Corrigé sujet Programmation Linéaire Session Mai 2017

Exercice 1

$$1\begin{cases} x+2y-z=5 & \color{red}{(1)} \\ 3x+y+2z=10 & \color{red}{(2)} \\ 2x-2y-3z=-10 & \color{red}{(3)} \end{cases} \quad \Rightarrow \quad 2\begin{cases} x+2y-z=5 & \color{red}{(1)} \\ 0x-5y+5z=-5 & \color{red}{(2)-3(1)} \\ 0x-6y-z=-20 & \color{red}{(3)-2(1)} \end{cases}$$

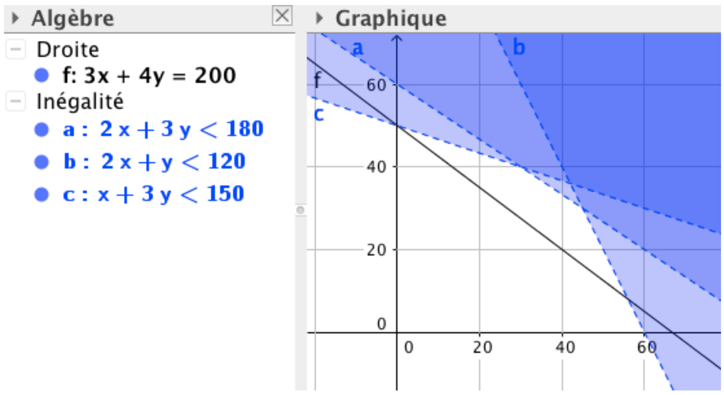
$$3\begin{cases} x+2y-z=5 & \color{red}{(1)} \\ 0x+y-z=1 & \color{red}{-(2)/5} \\ 0x+0y-35z=-70 & \color{red}{5(3)-6(2)} \end{cases} \quad \Rightarrow \quad 4\begin{cases} 1x+0y+0z=1 \\ 0x+1y+0z=3 \\ 0x+0y+1z=2 \end{cases}$$

Exercice 2

Soit *x* la quantité de potion A et *y* la quantité de potion B.

On a la PL suivant et sa représentation graphique :

$$\begin{cases} Max\ \ 30x \ +40y \\ 2x \ +3y \ \leq 180 \ \ (larmes) \\ 2x \ +1y \ \leq 120 \ \ (pattes) \\ x \ +3y \ \leq 150 \ \ (yeux) \\ \quad x\ et\ y \geq 0 \end{cases}$$



On ajoute 3 slacks variables et on obtient :

	x	y	s_1	s_2	s_3	
	-2	-3	1			180
	-2	-1		1		120
	-1	-3			1	150
	30	40				0

La base est (s_1,s_2,s_3), la solution initiale en (x,y) est (0,0), la fonction de coût vaut 0.

La plus grand coût réduit est 40 pour y, le plus petit pivot -(b_j/a_{kj}) est 50 pour s_3. On pivote

	x	s_3	
s_1	-1	1	30
s_2	-5/3	1/3	70
y	-1/3	-1/3	50
	50/3	-40/3	2000

La base est (s_1,s_2,y), la solution est (0,50) avec un coût de 2000. On pivote sur x avec s_1.

	s_1	s_3	
x	-1	1	30
s_2	5/3	-4/3	20
y	1/3	-2/3	40
	-50/3	10/3	2500

La base est (x,s_2,y), la solution est (30,40) avec un coût de 2500. On est au niveau 10 mais on peut encore faire mieux, on pivote sur s_3 avec s_2 !

	s_1	s_2	
x	¼	-3/4	45
s_3	5/4	-3/4	15
y	-1/2	1/2	30
	-25/2	-5/2	2550

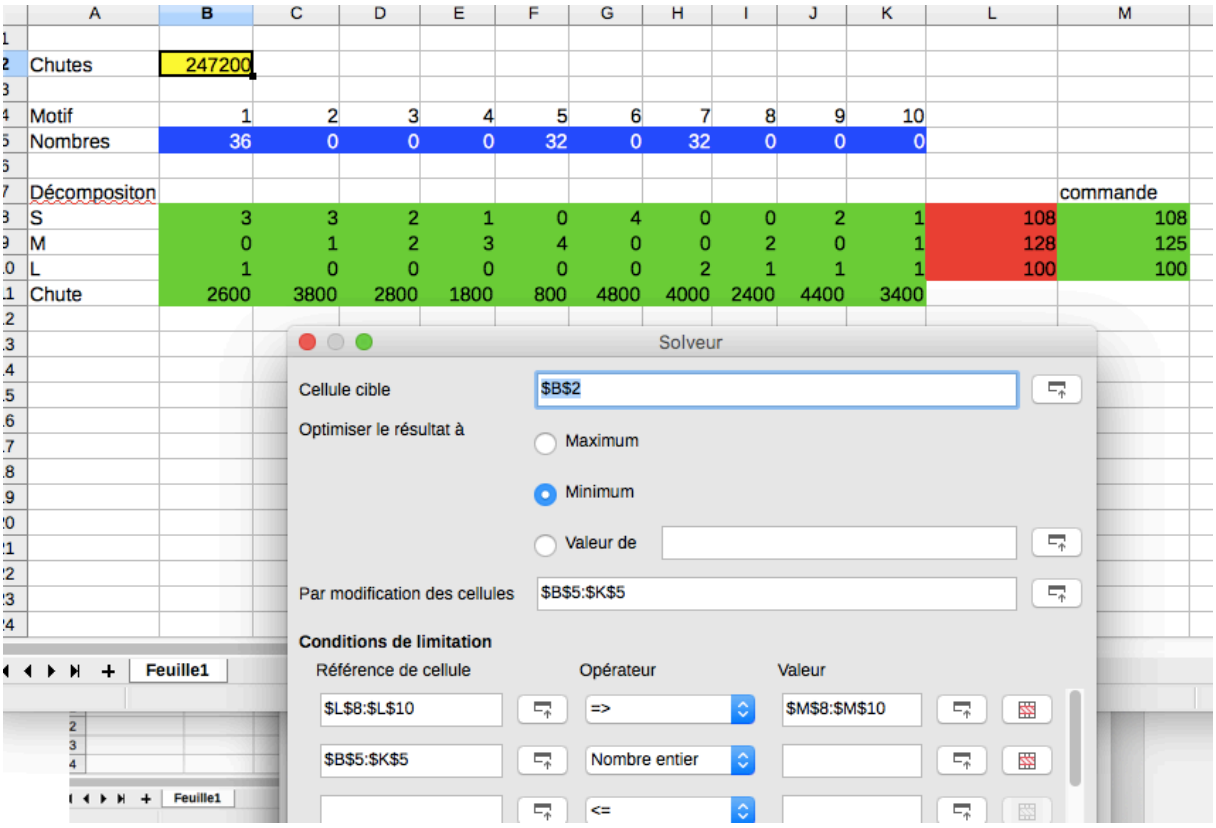
La solution optimale en (45,30) fournie 2550 XP (tous les coûts réduits sont négatifs).

Pour atteindre le niveau 9, il manque 500 points d'XP, soit 17 potions A ; En posant x>17, on voit que l'on trouve le même optimum, mais le démarrage de la résolution se fait avec les deux phases en ajoutant une variable artificielle.

Exercice 3

On voit que l'on va découper les grandes planches en 4. Chaque « coin » pouvant avoir 3 tailles. Nous avons donc 4x3=12 variables par planche à découper.
Pour approximer leur nombre, nous avons 27 (108/4 pour S) + 32 (125/4 M) + 100 (L) = 159 planches. Ce qui fait au pire 12x159 = 1908 variables 0/1
Remarque : Il est possible de réduire ce nombre de variables grâce aux symétries.
Il faut créer 4 équations pour respecter la taille en ligne et en colonnes de la planche.
Au lieu d'ajouter une variable de présence de la planche, on minimise la somme totale, mais ce programme linéaire sera lourd.
De plus nous n'avons pas pris en compte les rotations possibles.

Nous n'avons que peu motifs possibles (indépendants des symétries) : 1L+3S ; 1M+3S ; 2M+2S ; 3M+1S ; 4M et 4S si on n'autorise pas les rotations. Il faut ajouter 2L, 1L+2M, 1L+2S et 1L1M1S avec rotations. Il suffit de créer une variable par motifs (entières) et de poser les 3 contraintes de production. La fonction de coût dépend de la surface des chutes.



	A	B	C	D	E	F	G	H	I	J	K	L	M
2	Chutes	247200											
4	Motif	1	2	3	4	5	6	7	8	9	10		
5	Nombres	36	0	0	0	32	0	32	0	0	0		
7	Décompositon												commande
8	S	3	3	2	1	0	4	0	0	2	1	108	108
9	M	0	1	2	3	4	0	0	2	0	1	128	125
10	L	1	0	0	0	0	0	2	1	1	1	100	100
11	Chute	2600	3800	2800	1800	800	4800	4000	2400	4400	3400		

On voit que seuls 3 motifs sont nécessaires (peu de pivotages à la main sont nécessaires sur la matrice 3x10) comme l'avait expliqué monsieur Sevaux (base 3x3 quelle que soit le nombre de motifs !)