

## Installation d'entrepôts

Une entreprise s'installant en France et y créant un réseau de distribution souhaite y construire 4 entrepôts. Elle a pour l'instant retenu 14 emplacements possibles et estimé pour chacun d'entre eux le coût de la construction et de l'installation. Ces coûts, exprimés en millions d'Euros sont résumés dans la table ci-dessous.

Ville	Coût	Ville	Cout	Ville	Coût
Bordeaux	32	Clermont	12	Dijon	18
Limoges	14	Lyon	34	Marseille	28
Montelimar	22	Montpellier	27	Orleans	15
Paris	40	Poitiers	16	Reims	17
Toulouse	24	Troyes	13		

L'entreprise souhaite minimiser le coût total d'installation et le problème serait donc enfantin si elle ne souhaitait aussi que les sites choisis soient suffisamment distants les uns des autres (pour bien couvrir le territoire). Elle s'interdit ainsi que dans la solution, certains sites puissent être choisis conjointement. La liste suivante récapitule ainsi tous les couples de sites qui ne peuvent pas être choisis ensemble (car trop proches).

*(Bordeaux, Limoges), (Bordeaux, Toulouse), (Clermont, Limoges), (Clermont, Lyon), (Clermont, Poitiers), (Dijon, Lyon), (Dijon, Troyes), (Dijon, Reims), (Limoges, Poitiers), (Lyon, Montelimar), (Marseille, Montelimar), (Marseille, Montpellier), (Montelimar, Montpellier), (Montpellier, Toulouse), (Orleans, Paris), (Orleans, Troyes), (Orleans, Reims), (Orleans, Poitiers), (Paris, Troyes), (Reims, Troyes).*

1. Représenter les incompatibilités entre villes sous forme d'un graphe.
2. Modéliser mathématiquement le problème d'optimisation en utilisant des variables  $x_i$  indiquant si un entrepot est construit ou non sur le site  $i$ . Les couts sont notés  $c_i$  et les incompatibilités constituent l'ensemble  $I$ .
3. Quelles relaxations pouvez-vous envisager pour trouver une borne ? Au fait s'agit-il de trouver une borne inférieure ou une borne supérieure ?
4. Résoudre le problème de manière exacte en utilisant une méthode Branch and Bound.

Minimize

obj:  $32b + 12c + 18d + 14\text{ li} + 34\text{ ly} + 28\text{ ma} + 22\text{ mo} + 27\text{ mt} + 15\text{ o} + 40\text{ pa} + 16\text{ po} + 17\text{ r} + 24\text{ t} + 13\text{ tr}$

Subject To

c1:  $b + \text{li} \leq 1$

c2:  $b + \text{t} \leq 1$

c3:  $c + \text{li} \leq 1$

c4:  $c + \text{ly} \leq 1$

c5:  $d + \text{ly} \leq 1$

c6:  $d + \text{tr} \leq 1$

c7:  $d + \text{r} \leq 1$

c8:  $c + \text{po} \leq 1$

c9:  $\text{li} + \text{po} \leq 1$

c10:  $\text{mo} + \text{ly} \leq 1$

c11:  $\text{ma} + \text{mo} \leq 1$

c12:  $\text{ma} + \text{mt} \leq 1$

c13:  $\text{mo} + \text{mt} \leq 1$

c14:  $\text{mt} + \text{t} \leq 1$

c15:  $\text{o} + \text{pa} \leq 1$

c16:  $\text{o} + \text{tr} \leq 1$

c17:  $\text{o} + \text{r} \leq 1$

c18:  $\text{r} + \text{tr} \leq 1$

c19:  $\text{pa} + \text{tr} \leq 1$

c20:  $b+c+d+\text{li}+\text{ly}+\text{ma}+\text{mo}+\text{mt}+\text{o}+\text{pa}+\text{po}+\text{r}+\text{t}+\text{tr} = 4$

c21:  $\text{o} + \text{po} \leq 1$

Bounds

$0 \leq b \leq 1$

$0 \leq c \leq 1$

$0 \leq d \leq 1$

$0 \leq \text{li} \leq 1$

$0 \leq \text{ly} \leq 1$

$0 \leq \text{ma} \leq 1$

$0 \leq \text{mo} \leq 1$

$0 \leq \text{mt} \leq 1$

$0 \leq \text{o} \leq 1$

$0 \leq \text{pa} \leq 1$

$0 \leq \text{po} \leq 1$

$0 \leq \text{r} \leq 1$

$0 \leq \text{t} \leq 1$

$0 \leq \text{tr} \leq 1$

End

Problem:
Rows: 21
Columns: 14
Non-zeros: 54
Status: OPTIMAL
Objective: obj = 63.5 (Minimum)

Table with 6 columns: No., Row name, St, Activity, Lower bound, Upper bound, Marginal. It contains 21 rows of data for various activities (c1 to c21) and their associated constraints (b, c).

3	d	B	0.5	0	1	
4	li	B	0.5	0	1	
5	ly	NL	0	0	1	12
6	ma	NL	0	0	1	6
7	mo	B	0.5	0	1	
8	mt	NL	0	0	1	5
9	o	B	0.5	0	1	
10	pa	NL	0	0	1	18
11	po	B	0.5	0	1	
12	r	B	0.5	0	1	
13	t	NL	0	0	1	
14	tr	B	0.5	0	1	2

Karush-Kuhn-Tucker optimality conditions:

KKT.PE: max.abs.err = 0.00e+00 on row 0  
max.rel.err = 0.00e+00 on row 0  
High quality

KKT.PB: max.abs.err = 0.00e+00 on row 0  
max.rel.err = 0.00e+00 on row 0  
High quality

KKT.DE: max.abs.err = 0.00e+00 on column 0  
max.rel.err = 0.00e+00 on column 0  
High quality

KKT.DB: max.abs.err = 0.00e+00 on row 0  
max.rel.err = 0.00e+00 on row 0  
High quality

End of output

