PL résolution géométriq :

① tracer les droites de chq contraintes
② éliminer la zone selon le pt (0;0)
③ tracer le gradient de la fct° obj
④ trouver l'OPT en utilisant ⊥ au gradient
→ dernier pt atteint par la droite ⊥ en glissant le long du gradient (vers $+\infty$ si max, $-\infty$ si min)
⑤ OPT avec les équat° des droites
⑥ calculer la valeur correspondante

(possibilité de faire sans gradient en calculant tous les pts et choisir selon max ou min)

PL canoniq : $\geq$ ou $\leq$
PL standard : $=$ (2nd mb tjrs ⊕)

mettre sous forme standard :

$b \geq 0$, $e_1 \geq 0$

$ax \leq b \rightarrow ax + e_1 = b$

$ax \geq b \rightarrow ax - e_1 = b$

sommets du polygone :

① choisir n variables "base" (car n contraintes)
② exprimer var en base en fct° des autres ("hors base")
③ calculer val des var en base en remplaçant les var hors base par zéro

⇒ sommets réalisables si ∀ ⊕ (qu'importe max ou min)

$\binom{m}{n}$ intersect° possibles pour m var, n contraintes

algo simplex :

① mettre le prob sous f. canoniq
② choisir une base
③ var en base en fct° var hb
④ mettre à 0 les var hb
⑤ continuer tant q ∀ var en base ⊕/⊖ pour max/min

→ changemt de base :
↳ var entrante : plus ↗ profit
↳ sortante : celle qui s'annule en premier quand on ↗ var entrante

tableau simplex : (TD2 exo 11)

	coeff	x_1	x_2	e_1	e_2	e_3	hb = 0
en base	e_1	2	1	1	0	0	8
	e_2	1	2	0	1	0	7
	e_3	0	1	0	0	1	3
	fct° obj	4	5	0	0	0	0 = −profit

x_2 sort car $5 > 4$ (fct° obj)

① $\begin{cases} 8/1 = 8 \\ 7/2 = 3{,}5 \\ 3/1 = 3 \end{cases}$ diviser par val du col sortant

⇒ e_3 sort car min

chgmt base : x_2 en base
soustraire les lignes des autres var en base pour que la colonne de x_2 soit à 0 (sauf celle de x_2 qui vaut 1) ET faire algo simplex pour coeff entrant et quand hors-base = 0

	coeff	x_1	x_2	e_1	e_2	e_3	hb = 0
$(e_1 - e_3)$	e_1	2	0	1	0	−1	5
$(e_2 - 2e_3)$	e_2	1	0	0	1	−2	1
	x_2	0	1	0	0	1	3
	fct° obj	4	0	0	0	0	−15

x_1 entre $\quad 5/2 = 2{,}5$; $1/1 = 1$ ⇒ e_2 sort

Dualité : (cours 3 p. 34)

à l'optimum, les valeurs des fct° objectives (P et D) sont égaux

• sous forme canoniq :

primal : max $z = a x_1 + b x_2$

$\mathcal{P} \begin{cases} a_1 x_1 + b_1 x_2 \leq c_1 \\ a_2 x_1 + b_2 x_2 \leq c_2 \\ a_3 x_1 + b_3 x_2 \leq c_4 \end{cases}$ $\quad x_i \geq 0$

dual : min $z' = c_1 y_1 + c_2 y_2 + c_3 y_3$

$\mathcal{D} \begin{cases} a_1 y_1 + a_2 y_2 + a_3 y_3 \geq a \\ b_1 y_1 + b_2 y_2 + b_3 y_3 \geq b \end{cases}$ $\quad y_i \geq 0$

• sous forme standard :

dual : min $z' = [\dots]$ (inchangé)

contraintes $\leq a$ si min primal
$\geq a$ si max

max : $\sum a_{ij} x_j \leq c_i \rightarrow y_i \geq 0$
(TD3 exo 20) $\quad = c_i \rightarrow y_i \in \mathbb{R}$
$\geq c_i \rightarrow (-y_i) \geq 0$

min : $\leq c_i \rightarrow (-y_i) \geq 0$
$= c_i \rightarrow y_i \in \mathbb{R}$
$\geq c_i \rightarrow y_i \geq 0$

Th des écarts complémentaires :

x^* sol opti $\mathcal{P}$, y^* $\mathcal{D}$
si on connait x^*, il est possible d'obtenir y^* (inversm) si et seulement si :

- x^* et y^* réalisables
- $y_i^* \left(\sum a_{ij} x_j^* - b_i \right) = 0$
- $x_j^* \left(\sum a_{ij} y_i^* - c_j \right) = 0$

① faire le dual
② vérifier x^* réalisable
③ pour les contraintes primal c_i qui sont non saturées (val x^* n'est pas égale au 2nd mb), alors les $y_i^* = 0$
④ pour $x_j^* > 0$, alors les contraintes dual c_j sont saturées (→ égalité)
⑤ calculer y^*
⑥ vérifier y^* réalisable

ex : $\mathcal{P}$: min $z = 2x_1 + 3x_2$ $\quad$ (TD4 exo 24 Q°2)

$\begin{cases} 2x_1 + x_2 \geq 3 \\ 2x_1 - x_2 \geq 5 \\ x_1 + 4x_2 \geq 6 \end{cases}$ $\quad x_i \geq 0$

$\mathcal{D}$: max $z' = 3y_1 + 5y_2 + 6y_3$

$\begin{cases} 2y_1 + 2y_2 + y_3 \leq 2 \\ y_1 - y_2 + 4y_3 \leq 3 \end{cases}$ $\quad y_i \geq 0$

$x^* = (3;1)$:

$\left.\begin{array}{l} 2 \times 3 + 1 = 7 \geq 3 \\ 2 \times 3 - 1 = 5 \geq 5 \\ 3 + 4 \times 1 = 7 \geq 6 \end{array}\right\}$ x^* est réalisable

(c_1) et (c_3) non saturées donc $y_1^* = 0$ et $y_3^* = 0$

comme $x_1^* > 0$ et $x_2^* > 0$ alors les contraintes 1 et 2 du dual doivent être saturées :

$\begin{cases} 2y_1^* + 2y_2^* + y_3^* = 2 \\ y_1^* - y_2^* + 4y_3^* = 3 \end{cases}$

OR $y_1^* = 0$ et $y_3^* = 0$

donc $\begin{cases} 2y_2^* = 2 \\ -y_2^* = 3 \end{cases}$

$\Leftrightarrow \begin{cases} y_2^* = 1 \\ y_2^* = -3 \end{cases}$ impossible

Donc x^* non optimale

si chgt du problème comme nouvelle variable dans primal

$\Rightarrow$ TD4 exo 25 Q°3

Branch-and-Bound pour PLNE :

si soluto du PL $\bar{x} \notin \mathbb{N}$

① séparer $\bar{x}_k = \min_i (\bar{x}_i) \notin \mathbb{N}$

$\rightarrow$ ajout de l'une ou l'autre contrainte :

$x_k \leq \lfloor \bar{x}_k \rfloor$ ou $x_k \geq \lceil \bar{x}_k \rceil$

② calculer la valeur des versions relaxées des 2 sous-prob corresp aux 2 nouv sommets

③ si nouv $\bar{x}$ pas entière, retourner en ① en continuant le sommet le plus prometteur

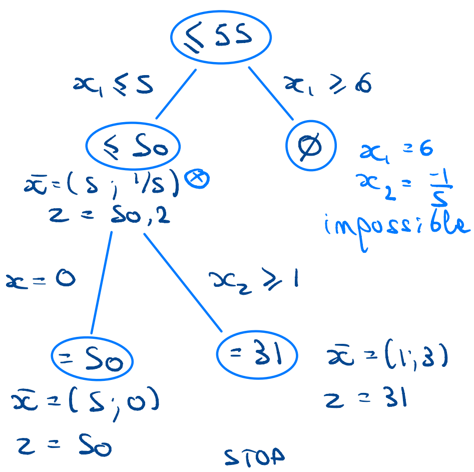
ex: TD5 exo 31 Q°1

max $z = 10x_1 + x_2$

$2x_1 + 5x_2 \leq 11 \qquad x_i \in \mathbb{N}$

$\bar{x} = (11/2 \,;\, 0) \qquad z = 55$

$\bar{x} = (5.5 \,;\, 0)$



Donc OPT : $\bar{x} = (5;0)$, $z = 50$

⊛ $x_1 = 5$

$(c_1) \Rightarrow 2 \times 5 + 5x_2 \leq 11$

$5x_2 \leq 11 - 10$

$x_2 \leq \frac{1}{5}$

PL binaire :

- exclusion mutuelle :
  un seul parmi $\{p_1, p_2, p_3\}$
  $\rightarrow x_1 + x_2 + x_3 = 1$
- cardinalité : pas plus de 2
  $\rightarrow x_1 + x_2 + x_3 \leq 2$
- si p_1, alors $p_3 \rightarrow x_1 \leq x_3$
- si p_1, alors p_2 et p_3
  $\rightarrow x_1 \leq x_2$ et $x_1 \leq x_3$
- p_2 ssi $p_4 \rightarrow x_2 = x_4$
- p_1 xor $p_4 \rightarrow x_1 + x_4 = 1$
- si p_1 et p_4, alors p_2
  $x_1 + x_4 - 1 \leq x_2$
- si $f(x) < a$ ou $g(x) < b$
  $\rightarrow f(x) \leq a + yM$
  $g(x) \leq b + kM$
  $y + z \leq 1 \quad (y, z \in \{0;1\})$

① soluto de la relaxation continue comme borne

② ordonner les variables selon le rapport v_i / w_i

v_i : coeff fct° obj de x_i

w_i : coeff contraintes pour x_i

③ affecter les variables dans l'ordre indit en ②

④ saturer les contraintes (la dernière variable peut être fractionnaire)

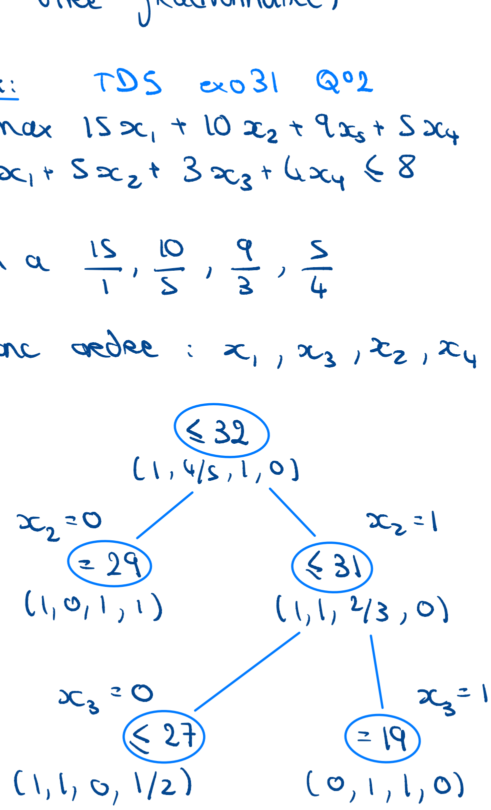
ex: TD5 exo31 Q°2

max $15x_1 + 10x_2 + 9x_3 + 5x_4$

$x_1 + 5x_2 + 3x_3 + 4x_4 \leq 8$

on a $\frac{15}{1}, \frac{10}{5}, \frac{9}{3}, \frac{5}{4}$

donc ordre : x_1, x_3, x_2, x_4



(car 29 > 27)

Donc OPT : $(1,0,1,1) \rightarrow 29$

Dijkstra : (poids positifs)

$G = (V, E)$ $\quad V$ ens sommets

E ens arcs

S ens sommets déjà explorés

$\lambda^*(u)$: distance opti de s à u

s : sommet source

init : $S = \{s\}$, $\lambda^*(s) = 0$

tant q $S \neq V$:

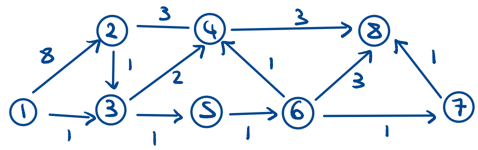
. choisir un sommet non-explo v minimisant $\lambda(v)$ tq
$\lambda(v) = \min(\lambda^*(u) + d(u,v))$ ⊛
où $u \in S$

. Rajouter v à S

. poser $\lambda^*(v) = \lambda(v)$

⊛ plus court chemin à un sommet u dans graphe explo, suivi d'un arc uniq (u, v)

ex: TD6 exo37 Q°1



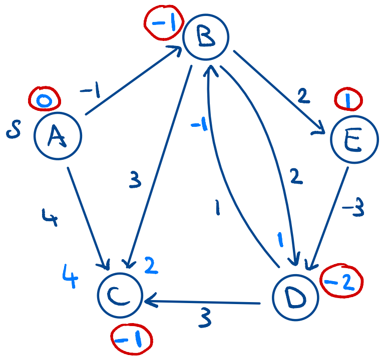
k		1	2	3	4	5	6	7	8
init	coût	0	∞	∞	∞	∞	∞	∞	∞
	père	–	/	/	/	/	/	/	/
1		–	8	1	∞	∞	∞	∞	∞
		–	1	1	/	/	/	/	/
2		–	8	–	3	2	∞	∞	∞
		–	1	–	3	3	/	/	/
3		–	8	–	3	–	3	∞	∞
		–	1	–	3	–	5	/	/
4		–	6	–	–	–	3	∞	6
		–	4	–	–	–	5	/	4
5		–	6	–	–	–	–	4	6
		–	4	–	–	–	–	6	4
6		–	–	–	–	–	–	–	5
		–	4	–	–	–	–	–	7
7		–	6	–	–	–	–	–	–
		–	4	–	–	–	–	–	–

Bellman-Ford : (poids négatifs)

calcule du plus cour chemin de s vers tout $v \in S$ ou déclare l'existence d'un cycle de poids négatifs ou cycle absorbant algo Bellman-Ford :

pour chq $u \in V$:
$\lambda(0,u) = +\infty$
$\lambda(0,s) = 0$
pour k de 1 à n :
pour $\forall u \in V$:
$$\lambda(k+1,u) = \min\{ \lambda(k,u), \min_{(v,u)\in E} (\lambda(k,v) + d(v,u)) \}$$
si $\lambda(n,v) < \lambda(n-1,v)$
alors circuit absorbant
sinon $\lambda(n-1,u)$ = valeur d'un PCCH de s à u

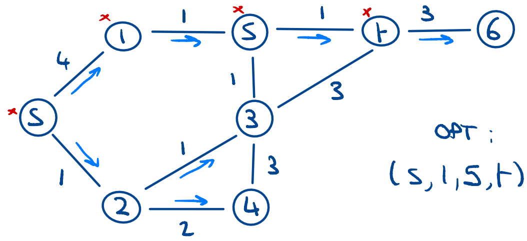
ex: cours 6 p.39



k	A	B	C	D	E
init	0	∞	∞	∞	∞
	–	/	/	/	/
1	0	−1	4	∞	∞
	–	A	A	/	/
2	0	−1	2	1	1
	–	A	B	B	B
3	0	−1	2	−2	1
	–	A	B	E	B
4	0	−1	1	−2	1
	–	A	D	E	B
5	0	−1	1	−2	1
	–	A	B	E	B

Parcours en largeur (BFS) :

$L_0 = \{s\}$
L_1 = tous les voisins de L_0
L_2 = tous les sommets n'appartenant ni à L_0 ni à L_1, et ayant une arête vers un sommet de L_1
L_{i+1} = tous les sommets n'appartenant pas au niveau précédent, et ayant une arête vers un sommet de L_i



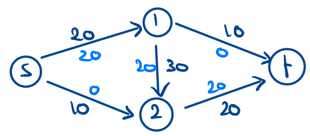
OPT : (s,1,5,t)

k	parcours	frontière
init	∅	s
1	s	1,2
2	[s],1	2,5
3	[s,1],2	5,3,4
4	[s,1,2],5	3,4,t
5	[s,1,2,5],3	4,t
6	[s,1,2,5,3],4	t
7	[s,1,2,5,3,4],t	6
8	[s,1,2,5,3,4,t],6	∅

Ford-Fulkerson : flot max / coupe min

• Constructᵒ d'un flot complet :
algo glouton (opt local)
① $f(e) = 0 \quad \forall e \in E$
② déterminer un chemin P ($s \to t$) tq $\forall e \in P$, $f(e) < c(e)$
③ augmenter le flot le long du chemin P
④ répéter jusqu'à qu'on ne puisse plus trouver de chemin

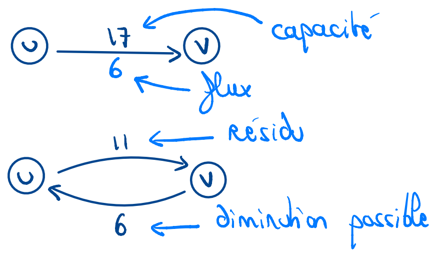
ex:



flot = 20
(opt = 30)

• graphe d'écart :
à partir d'un flot f d'un graph G, construire ∀ les possibilités de modif des flux (↗ ou ↘)

ex:



graphe d'écart long à faire
⇒ faire un parcours implicite directem dans G
(choisir un flot arbitraire puis vérifier s'il est max)

• déterminer si flot max :
① marquer les sommets où on peut augmenter le flux (+, sommet père) ou diminuer le flux (−, père)
② si t est marqué, alors on peut ↗ le flot, sinon flot max et $C(S,T)$ une coupe min (S = sommets marqués)
③. vérification par calcul :
capacité coupe $C = \sum_e c(e)$
où $e \in$ arcs sortants de C
flot $= \sum_a f(a)$
où $a \in$ flux arrivant vers t
si capacité = flot
alors flot max (coupe min)

• avec le graphe d'écart :
flot max si dans graphe d'écart, il n'y a pas de chemin $s \to t$

Programmation dynamique :

① décomposer le problème en une seq de ss-prob indicés $k = 0, \dots, t-1$
② pour chq ss-prob :
(2.1) ens S de situatᵒ/états possibles
(2.2) calculer $\lambda(k,v)$ $v \in S$: meilleure manière de parvenir dans l'état v du ss-prob k

étape ① : décompositᵒ tq
- on sait initialiser $\lambda(0,.)$
- on a une relatᵒ de récurrence permettant de calculer $\lambda(k,.)$ à partir de $\lambda(k-1,.)$ ou des $\lambda(i,.)$ avec $i < k$
- calcul des $\lambda(k,v)$ permet de répondre au problème initial

Méthode hongroise : (affectatᵒ)

matrice C représentant les états du problème (graphe biparti)
(algo primal-dual = hongrois)
① à partir de C, construire C' en soustrayant à chq ligne son élément minimum
② à partir de C', construire C'' en soustrayant à chq colonne son élément minimum

⇒ donne "graphe des paires nulles"

algo hongrois suite :

③ chercher un flot max dans le "graphe des paires nulles" avec Ford-Fulkerson à partir du flot de l'itérat° précédente

④ . si flot max = n, alors couplage parfait, soit affectation optimale

. sinon

soit S_L et S_C ens sommets (lignes et colonnes) marqués à la dernière étape

soit δ min éléments à l'intersect° d'une ligne marquée et d'une colonne non marquée

(4.1) faire $-\delta$ sur les lignes de S_L

(4.2) faire $+\delta$ sur les colonnes de S_C

⑤ mettre à jour le graphe et itérer

ex : TD9 exo 55

	BD	IT	MO	AL	
I1	8	1	5	5	(-1)
I2	5	6	4	11	(-4)
I3	6	4	3	9	(-3)
I4	3	3	7	7	(-3)

①

7	0	4	4
1	2	0	7
3	1	0	6
0	0	4	4

(-4)

②

7	0	4	0
1	2	0	4
3	1	0	3
0	0	4	0

③ coupe : (S, I_2, I_3, M)

flot = 3 ≠ 4

=> flot non max

④

	BD	IT	MO	AL	
I1	7	0	4	0	(4.1)
I2	1	2	0	4	(-1)
I3	3	1	0	3	(-1)
I4	0	0	4	0	
	(4.2)	(+1)			

Busacker - Gowen : (transport)

reproduire Ford-Fulkerson, mais en prenant à chq étape le chemin augmentant le coût minimum

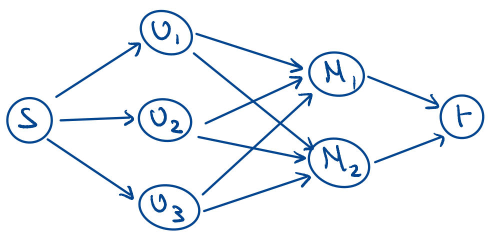
① acheminer une ou plus unités de flot avec un coût unitaire minimum

② itérer jusqu'à obtenir flot max

③ montrer que le coût total est minimum

ex : TD9 exo 56

	M1	M2	Qté dispo
U1	12	3	30
U2	14	4	40
U3	15	8	20
besoin	30	60	

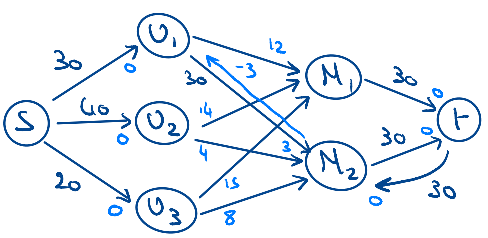


(1) : $S \xrightarrow{30} U_1 \xrightarrow{30} M_2 \xrightarrow{30} t$

coût = 3   flot = 30

f = 30   c = 3 × 30 = 90

graphe d'écart :



(2) : $S \to U_2 \to M_2 \to t$

coût = 4   flot = 30

f = 30 + 30 = 60

c = 90 + 4 × 30 = 210

(3) : $S \to U_2 \to M_2 \to U_1 \to M_1 \to t$

coût = 13   flot = 10

f = 60 + 10 = 70

c = 210 + 13 × 10 = 340

(4) : $S \to U_3 \to M_2 \to t$

coût = 15   flot = 20

f = 90   c = 640   STOP