Algorithmique Avancée

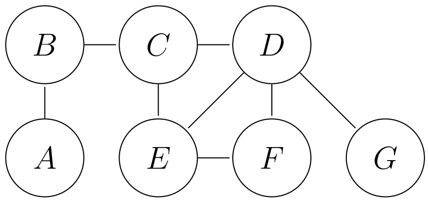
TD3
Programmation Linéaire en Nombres Entiers

Elana Courtines
courtines.e@gmail.com
https://github.com/irinacake

Séance 1 - 10 octobre 2022
Séance 1 - 17 octobre 2022

Jerome Mengin - jerome.mengin@univ-tlse3.fr

Exercice 1 :



Dans l'exemple de l'exercice, une solution serait : $\{B, D, E\}$.

Dans le cas général :

- sommets : $\{1, 2..., n\}$
- arêtes : paires $\{i, j\}$, $i \neq j$
  $\rightarrow$ d'où un boulean x_i
- $x_1 = 1$ si sommet i sélectionné
  $\rightarrow$ on cherche : $min(\sum_{i=1}^{n} x_i)$
- $\forall \{i, j\}$ arête du graphe, $x_i + x_j \geq 1$

Exercice 2 :

Question 2.1 :

Variables :

$$\begin{cases} x_{ij} = & 1 \text{ si la tâche } i \text{ a été réalisée sur la machine } j \\ & 0 \text{ sinon} \end{cases}$$

Objectif :

$min(\sum_{j=1}^{p} c_j(\sum_{i=1}^{k} d_{ij} * x_{ij}))$

Contraintes :

$$\begin{cases} \text{Durée sur machine} \leq D : & \forall j \in \{1, .., p\} \text{ machine }, \sum_{i=1}^{k} d_{ij} * x_{ij} \leq D \\ \text{Unicité des tâches :} & \forall i \in \{1, ..., k\} \text{ tâche }, \sum_{j=1}^{p} x_{ij} = 1 \end{cases}$$

Note : si la notation $\sum_{j=1}^{p} x_{ij} = 1$ est acceptable en "langage humain", elle *peut* ne pas l'être pour un solveur. En théorie il faudrait donner cette contrainte sous forme de deux contraintes avec inégalité : l'une présentant '$\leq$' et l'autre présentant '$\geq$'.
En pratique cela dit, un solveur devrait être capable de convertir une telle égalité en deux inégalités.

Question 2.2 :

Variables :

$$\begin{cases} x_{ij} = & 1 \text{ si la tâche } i \text{ a été réalisée sur la machine } j \\ & 0 \text{ sinon} \\ d_{max} & \geq 0, \text{ durée sur la machine qui finit en dernier} \\ & \text{également appelé } maxspan \text{ en anglais} \end{cases}$$

Objectif : $min(d_{max})$

Contraintes :

$$\begin{cases} \forall j \in \{1, .., p\} \text{ machine }, \sum_{i=1}^{k} d_{ij} * x_{ij} \leq d_{max} \\ \forall i \in \{1, ..., k\} \text{ tâche }, \sum_{j=1}^{p} x_{ij} = 1 \end{cases}$$

Exercice 3 :

Question 3.1 :

Variables : $\begin{cases} x_{ij} = & 1 \text{ si l'équipe i va au site j} \\ & 0 \text{ sinon} \\ y_{ijk} = & 1 \text{ si equipe } i \text{ visite sites } j \textbf{ et } k,\ j \leq k \\ & 0 \text{ sinon} \end{cases}$

Objectif : $min(\sum_{i,j} t_{ij} * x_{ij} + \sum_{i,j,k} d_{jk} * y_{ijk})$

$= min(\sum_{i,j} t_{ij} * x_{ij} + \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=j+1}^{n} d_{jk} * y_{ijk})$

Contraintes :

$$\begin{cases} \forall i\ ,\ \sum_j x_{ij} \leq 2 & \text{Chaque équipe va sur 2 sites au plus} \\ \forall j\ ,\ \sum_i x_{ij} = 1 & \text{Chaque site visité par une seule équipe} \\ \forall i,j,k,\ y_{ijk} \geq x_{ij} + x_{ik} - 1 & y_{ijk} = 1 \text{ si } x_{ij} = x_{ik} = 1 \\ & \text{si } x_{ij} = 0 \text{ ou } x_{ik} = 0 \text{, la contrainte devient :} \\ & y_{ijk} \geq 0 \text{ , car } y_{ijk} \text{ ne peut pas valoir -1} \\ \forall i,j,k,\ y_{ijk} \leq x_{ij} & \\ \forall i,j,k,\ y_{ijk} \leq x_{ik} & \end{cases}$$

Question 3.2 :

Variables : $\begin{cases} x_{ij} = & 1 \text{ si l'équipe i va au site j} \\ & 0 \text{ sinon} \\ y_{ijk} = & 1 \text{ si equipe } i \text{ visite sites } j \textbf{ et } k,\ j \leq k \\ & 0 \text{ sinon} \\ t_{max} & \geq 0 \text{, temps pris par l'équipe qui termine en dernier} \end{cases}$

Objectif : $min(t_{max})$

Contraintes :

- $\forall i\ ,\ \sum_j x_{ij} \leq 2$, Chaque équipe va sur 2 sites au plus
- $\forall j\ ,\ \sum_i x_{ij} = 1$, Chaque site visité par une seule équipe
- $\forall i,j,k,\ y_{ijk} \geq x_{ij} + x_{ik} - 1$
  $y_{ijk} = 1$ si $x_{ij} = x_{ik} = 1$
  si $x_{ij} = 0$ ou $x_{ik} = 0$, la contrainte devient :
  $y_{ijk} \geq 0$, car y_{ijk} ne peut pas valoir -1
- $\forall i,j,k,\ y_{ijk} \leq x_{ij}$
- $\forall i,j,k,\ y_{ijk} \leq x_{ik}$

\+ $\forall i\ (\sum_j t_{ij} * x_{ij}) + (\sum_{jk} d_{jk} * y_{ijk}) \leq d_{max}$

Exercice 5 :

Question 5.1 :

- T_{min} et T_{max} : $\begin{cases} \sum_j x_{Aj} * 100 \geq 200 \\ \sum_j x_{Aj} * 100 \leq 500 \end{cases}$, Soit 2 contraintes.
  On veut idéalement multiplier les valeurs par 100 (plutôt que de garder des petits entiers) car il s'agit simplement ici d'une simplification. Dans un univers plus réaliste, des valeurs aussi petites n'auraient aucun sens.

- B ne travaille pas le samedi : $x_{B5} = 0$.
  Note : on a fait la contrainte sur B ici à titre d'exemple, puisque A n'a aucune contrainte sur les jours off.

- *MinConsOff* : Quand A est de repos, c'est au moins 2 jours consécutifs.
  $\forall j = 0, ..., 4$, $\begin{cases} x_{Aj} + x_{A(j+2)} - 1 \leq x_{A(j+1)} \end{cases}$, Soit 5 contraintes.

- *MaxCons* : Quand A travaille, c'est au plus 5 jours consécutifs.
  $\begin{cases} x_{A0} + x_{A1} + x_{A2} + x_{A3} + x_{A4} + x_{A5} \leq 5 \\ x_{A1} + x_{A2} + x_{A3} + x_{A4} + x_{A5} + x_{A6} \leq 5 \end{cases}$, Soit 2 contraintes.

- *MinCons* : Quand A travaille, c'est au moins 3 jours consécutifs → on cherche à bloquer les séquences "Off-On-Off" et "Off-On-On-Off" :
  $\begin{cases} \forall j = 0, ..., 4 \quad x_{Aj} + x_{A(j+2)} \geq x_{A(j+1)} \\ \forall j = 0, ..., 3 \quad x_{Aj} + x_{A(j+3)} \leq x_{A(j+1)} + x_{A(j+2)} - 1 \end{cases}$, Soit (5+4 contraintes)

Pour A, il y a donc au total $2 + 0 + 5 + 2 + (5 + 4) = 18$ contraintes.

Question 5.2 : Contraintes sur la planification :

$$\begin{cases} x_{A0} + x_{B0} + x_{C0} = 2 & \text{, deux personnes le Lundi} \\ x_{A1} + x_{B1} + x_{C1} = 2 & \text{, deux personnes le Mardi} \\ x_{A2} + x_{B2} + x_{C2} = 2 & \text{, deux personnes le Mercredi} \\ x_{A3} + x_{B3} + x_{C3} = 2 & \text{, deux personnes le Jeudi} \\ x_{A4} + x_{B4} + x_{C4} = 2 & \text{, deux personnes le Vendredi} \\ x_{A5} + x_{B5} + x_{C5} = 1 & \text{, une personne le Samedi} \\ x_{A6} + x_{B6} + x_{C6} = 1 & \text{, une personne le Dimanche} \end{cases}$$

Question 5.3 :
La solution 2 ne vérifie pas toutes les contraintes, en effet :

- Pour A : *MinConsOff* - il ne fait une pause que de 1 jour le mercredi

- Pour B : *MinConsOff* - il ne fait une pause que de 1 jour le mardi
  MinCons - il ne travaille qu'un seul jour consécutif le mercredi

Question 5.4 :

- Pénalités pour la 1^{er} solution = 7
  - pénalité si repos : $pen[B,3] = 5$
  - pénalité si service : $pen[C,4] = 2$
- *La 2^e solution ne respecte pas les contraintes de base*
- Pénalité pour la 3^e solution = 7 [...]
- Pénalité pour la 4^e solution = 9 [...]

Si on suppose que les solutions 1 et 3 sont optimales (c'est à dire qui minimisent les pénalités), alors c'est la 4^e solution qui n'est pas optimale (elle ne minimise pas les pénalités).

Question 5.5 : Objectif

On chercher à minimiser la fonction suivante :
$min = 5*(1-x_{A0}) + 2*(1-x_{A4}) + 5*(1-x_{B3}) + 5*(1-x_{C2}) + 5*(x_{A2}) + 2*(x_{C4})$

Exercice 6 : Généralisation de toutes les contraintes de l'exercice 5

- T_{min} et T_{max} :

$$\begin{cases} \forall p \in P \quad \sum_{j=0}^{h-1} x_{pj} * 100 \leq TMax[p] \\ \qquad\qquad \sum_{j=0}^{h-1} x_{pj} * 100 \geq TMin[p] \end{cases}$$

- *Jours de repos* :
  $\forall p \in P \ , \ \forall j \in H \ , \ x_{pj} = 1 - Off[p,j]$
  Attention, c'est bien $Off[p,j] = 1$ *si* p *ne travaille* ***pas*** *le jour* j

- *MinConsOff* : $\forall p \in P \ , \ \forall k \in \{2,..,MinConsOff[p])\} \ , \ \forall j \in \{0,..,h-1-k\}$

  $x_{pj} + x_{p(j+k)} - 1 \leq \sum_{n=j+1}^{j+k-1} x_{pn}$

- *MaxCons* :
  $\forall p \in P \ , \ \forall j \in \{0,..,(h-1-MaxCons[p])\}$

  $\sum_{k=j}^{j+MaxCons[p]} x_{pk} \leq MaxCons[p]$

- *MinCons* :
  $\forall p \in P \ , \ \forall k \in \{2,..,MinCons[p])\} \ , \ \forall j \in \{0,..,h-1-k\}$

  $x_{pj} + x_{p(j+k)} \geq \sum_{n=j+1}^{j+k-1} x_{pn} - (k-2)$