# Cours de méthodes de la recherche opérationnelle

Mini-projet :

*Heuristiques pour la résolution des problèmes combinatoires difficiles*

## I. Le problème du voyageur de commerce :

### *1. Question préliminaire :*

On effectue un cycle pour le ramassage, ce qui nous permet de choisir le sommet de départ. De ce fait, il n'y a plus que 5 éléments à prendre en compte pour la permutation.
**On obtient donc 5! Permutations, ce qui nous fait 120 permutations.**

### *2. Cœur du problème :*

$$Min \sum_{i \in \chi} \sum_{j \in \chi} Cij\ Xij$$
$$\sum_{j \in \chi} xij = 1, \forall i \in \chi$$
$$\sum_{i \in \chi} xij = 1, \forall j \in \chi$$
$$xij \geq 0, \forall i,j \in \chi$$

Dans notre cas on a nombre de ville égale à 6, ce qui nous donne une matrice de 12 lignes et de 36 colonnes.

**Déterminons le rang**

Matrice de 12 lignes et de 36 colonnes = *matrice problème*

Le rang sera au plus égale au nombre de lignes, soit 12.

| | | |
|---|---|---|
| 1 1 1 1 1 1 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 |
| 0 0 0 0 0 0 1 1 1 1 1 1 | 0 0 0 0 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 |
| 0 0 0 0 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 |
| 0 0 0 0 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 1 1 1 1 1 1 | 0 0 0 0 0 0 0 0 0 0 0 0 |
| 0 0 0 0 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 0 0 0 0 0 0 |
| 0 0 0 0 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 1 1 1 1 1 1 |
| 1 0 0 0 0 0 1 0 0 0 0 0 | 1 0 0 0 0 0 1 0 0 0 0 0 | 1 0 0 0 0 0 1 0 0 0 0 0 |
| 0 1 0 0 0 0 0 1 0 0 0 0 | 0 1 0 0 0 0 0 1 0 0 0 0 | 0 1 0 0 0 0 0 1 0 0 0 0 |

De plus,

$$ligne\ 1 = \frac{\sum_{i=7}^{12} ligne\ i}{6} - \sum_{i=2}^{6} ligne\ i$$

**Donc le rang est au plus de 11**

Vérifions que le rang est exactement de 11 :

Pour ce faire, nous allons calculer $\sum_{i=1}^{11} lambda\ i * Li = 0$

Li= ligne i de la matrice (11,36) = Matrice A (c'est la matrice que l'on a construit en enlevant la première ligne de la *matrice problème* car elle est combinaison des autres)

Avec lambda i ≠ lambda j  quelque soit i et j

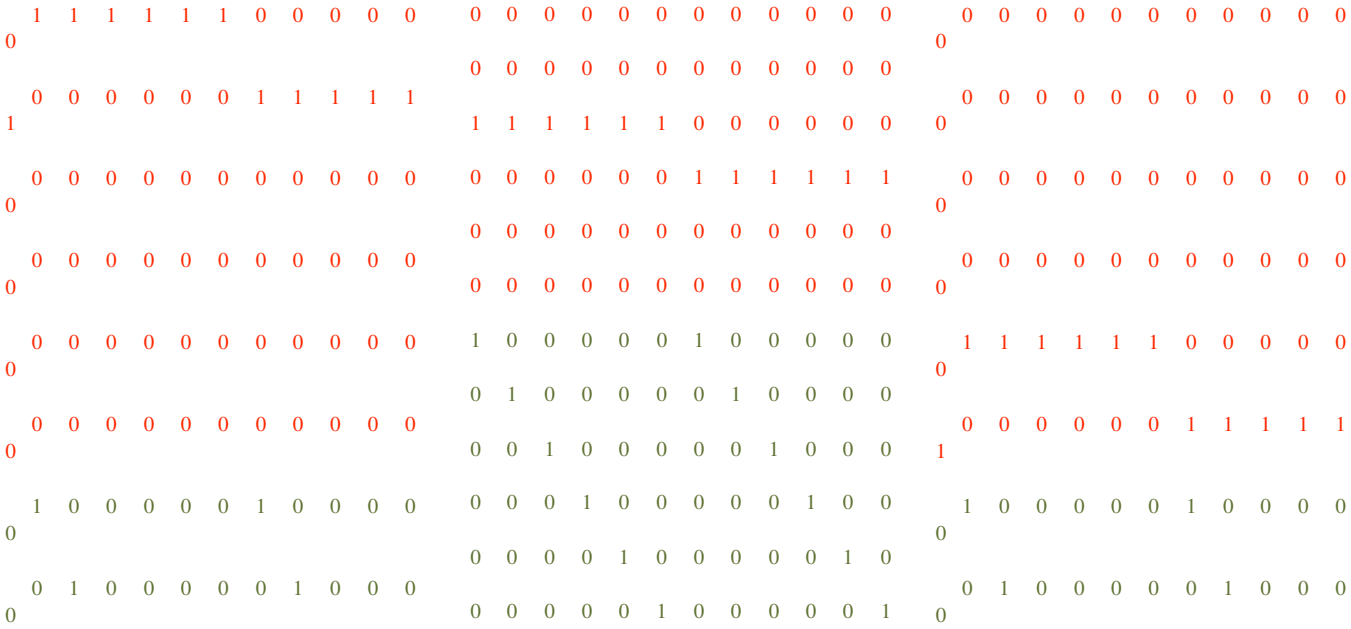
On trouve tous les lambda i nul, donc la matrice A est libre.

**Ainsi, la *matrice problème* est de rang exactement égale à 11.**

Le déterminant des sous-matrices est un déterminant calculé en extrayant un nombre de lignes et de colonnes, différente, inférieur où égal au rang de la matrice.

**Justification de la valeur du sous-déterminant : -1, 0, ou 1**

Nous avons notre *matrice problème* (12,36), de rang 11.



**Pour calculer les sous déterminants de cette matrice, nous devons extraire de la matrice problème une matrice de taille (11,11).**
Pour ce faire, nous allons choisir 11 lignes parmi les 12 :
exemple : lignes 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12  On obtient :

Maintenant, nous allons choisir 11 colonnes parmi les 36 :
Par exemple : colonne 1, 3, 7, 9, 12, 14, 16, 20, 25, 33, 35, on obtient :

1 1 0 0 0 0 0 0 0 0 0

0 0 1 1 1 0 0 0 0 0 0

0 0 0 0 0 1 1 0 0 0 0

0 0 0 0 0 0 0 1 0 0 0

0 0 0 0 0 0 0 0 0 1 1

1 0 1 0 0 0 0 0 1 0 0

0 0 0 0 0 1 0 1 0 0 0

0 1 0 1 0 0 0 0 0 1 0

En rouge = partie haute
En vert = partie base

**Calcul du sous-déterminant :**

Nous avons construit des sous déterminants de taille (11,11) [méthode de construction ci dessus].
Mais en fait le **sous déterminant est un déterminant d'une matrice extraite de la *matrice problème* de taille (n,n) avec n≤11.**
Pour cette matrice extraite de taille (n,n), définissons Lh et Lb :

**On somme toutes les lignes de la partie haute sur une ligne = Lh** (partie haute = partie verte dans la *matrice problème*)
**On somme toutes les lignes de la partie basse sur une ligne = Lb** (partie rouge dans la *matrice problème).*

Partant de la matrice (n,n) :

1er cas : Il y a un élément de Lh qui est nulle.

*1er sous cas :* L'élément de Lb qui est dans la même colonne que l'élément de Lh n'est pas nul. Que l'on ait un élément de Lb nul où non (autre que dans la colonne de l'élément nul de Lh), nous avons au moins une colonne qui ne contient que un seul « 1 » **(et pas de colonne nulle)**.
On développe par rapport à cette colonne. On obtient (-1)^(i+j)det(matrice(n-1,n-1)).
Où i et j sont les numéro de ligne et de colonne de cet éléments dans notre matrice (n,n).

*2ieme sous cas :* L'élément de Lb qui est dans la même colonne que l'élément de Lh est aussi nul.
Nous avons donc une colonne ne contenant que des zéros.
**Le déterminant est donc nul.**

2ieme cas : Aucun élément de Lh n'est nul.

*1er sous cas :* 1 élément de Lb est nul.

Cela signifie que nous avons une colonne qui contient qu'un seul « 1 ».
On développe par rapport à cette colonne. On obtient (-1)^(i+j)*det(matrice(n-1,n-1)).
Où i et j sont les numéros de ligne et de colonne de cet éléments dans notre matrice (n,n).

*2ieme sous cas :* Aucun élément de Lb n'est nul.

*1er sous-sous cas :* il y a une ligne qui ne contient qu'un seul « 1 ». On développe par rapport à cette ligne. On obtient (-1)^(i+j)det(matrice(n-1,n-1)).
Où i et j sont les numéro de ligne et de colonne de ce « 1 » dans notre matrice (n,n).

*2ieme sou-sous cas :* Toutes les lignes on au moins deux éléments « 1 », ainsi que toutes les colonnes.
D'après la construction de notre matrice (dont on se sert pour le calcul du sous déterminant) Lh=Lb ;
La matrice est donc liée. **Le déterminant est donc nul.**

Pour les sous cas où on a calculé (-1)^(i+j)det(matrice(n-1,n-1)), on réapplique les cas à la matrice(n-1,n-1).

Bilan : On obtient deux types de valeur pour les sous déterminants :

**Déterminant nul**

Déterminant de la forme (-1)^k où k est un entier naturel.

## Conclusion : Tout sous déterminant de la matrice est égal à -1,0, ou 1

## Justification : toute solution en nombre entiers de (PL) correspond à une permutation σ : xij=1 si et seulement si j=σ(i).

Une permutation σ solution impose que l'on parcourt le cycle ABCDEF une et une seule fois (avec deux boucles ou non).
De plus les xij ne peuvent être qu'égale à 1 ou à 0 : soit on va de i vers j ; soit on n'y va pas.
Enfin, les contraintes imposent que pour un « j » fixé, il n'y a que un seul « i » qui corresponde à xij=1. De même, pour un « i » fixé, il n'y a que un seul « j » qui corresponde à ce que xij=1. Ainsi, avec les contrainte, **on aura seulement 6 xij non nul, ce qui correspond aussi au nombre de permutation**.
Si on a xij=1, cela signifie que l'on va de i vers j. Ce qui impose σ(i)=j.
Si on a σ(i)=j, cela signifie que l'on parcourt le cycle en passant de i à j, et donc xij=1.

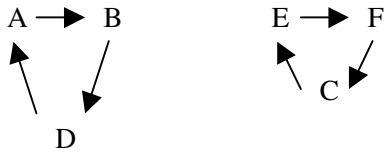
**On a donc bien xij=1 si et seulement si σ(i)=j.**

## Résolvons le problème en utilisant maple :

Ayant chargé la librairie simplex, on résout avec le programme minimize qui fonctionne de cette façon :
minimize (fonction linéaire à réduire, {équation des contraintes}, NONNEGATIVE car les xij sont positif). (cf programme en annexe)

On trouve donc :
x12=1
x24=1
x35=1
x41=1
x56=1
x63=1
Tous les autres xij sont nuls.
Le coût total est de 15.

Ce qui donne le cycle suivant :




On trouve deux boucles, ce qui pose problème : Il faudrait 2 voyageurs de commerce.
Il faut donc utiliser la séparation.

## *3. Séparation :*

La permutation consiste à imposer un ou plusieurs éléments de la permutation σ solution de (PL) : on fait une sorte de « forçage » en modifiant les coûts.

Dans notre exemple (que l'on va expliciter), on veut imposer σ(A)=E et σ(C)=B.

D'après la propriété suivante :

σ : xij=1 si et seulement si j=σ(i).

On a nécessairement xAE=1 et xCB=1, puisque l'on veut imposer σ(A)=E et σ(C)=B.
D'après les contraintes, on aura xAj=0 pour tous « j » différent de E, xCj=0 pour tout « j » différent de B, xiE=0 pour tout « i » différent de A, et xiB=0, pour tout « i » différent de C.

**Un moyen sûr d'annuler ces xij (et donc d'effectuer une séparation) est de leur associé des coûts infinis.** On a toujours les coûts des xii infinis puisque « qu'one ne va pas faire du sur-place ».
D'où le tableau des coûts ci-dessous :

| x\y | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | ∞ | ∞ | ∞ | ∞ | 7*(σ(A)=E)* | ∞ |
| B | 9 | ∞ | 7 | 4 | ∞ | 10 |
| C | ∞ | 9*(σ(C)=B)* | ∞ | ∞ | ∞ | ∞ |
| D | 2 | ∞ | 15 | ∞ | ∞ | 9 |
| E | 20 | ∞ | 12 | 8 | ∞ | 2 |
| F | 7 | ∞ | 3 | 9 | ∞ | ∞ |

Nous allons maintenant effectuer une séparation par ligne.
Etant donné que nous avons un cycle hamiltonien, nous choisissons arbitrairement de partir du point A.

**Pour la séparation qui contient l'arc (A-B), nous avons le tableau des coûts suivant :**

| x\y | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | ∞ | 1 | ∞ | ∞ | ∞ | ∞ |
| B | 9 | ∞ | 7 | 4 | 12 | 10 |
| C | 15 | ∞ | ∞ | 9 | 1 | 8 |
| D | 2 | ∞ | 15 | ∞ | 8 | 9 |
| E | 20 | ∞ | 12 | 8 | ∞ | 2 |
| F | 7 | ∞ | 3 | 9 | 7 | ∞ |

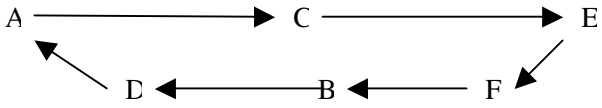
On obtient x41=x35=x63=x56=x24=x12=1 et tous les autres xij nul.
Coût total =15.
Représentation de l'arc :


Toujours 2 boucles indépendantes.

**Pour la séparation qui contient l'arc (A-C), nous avons le tableau des coûts suivant :**

| x\y | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | ∞ | ∞ | 8 | ∞ | ∞ | ∞ |
| B | 9 | ∞ | ∞ | 4 | 12 | 10 |
| C | 15 | 9 | ∞ | 9 | 1 | 8 |
| D | 2 | 12 | ∞ | ∞ | 8 | 9 |
| E | 20 | 10 | ∞ | 8 | ∞ | 2 |
| F | 7 | 6 | ∞ | 9 | 7 | ∞ |

On obtient x62=x56=x24=x41=x25=x13=1, et les autres xij nul.
Coût total =27
Représentation de l'arc :

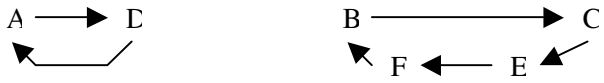



Nous avons un seul arc : ce qui permet concrètement de faire la tourné avec un seul véhicule.

**Pour la séparation qui contient l'arc (A-D), nous avons le tableau des coûts suivant :**

| x\y | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | ∞ | ∞ | ∞ | 10 | ∞ | ∞ |
| B | 9 | ∞ | 7 | ∞ | 12 | 10 |
| C | 15 | 9 | ∞ | ∞ | 1 | 8 |
| D | 2 | 12 | 15 | ∞ | 8 | 9 |
| E | 20 | 10 | 12 | ∞ | ∞ | 2 |
| F | 7 | 6 | 3 | ∞ | 7 | ∞ |

On obtient x14=x41=x23=x56=x62=x35=1, et les autres xij nul.
Coût total = 28
Représentation :

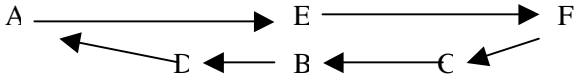



Deux boucles indépendantes.

**Pour la séparation qui contient l'arc (A-E), nous avons le tableau des coûts suivant :**

| x\y | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | ∞ | ∞ | ∞ | ∞ | 7 | ∞ |
| B | 9 | ∞ | 7 | 4 | ∞ | 10 |
| C | 15 | 9 | ∞ | 9 | ∞ | 8 |
| D | 2 | 12 | 15 | ∞ | ∞ | 9 |
| E | 20 | 10 | 12 | 8 | ∞ | 2 |
| F | 7 | 6 | 3 | 9 | ∞ | ∞ |

On obtient x24=x32=x41=x56=x63=x15=1 et tous les autres xij nul.
Coût total =27
Représentation :

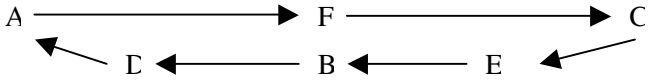



Une seule boucle.

**Pour la séparation qui contient l'arc (A-F), nous avons le tableau des coûts suivant :**

| x\y | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | ∞ | ∞ | ∞ | ∞ | ∞ | 9 |
| B | 9 | ∞ | 7 | 4 | 12 | ∞ |
| C | 15 | 9 | ∞ | 9 | 1 | ∞ |
| D | 2 | 12 | 15 | ∞ | 8 | ∞ |
| E | 20 | 10 | 12 | 8 | ∞ | ∞ |
| F | 7 | 6 | 3 | 9 | 7 | ∞ |

On obtient x63=x41=x24=x52=x16=x35=1 et les autres xij nul.
Coût total =29
Représentation :




Une seule boucle.

**Conclusion :** Notre but est d'effectuer une seule boucle.
Donc nous gardons les séparations qui contiennent les arcs : (A-C), (A-E), et (A-F).
De plus, nous choisissons l'arc de coût minimum.
Finalement, nous avons le choix entre les arcs (A-C, et (A-E), dont le coût total est de 27.

# II. La méthode colonie de fourmis :

## Résolution d'un exemple :

Nous avons pu récupérer un algorithme « ant colony » sur le site MATLAB Exchange, cependant il a fallu l'adapter pour un problème définissant des coûts et non seulement des positons de points. Les modifications apportées sont le plus souvent marquées par des commentaires.

*Variables utilisées :*

d = distance ( sous forme d'une matrice $c_{ij}$ )

t représente $\tau_{ij}$, le taux de phéromone

iter = nombre de cycle

m = nombre de fourmis

n = nombre de points considérés

e = coefficient d'évaporation

alpha = intensité

beta = effet sur la visibilité

h = matrice de visibilité : $\eta_{ij} = \frac{1}{d_{ij}}$

el = coefficient d'élimination du coût commun

app= c'est la position de chaque fourmi ( par ligne ).

*NB. Les feuilles Matlab sont données en annexes de la partie II.*

*Problème donné :*

l(a)=1500
l(b)=1460
l(c)=1580
l(d)=1620
L=5000

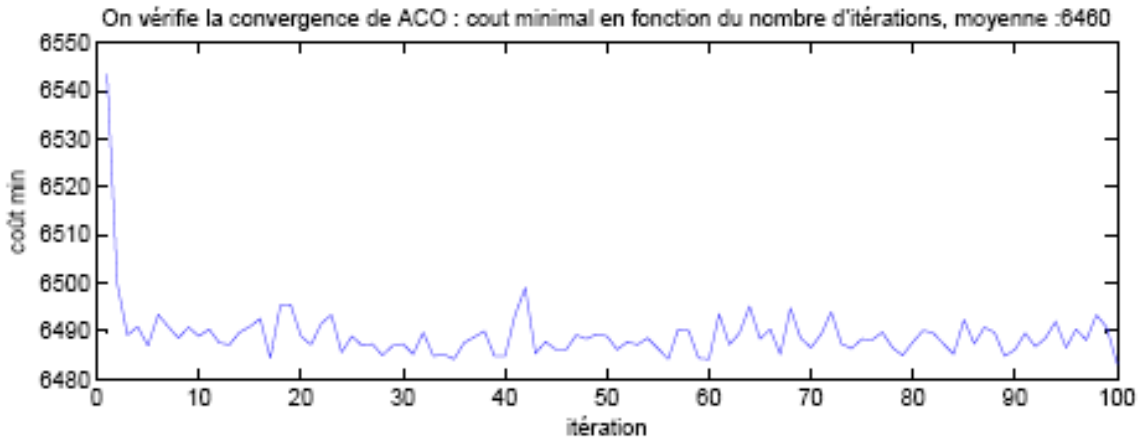
On veut montrer que notre fonction établit que le circuit le plus court est B+L.

On fournit la matrice suivante au programme :

```
cl =

     0     1500     1460     1580     1620
  5000        0        0        0        0
  5000        0        0        0        0
  5000        0        0        0        0
  5000        0        0        0        0
```

*Résultats :*




On peut effectivement montrer la convergence de la méthode avec le graphique.

Nous allons ensuite appliquer ce même programme pour chercher le circuit Hamiltonien optimum du problème, les données sont fournies dans le fichier *ant_informations.m* et notamment la matrice suivante :

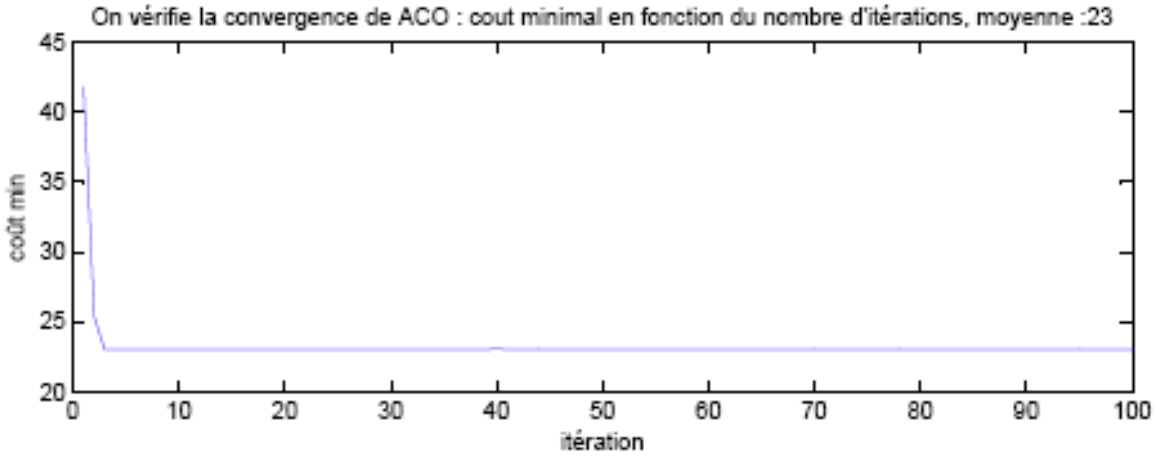
```
cl =

     0     3     8    10     7     9
     9     0     7     4    12    10
    15     9     0     9     1     8
     2    12    15     0     8     9
    20    10    12     8     0     2
     7     6     3     9     7     0
```

Les résultats montre la convergence vers la solution :

```
le circuit optimal est : 1   3   5   6   2   4   1
>>
```

Ce qui correspond à : A→C→E→F→B→D→A, pour un coût de 23.




## III. Ordonnancement avec contraintes :

Question 1 :

La résolution manuelle du problème nous donne :

| sommets | Coût des arcs en sens direct | Coût des arcs en sens indirect | Temps avant le début de la tâche |
|---|---|---|---|
| A | 0 | 44 | 0 (tâche critique) |
| B | 6 | 38 | 0 (tâche critique) |
| C | 5 | 37 | 2 |
| D | 20 | 22 | 2 |
| E | 20 | 24 | 0 (tâche critique) |
| F | 15 | 24 | 5 |
| G | 32 | 12 | 0 (tâche critique) |

| | | | |
|---|---|---|---|
| H | 25 | 16 | 2 |
| I | 44 (temps minimal) | 0 | 0 (tâche critique) |

Question 2 :

Si B et C ne peuvent être effectué en même temps on doit alors attendre la fin d'une des taches avant de commencer l'autre.

Si l'on commence par B, C ne pourra être effectué qu'à un temps 13.

De même si l'on commence par C, B ne pourra être commencé qu'à un temps 12.

On voit donc que ces changements influent directement sur la durée minimale totale du projet. La durée de 44 n'est donc plus valable.

Résolution du problème :

$$(1) \sum_{h \in Succ(s)} xsh = n - 1$$

$$(2) \sum_{k \in Pred(i)} xki - \sum_{h \in succ(i)} xih = 1, \forall i \in s$$

Montrons la redondance de l'équation (1) et (2) :

$$(2) \leftrightarrow \sum_{k \in Pred(i)} (xks + xsi) - \sum_{h \in succ(i)} (xis + xsh) = 1$$

$$\leftrightarrow \sum_{k \in Pred(i)} xks - \sum_{h \in succ(i)} xsh = 1$$

$$\leftrightarrow n - \sum_{h \in succ(i)} xsh = 1 \leftrightarrow (1)$$

$$\sum_{h \in Succ(s)} x^i_{sh} = 1$$

$$(3) \sum_{k \in Pred(i)} x^i_{kl} - \sum_{h \in Succ(i)} x^i_{ih} = 0, \quad \forall l \neq s, i$$

$$\sum_{k \in Pred(i)} x^i_{ki} - \sum_{h \in Succ(i)} x^i_{ih} = 1$$

La première s'explique par le fait qu'en partant du point initial il existe forcement un arc partant de ce point qui appartient au chemin le plus long.

La deuxième illustre le fait que lorsque le point l appartient au chemin le plus long on a forcement un unique prédécesseur et un unique successeur, ce qui nous fait 1 ôté à 1, soit 0 !

Dans le cas ou l n'appartient pas au chemin le plus long il n'existe pas de prédécesseur et encore moins de successeur, ce qui fait également 0.

La troisième est évidente puisque $\sum_{h \in Succ(i)} x_{ih}^{i} = 0$. Les arcs (i, h) ne peuvent appartenir au chemin de s à i !

Résolution du problème :

$$Max \sum_{(i,j) \in U} dijxij$$

$$\sum_{k \in Pred(i)} xki - \sum_{h \in Succ(i)} xih = 1, \quad \forall i \neq s$$

$$xij \geq 0 \quad \forall (i,j) \in U$$

Pour résoudre ce problème on choisit d'utiliser le tableur excel :

Au commencement il y avait ceci :

| (i,j) | dij | xij | xij*dij | | contraintes |
|---|---|---|---|---|---|
| A,B | 6 | | 0 | | 0 |
| A,C | 5 | | 0 | | 0 |
| B,E | 14 | | 0 | | 0 |
| B,F | 9 | | 0 | | 0 |
| C,D | 15 | | 0 | | 0 |
| C,E | 10 | | 0 | | 0 |
| D,G | 10 | | 0 | | 0 |
| E,G | 12 | | 0 | | 0 |
| E,H | 6 | | 0 | | 0 |
| F,H | 8 | | 0 | | 0 |
| G,I | 12 | | 0 | | 0 |
| H,I | 16 | | 0 | | 0 |
| | | somme | 0 | | |

Nue d'apparence, la colonne des xij représente la variable à trouver.

Les contraintes sont dépendantes de la colonne des xij.

Vint ensuite la merveille des merveilles : le solveur !

L'outil, saint Graal de la résolution de ce problème nous donne le résultat miracle suivant :

| (i,j) | dij | xij | xij*dij | | contraintes |
|---|---|---|---|---|---|
| A,B | 6 | 6 | 36 | | 1 |
| A,C | 5 | 2 | 10 | | 1 |
| B,E | 14 | 4 | 56 | | 1 |
| B,F | 9 | 1 | 9 | | 1 |
| C,D | 15 | 1 | 15 | | 1 |
| C,E | 10 | 0 | 0 | | 1 |
| D,G | 10 | 0 | 0 | | 1 |
| E,G | 12 | 2 | 24 | | 1 |
| E,H | 6 | 1 | 6 | | 1 |
| F,H | 8 | 0 | 0 | | 1 |
| G,I | 12 | 1 | 12 | | 1 |
| H,I | 16 | 0 | 0 | | 1 |
| | | somme | 168 | | |

De cette manière on obtient le chemin le plus court en reliant les points non nul de la colonne des xij.

Ainsi on retrouve, en sommant les couts des arcs, le résultat de la question 1 qui est de 44.

Pour répondre à la question 2 il faut refaire les même calcul excel mais en faisant intervenir les arcs B->C ou C->B et D->E->F, D->F->E, E->D->F, E->F->D, F->D->E ou F->E->D.

Ceci nous donne un total de 12 feuilles excel : un vrai bonheur !

Ainsi effectué on choisit celle qui a la somme des xij*dij la plus faible soit la solution du chemin le plus court des chemins les plus longs !

Voici l'heureux gagnant du concours excel :

B->C
F->E->D

| (i,j) | dij | xij | xij*dij | | contraintes |
|---|---|---|---|---|---|
| A,B | 6 | 8 | 48 | | 1 |
| A,C | 5 | 0 | 0 | | 1 |
| B,E | 14 | 0 | 0 | | 1 |
| B,F | 9 | 6 | 54 | | 1 |
| C,D | 15 | 0 | 0 | | 1 |
| C,E | 10 | 0 | 0 | | 1 |
| D,G | 10 | 2 | 20 | | 1 |
| E,G | 12 | 0 | 0 | | 1 |

| | | | | | |
|---|---|---|---|---|---|
| E,H | 6 | 1 | 6 | | 1 |
| F,H | 8 | 0 | 0 | | 1 |
| G,I | 12 | 1 | 12 | | 1 |
| H,I | 16 | 0 | 0 | | 1 |
| B,C | 6 | 1 | 6 | | 1 |
| F,E | 8 | 5 | 40 | | 1 |
| E,D | 6 | 3 | 18 | | 1 |
| | | somme | 204 | | |

En construisant l'arc on trouve une durée maximale de 51.

Question 3 :

La question piège du projet ! Mais ne serai ce pas la question 2 précédentes tournées autrement ? Mais bien sur, on remarque que les taches A et B sont incompatibles ainsi que les taches E, F et D.

Quel soulagement ! Le projet s'achève ici.

# ANNEXE PARTIE I

```
> restart;

> with(simplex):
Warning, the protected names maximize and minimize have been
redefined and unprotected
```

Rentrons la fonction linéaire à minimiser

```
>
F:=infinity*x11+3*x12+8*x13+10*x14+7*x15+9*x16+9*x21+infinity*
x22+7*x23+4*x24+12*x25+10*x26+15*x31+9*x32+infinity*x33+9*x34+
x35+8*x36+2*x41+12*x42+15*x43+infinity*x44+8*x45+9*x46+20*x51+
10*x52+12*x53+8*x54+infinity*x55+2*x56+7*x61+6*x62+3*x63+9*x64
+7*x65+infinity*x66;
```

$$F := 3x12 + 8x13 + 10x14 + 7x15 + 9x16 + 9x21 + 7x23 + 4x24 + 12x25 + 10x26 + 15x31 + 9x32 + 9x34 + x35 + 8x36 + 2x41 + 12x42 + 15x43 + 8x45 + 9x46 + 20x51 + 10x52 + 12x53 + 8x54 + 2x56 + 7x61 + 6x62 + 3x63 + 7x65 + 9x64$$

Rentrons les contraintes :

```
> G1:=x11+x12+x13+x14+x15+x16:
> G2:=x21+x22+x23+x24+x25+x26:
> G3:=x31+x32+x33+x34+x35+x36:
> G4:=x41+x42+x43+x44+x45+x46:
> G5:=x51+x52+x53+x54+x55+x56:
> G6:=x61+x62+x63+x64+x65+x66:
> H1:=x11+x21+x31+x41+x51+x61:
> H2:=x12+x22+x32+x42+x52+x62:
> H3:=x13+x23+x33+x43+x53+x63:
> H4:=x14+x24+x34+x44+x54+x64:
> H5:=x15+x25+x35+x45+x55+x65:
> H6:=x16+x26+x36+x46+x56+x66:
```

Minimisons

```
> minimize( F,
{G1=1,G2=1,G3=1,G4=1,G5=1,G6=1,H1=1,H2=1,H3=1,H4=1,H5=1,H6=1},
NONNEGATIVE );assign(%);
```

$$\{x11 = 0, x21 = 0, x33 = 0, x51 = 0, x61 = 0, x13 = 0, x14 = 0, x15 = 0, x16 = 0, x22 = 0, x23 = 0, x26 = 0, x42 = 0, x36 = 0, x44 = 0, x45 = 0, x46 = 0, x53 = 0, x54 = 0, x55 = 0, x62 = 0, x65 = 0, x66 = 0, x64 = 0, x31 = 0, x32 = 0, x25 = 0, x34 = 0, x43 = 0, x52 = 0, x41 = 1, x24 = 1, x12 = 1, x35 = 1, x56 = 1, x63 = 1\}$$

coût du cycle

```
>
G:=10^5*x11+3*x12+8*x13+10*x14+7*x15+9*x16+9*x21+10^5*x22+7*x2
3+4*x24+12*x25+10*x26+15*x31+9*x32+10^5*x33+9*x34+x35+8*x36+2*
x41+12*x42+15*x43+10^5*x44+8*x45+9*x46+20*x51+10*x52+12*x53+8*
x54+10^5*x55+2*x56+7*x61+6*x62+3*x63+9*x64+7*x65+10^5*x66;
```

$$G := 15$$

## Cas séparation avec l'arc (A-B)

```
> restart;
> with(simplex):
Warning, the protected names maximize and minimize have been
redefined and unprotected
```

Rentrons la fonction linéaire à minimiser

```
>
F:=infinity*x11+3*x12+infinity*x13+infinity*x14+infinity*x15+i
nfinity*x16+9*x21+infinity*x22+7*x23+4*x24+12*x25+10*x26+15*x3
1+infinity*x32+infinity*x33+9*x34+x35+8*x36+2*x41+infinity*x42
+15*x43+infinity*x44+8*x45+9*x46+20*x51+infinity*x52+12*x53+8*
x54+infinity*x55+2*x56+7*x61+infinity*x62+3*x63+9*x64+7*x65+in
finity*x66;
```

$$F := 3\,x12 - 9\,x21 + 7\,x23 + 4\,x24 - 12\,x25 - \infty\,x11 + \infty\,x14 + \infty\,x13 - \infty\,x15 + \infty\,x16 + \infty\,x22 - \infty\,x32 + \infty\,x33 - \infty\,x42 + \infty\,x44 + \infty\,x52 - \infty\,x55$$

$$+ \infty\,x62 + \infty\,x66 + 10\,x26 + 15\,x31 + 9\,x34 + x35 + 8\,x36 + 2\,x41 + 15\,x43 + 8\,x45 + 9\,x46 + 20\,x51 + 12\,x53 + 8\,x54 + 2\,x56 + 7\,x61 + 3\,x63$$

$$- 9\,x64 - 7\,x65$$

Rentrons les contraintes :

```
> G1:=x11+x12+x13+x14+x15+x16:
> G2:=x21+x22+x23+x24+x25+x26:
> G3:=x31+x32+x33+x34+x35+x36:
> G4:=x41+x42+x43+x44+x45+x46:
> G5:=x51+x52+x53+x54+x55+x56:
> G6:=x61+x62+x63+x64+x65+x66:
> H1:=x11+x21+x31+x41+x51+x61:
> H2:=x12+x22+x32+x42+x52+x62:
> H3:=x13+x23+x33+x43+x53+x63:
> H4:=x14+x24+x34+x44+x54+x64:
> H5:=x15+x25+x35+x45+x55+x65:
> H6:=x16+x26+x36+x46+x56+x66:
```

Minimisons

```
> minimize( F,
{G1=1,G2=1,G3=1,G4=1,G5=1,G6=1,H1=1,H2=1,H3=1,H4=1,H5=1,H6=1},
NONNEGATIVE );assign(%);
```

$$\{x11 = 0, x21 = 0, x31 = 0, x51 = 0, x61 = 0, x13 = 0, x14 = 0, x15 = 0, x16 = 0, x22 = 0, x26 = 0, x32 = 0, x33 = 0, x36 = 0, x44 = 0, x45 = 0, x46 = 0,$$

$$x53 = 1, x54 = 0, x55 = 0, x62 = 1, x64 = 0, x65 = 1, x66 = 0, x23 = 0, x41 = 1, x25 = 0, x34 = 0, x43 = 1, x42 = 1, x52 = 0, x35 = 1, x63 = 1, x56 = 1,$$

$$x24 = 1, x12 = 1\}$$

coût du cycle :

```
>
G:=10^5*x11+3*x12+8*x13+10*x14+7*x15+9*x16+9*x21+10^5*x22+7*x2
3+4*x24+12*x25+10*x26+15*x31+9*x32+10^5*x33+9*x34+x35+8*x36+2*
x41+12*x42+15*x43+10^5*x44+8*x45+9*x46+20*x51+10*x52+12*x53+8*
x54+10^5*x55+2*x56+7*x61+6*x62+3*x63+9*x64+7*x65+10^5*x66;
```

$$G := 15$$

## Cas séparation avec l'arc (A-C)

```
> restart;
> with(simplex):
Warning, the protected names maximize and minimize have been
redefined and unprotected
```

Rentrons la fonction linéaire à minimiser

```
>
F:=infinity*x11+infinity*x12+8*x13+infinity*x14+infinity*x15+i
nfinity*x16+9*x21+infinity*x22+infinity*x23+4*x24+12*x25+10*x2
6+15*x31+9*x32+infinity*x33+9*x34+x35+8*x36+2*x41+12*x42+infin
ity*x43+infinity*x44+8*x45+9*x46+20*x51+10*x52+infinity*x53+8*
x54+infinity*x55+2*x56+7*x61+6*x62+infinity*x63+9*x64+7*x65+in
finity*x66;
```

$$F := \infty\, x11 + \infty\, x12 + 8\, x13 + \infty\, x14 + \infty\, x15 + \infty\, x16 + \infty\, x22 + \infty\, x23 + \infty\, x33 + 4\, x24 + 12\, x25 + 10\, x26 + 15\, x31 + 9\, x32 + 9\, x34 + x35 + 8\, x36 + 2\, x41 + 12\, x42 + \infty\, x43 + \infty\, x44 + 8\, x45 + 9\, x46 + 20\, x51 + 10\, x52 + \infty\, x53 + 8\, x54 + \infty\, x55 + 2\, x56 + 7\, x61 + 6\, x62 + \infty\, x63 + 9\, x64 + 7\, x65 + \infty\, x66$$

Rentrons les contraintes :

```
> G1:=x11+x12+x13+x14+x15+x16:
> G2:=x21+x22+x23+x24+x25+x26:
> G3:=x31+x32+x33+x34+x35+x36:
> G4:=x41+x42+x43+x44+x45+x46:
> G5:=x51+x52+x53+x54+x55+x56:
> G6:=x61+x62+x63+x64+x65+x66:
> H1:=x11+x21+x31+x41+x51+x61:
> H2:=x12+x22+x32+x42+x52+x62:
> H3:=x13+x23+x33+x43+x53+x63:
> H4:=x14+x24+x34+x44+x54+x64:
> H5:=x15+x25+x35+x45+x55+x65:
> H6:=x16+x26+x36+x46+x56+x66:
```

Minimisons

```
> minimize( F,
{G1=1,G2=1,G3=1,G4=1,G5=1,G6=1,H1=1,H2=1,H3=1,H4=1,H5=1,H6=1},
NONNEGATIVE );assign(%);
```

$$\{x46 = 0, x53 = 0, x54 = 0, x55 = 0, x63 = 0, x64 = 0, x65 = 0, x66 = 0, x42 = 0, x15 = 0, x33 = 0, x51 = 0, x23 = 0, x26 = 0, x31 = 0, x32 = 0, x36 = 0, x44 = 0, x45 = 0, x16 = 0, x25 = 0, x34 = 0, x62 = 1, x56 = 1, x52 = 0, x43 = 0, x24 = 1, x61 = 0, x41 = 1, x35 = 1, x13 = 1, x11 = 0, x12 = 0, x14 = 0, x21 = 0, x22 = 0\}$$

coût du cycle

```
>
G:=10^5*x11+3*x12+8*x13+10*x14+7*x15+9*x16+9*x21+10^5*x22+7*x2
```

```
3+4*x24+12*x25+10*x26+15*x31+9*x32+10^5*x33+9*x34+x35+8*x36+2*
x41+12*x42+15*x43+10^5*x44+8*x45+9*x46+20*x51+10*x52+12*x53+8*
x54+10^5*x55+2*x56+7*x61+6*x62+3*x63+9*x64+7*x65+10^5*x66;
```

$$G := 23$$

## Cas séparation avec l'arc (A-D)

```
> restart;
> with(simplex):
Warning, the protected names maximize and minimize have been
redefined and unprotected
```

Rentrons la fonction linéaire à minimiser

```
>
F:=infinity*x11+infinity*x12+infinity*x13+10*x14+infinity*x15+
infinity*x16+9*x21+infinity*x22+7*x23+infinity*x24+12*x25+10*x
26+15*x31+9*x32+infinity*x33+infinity*x34+x35+8*x36+2*x41+12*x
42+15*x43+infinity*x44+8*x45+9*x46+20*x51+10*x52+12*x53+infini
ty*x54+infinity*x55+2*x56+7*x61+6*x62+3*x63+infinity*x64+7*x65
+infinity*x66;
```

$$F := 12\,x25 + 10\,x26 + 15\,x31 + 9\,x32 + x35 + 8\,x36 + 2\,x41 + 12\,x42 + 15\,x43 + 8\,x45 + 9\,x46 + 20\,x51 + 10\,x52 + 12\,x53 + 2\,x56 + 7\,x61 + 6\,x62 + 3\,x63 + 7\,x65 + 10\,x14 + 9\,x21 + 7\,x23 + \infty\,x11 + \infty\,x12 + \infty\,x13 + \infty\,x15 + \infty\,x16 + \infty\,x22 + \infty\,x24 + \infty\,x33 + \infty\,x34 + \infty\,x44 + \infty\,x54 + \infty\,x55 + \infty\,x64 + \infty\,x66$$

Rentrons les contraintes :

```
> G1:=x11+x12+x13+x14+x15+x16:
> G2:=x21+x22+x23+x24+x25+x26:
> G3:=x31+x32+x33+x34+x35+x36:
> G4:=x41+x42+x43+x44+x45+x46:
> G5:=x51+x52+x53+x54+x55+x56:
> G6:=x61+x62+x63+x64+x65+x66:
> H1:=x11+x21+x31+x41+x51+x61:
> H2:=x12+x22+x32+x42+x52+x62:
> H3:=x13+x23+x33+x43+x53+x63:
> H4:=x14+x24+x34+x44+x54+x64:
> H5:=x15+x25+x35+x45+x55+x65:
> H6:=x16+x26+x36+x46+x56+x66:
```

Minimisons

```
> minimize( F,
{G1=1,G2=1,G3=1,G4=1,G5=1,G6=1,H1=1,H2=1,H3=1,H4=1,H5=1,H6=1},
NONNEGATIVE );assign(%);
```

$$\{x16 = 0, x25 = 0, x34 = 0, x26 = 0, x56 = 0, x44 = 0, x45 = 0, x46 = 0, x53 = 0, x54 = 0, x55 = 0, x63 = 0, x64 = 0, x65 = 0, x66 = 0, x11 = 0, x12 = 0, x13 = 0, x15 = 0, x21 = 0, x22 = 0, x24 = 0, x42 = 0, x31 = 0, x51 = 0, x61 = 0, x32 = 0, x33 = 0, x43 = 0, x14 = 1, x41 = 1, x23 = 1, x36 = 1, x62 = 1, x52 = 0, x35 = 1\}$$

coût du cycle

```
>
G:=10^5*x11+3*x12+8*x13+10*x14+7*x15+9*x16+9*x21+10^5*x22+7*x2
3+4*x24+12*x25+10*x26+15*x31+9*x32+10^5*x33+9*x34+x35+8*x36+2*
```

```
x41+12*x42+15*x43+10^5*x44+8*x45+9*x46+20*x51+10*x52+12*x53+8*
x54+10^5*x55+2*x56+7*x61+6*x62+3*x63+9*x64+7*x65+10^5*x66;
```

$$G := 28$$

## Cas séparation avec l'arc (A-E)

```
> restart;
> with(simplex):
Warning, the protected names maximize and minimize have been
redefined and unprotected
```

Rentrons la fonction linéaire à minimiser

```
>
F:=infinity*x11+infinity*x12+infinity*x13+infinity*x14+7*x15+i
nfinity*x16+9*x21+infinity*x22+7*x23+4*x24+infinity*x25+10*x26
+15*x31+9*x32+infinity*x33+9*x34+infinity*x35+8*x36+2*x41+12*x
42+15*x43+infinity*x44+infinity*x45+9*x46+20*x51+10*x52+12*x53
+8*x54+infinity*x55+2*x56+7*x61+6*x62+3*x63+9*x64+infinity*x65
+infinity*x66;
```

$$F := 7x15 + 9x21 + 7x23 + 4x24 + 10x26 + 15x31 + 9x32 + 9x34 + 8x36 + 2x41 + 12x42 + 15x43 + 9x46 + 20x51 + 10x52 + 12x53 + 8x54 + 2x56 + 7x61 + 6x62 + 3x63 + 9x64 + \infty x11 + \infty x12 + \infty x13 + \infty x14 + \infty x16 + \infty x22 + \infty x25 + \infty x33 + \infty x35 + \infty x44 + \infty x45 + \infty x55 + \infty x65 + \infty x66$$

Rentrons les contraintes :

```
> G1:=x11+x12+x13+x14+x15+x16:
> G2:=x21+x22+x23+x24+x25+x26:
> G3:=x31+x32+x33+x34+x35+x36:
> G4:=x41+x42+x43+x44+x45+x46:
> G5:=x51+x52+x53+x54+x55+x56:
> G6:=x61+x62+x63+x64+x65+x66:
> H1:=x11+x21+x31+x41+x51+x61:
> H2:=x12+x22+x32+x42+x52+x62:
> H3:=x13+x23+x33+x43+x53+x63:
> H4:=x14+x24+x34+x44+x54+x64:
> H5:=x15+x25+x35+x45+x55+x65:
> H6:=x16+x26+x36+x46+x56+x66:
```

Minimisons

```
> minimize( F,
{G1=1,G2=1,G3=1,G4=1,G5=1,G6=1,H1=1,H2=1,H3=1,H4=1,H5=1,H6=1},
NONNEGATIVE );assign(%);
```

$$\{x11 = 0, x12 = 0, x13 = 0, x14 = 0, x21 = 0, x22 = 0, x23 = 0, x26 = 0, x31 = 0, x33 = 0, x36 = 0, x44 = 0, x45 = 0, x46 = 0, x53 = 0, x54 = 0, x55 = 0, x62 = 0, x64 = 0, x65 = 0, x66 = 0, x42 = 0, x33 = 0, x51 = 0, x16 = 0, x43 = 0, x52 = 0, x24 = 1, x25 = 0, x61 = 0, x32 = 1, x34 = 0, x41 = 1, x56 = 1, x63 = 1, x15 = 1\}$$

coût du cycle

```
>
G:=10^5*x11+3*x12+8*x13+10*x14+7*x15+9*x16+9*x21+10^5*x22+7*x2
3+4*x24+12*x25+10*x26+15*x31+9*x32+10^5*x33+9*x34+x35+8*x36+2*
x41+12*x42+15*x43+10^5*x44+8*x45+9*x46+20*x51+10*x52+12*x53+8*
x54+10^5*x55+2*x56+7*x61+6*x62+3*x63+9*x64+7*x65+10^5*x66;
```

$$G := 27$$

## Cas séparation avec l'arc (A-F)

```
> restart;
> with(simplex):
Warning, the protected names maximize and minimize have been
redefined and unprotected
```

Rentrons la fonction linéaire à minimiser

```
>
F:=infinity*x11+infinity*x12+infinity*x13+infinity*x14+infinit
y*x15+9*x16+9*x21+infinity*x22+7*x23+4*x24+12*x25+infinity*x26
+15*x31+9*x32+infinity*x33+9*x34+x35+infinity*x36+2*x41+12*x42
+15*x43+infinity*x44+8*x45+infinity*x46+20*x51+10*x52+12*x53+8
*x54+infinity*x55+infinity*x56+7*x61+6*x62+3*x63+9*x64+7*x65+i
nfinity*x66;
```

$$F := \infty\, x11 + \infty\, x12 + \infty\, x14 + \infty\, x13 + \infty\, x15 + \infty\, x22 + \infty\, x26 + \infty\, x33 + \infty\, x36 + \infty\, x44 + \infty\, x46 + 9\, x16 + 9\, x21 + 7\, x23 + 4\, x24 + 12\, x25 + 15\, x31 + 9\, x32 + 9\, x34 + x35 + 2\, x41 + 12\, x42 + 15\, x43 + 8\, x45 + 20\, x51 + 10\, x52 + 12\, x53 + 8\, x54 + 7\, x61 + 6\, x62 + 3\, x63 + 9\, x64 + 7\, x65 + \infty\, x55 + \infty\, x56 + \infty\, x66$$

Rentrons les contraintes :

```
> G1:=x11+x12+x13+x14+x15+x16:
> G2:=x21+x22+x23+x24+x25+x26:
> G3:=x31+x32+x33+x34+x35+x36:
> G4:=x41+x42+x43+x44+x45+x46:
> G5:=x51+x52+x53+x54+x55+x56:
> G6:=x61+x62+x63+x64+x65+x66:
> H1:=x11+x21+x31+x41+x51+x61:
> H2:=x12+x22+x32+x42+x52+x62:
> H3:=x13+x23+x33+x43+x53+x63:
> H4:=x14+x24+x34+x44+x54+x64:
> H5:=x15+x25+x35+x45+x55+x65:
> H6:=x16+x26+x36+x46+x56+x66:
```

Minimisons

```
> minimize( F,
{G1=1,G2=1,G3=1,G4=1,G5=1,G6=1,H1=1,H2=1,H3=1,H4=1,H5=1,H6=1},
NONNEGATIVE );assign(%);
```

$$\{x64 = 0, x65 = 0, x66 = 0, x63 = 1, x41 = 1, x24 = 1, x54 = 0, x32 = 0, x52 = 1, x62 = 0, x15 = 0, x35 = 1, x56 = 0, x16 = 1, x61 = 0, x11 = 0, x13 = 0, x14 = 0, x21 = 0, x22 = 0, x23 = 0, x25 = 0, x26 = 0, x31 = 0, x33 = 0, x34 = 0, x36 = 0, x12 = 0, x43 = 0, x44 = 0, x45 = 0, x46 = 0, x51 = 0, x53 = 0, x55 = 0, x12 = 0\}$$

coût du cycle

```
>
G:=10^5*x11+3*x12+8*x13+10*x14+7*x15+9*x16+9*x21+10^5*x22+7*x2
3+4*x24+12*x25+10*x26+15*x31+9*x32+10^5*x33+9*x34+x35+8*x36+2*
x41+12*x42+15*x43+10^5*x44+8*x45+9*x46+20*x51+10*x52+12*x53+8*
x54+10^5*x55+2*x56+7*x61+6*x62+3*x63+9*x64+7*x65+10^5*x66;
```

$$G := 25$$

# ANNEXE PARTIE II

Algorithme des fourmis.

```
[d,t,h,iter,alpha,beta,e,m,n,el,Q,tparcours]=ants_information;
sequence=[];
MIN=inf;
for i=1:iter
    [app]=ants_primaryplacing(m,n);
    [at]=ants_cycle(app,m,n,h,t,alpha,beta,tparcours);
    at=horzcat(at,at(:,1));
    [cost,f]=ants_cost(m,n,d,at,el,tparcours);
    [t]=ants_traceupdating(m,n,t,at,f,e,Q,tparcours);

    costoa(i)=mean(cost);
    [mincost(i),number]=min(cost);

    if mincost(i)<MIN
        % on enregistre la séquence qui a obtenu le cout minimal.
        sequence=[at(number,:)];
        MIN=mincost(i);
    end

    iteration(i)=i;
end
% la séquence de coût minimal est en at(
MOY=num2str(mean(mincost));
subplot(2,1,1);plot(iteration,costoa);
title(['On vérifie la convergence de ACO : cout minimal en fonction du nombre ↙
d''itérations, moyenne :', MOY]);
xlabel('itération');
ylabel('coût min');


disp(['le circuit optimal est : ',num2str(sequence)])
```

```
function [d,t,h,iter,alpha,beta,e,m,n,el,Q,tparcours]=ants_information;
iter=100;%number of cycles.
m=200;%number of ants.

Q=1; % paramètre
tparcours=6; %nombre de segments du parcours hamiltonien
n=6; % nombre de points
d=[0,3,8,10,7,9;9,0,7,4,12,10;15,9,0,9,1,8;2,12,15,0,8,9;20,10,12,8,0,2;7,6,3,9,7,0];

e=.1;%evaporation coefficient.
alpha=1;%order of effect of ants' sight.
beta=5;%order of trace's effect.
for i=1:n%generating sight matrix.
    for j=1:n
        if d(i,j)==0
            h(i,j)=0;
        else
            h(i,j)=1/d(i,j);
        end
    end
end
t=0.0001*ones(n);%primary tracing.
el=.96;%coefficient of common cost elimination.
```

```
function [app]=ants_primaryplacing(m,n);
 % on part arbitrairement du point 1
app=ones(m+1,1);
```

```
function [at]=ants_cycle(app,m,n,h,t,alpha,beta,tparcours);
at=app;
i=1;
p=[];
while i~=m+1
    mh=h;

    for j=1:(tparcours-1)

        [at,mh]=choix(at,m,i,j,mh,n,alpha,beta,t);

    end
 i=i+1;

end
% generation of ants tour matrix during a cycle.

% on enlève la m eme ligne, qui pose probleme. Erreur de la boucle for ,
% elle repasse de 200 à 1.
at=at(1:(m),:);
```

```
function [at,mh]=choix(at,m,i,j,mh,n,alpha,beta,t)

% cette fonction renvoie la valeur de at(i,j+1)

c=at(i,j);
mh(:,c)=0;

     if mh(c,:)==[zeros(1,n)]
           at(i,j+1)=at(i,j);
           % si aucun point n'est visible, la fourmi s'arrête
        else

        temp=(t(c,:).^beta).*(mh(c,:).^alpha);
        s=(sum(temp));
        p=(1/s).*temp;

        % la fourmi va sur le point qui a la plus grande probabilité
        % on cherche la position du max dans le vecteur p

            if p==[zeros(1,n)]
                % on change de position au hasard
                at(i,j+1)=floor(n*rand+1);
            else
                r=rand;
                s=0;
                     for k=1:n
                     s=s+p(k);
                          if r<=s
                          at(i,j+1)=k;
                          break
                          end
                     end

            end

        end
```

```
function [cost,f]=ants_cost(m,n,d,at,el,tparcours)
for i=1:m
    s=0;
    for j=1:tparcours
        s=s+d(at(i,j),at(i,j+1));
    end
    f(i)=s;
end
cost=f;
f=f-el*min(f);%elimination of common cost.
```

```
function [t]=ants_traceupdating(m,n,t,at,f,e,Q,tparcours);
for i=1:m
    for j=1:tparcours
        % ici on peut mettre le paramètre Q tel que dt=Q/f(i);
        % on rajoute aussi e*dt
        dt=Q/f(i);
        t(at(i,j),at(i,j+1))=(1-e)*t(at(i,j),at(i,j+1))+e*dt; %updating traces.
    end
end
```