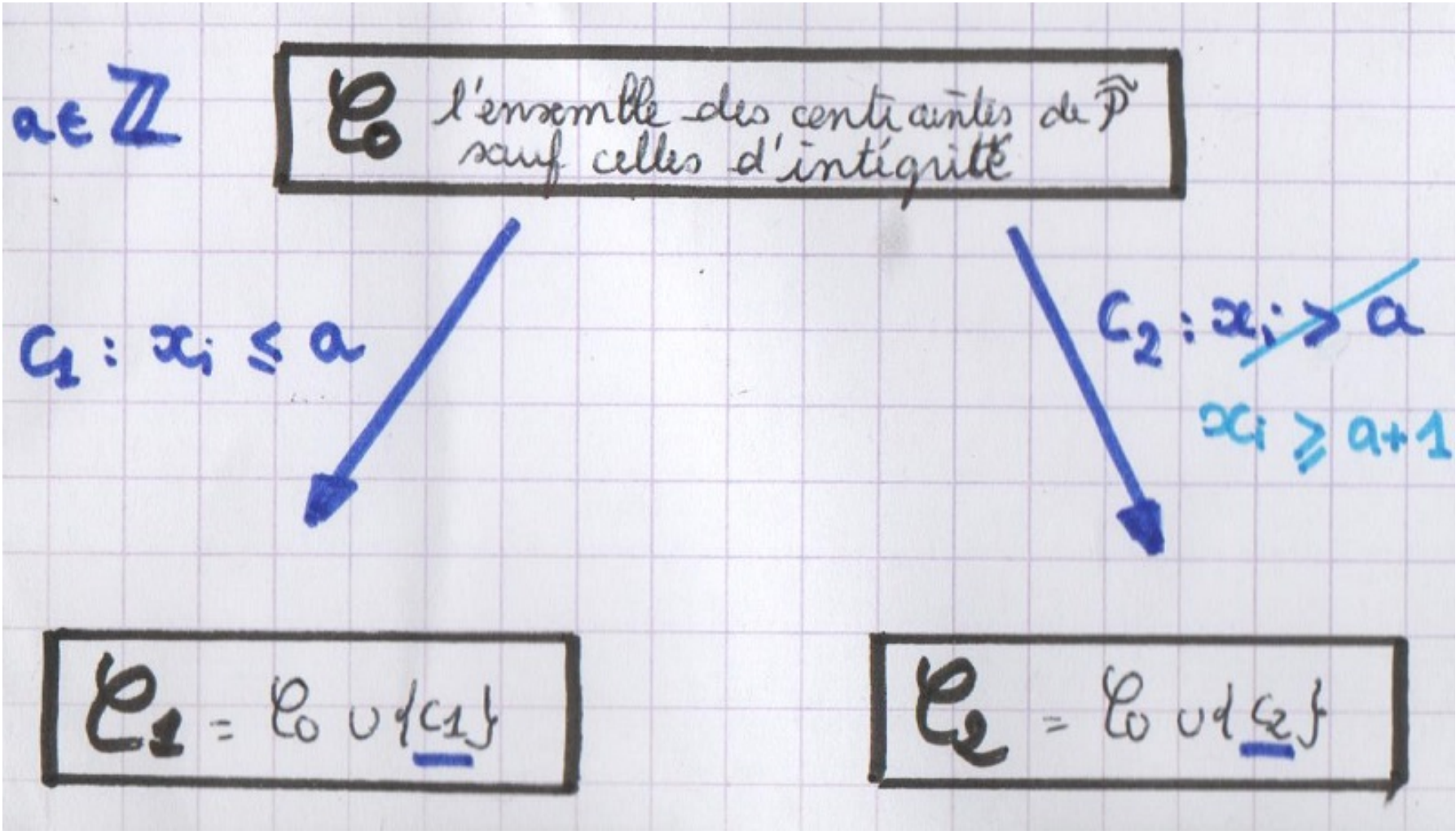
Disjonction des cas par ajout de contrainte



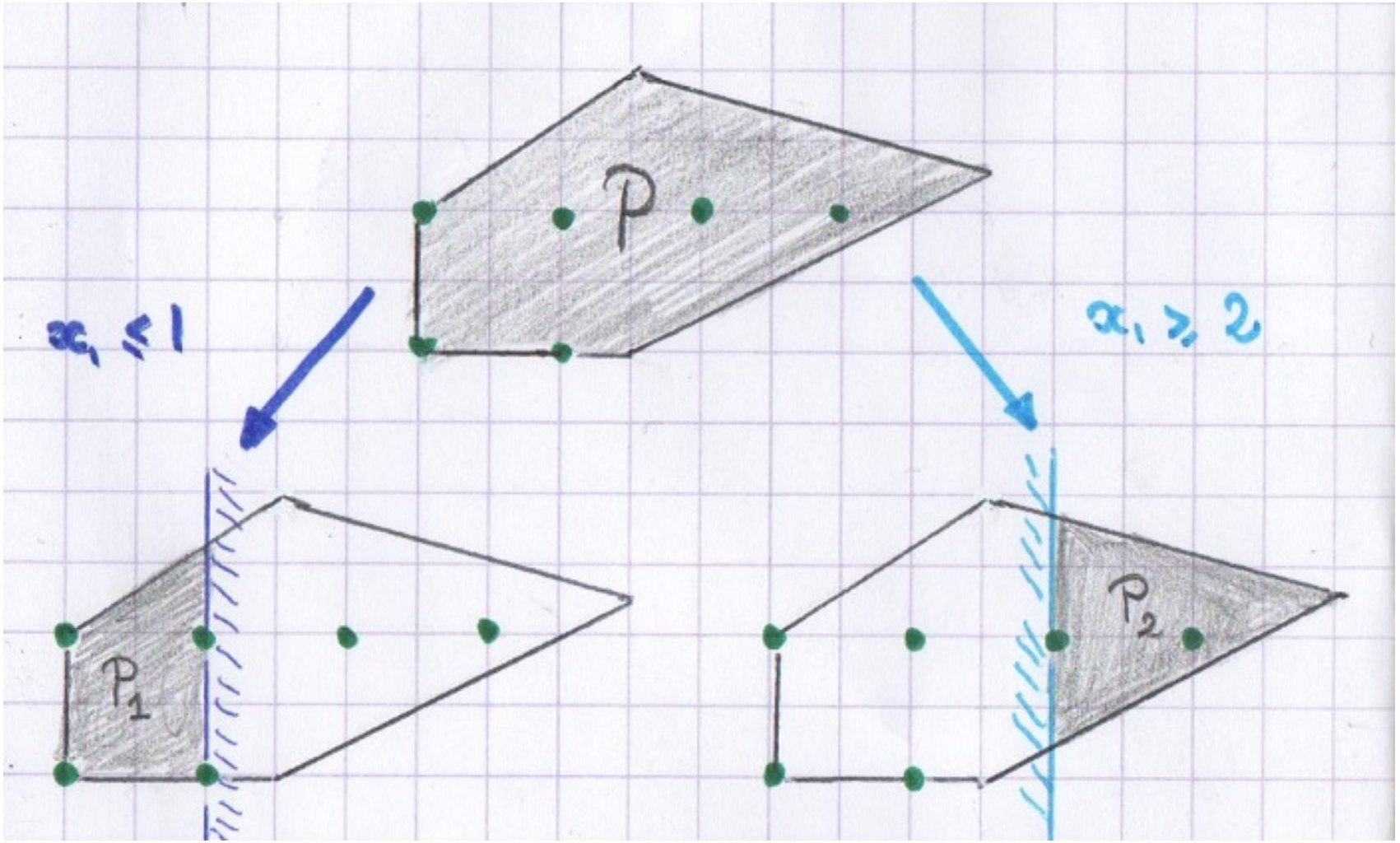
Disjonction des cas par ajout de contrainte



Disjonction des cas par ajout de contrainte



Disjonction des cas par ajout de contrainte



$$P \cap \mathbb{Z}^2 = (P_1 \cap \mathbb{Z}^2) \sqcup (P_2 \cap \mathbb{Z}^2)$$

Quel schéma de branchement ?

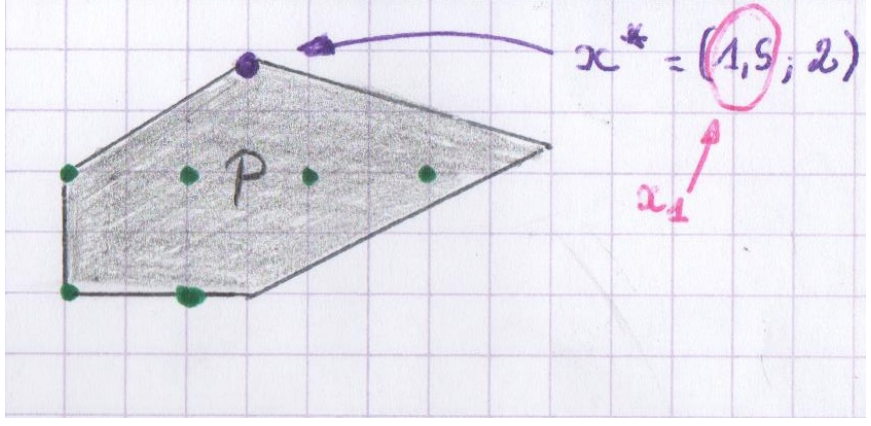
- Sur quelle variable brancher ?

- Sur quelle valeur brancher ?

Quel schéma de branchement ?

- Sur quelle variable brancher ?
$\longrightarrow$ sur la plus fractionnaire dans l'optimum

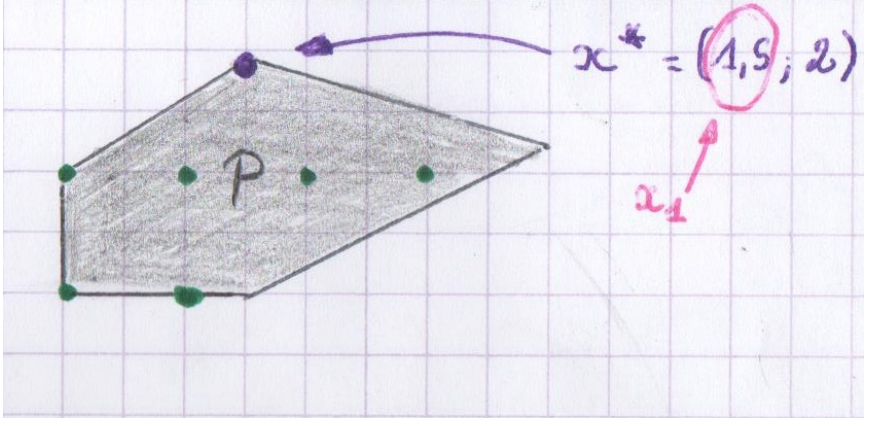
- Sur quelle valeur brancher ?



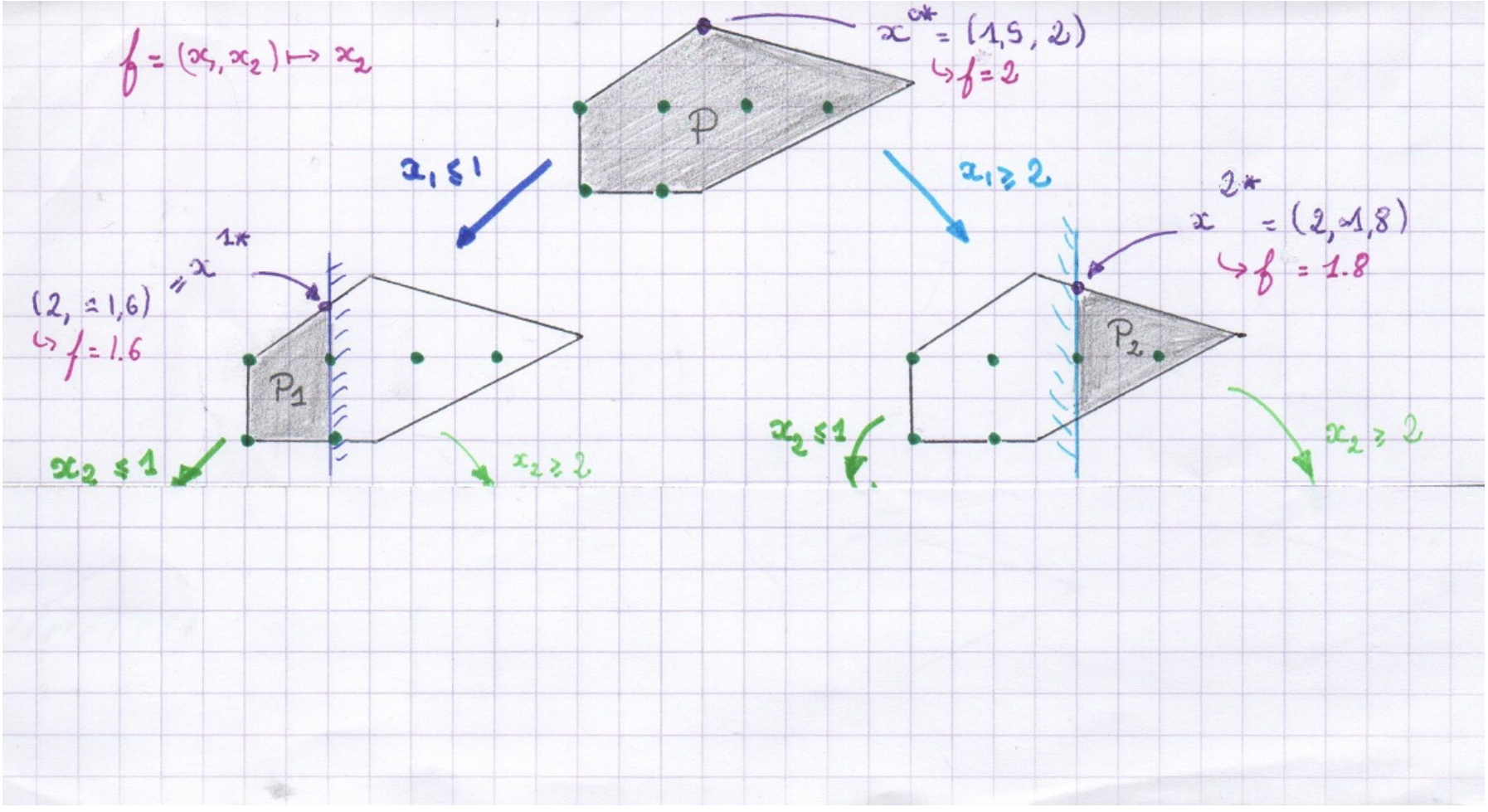
Quel schéma de branchement ?

- Sur quelle variable brancher ?
$\longrightarrow$ sur la plus fractionnaire dans l'optimum

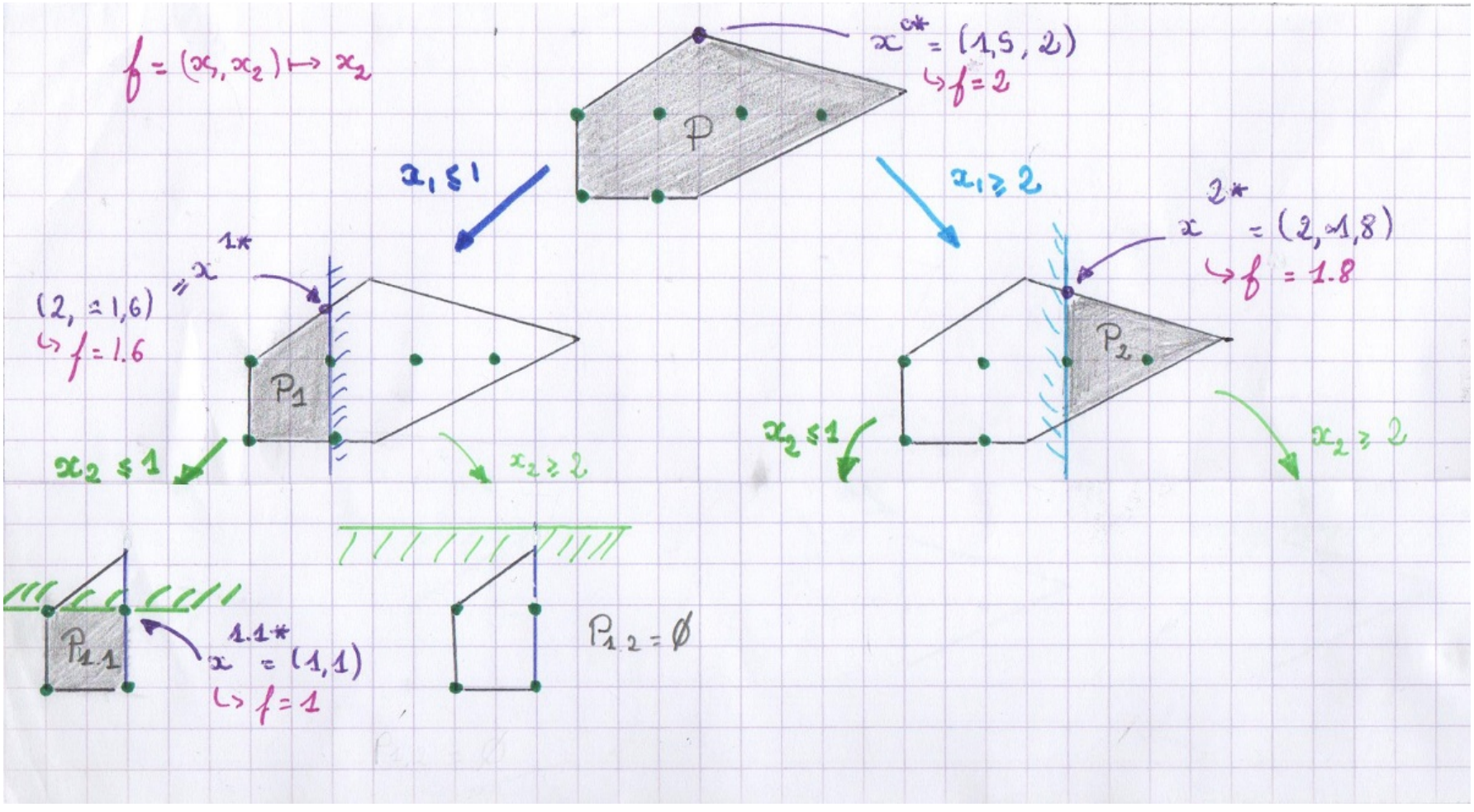
- Sur quelle valeur brancher ?
$\longrightarrow$ sur la partie entière de sa valeur dans l'optimum



Exemple complet d'un "Branch-and-Bound"



Exemple complet d'un "Branch-and-Bound"



Exemple complet d'un "Branch-and-Bound"

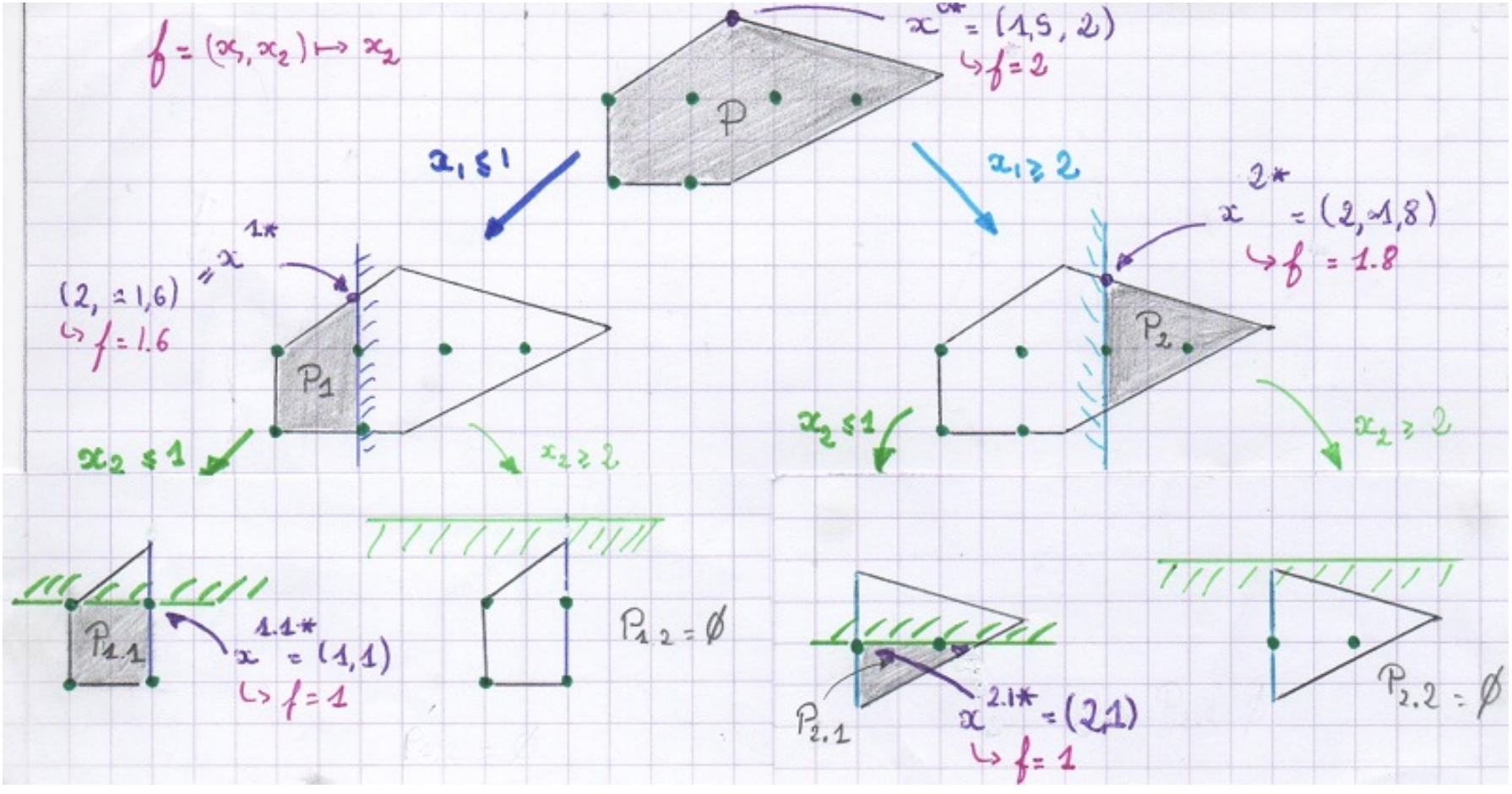


Schéma global de "Branch-and-Bound" en maximisation

La valeur de l'optimum en chaque nœud donne une borne sup pour le cas considéré.

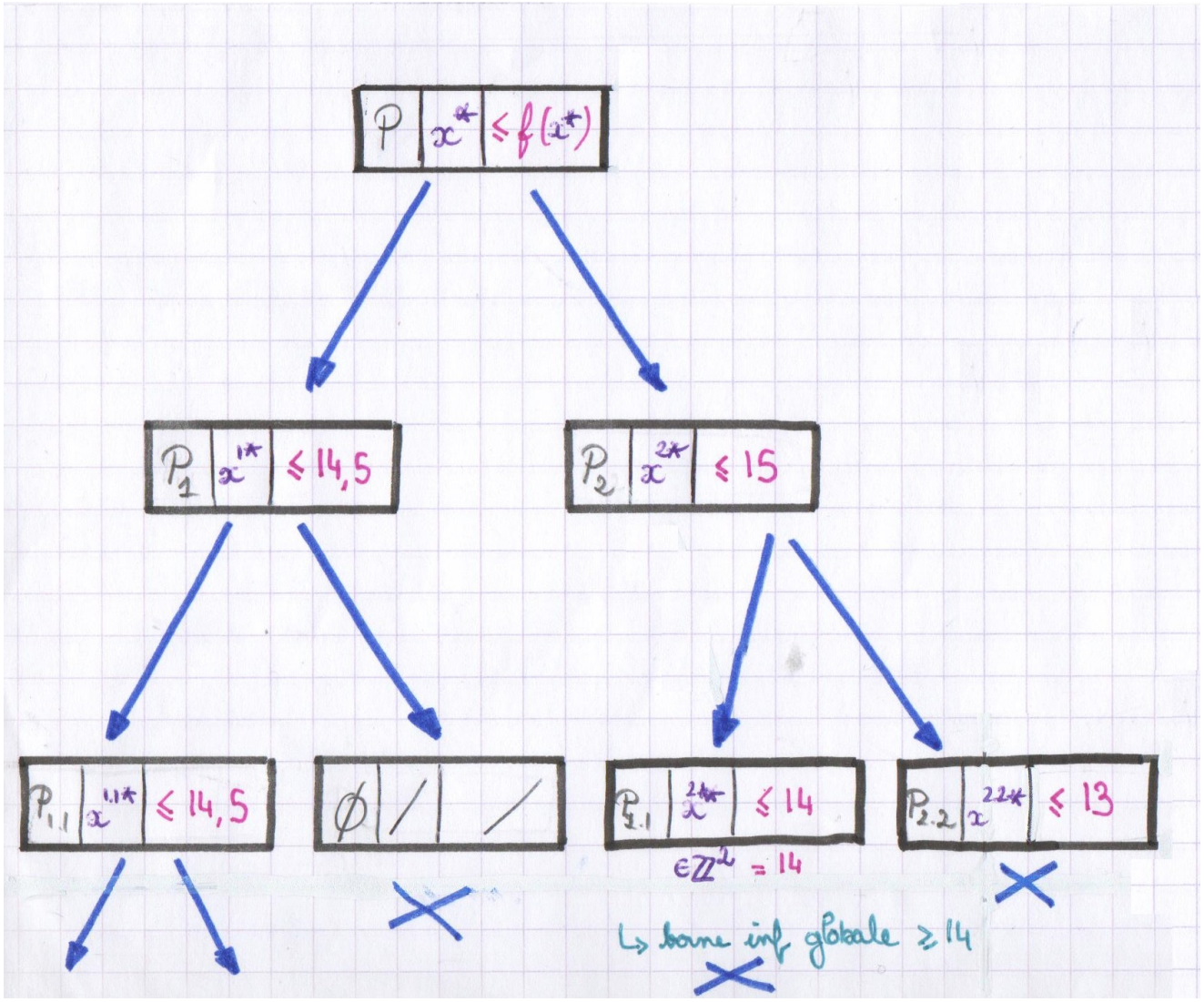


Schéma global de "Branch-and-Bound" en maximisation

La valeur de l'optimum en chaque nœud donne une borne sup pour le cas considéré.

Cette borne sup reste valide dans tout le sous arbre issu du nœud.

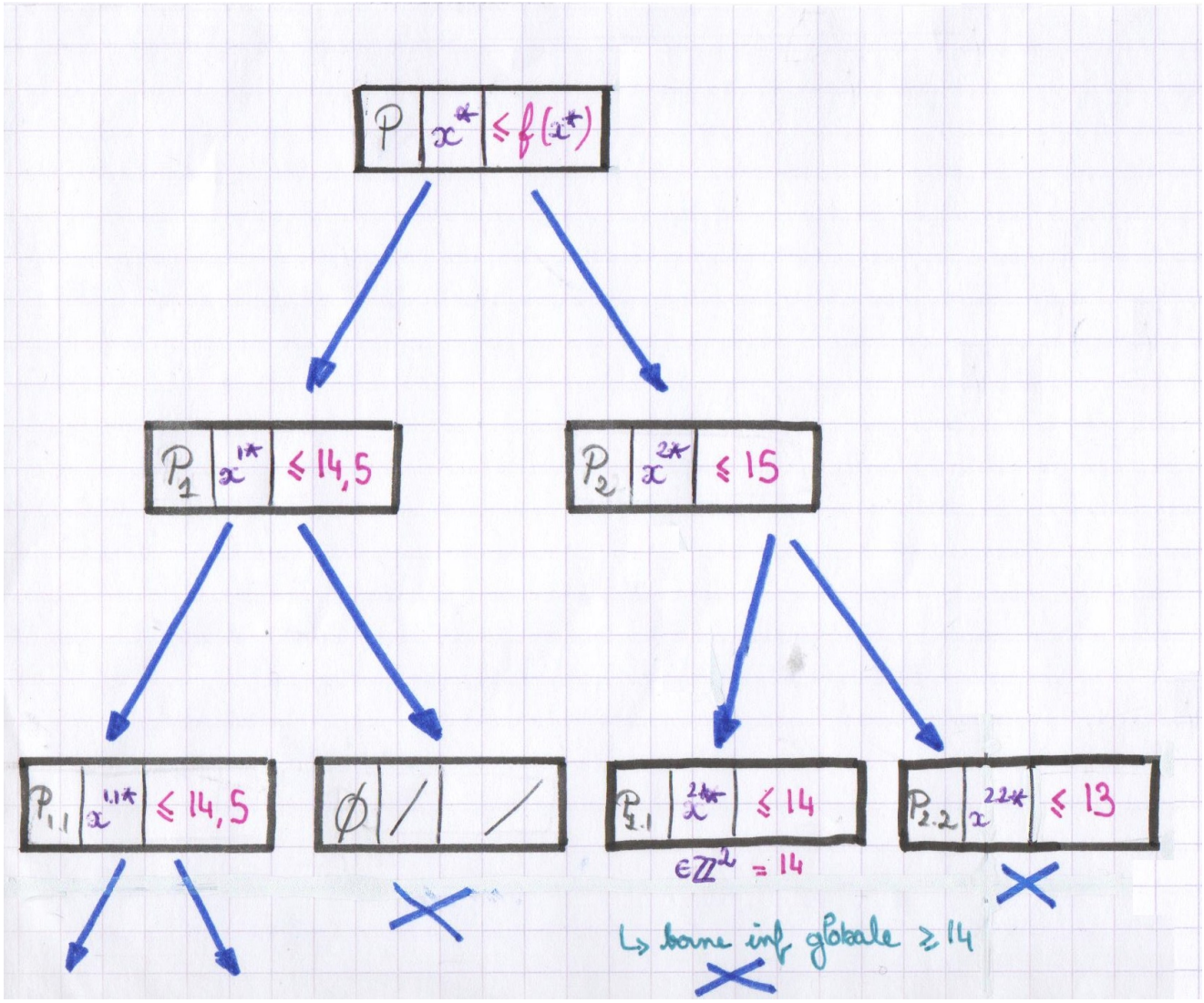


Schéma global de "Branch-and-Bound" en maximisation

La valeur de l'optimum en chaque nœud donne une borne sup pour le cas considéré.

Cette borne sup reste valide dans tout le sous arbre issu du nœud.

Pour avoir une borne sup globale il faut faire le max sur toute une largeur.

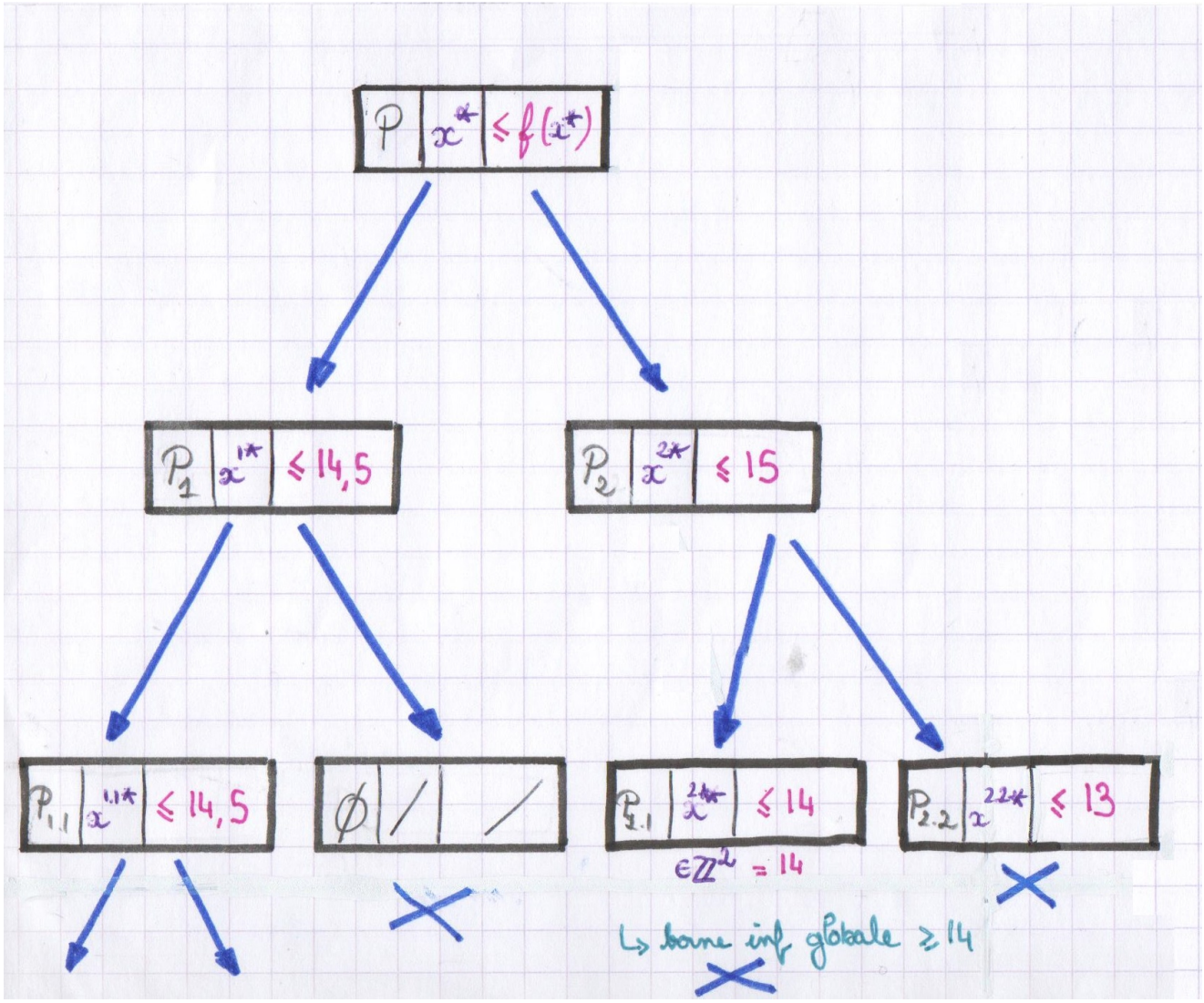


Schéma global de "Branch-and-Bound" en maximisation

Un nœud est stérile soit parce que le PL est vide, soit parce que l'optimum est entier.

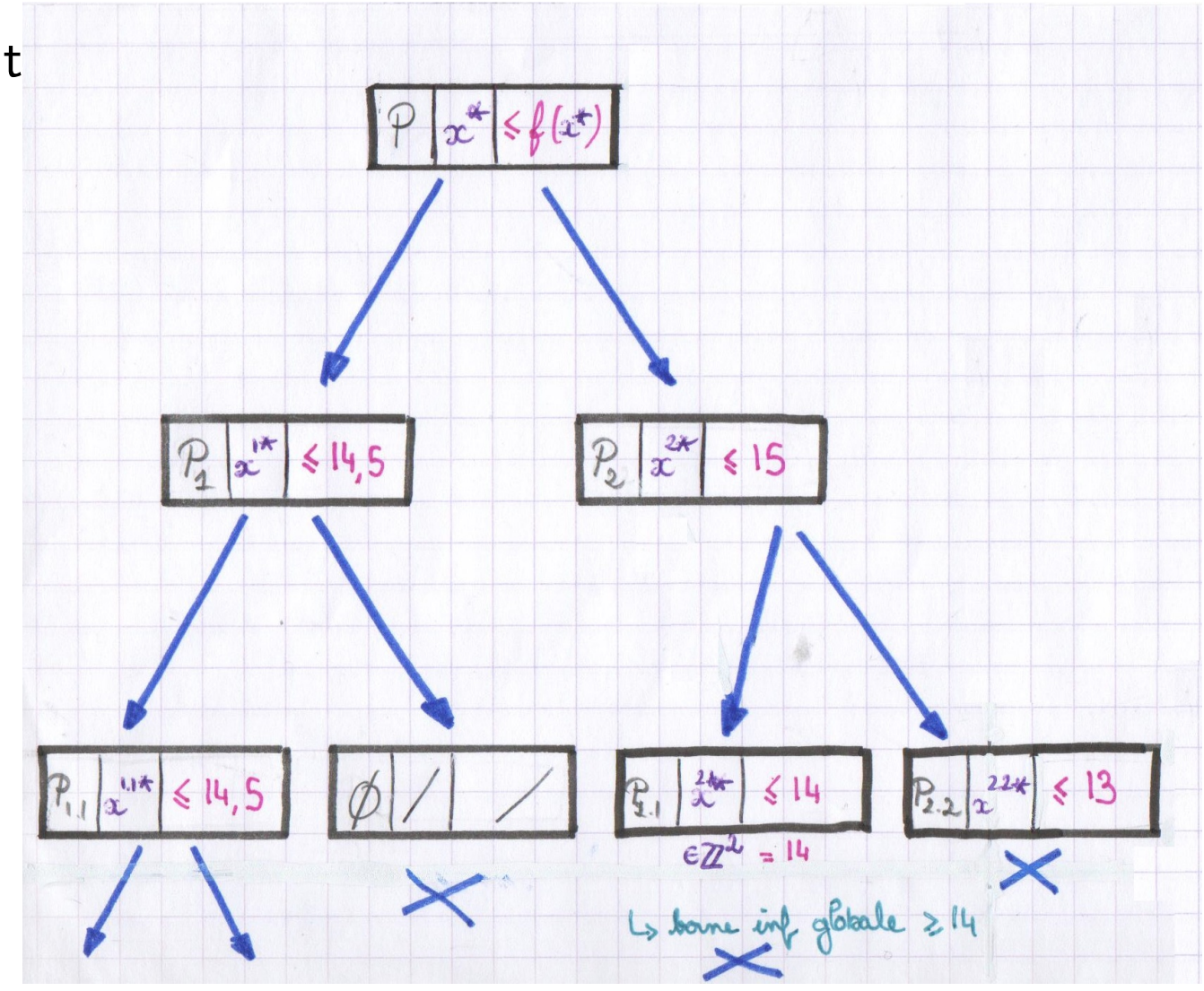


Schéma global de "Branch-and-Bound" en maximisation

Un nœud est stérile soit parce que le PL est vide, soit parce que l'optimum est entier.

La valeur d'un optimum **entier** donne une borne inf globale.

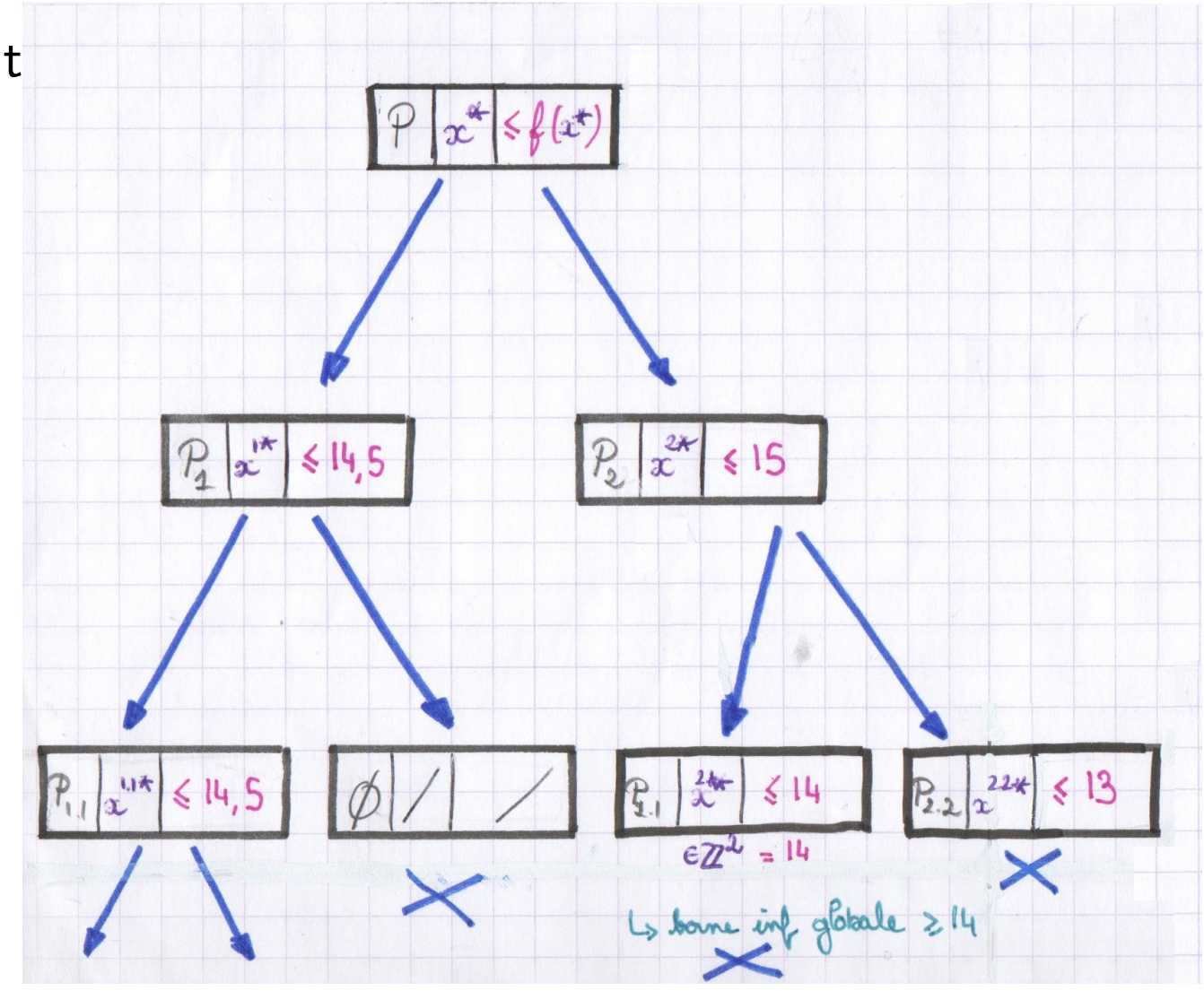


Schéma global de "Branch-and-Bound" en maximisation

Un nœud est stérile soit parce que le PL est vide, soit parce que l'optimum est entier.

La valeur d'un optimum **entier** donne une borne inf globale.

On peut couper une branche si sa borne sup est moindre que la borne inf.

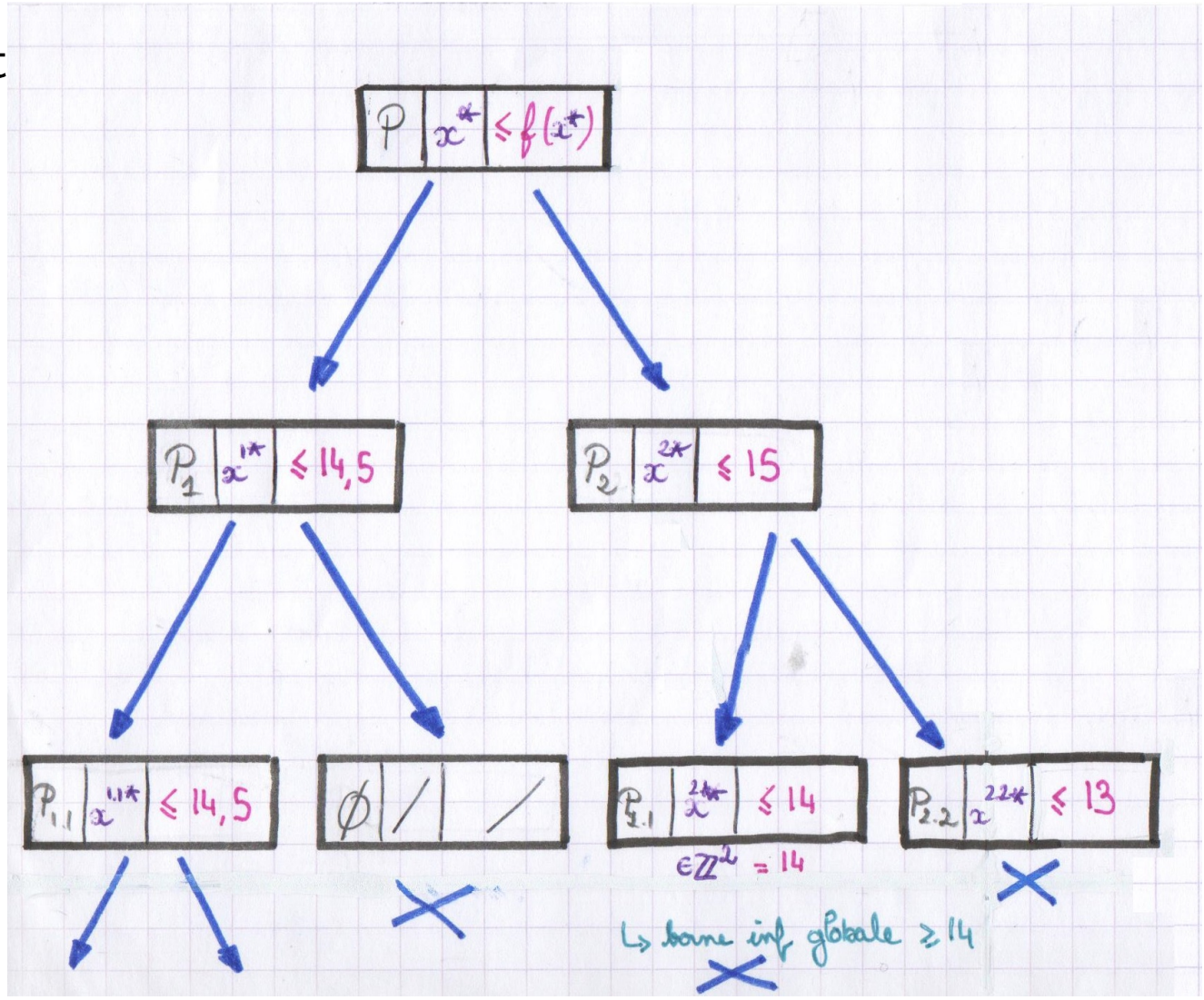
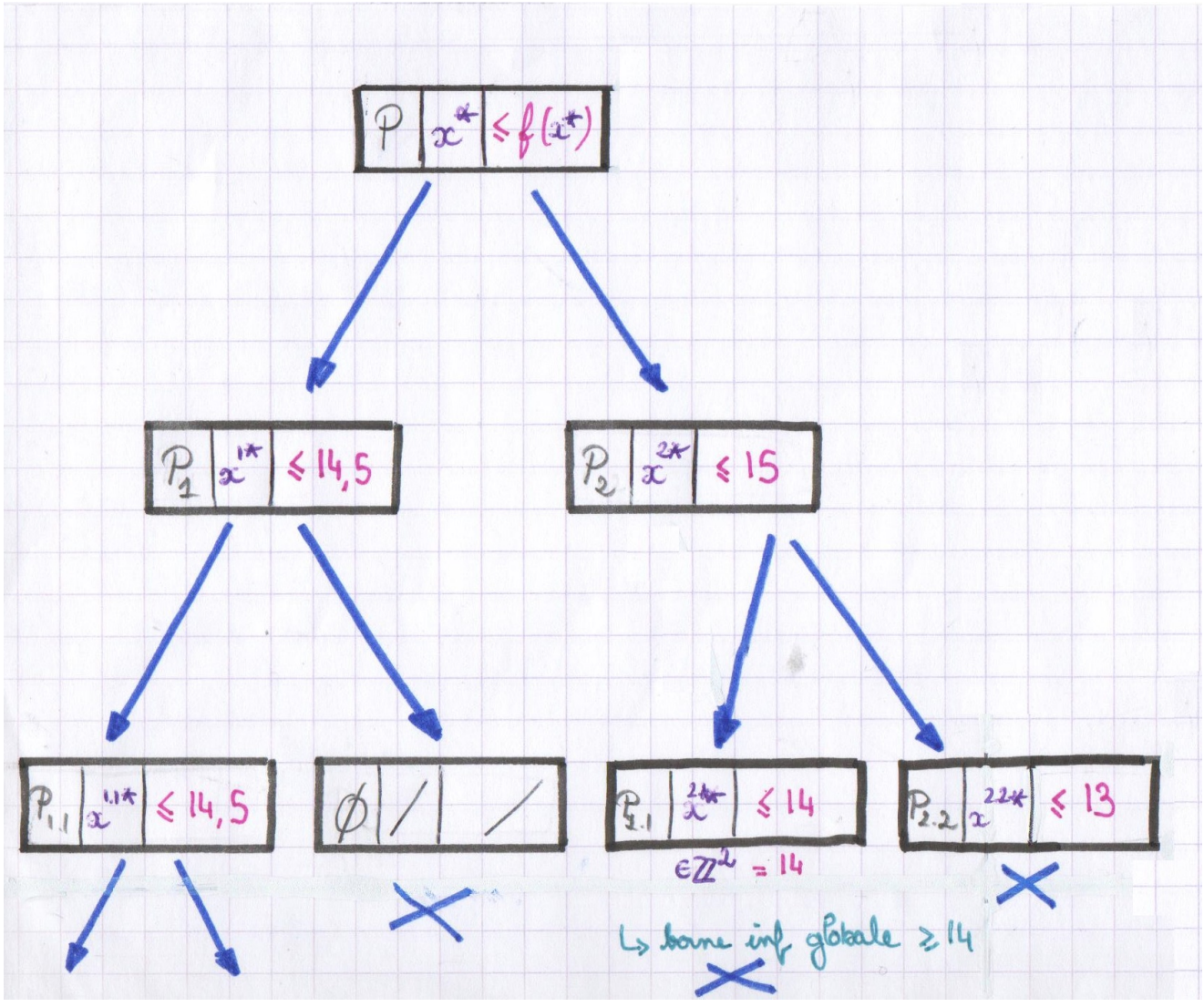


Schéma global de "Branch-and-Bound" en maximisation

Résultats avec garantie de performance : on connaît le "gap".



Résultats avec Cplex

1. Récupérer une instance (un fichier .txt)
2. Rentrer le PL sous Cplex (via C++ et Concert Technology)
3. Lancer la résolution (par Cplex)
4. L'interrompre après 5min car le gap ne réduit plus

Instances A			
Nom de l'instance	Statut	Valeur de l'objectif	Gap(%)
A_014_ABS96_15_1	Réalisable	605 719	0.39
A_014_ABS96_15_2	Réalisable	506 096	0.21
A_014_ABS1_15_2	Réalisable	40 390	1.24
A_050_ABS96_50_1	Réalisable	2 351 820	23
A_050_ABS1_50_1	Réalisable	127 987	12.72
A_050_ABS1_50_3	Réalisable	115 878	12.50

1. Le problème concret

2. Modélisation par un PL mixte

3. Comment résoudre une formulation compacte ?

4. Une reformulation non compacte
   - Formulation non compacte avec les contraintes (17)
   - Apparté sur le "Branch-and-Cut"
   - Un algorithme de séparation astucieux

5. Conclusion

Une nouvelle formulation

On remplace les contraintes

$$\begin{cases} \forall t \in T, \forall i \in N_c, \forall j \in N, \;\; w_{i,t} - w_{j,t} \geqslant q_{i,t} - \tilde{M}_{j,t}(1 - x_{i,j,t}) & (11) \\ \forall t \in T, \forall i \in N_c, \;\; 0 \leqslant w_{i,t} \leqslant Q z_{i,t} & (12) \end{cases}$$

par les contraintes

$$\forall t, \forall S \subset N_c, S \neq \emptyset, \;\; \sum_{i \notin S} \sum_{j \in S} x_{i,j,t} \geqslant \sum_{j \in S} \frac{q_{j,t}}{Q} \quad (17)$$

Validité des contraintes (17)

$$\forall t, \forall S \subset N_c, S \neq \emptyset, \quad \sum_{i \notin S} \sum_{j \in S} x_{i,j,t} \geqslant \sum_{j \in S} \frac{q_{j,t}}{Q} \quad (17)$$

$$\Leftrightarrow \forall t, \forall S \subset N_c, S \neq \emptyset, \quad \sum_{i \notin S} \sum_{j \in S} x_{i,j,t} \geqslant \frac{\sum_{j \in S} q_{j,t}}{Q} \quad (17)$$

$$\Leftrightarrow \forall t, \forall S \subset N_c, S \neq \emptyset, \quad \begin{array}{c}\text{le nombre de fois} \\ \text{qu'un camion entre} \\ \text{dans la zone } S\end{array} \geqslant \frac{\begin{array}{c}\text{quantité livrée} \\ \text{dans la zone } S\end{array}}{Q} \quad (17)$$

Validité des contraintes (17)

$$\Leftrightarrow \forall t, \forall S \subset N_c, S \neq \emptyset, \quad \begin{array}{c}\text{le nombre de fois}\\ \text{qu'un camion entre}\\ \text{dans la zone } S\end{array} \geqslant \frac{\begin{array}{c}\text{quantité livrée}\\ \text{dans la zone } S\end{array}}{Q} \quad (17)$$

Or on sait que pour des valeurs réelles :

$$Q \begin{array}{c}\text{le nombre de fois}\\ \text{qu'un camion entre}\\ \text{dans la zone } S\end{array} \geqslant Q \begin{array}{c}\text{le nombre de}\\ \text{camions qui entrent}\\ \text{dans la zone } S\end{array} \geqslant \begin{array}{c}\text{quantité livrée}\\ \text{dans la zone } S\end{array}$$

Validité des contraintes (17)

$$\Leftrightarrow \forall t, \forall S \subset N_c, S \neq \emptyset, \quad \begin{array}{c}\text{le nombre de fois} \\ \text{qu'un camion entre} \\ \text{dans la zone } S\end{array} \geqslant \frac{\begin{array}{c}\text{quantité livrée} \\ \text{dans la zone } S\end{array}}{Q} \quad (17)$$

Or on sait que pour des valeurs réelles :

$$Q \begin{array}{c}\text{le nombre de fois} \\ \text{qu'un camion entre} \\ \text{dans la zone } S\end{array} \geqslant Q \begin{array}{c}\text{le nombre de} \\ \text{camions qui entrent} \\ \text{dans la zone } S\end{array} \geqslant \begin{array}{c}\text{quantité livrée} \\ \text{dans la zone } S\end{array}$$

$\longrightarrow$ Donc les contraintes (17) sont **valides**.

Validité des contraintes (17)

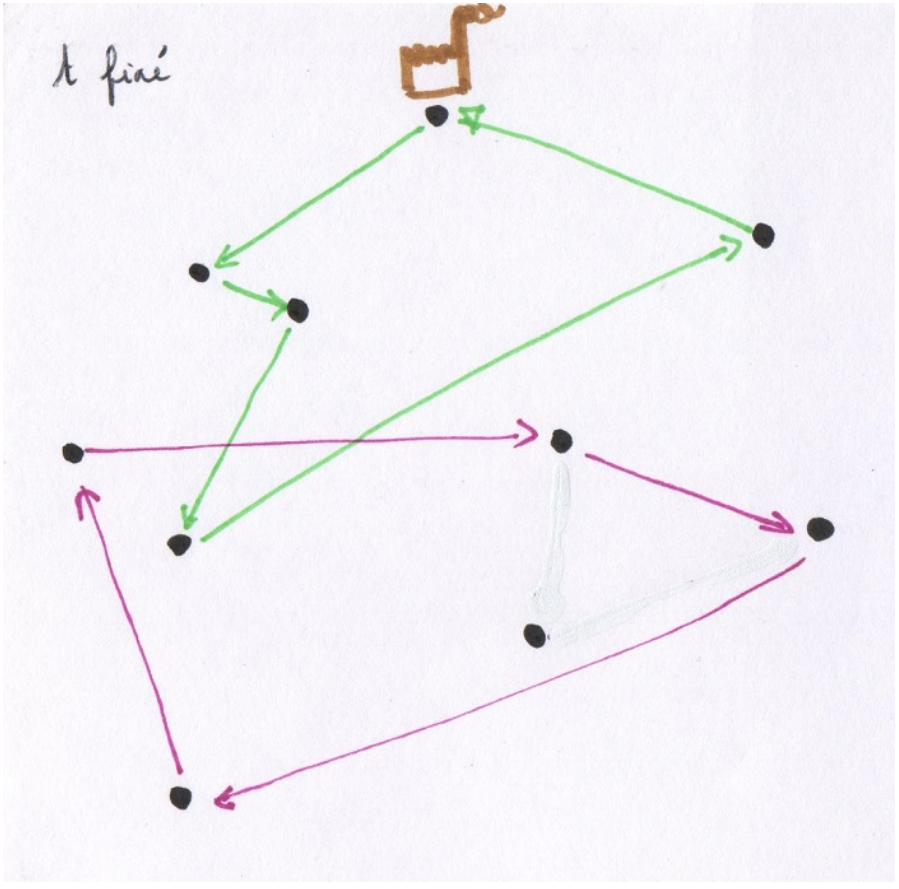
$$\Leftrightarrow \forall t, \forall S \subset N_c, S \neq \emptyset, \quad \begin{array}{c}\text{le nombre de fois} \\ \text{qu'un camion entre} \\ \text{dans la zone } S\end{array} \geqslant \frac{\begin{array}{c}\text{quantité livrée} \\ \text{dans la zone } S\end{array}}{Q} \quad (17)$$

Or on sait que pour des valeurs réelles :

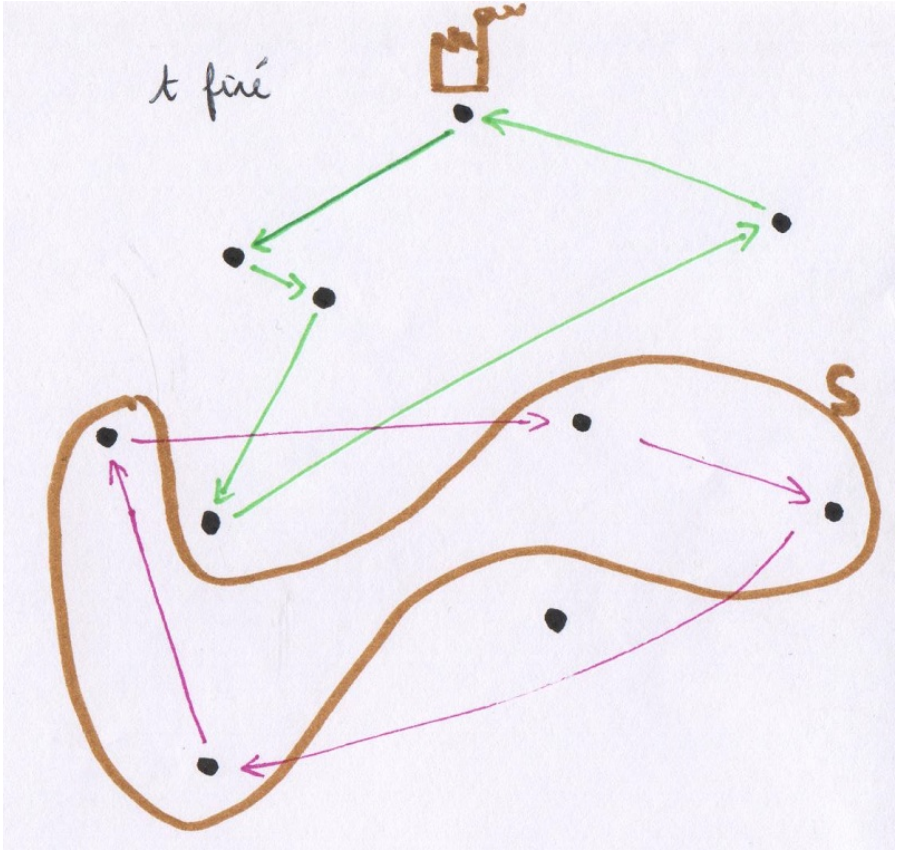
$$Q \begin{array}{c}\text{le nombre de fois} \\ \text{qu'un camion entre} \\ \text{dans la zone } S\end{array} \geqslant Q \begin{array}{c}\text{le nombre de} \\ \text{camions qui entrent} \\ \text{dans la zone } S\end{array} \geqslant \begin{array}{c}\text{quantité livrée} \\ \text{dans la zone } S\end{array}$$

$\longrightarrow$ Donc les contraintes (17) sont **valides**.
Suffisent-elles à remplacer les contraintes (11) et (12) ?

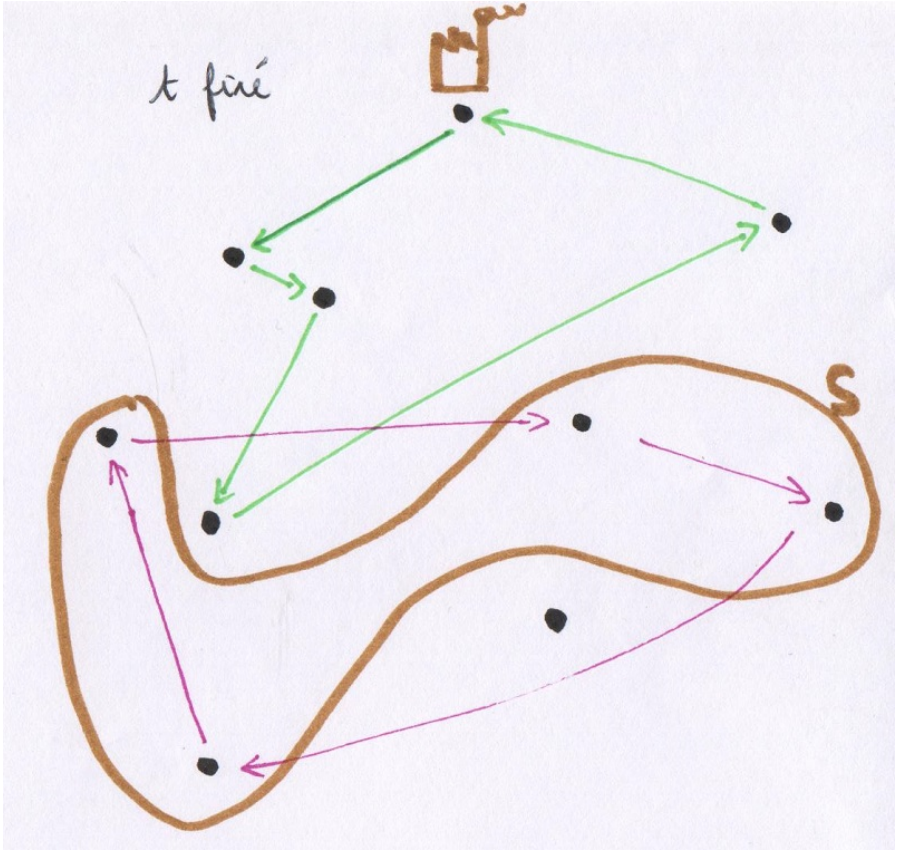
Les contraintes (17) éliminent-elles les sous-tours ?



Les contraintes (17) éliminent-elles les sous-tours ?

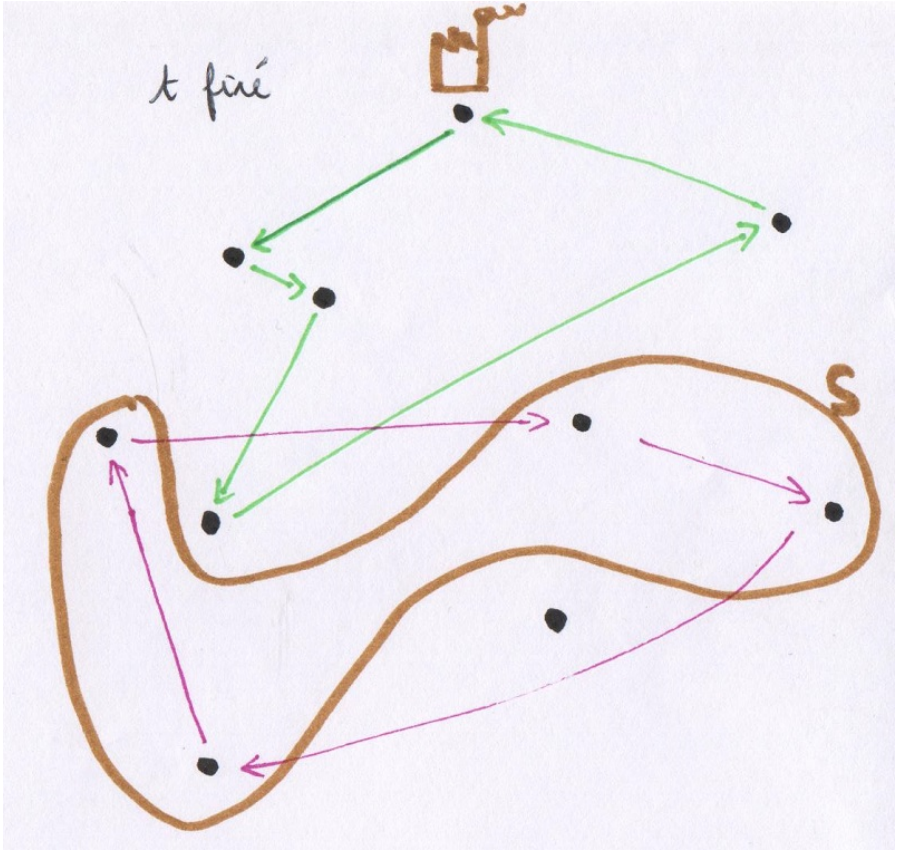


Les contraintes (17) éliminent-elles les sous-tours ?



$$\underbrace{\sum_{i \notin S} \sum_{j \in S} x_{i,j,t}}_{=0} \geqslant \sum_{j \in S} \frac{q_{j,t}}{Q} \quad (17_{s,t})$$

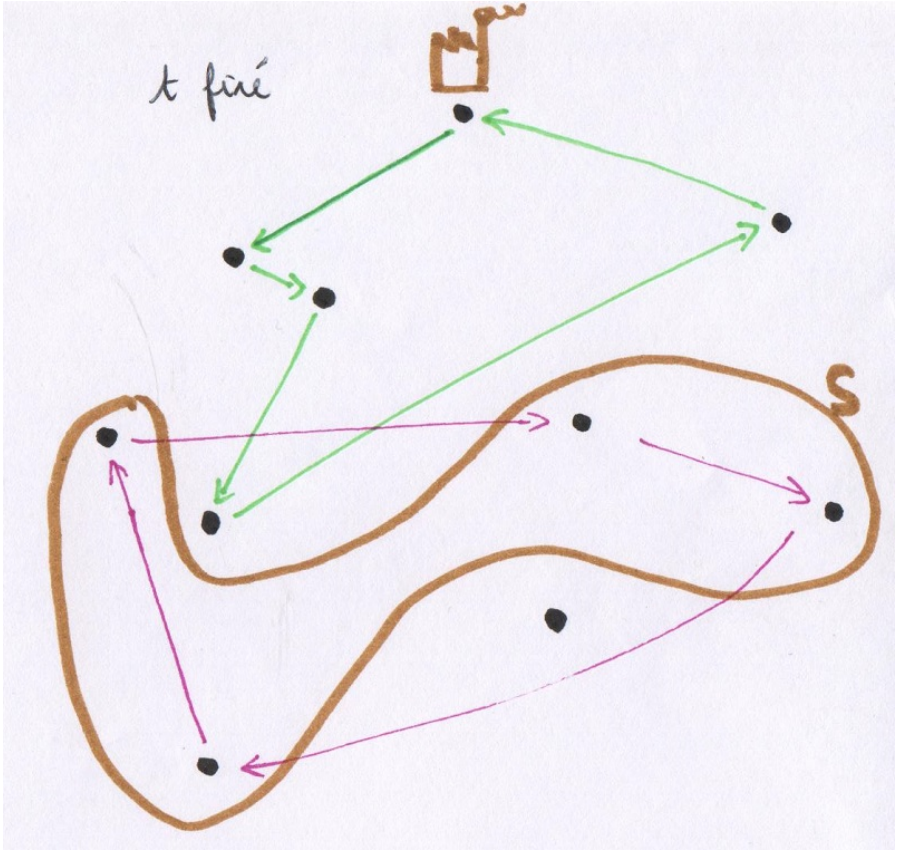
Les contraintes (17) éliminent-elles les sous-tours ?



$$\underbrace{\sum_{i \notin S} \sum_{j \in S} x_{i,j,t}}_{=0} \geqslant \sum_{j \in S} \frac{q_{j,t}}{Q} \quad (17_{s,t})$$

donc $\forall j \in S, \quad q_{j,t} = 0$

Les contraintes (17) éliminent-elles les sous-tours ?

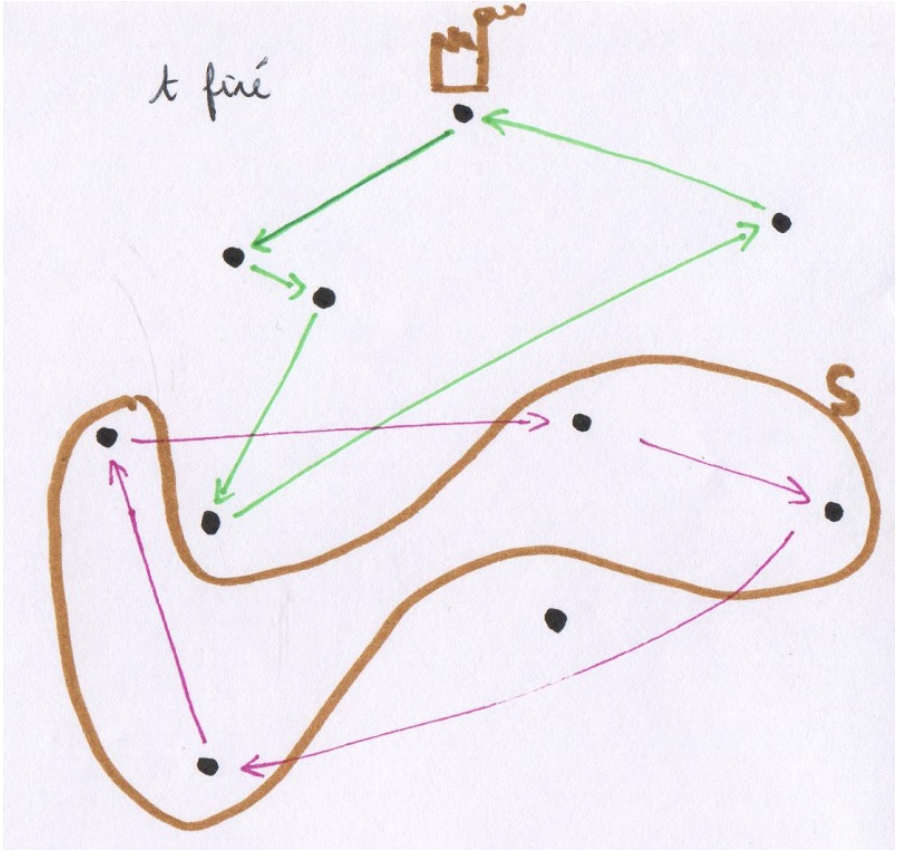


$$\underbrace{\sum_{i \notin S} \sum_{j \in S} x_{i,j,t}}_{=0} \geqslant \sum_{j \in S} \frac{q_{j,t}}{Q} \quad (17_{s,t})$$

donc $\forall j \in S, \quad q_{j,t} = 0$

donc $\forall j \in S, \quad z_{j,t} = 0$
Absurde

Les contraintes (17) éliminent-elles les sous-tours ?



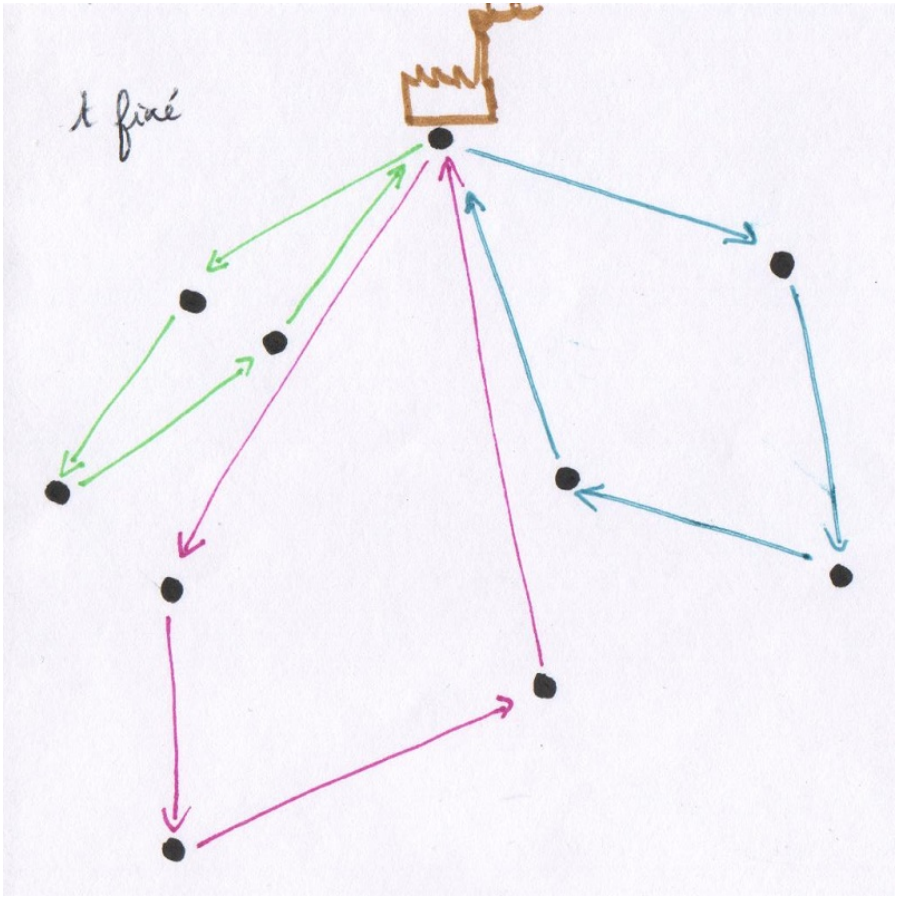
$$\underbrace{\sum_{i \notin S} \sum_{j \in S} x_{i,j,t}}_{=0} \geqslant \sum_{j \in S} \frac{q_{j,t}}{Q} \quad (17_{s,t})$$

donc $\forall j \in S, \quad q_{j,t} = 0$

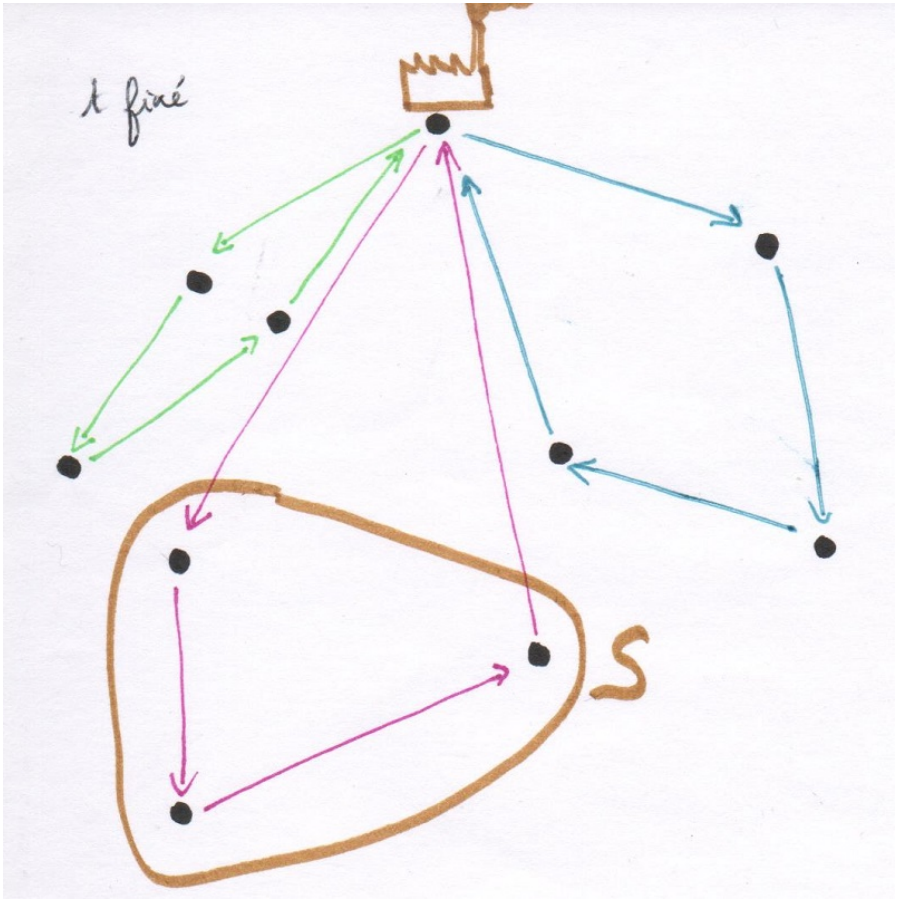
donc $\forall j \in S, \quad z_{j,t} = 0$
Absurde

$\longrightarrow$ Donc les contraintes (17) éliminent les sous-tours

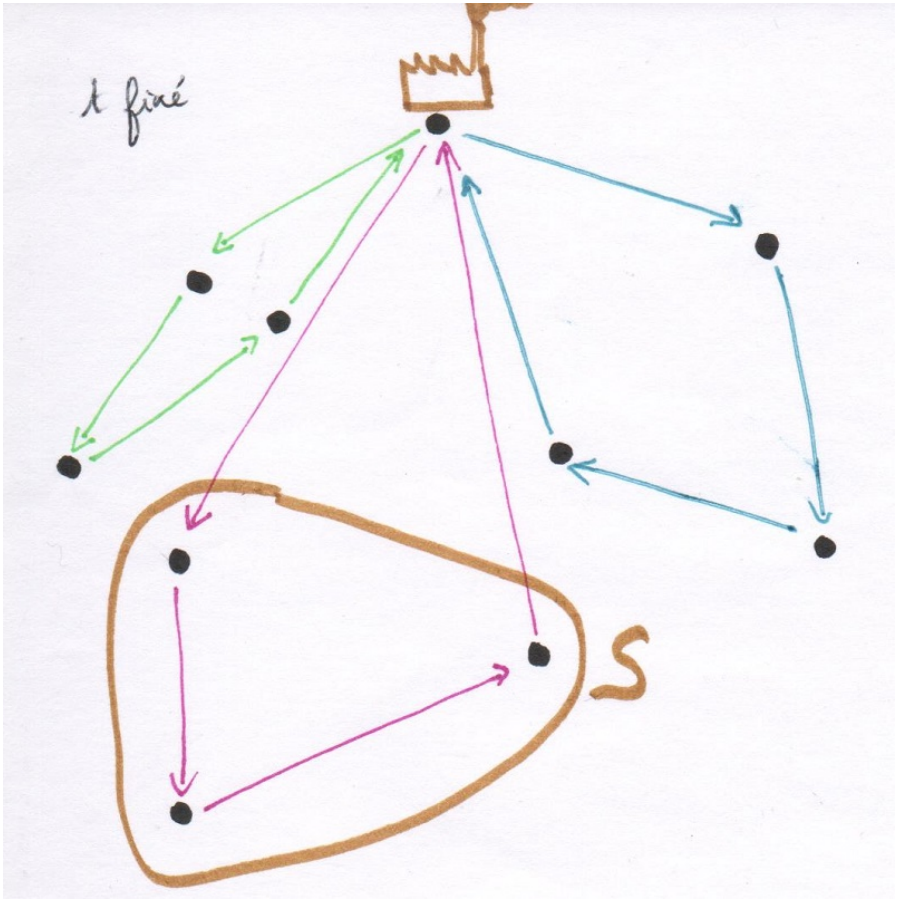
Assurent-elles que les chargements sont conformes ?



Assurent-elles que les chargements sont conformes ?

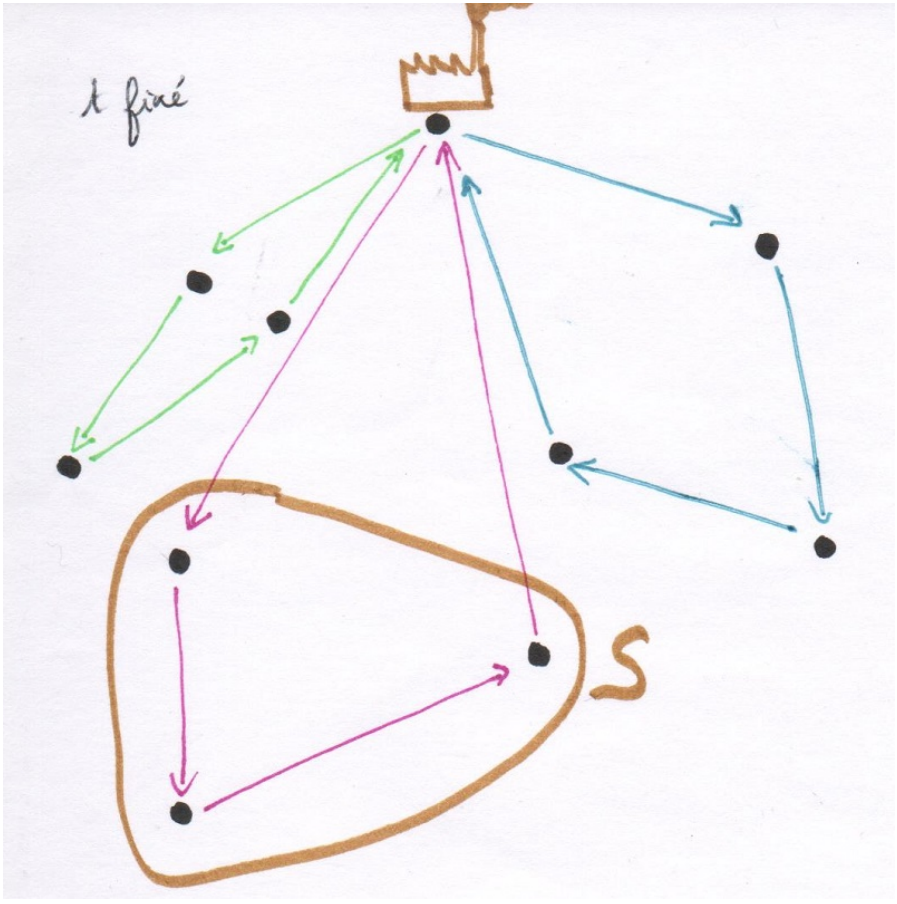


Assurent-elles que les chargements sont conformes ?



$$\underbrace{\sum_{i \notin S} \sum_{j \in S} x_{i,j,t}}_{=1} \geqslant \sum_{j \in S} \frac{q_{j,t}}{Q} \quad (17_{s,t})$$

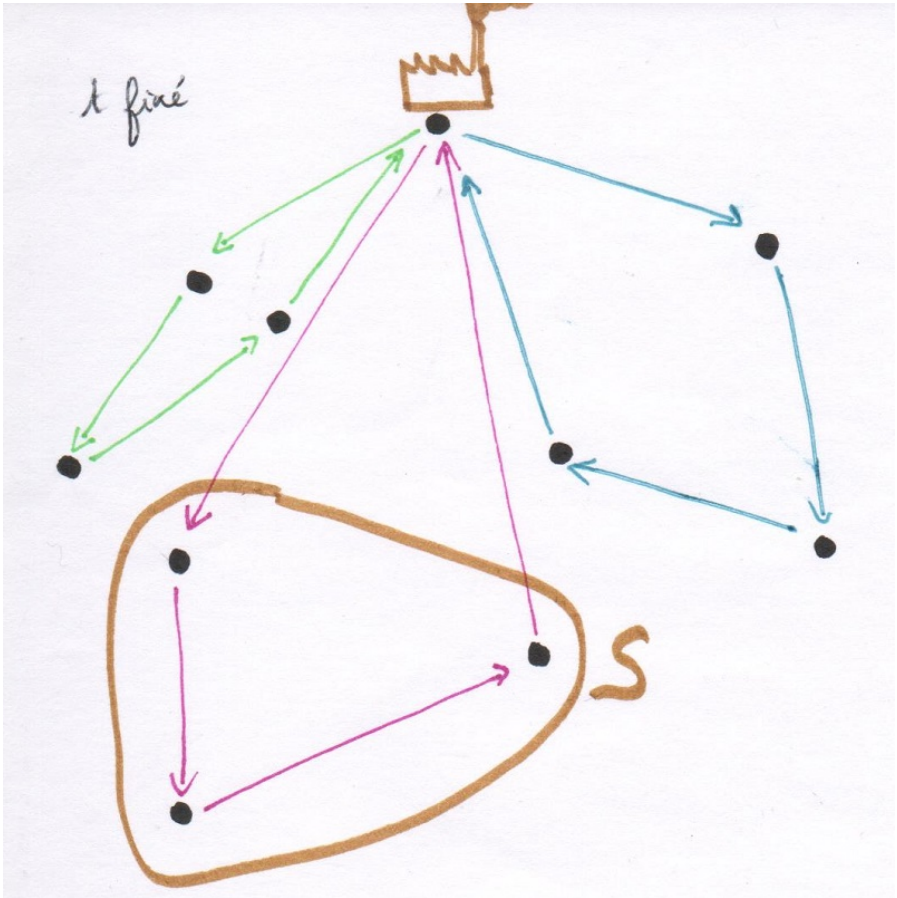
Assurent-elles que les chargements sont conformes ?



$$\underbrace{\sum_{i \notin S} \sum_{j \in S} x_{i,j,t}}_{=1} \geqslant \sum_{j \in S} \frac{q_{j,t}}{Q} \quad (17_{s,t})$$

donc $Q \geqslant \underbrace{\sum_{j \in S} q_{j,t}}_{\text{chargement global de la tournée}}$

Assurent-elles que les chargements sont conformes ?



$$\underbrace{\sum_{i \notin S} \sum_{j \in S} x_{i,j,t}}_{=1} \geqslant \sum_{j \in S} \frac{q_{j,t}}{Q} \quad (17_{s,t})$$

donc $Q \geqslant \underbrace{\sum_{j \in S} q_{j,t}}_{\text{chargement global de la tournée}}$

$\longrightarrow$ Donc les contraintes (17) assurent que les chargements ne dépassent pas la capacité d'un camion

Qu'est ce que ça change ?

$\rightarrow$ Nombre exponentiel de contraintes

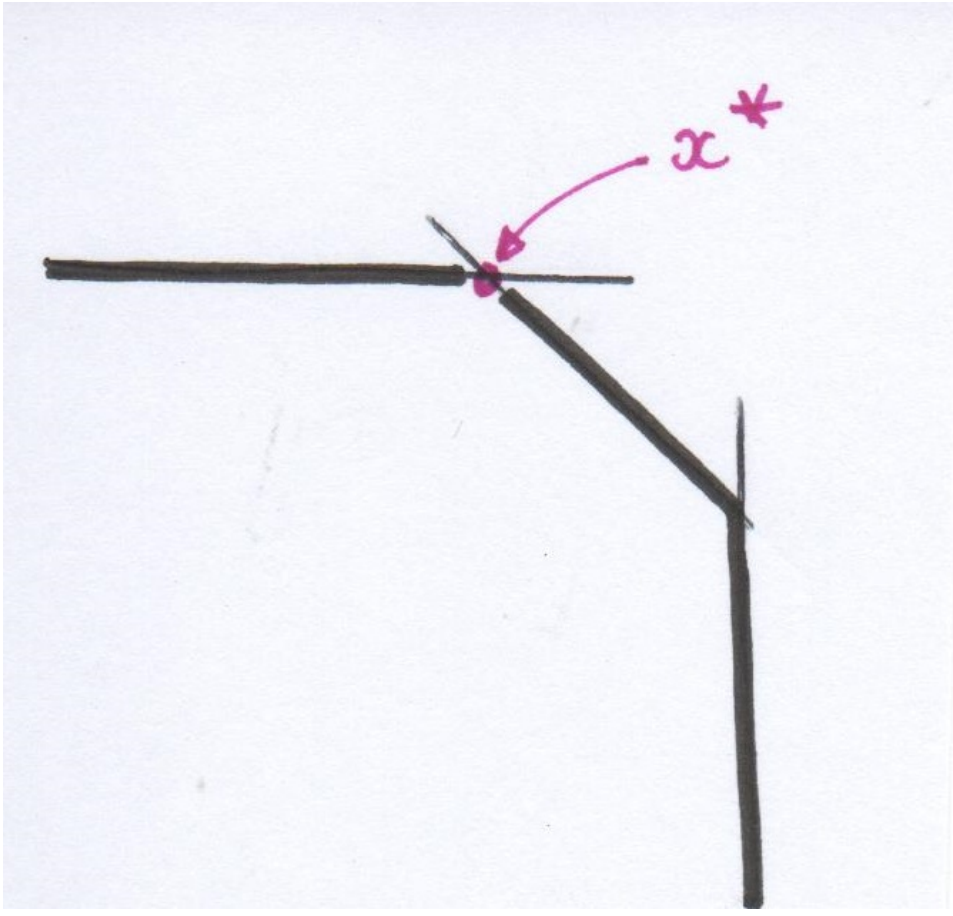
Qu'est ce que ça change ?

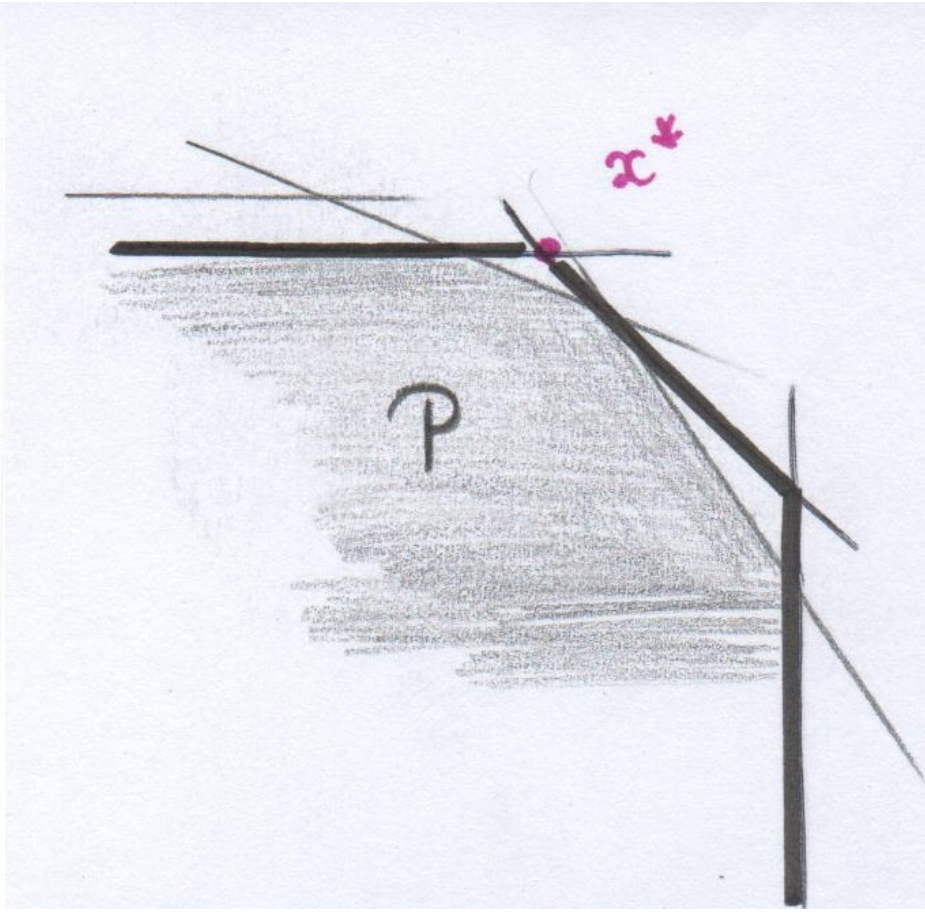
→ Nombre exponentiel de contraintes
→ Procédure de "Branch-and-Cut"

Qu'est ce que ça change ?

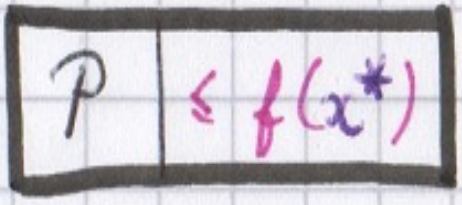
→ Nombre exponentiel de contraintes
→ Procédure de "Branch-and-Cut"
→ Algorithme de coupe

Image à avoir en tête pour une coupe

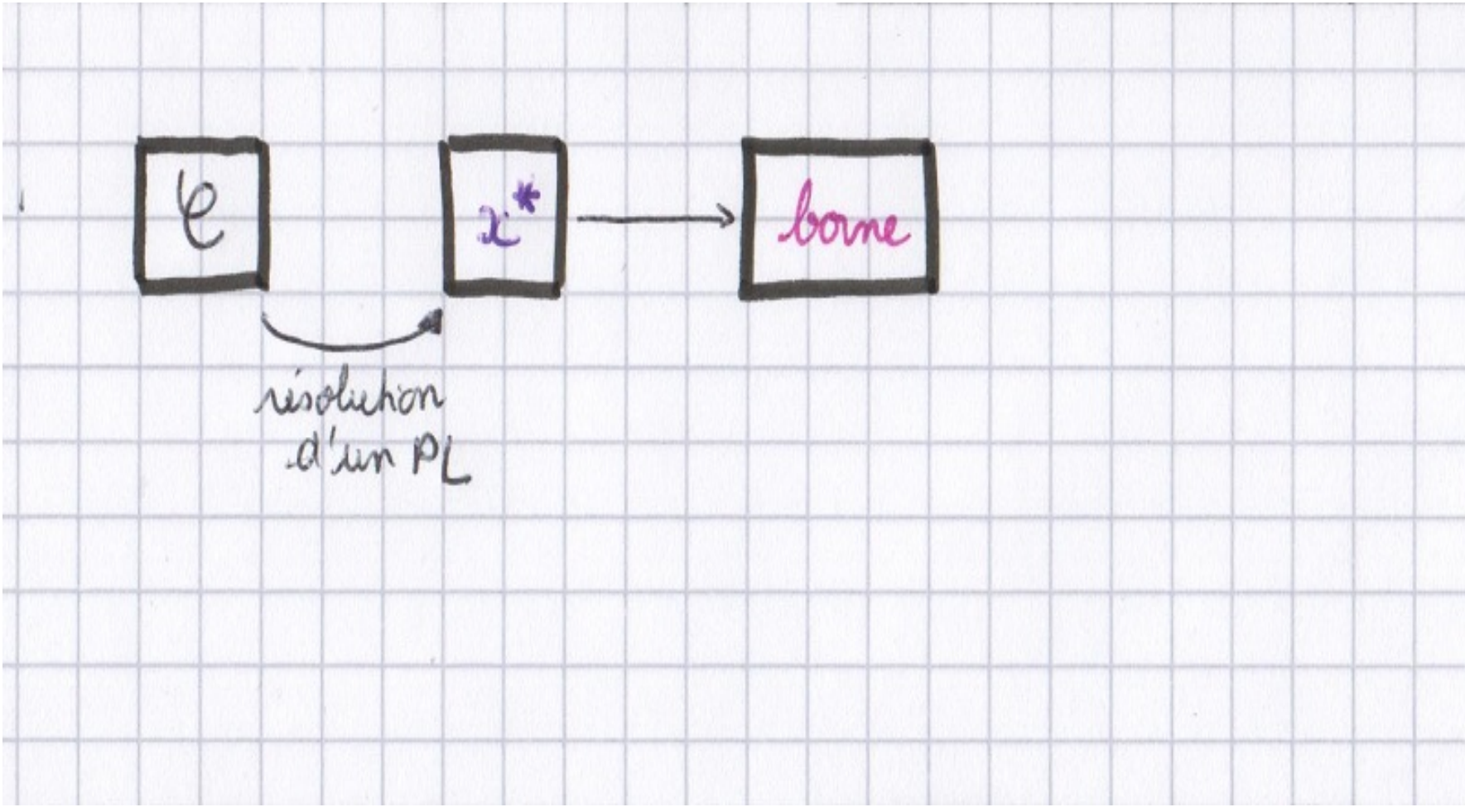




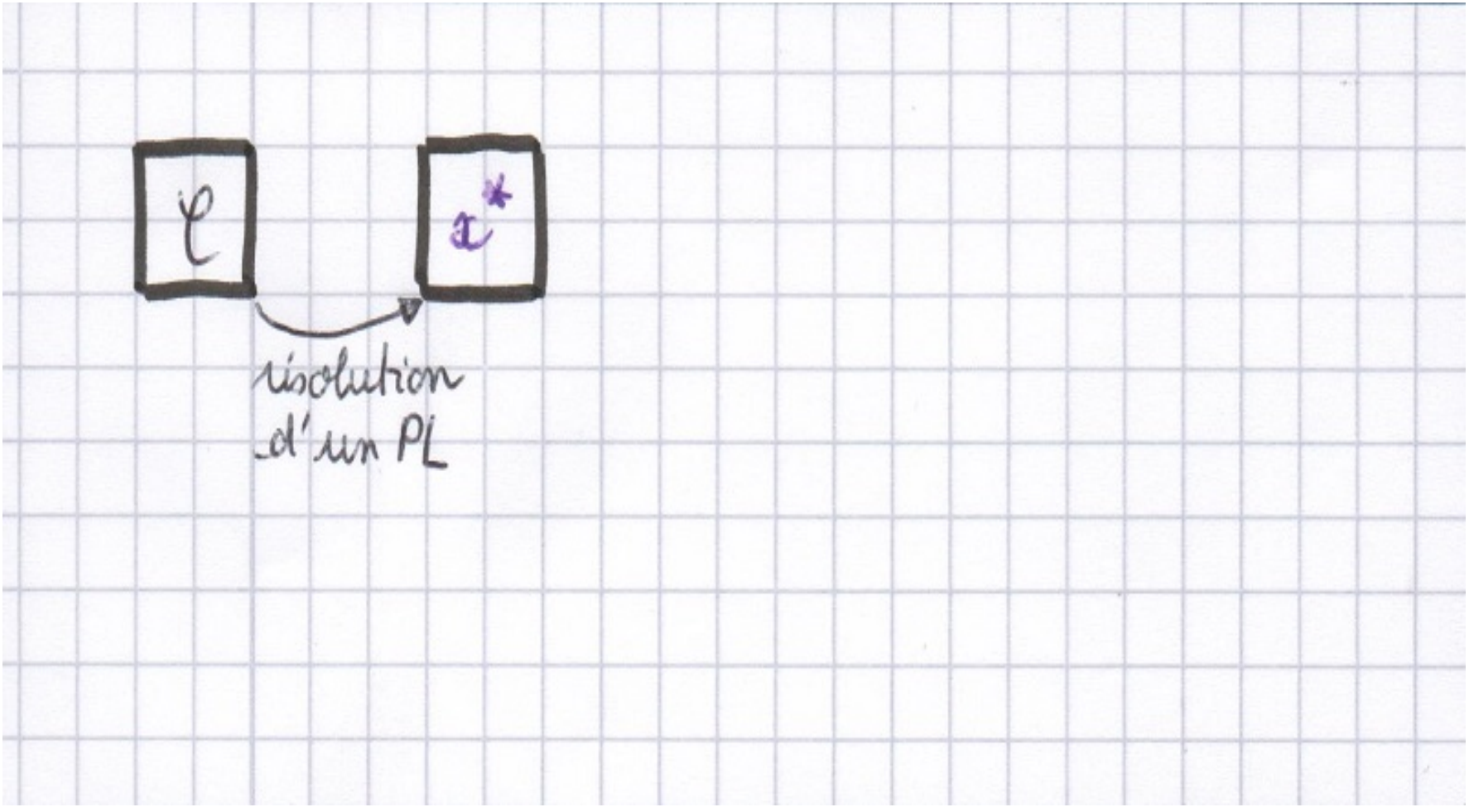
Qu'est ce que le "Branch-and-Cut" ?



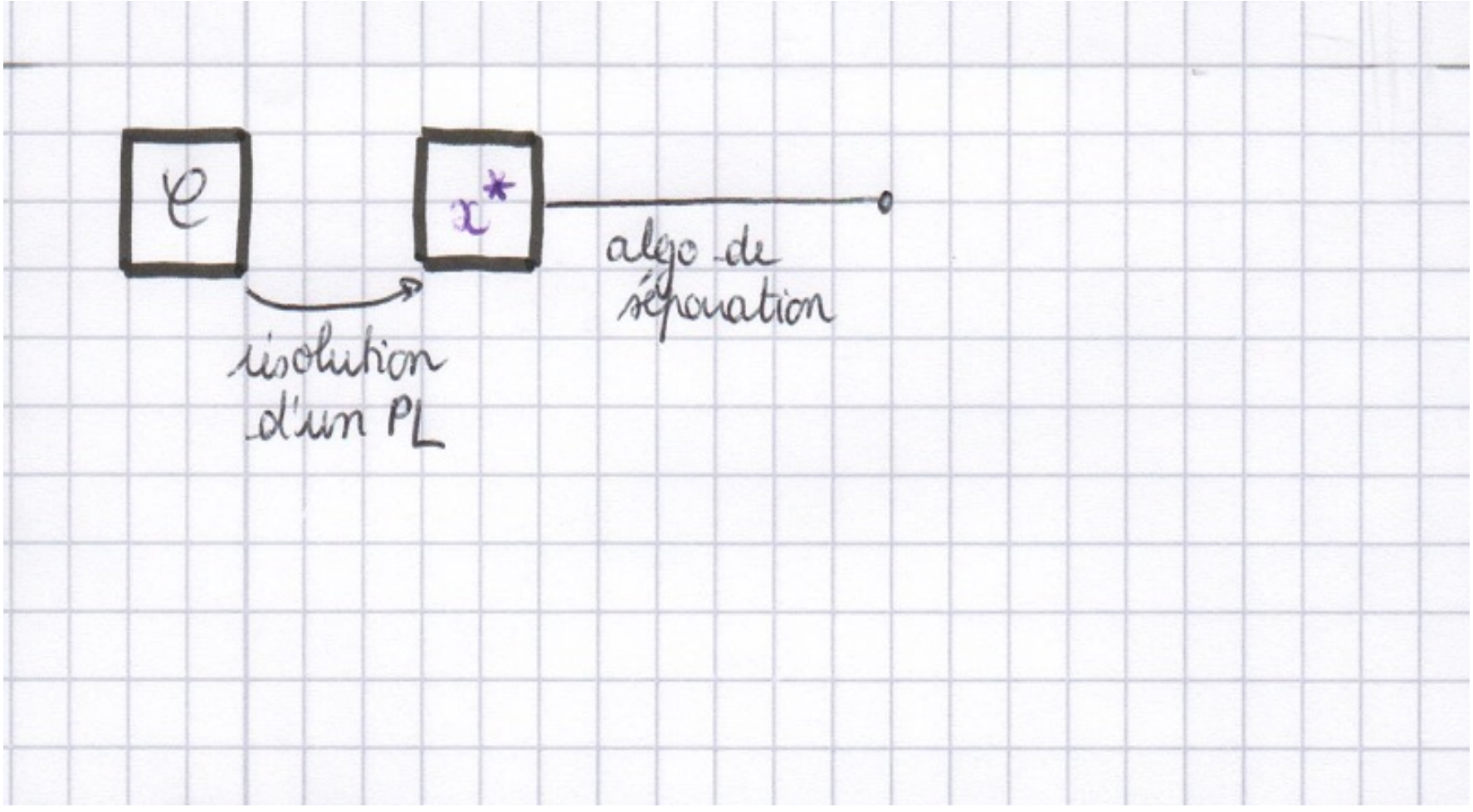
Qu'est ce que le "Branch-and-Cut" ?



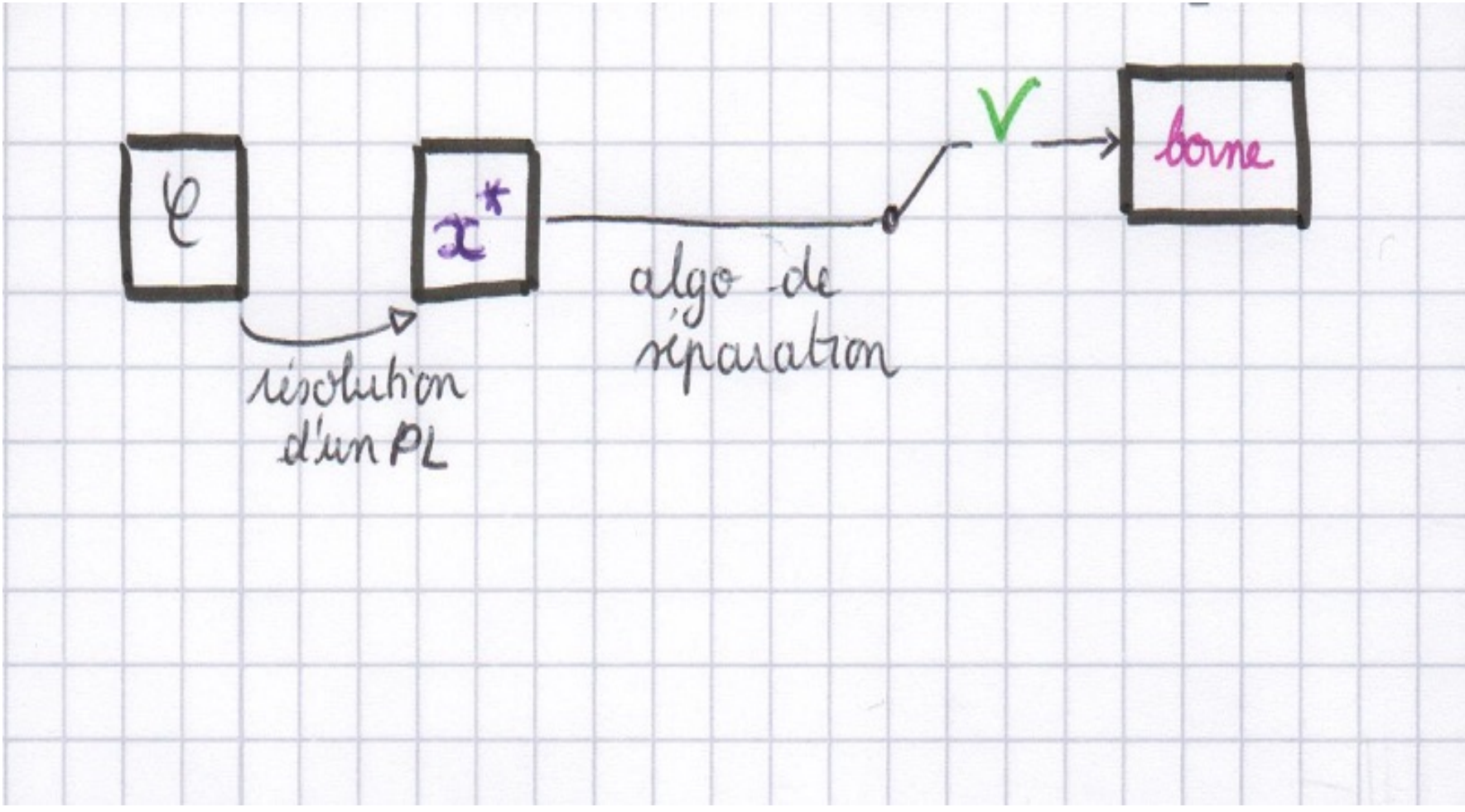
Qu'est ce que le "Branch-and-Cut" ?



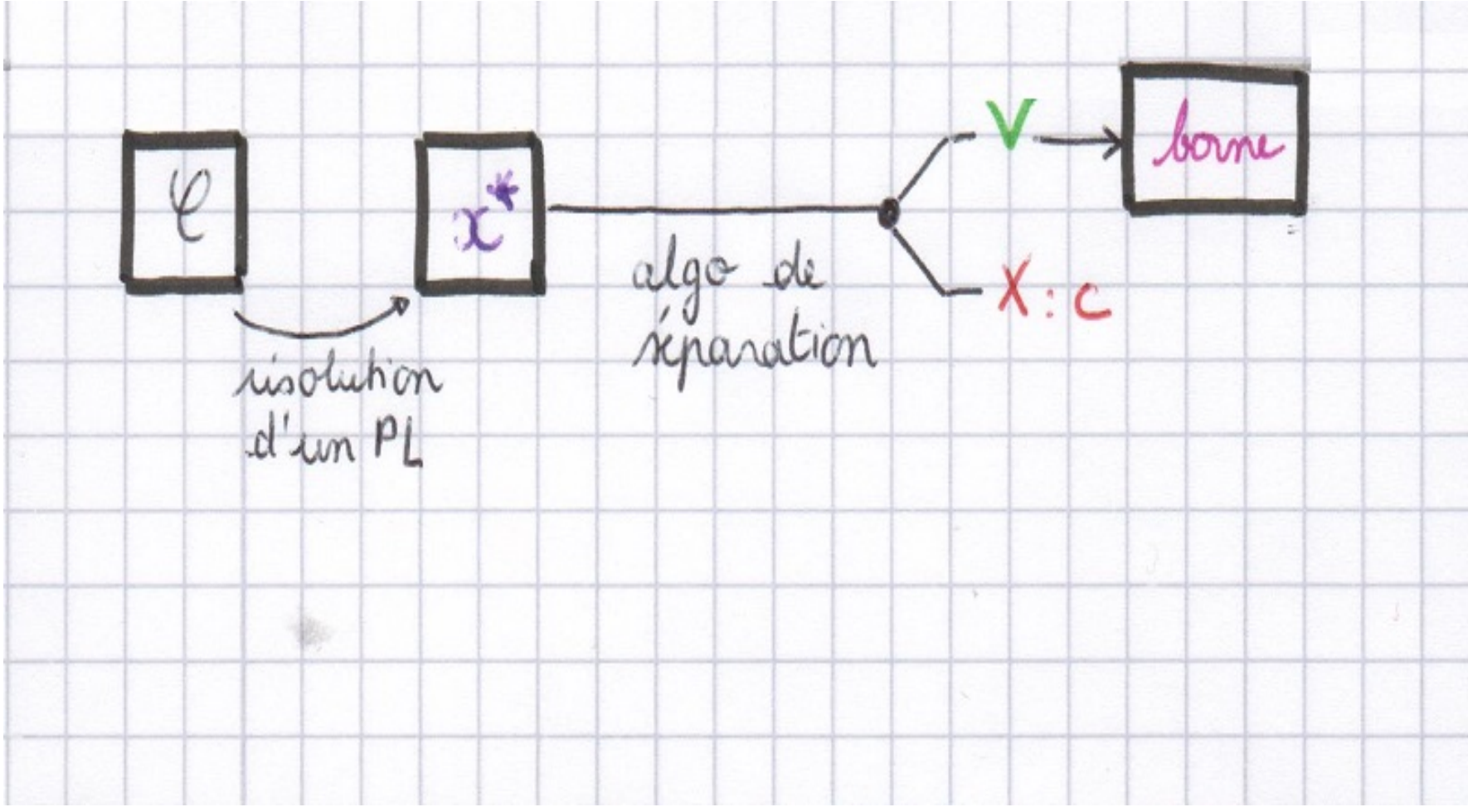
Qu'est ce que le "Branch-and-Cut" ?



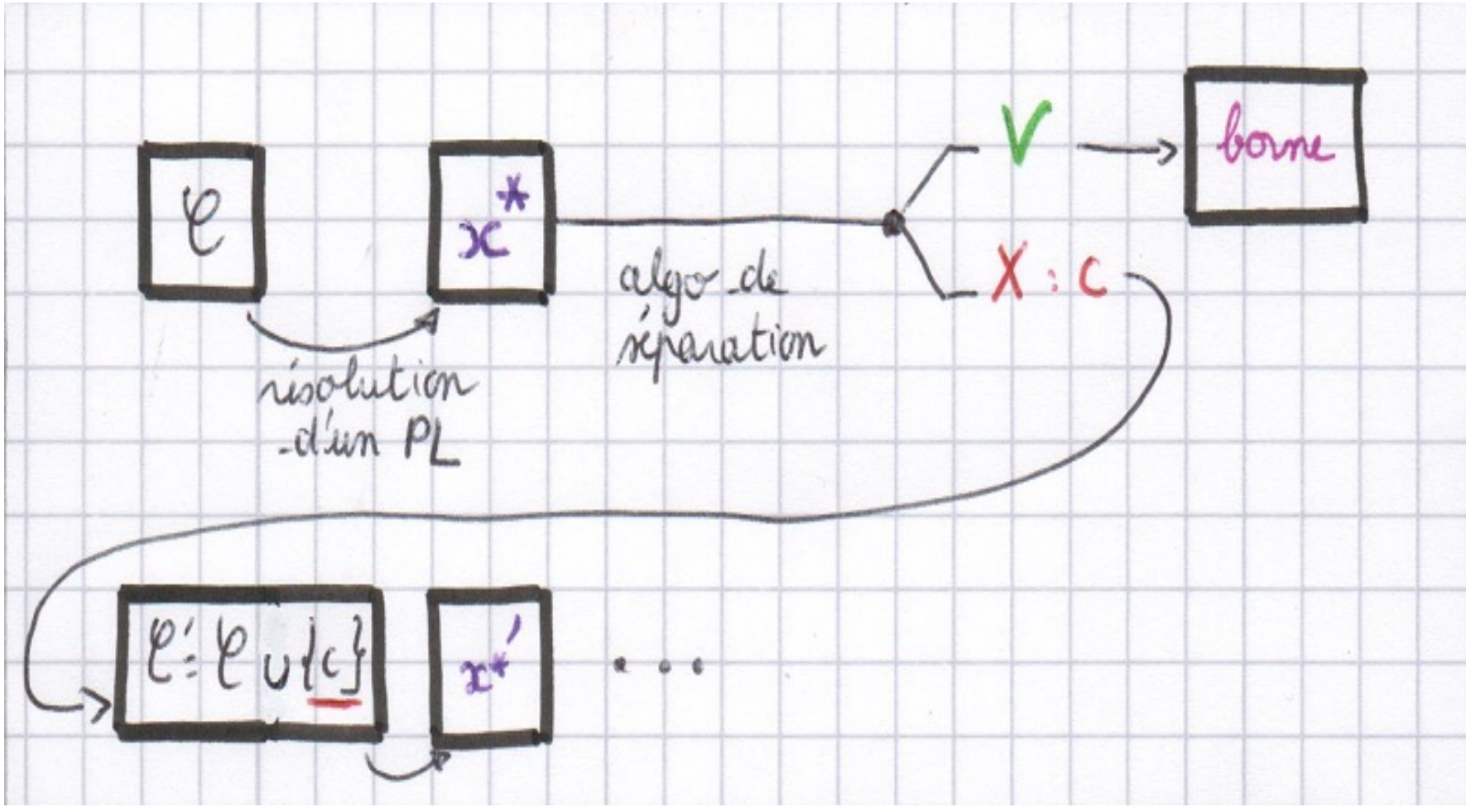
Qu'est ce que le "Branch-and-Cut" ?



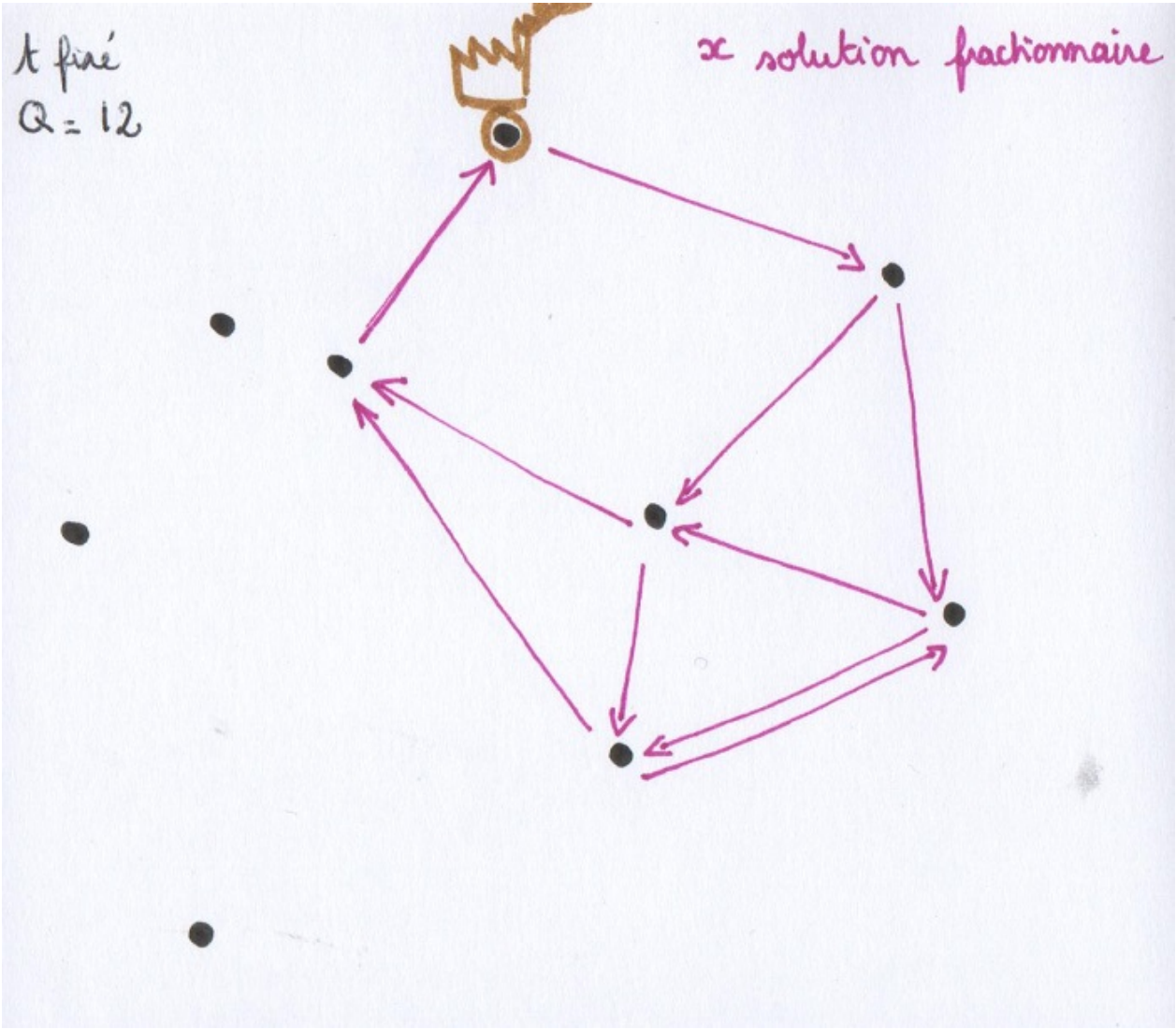
Qu'est ce que le "Branch-and-Cut" ?



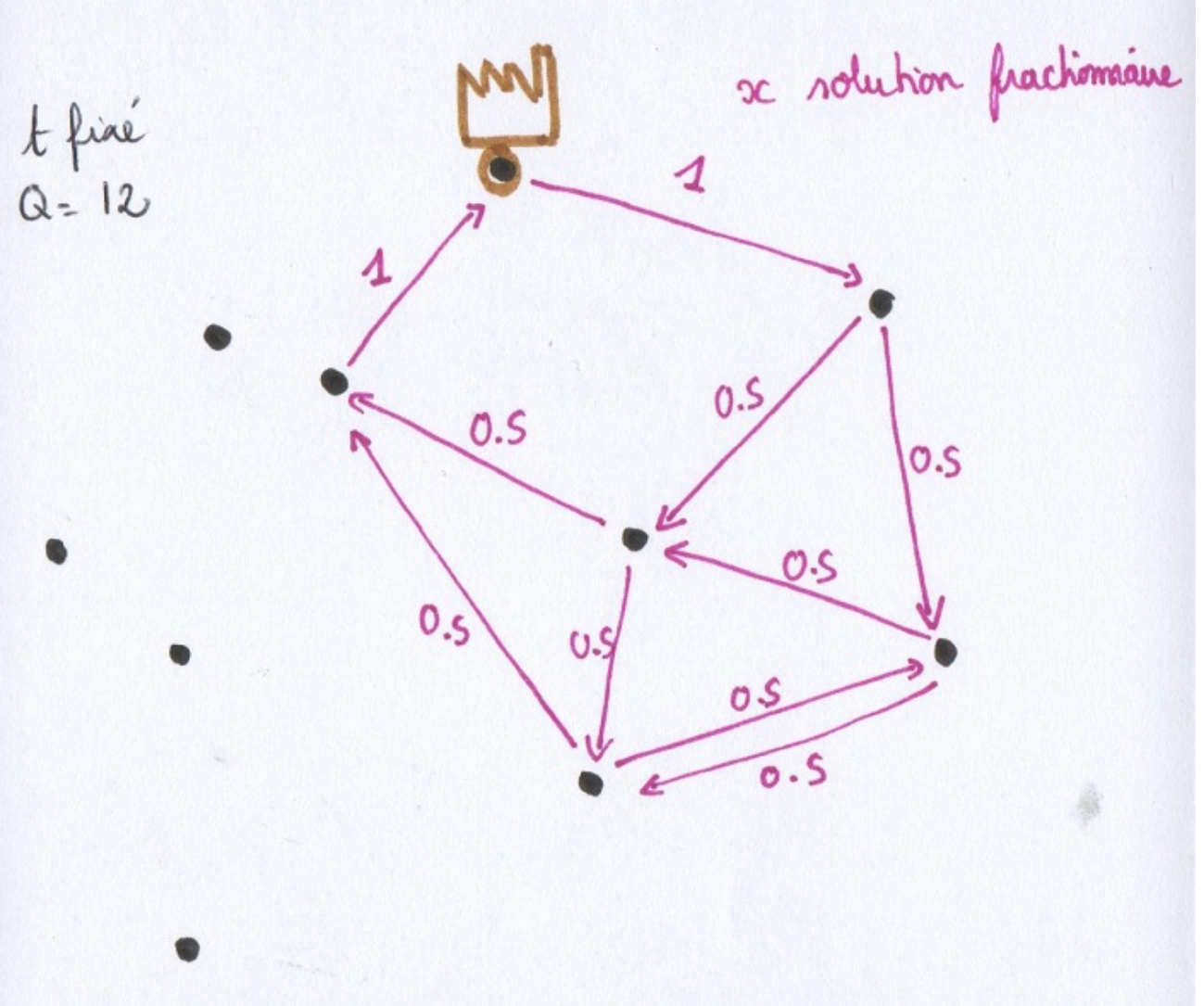
Qu'est ce que le "Branch-and-Cut" ?



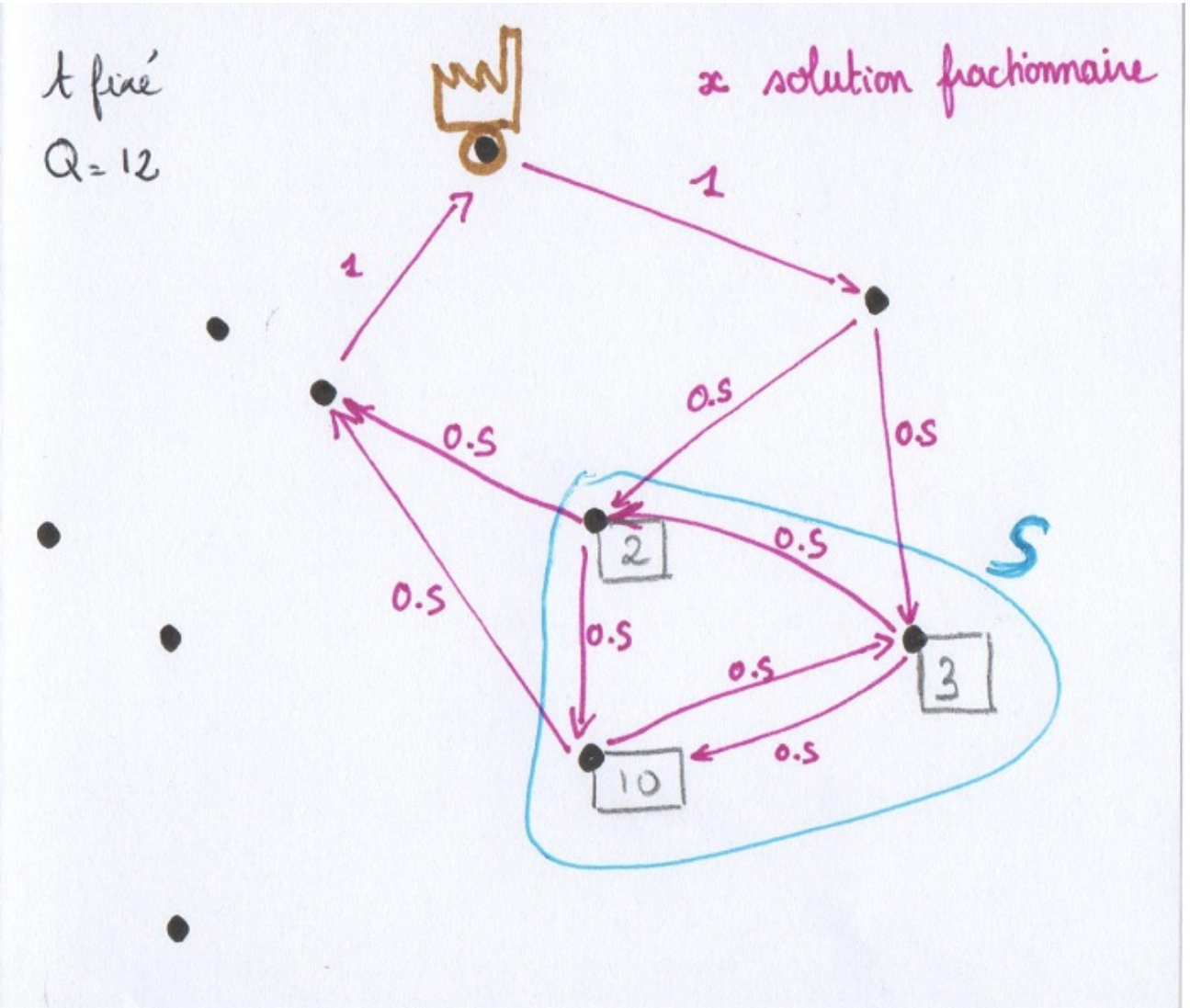
Partir d'une solution fractionnaire...



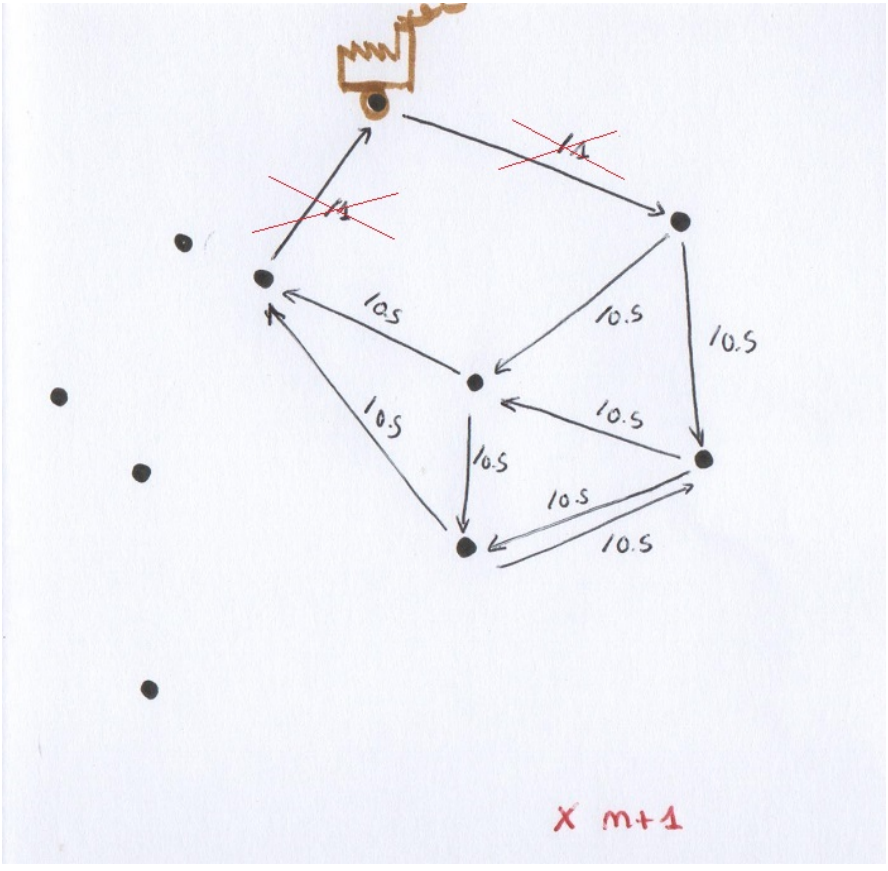
Partir d'une solution fractionnaire...



Partir d'une solution fractionnaire...



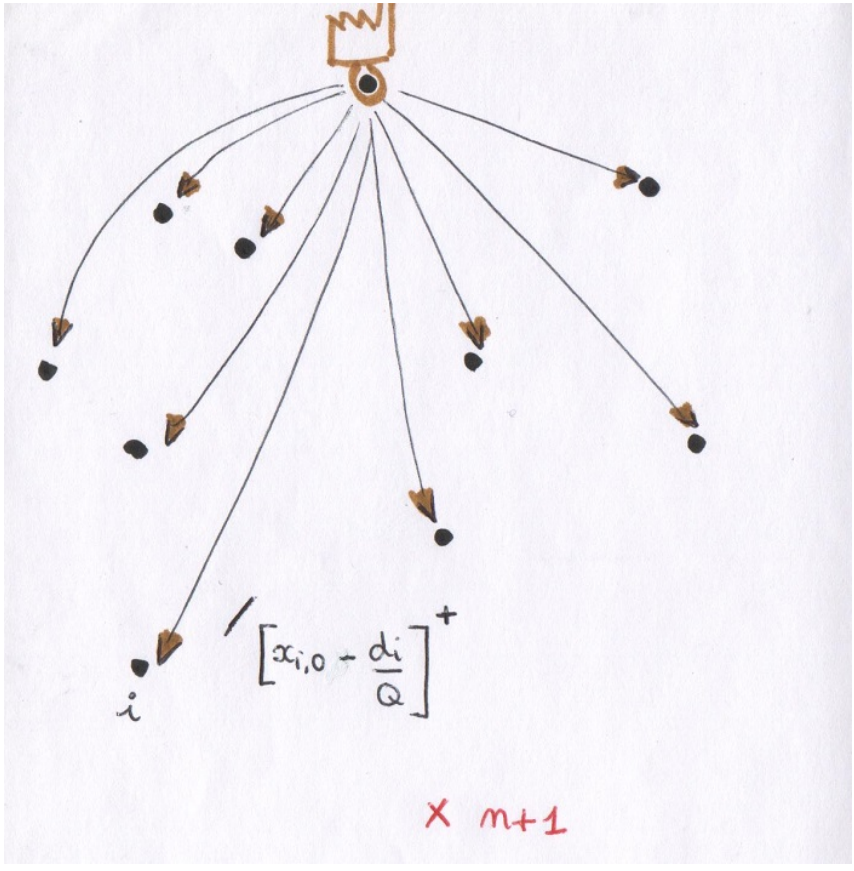
...considérer un graphe ad hoc ...



$$\forall (i,j) \in N_c^2, \quad \kappa_{i,j} = x_{i,j}$$

$$G^t : \begin{cases} V = \{0\} \cup [1..n] \cup \{n+1\} \\ E^t = \{(i,0) \mid i \in N\} \cup \{(i,j) \in N_c^2 \mid x_{i,j,t} > 0\} \cup \{(n+1,j) \mid j \in N_c\} \end{cases}$$

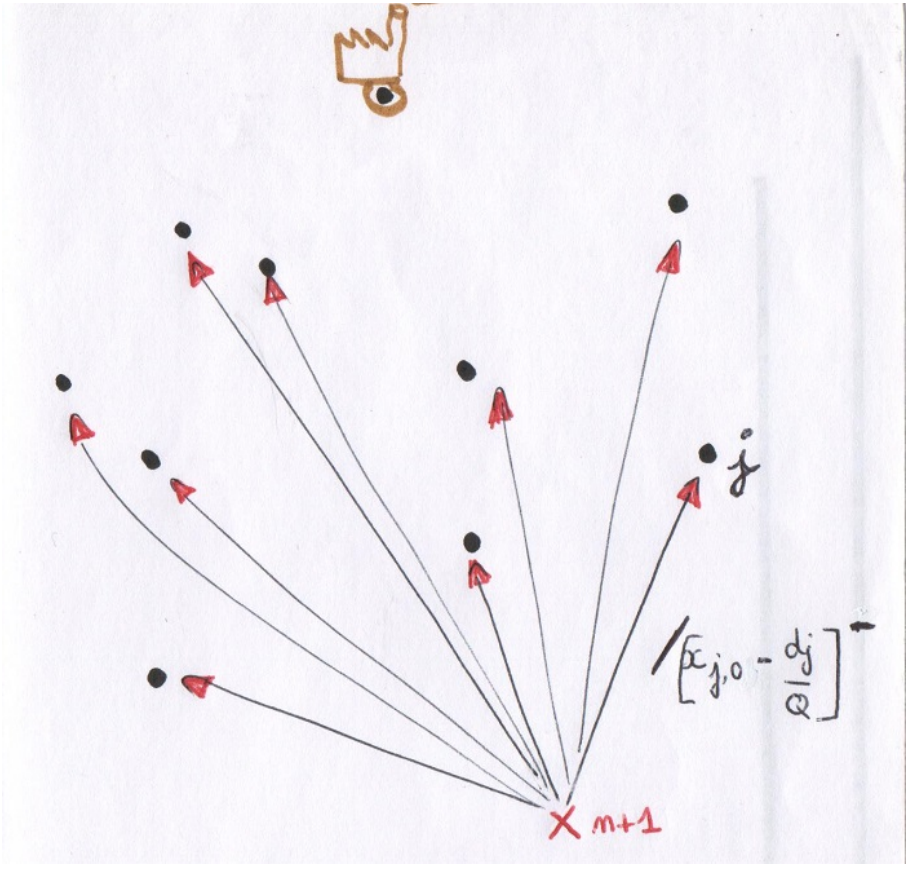
...considérer un graphe ad hoc ...



$$\forall i \in N_c, \quad \kappa_{i,0} = \left[x_{i,0} - \frac{d_i}{Q}\right]^+$$

$$G^t : \begin{cases} V = \{0\} \cup [1..n] \cup \{n+1\} \\ E^t = \{(i,0) \mid i \in N\} \cup \{(i,j) \in N_c^2 \mid x_{i,j,t} > 0\} \cup \{(n+1,j) \mid j \in N_c\} \end{cases}$$

...considérer un graphe ad hoc ...



$$\forall j \in N_c, \quad \kappa_{n+1,j} = \left\lceil x_{j,0} - \frac{d_j}{Q} \right\rceil^-$$

$$G^t : \begin{cases} V = \{0\} \cup [1..n] \cup \{n+1\} \\ E^t = \{(i,0) \mid i \in N\} \cup \{(i,j) \in N_c^2 \mid x_{i,j,t} > 0\} \cup \{(n+1,j) \mid j \in N_c\} \end{cases}$$

...chercher une coupe min et comparer la valeur

1. Considérer une solution x
2. À t fixé, créer le graphe G_t avec ses capacités κ

...chercher une coupe min et comparer la valeur

1. Considérer une solution x
2. À t fixé, créer le graphe G_t avec ses capacités κ
3. Trouver une coupe min entre 0 et $n+1$

...chercher une coupe min et comparer la valeur

1. Considérer une solution x
2. À t fixé, créer le graphe G_t avec ses capacités κ
3. Trouver une coupe min entre 0 et $n+1$
4. Comparer la valeur de la coupe à $P = \sum_{j \in N_c} \kappa_{n+1,j}$

...chercher une coupe min et comparer la valeur

1. Considérer une solution x
2. À t fixé, créer le graphe G_t avec ses capacités κ
3. Trouver une coupe min entre 0 et $n+1$
4. Comparer la valeur de la coupe à $P = \sum_{j \in N_c} \kappa_{n+1,j}$
5. En déduire si x vérifie toutes les contraintes, ou sinon en renvoyer une transgressée

Comment ça marche ?

Considérons $V = U \sqcup W$ la coupe min, avec $U \ni n{+}1$
On pose $S = U_{\setminus \{n+1\}}$ ainsi $V = S \sqcup \{n{+}1\} \sqcup W$ et $S \subset N_c$

Comment ça marche ?

Considérons $V = U \sqcup W$ la coupe min, avec $U \ni n{+}1$
On pose $S = U_{\setminus\{n+1\}}$ ainsi $V = S \sqcup \{n{+}1\} \sqcup W$ et $S \subset N_c$

$$\kappa(U, W) - P =$$

Comment ça marche ?

Considérons $V = U \sqcup W$ la coupe min, avec $U \ni n{+}1$
On pose $S = U_{\setminus \{n+1\}}$ ainsi $V = S \sqcup \{n{+}1\} \sqcup W$ et $S \subset N_c$

$$\kappa(U, W) - P = \sum_{(i,j) \in U \times W} \kappa_{i,j} - \sum_{j \in N_c} \kappa_{n+1,j}$$

Comment ça marche ?

Considérons $V = U \sqcup W$ la coupe min, avec $U \ni n{+}1$
On pose $S = U_{\setminus \{n+1\}}$ ainsi $V = S \sqcup \{n{+}1\} \sqcup W$ et $S \subset N_c$

$$\kappa(U, W) - P = \sum_{(i,j) \in U \times W} \kappa_{i,j} - \sum_{j \in N_c} \kappa_{n+1,j}$$

or $U = S \sqcup \{n{+}1\}$

Comment ça marche ?

Considérons $V = U \sqcup W$ la coupe min, avec $U \ni n{+}1$
On pose $S = U_{\setminus \{n+1\}}$ ainsi $V = S \sqcup \{n{+}1\} \sqcup W$ et $S \subset N_c$

$$\begin{aligned}\kappa(U,W) - P &= \sum_{(i,j)\in U\times W} \kappa_{i,j} - \sum_{j\in N_c} \kappa_{n+1,j} \\ &= \sum_{(i,j)\in S\times W} \kappa_{i,j} + \sum_{j\in W} \kappa_{n+1,j} - \sum_{j\in N_c} \kappa_{n+1,j}\end{aligned}$$

Comment ça marche ?

Considérons $V = U \sqcup W$ la coupe min, avec $U \ni n{+}1$
On pose $S = U_{\setminus\{n+1\}}$ ainsi $V = S \sqcup \{n{+}1\} \sqcup W$ et $S \subset N_c$

$$\begin{aligned}\kappa(U,W) - P &= \sum_{(i,j)\in U\times W} \kappa_{i,j} - \sum_{j\in N_c} \kappa_{n+1,j} \\ &= \sum_{(i,j)\in S\times W} \kappa_{i,j} + \sum_{j\in W} \kappa_{n+1,j} - \sum_{j\in N_c} \kappa_{n+1,j}\end{aligned}$$

or $W = \{0\} \sqcup N_{c\setminus S}$ et $(n{+}1, 0) \notin E$

Comment ça marche ?

Considérons $V = U \sqcup W$ la coupe min, avec $U \ni n{+}1$
On pose $S = U_{\setminus\{n+1\}}$ ainsi $V = S \sqcup \{n{+}1\} \sqcup W$ et $S \subset N_c$

$$\begin{aligned}\kappa(U,W) - P &= \sum_{(i,j)\in U\times W} \kappa_{i,j} - \sum_{j\in N_c} \kappa_{n+1,j} \\ &= \sum_{(i,j)\in S\times\{0\}\sqcup N_{c\setminus S}} \kappa_{i,j} + \sum_{j\in N_{c\setminus S}} \kappa_{n+1,j} - \sum_{j\in N_c} \kappa_{n+1,j}\end{aligned}$$

Comment ça marche ?

Considérons $V = U \sqcup W$ la coupe min, avec $U \ni n{+}1$
On pose $S = U_{\setminus\{n+1\}}$ ainsi $V = S \sqcup \{n{+}1\} \sqcup W$ et $S \subset N_c$

$$\kappa(U,W) - P = \sum_{(i,j)\in U\times W} \kappa_{i,j} - \sum_{j\in N_c} \kappa_{n+1,j}$$

$$= \sum_{(i,j)\in S\times\{0\}\sqcup N_c\setminus S} \kappa_{i,j} + \underbrace{\sum_{j\in N_c\setminus S} \kappa_{n+1,j} - \sum_{j\in N_c} \kappa_{n+1,j}}$$

Comment ça marche ?

Considérons $V = U \sqcup W$ la coupe min, avec $U \ni n{+}1$
On pose $S = U_{\setminus\{n+1\}}$ ainsi $V = S \sqcup \{n{+}1\} \sqcup W$ et $S \subset N_c$

$$\begin{aligned}
\kappa(U,W) - P &= \sum_{(i,j)\in U\times W} \kappa_{i,j} - \sum_{j\in N_c} \kappa_{n+1,j} \\
&= \sum_{(i,j)\in S\times\{0\}\sqcup N_{c\setminus S}} \kappa_{i,j} + \sum_{j\in N_{c\setminus S}} \kappa_{n+1,j} - \sum_{j\in N_c} \kappa_{n+1,j} \\
&= \sum_{(i,j)\in S\times\{0\}\sqcup N_{c\setminus S}} \kappa_{i,j} - \sum_{j\in N_c} \kappa_{n+1,j}
\end{aligned}$$

Comment ça marche ?

Considérons $V = U \sqcup W$ la coupe min, avec $U \ni n{+}1$
On pose $S = U_{\setminus\{n+1\}}$ ainsi $V = S \sqcup \{n{+}1\} \sqcup W$ et $S \subset N_c$

$$\begin{aligned}
\kappa(U,W) - P &= \sum_{(i,j)\in U\times W} \kappa_{i,j} - \sum_{j\in N_c} \kappa_{n+1,j} \\
&= \sum_{(i,j)\in S\times\{0\}\sqcup N_{c\setminus S}} \kappa_{i,j} + \sum_{j\in N_{c\setminus S}} \kappa_{n+1,j} - \sum_{j\in N_c} \kappa_{n+1,j} \\
&= \underline{\sum_{(i,j)\in S\times\{0\}\sqcup N_{c\setminus S}} \kappa_{i,j} - \sum_{j\in N_c} \kappa_{n+1,j}}
\end{aligned}$$

Comment ça marche ?

Considérons $V = U \sqcup W$ la coupe min, avec $U \ni n{+}1$
On pose $S = U_{\setminus\{n+1\}}$ ainsi $V = S \sqcup \{n{+}1\} \sqcup W$ et $S \subset N_c$

$$\begin{aligned}
\kappa(U,W) - P &= \sum_{(i,j)\in U\times W} \kappa_{i,j} - \sum_{j\in N_c} \kappa_{n+1,j} \\
&= \sum_{(i,j)\in S\times\{0\}\sqcup N_{c\setminus S}} \kappa_{i,j} + \sum_{j\in N_{c\setminus S}} \kappa_{n+1,j} - \sum_{j\in N_c} \kappa_{n+1,j} \\
&= \sum_{(i,j)\in S\times\{0\}\sqcup N_{c\setminus S}} \kappa_{i,j} - \sum_{j\in N_c} \kappa_{n+1,j} \\
&= \sum_{i\in S} \kappa_{i,0} + \sum_{(i,j)\in S\times N_{c\setminus S}} \kappa_{i,j} + \sum_{j\in S} \kappa_{n+1,j}
\end{aligned}$$

Comment ça marche ?

Considérons $V = U \sqcup W$ la coupe min, avec $U \ni n{+}1$
On pose $S = U_{\setminus\{n+1\}}$ ainsi $V = S \sqcup \{n{+}1\} \sqcup W$ et $S \subset N_c$

$$\begin{aligned}
\kappa(U,W) - P &= \sum_{(i,j)\in U\times W} \kappa_{i,j} - \sum_{j\in N_c} \kappa_{n+1,j} \\
&= \sum_{(i,j)\in S\times\{0\}\sqcup N_{c\setminus S}} \kappa_{i,j} + \sum_{j\in N_{c\setminus S}} \kappa_{n+1,j} - \sum_{j\in N_c} \kappa_{n+1,j} \\
&= \sum_{(i,j)\in S\times\{0\}\sqcup N_{c\setminus S}} \kappa_{i,j} - \sum_{j\in N_c} \kappa_{n+1,j} \\
&= \underline{\sum_{i\in S} \kappa_{i,0}} + \sum_{(i,j)\in S\times N_{c\setminus S}} \kappa_{i,j} - \underline{\sum_{j\in S} \kappa_{n+1,j}}
\end{aligned}$$

Comment ça marche ?

Considérons $V = U \sqcup W$ la coupe min, avec $U \ni n{+}1$
On pose $S = U_{\setminus\{n+1\}}$ ainsi $V = S \sqcup \{n{+}1\} \sqcup W$ et $S \subset N_c$

$$\begin{aligned}
\kappa(U,W) - P &= \sum_{(i,j)\in U\times W} \kappa_{i,j} - \sum_{j\in N_c} \kappa_{n+1,j} \\
&= \sum_{(i,j)\in S\times\{0\}\sqcup N_{c\setminus S}} \kappa_{i,j} + \sum_{j\in N_{c\setminus S}} \kappa_{n+1,j} - \sum_{j\in N_c} \kappa_{n+1,j} \\
&= \sum_{(i,j)\in S\times\{0\}\sqcup N_{c\setminus S}} \kappa_{i,j} - \sum_{j\in N_c} \kappa_{n+1,j} \\
&= \sum_{i\in S} \kappa_{i,0} + \sum_{(i,j)\in S\times N_{c\setminus S}} \kappa_{i,j} + \sum_{j\in S} \kappa_{n+1,j} \\
&= \sum_{(i,j)\in S\times N_{c\setminus S}} \kappa_{i,j} + \sum_{i\in S} \left(\kappa_{i,0} - \kappa_{n+1,i}\right)
\end{aligned}$$

Comment ça marche ?

$$\kappa(U, W) - P = \sum_{(i,j) \in S \times N_c \setminus S} \kappa_{i,j} + \sum_{i \in S} (\kappa_{i,0} - \kappa_{n+1,i})$$

Comment ça marche ?

$$\kappa(U,W) - P = \sum_{(i,j)\in S\times N_c\setminus S} \kappa_{i,j} + \sum_{i\in S} \underbrace{(\kappa_{i,0} - \kappa_{n+1,i})}_{\left[x_{i,0}-\frac{d_i}{Q}\right]^+ - \left[x_{i,0}-\frac{d_i}{Q}\right]^-}$$

or $[a]^+ - [a]^- = a$

Comment ça marche ?

$$\kappa(U,W) - P = \sum_{(i,j)\in S\times N_c\setminus S} \kappa_{i,j} + \sum_{i\in S} (\kappa_{i,0} - \kappa_{n+1,i})$$

$$= \sum_{(i,j)\in S\times N_c\setminus S} x_{i,j} + \sum_{i\in S} \left(x_{i,0} - \frac{d_i}{Q} \right)$$

Comment ça marche ?

$$\begin{aligned}
\kappa(U,W) - P &= \sum_{(i,j)\in S\times N_c\setminus S} \kappa_{i,j} + \sum_{i\in S} (\kappa_{i,0} - \kappa_{n+1,i}) \\
&= \sum_{(i,j)\in S\times N_c\setminus S} x_{i,j} + \sum_{i\in S} \left(x_{i,0} - \frac{d_i}{Q} \right) \\
&= \sum_{(i,j)\in S\times N_c\setminus S} x_{i,j} + \sum_{i\in S} x_{i,0} - \sum_{i\in S} \frac{d_i}{Q}
\end{aligned}$$

Comment ça marche ?

$$\begin{aligned}
\kappa(U,W) - P &= \sum_{(i,j)\in S\times N_c\setminus S} \kappa_{i,j} + \sum_{i\in S}(\kappa_{i,0} - \kappa_{n+1,i}) \\
&= \sum_{(i,j)\in S\times N_c\setminus S} x_{i,j} + \sum_{i\in S}\left(x_{i,0} - \frac{d_i}{Q}\right) \\
&= \sum_{(i,j)\in S\times N_c\setminus S} x_{i,j} + \sum_{i\in S} x_{i,0} - \sum_{i\in S}\frac{d_i}{Q} \\
&\text{or } N = \{0\} \sqcup N_c \text{ et } 0 \notin S
\end{aligned}$$

Comment ça marche ?

$$\begin{aligned}
\kappa(U,W) - P &= \sum_{(i,j)\in S\times N_{c\setminus S}} \kappa_{i,j} + \sum_{i\in S} (\kappa_{i,0} - \kappa_{n+1,i}) \\
&= \sum_{(i,j)\in S\times N_{c\setminus S}} x_{i,j} + \sum_{i\in S} \left(x_{i,0} - \frac{d_i}{Q} \right) \\
&= \sum_{(i,j)\in S\times N_{c\setminus S}} x_{i,j} + \sum_{i\in S} x_{i,0} - \sum_{i\in S} \frac{d_i}{Q} \\
&= \sum_{(i,j)\in S\times N_{\setminus S}} x_{i,j} - \sum_{i\in S} \frac{d_i}{Q}
\end{aligned}$$

Comment on "coupe" ?

$$\kappa(U,W) - P = \sum_{(i,j)\in S\times N\setminus S} x_{i,j,t} - \sum_{i\in S} \frac{d_{i,t}}{Q}$$

- Si $\kappa(U,W) - P < 0$, alors $\sum_{i,\in S}\sum_{j\notin S} x_{i,j,t} < \sum_{i\in S} \frac{d_{i,t}}{Q}$, la contrainte $(17_{S,t})$ est transgressée par la solution $x_{.,.,t}$.

- Sinon, par minimalité de cette coupe, la solution $x_{.,.,t}$, vérifie toutes les contraintes $(17_{.,t})$

1. Le problème concret

2. Modélisation par un PL mixte

3. Comment résoudre une formulation compacte ?

4. Une reformulation non compacte

5. Conclusion

Conclusion

- La PL/PLNE est un outil qui permet de modéliser beaucoup

Conclusion

- La PL/PLNE est un outil qui permet de modéliser beaucoup
- Il a été beaucoup étudié/utilisé donc on dispose de solveurs

Conclusion

- La PL/PLNE est un outil qui permet de modéliser beaucoup
- Il a été beaucoup étudié/utilisé donc on dispose de solveurs
- MAIS il permet aussi de modéliser des problèmes NP-difficiles

Conclusion

- La PL/PLNE est un outil qui permet de modéliser beaucoup
- Il a été beaucoup étudié/utilisé donc on dispose de solveurs
- MAIS il permet aussi de modéliser des problèmes NP-difficiles
- Il faut alors mettre en place "à la main" des améliorations dédiées

Références

- Yossiri Adulyasak, Jean-François Cordeau et Raf Jans, *The production routing problem : A review of formulations and solution algorithms*, Computers & Operations Research, vol 55, Mars 2015
- P. Augerat, J-M. Belenguer, A. Corberàn et D. Naddef, *Separating capacity constraints in the CVRP using tabu search*, European journl of operationnal research, vol 106, 1997