

Notes de cours de ARO

Yann Miguel

13 octobre 2020

Table des matières

1	Introduction	2
2	Cours 1	3
2.1	La programmation linéaire	3
2.2	La méthode simplex	3
2.2.1	Cas spéciaux	5
3	Cours 2	7
3.1	La méthode simplex à deux phases	7
3.2	L'algorithme du simplex en cas général	7
3.3	Dictionnaires, cas général	8
4	Cours 3	11
4.1	Dualité dans la programmation linéaire	11
4.2	Théorème des écarts complémentaire	13
4.3	La signification économique du dual	14
4.4	Problème dual-réalisable	15
5	Cours 4	16
5.1	Modélisation par flots	16
5.2	Circulation et flots avec demandes et bornes inférieures et bornes supérieures	16
6	Cours 5	21
6.1	Beyond flows	21
6.1.1	MAX-SAT comme PL(en nombre entiers)	22
6.1.2	Couverture de sommets	23
7	Informations importantes	24

1 Introduction

Cette UE consiste à modéliser les problèmes en flot, puis à les programmer en linéaire.

2 Cours 1

2.1 La programmation linéaire

Exemple de problème linéaire: maximiser $x_1 + x_2$ avec:

- $x_2 - x_1 \leq 1$
- $x_1 + 6x_2 \leq 15$
- $4x_1 - x_2 \leq 10$
- $x_1 \geq 0$, et $x_2 \geq 0$

Si on dessine ce problème sur un plan, on observe une zone du plan correspondant à ces contraintes, et la solution optimale avec ces contraintes est 5, avec $x_1 = 3$ et $x_2 = 2$.

2.2 La méthode simplex

Il s'agit d'aller de solution de base en solution de base.

Une solution de base est l'ensemble des sommets du polyhèdre, et le polyhèdre est l'ensemble des solutions admissibles.

La méthode simplex est donc une méthode de voyage entre les sommets du polyhèdre solution.

Forme générale du problème linéaire: maximiser $c_1x_1 + c_2x_2 + \dots + c_nx_n$ avec:

- $a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1$
- $a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m$
- $x_1, \dots, x_n \geq 0$

Chaque problème de maximisation a un problème de minimisation associé.

La programmation linéaire date de Fourrier, alors que la méthode suplex a été inventée dans les années 50 par Dantzig.

L'algorithme de la méthode simplex est exponentiel dans le pire des cas, et polynomial en moyenne.

L'algorithme de Khachian, créé dans les années 70 est un algorithme polynomial un peu moins efficace que la méthode suplex.

Prenons l'exemple de la diète. Notre corps à besoin de:

- Vitamine A : 0,5mg
- Vitamine C : 15mg
- Fibres : 4g

Chaque nourriture à un prix, et notre corps à des demandes nutritionnelles.

Nom nutriment	Carottes(x_1)	Choux(x_2)	Cornichons(x_3)	Demandes
Vitamine A	35mg	0,5mg	0,5mg	0,5mg
Vitamine C	60mg	30mg	10mg	15mg
Fibres	30g	20g	10g	4g
prix	0,75 euro	0,5 euro	0,15 euro	

Cette exemple demandera donc de MINIMISER la quantité d'aliments à prendre, afin de minimiser le coût, et d'avoir les apports minimaux requis.

Exemple de la méthode suplex:

Objet	Cendrier(x_1)	Bol(x_2)	Cruche(x_3)	Vase(x_4)	maximum
Moulage	2	4	5	7	42
Cuisson	1	1	2	2	17
Peinture	1	2	3	3	24
Bénéfice	7	9	18	17	

On veut donc, dans ce problème, maximiser $7x_1 + 9x_2 + 18x_3 + 17x_4$

avec:

- $2x_1 + 4x_2 + 5x_3 + 7x_4 \leq 42$
- $1x_1 + 1x_2 + 2x_3 + 2x_4 \leq 17$
- $1x_1 + 2x_2 + 3x_3 + 4x_4 \leq 24$
- $x_1, x_2, x_3, x_4 \geq 0$

Dictionnaire D1:

- $x_5 = 42 - 2x_1 - 4x_2 - 5x_3 - 7x_4$
- $x_6 = 17 - 1x_1 - 1x_2 - 3x_3 - 2x_4$
- $x_7 = 24 - 1x_1 - 2x_2 - 3x_3 - 3x_4$
- traitdivision
- $z = 7x_1 + 9x_2 + 18x_3 + 17x_4$

Les variables de base sont les x de 5 à 7, et les variables hors base sont les x de 1 à 4.

La solution de base admissible consiste à affecter à toutes les variables hors base la valeur 0, ce qui crée:

- $x_5 = 42$
- $x_6 = 17$
- $x_7 = 24$
- $z = 0$

La méthode suplex va consister à faire entrer une variable hors base dans la base, et à faire sortir une variable de base de la base.

On va donc faire entrer la variable x_3 , car c'est la variable qui donne le plus de bénéfices (technique glouton). La variable qui va sortir sera celle qui mets le plus de contraintes, donc qui limite le plus la variable entrante. Il s'agit de x_7 dans ce cas.

L'étape de pivot fait donc entrer x_3 et sortir x_7 . Donc, on a:

$$3x_3 = 24 - 1x_1 - 2x_2 - 3x_4 - x_7$$

Ce qui donne:

$$x_3 = 8 - \frac{1}{3}x_1 - \frac{2}{3}x_2 - x_4 - \frac{1}{3}x_7$$

Le dictionnaire D2 est crée en remplaçant x_3 par l'expression en haut sur toutes les lignes de D1. En le simplifiant, ça donne:

- $x_3 = 8 - \frac{1}{3}x_1 - \frac{2}{3}x_2 - x_4 - \frac{1}{3}x_7$
- $x_5 = 2 - \frac{1}{3}x_1 - \frac{2}{3}x_2 - 2x_4 - \frac{5}{3}x_7$
- $x_6 = 1 - \frac{1}{3}x_1 - \frac{1}{3}x_2 - \frac{2}{3}x_7$
- traitdedivision
- $z = 144 + x_1 - 3x_2 - x_4 - 6x_7$

La solution de base admissible dans ce cas est 144 euros gagnés.

La variable entrante de D2 est x_1 , car c'est la seule encore positif dans l'équation avec z. La variable sortante est x_6 . Ce qui donne:

$$\frac{1}{3}x_1 = 1 + \frac{1}{3}x_2 - x_6 + \frac{2}{3}x_7$$

Ce qui donne, après simplification:

$$x_1 = 3 + x_2 - 3x_6 + 2x_7$$

On obtient D3 en remplaçant x_1 par cette équation dans tout D2. Après simplification, ça donne:

- $x_1 = 3 + x_2 - 3x_6 + 2x_7$
- $x_3 = 7 - x_2 - x_4 + x_6 - x_7$
- $x_5 = 1 - x_2 - 2x_4 + x_6 + x_7$
- traitdedivision
- $z = 147 - 2x_2 - x_4 - 3x_6 - 4x_7$

Étant donné qu'il n'y a plus de variable positive dans l'équation avec z, la solution de base admissible de D3 est la solution optimale. Donc, optimalement, on fera 3 cendriers, 0 bols, 7 cruches et 0 vases, et on fera un bénéfice de 147 euros.

2.2.1 Cas spéciaux

Cas spécial 1: maximum non borné

Si le maximum n'est pas borné, alors maximiser retourne $+\infty$.

Si il n'y a pas de contrainte sur x_2 , alors le maximum va vers l'infini.

Cas spécial 2: Dégénérescence et cyclage

La dégénérescence, c'est quand la borne supérieure de la variable

entrante est 0. Dans ce cas, on fait entrer dans la base cette variable, même si on ne gagne rien.

Le cyclage, c'est quand des étapes dégénèrent et nous ramènent toujours aux mêmes dictionnaires. Il faudra appliquer la règle de Blund pour s'en sortir.

La règle de Blund consiste à toujours choisir les variables entrentantes et sortantes de plus petit indice , si jamais il y a plus d'une variable possible.

Théorème La méthode suplex avec la règle de Blund n'a pas de cyclage.

Cas spécial 3: Recherche d'un dictionnaire et infaillibilité
Il faut utiliser la méthode suplex à deux phases, ce qui permet, via l'introduction d'un programme linéaire auxiliaire, de gérer le cas où créer la solution de base admissible du dictionnaire donne des x négatifs.

3 Cours 2

3.1 La méthode simplex à deux phases

La méthode simplex à deux phases consiste à résoudre un programme linéaire auxiliaire avant de résoudre le premier. Si le résultat est 0, alors il n'y a aucune réparation à faire dans le programme principal. Sinon, il faut réparer le programme principal, et il n'a pas de solution admissible.

Remarque:

Un programme linéaire possède une solution admissible, et donc un dictionnaire réalisable, si et seulement si son programme linéaire auxiliaire a $x_0=0$ comme solution optimale. Par conséquent, x_0 est une variable dite de réparation pour le programme linéaire.

3.2 L'algorithme du simplex en cas général

Forme canonique:

maximiser $c^t x$ avec:

- $Ax \leq b$
- $x \geq 0$

Forme équationnelle:

maximiser $c^t x$ avec:

- $Ax = b$
- $x \geq 0$

On passe de la forme canonique à la forme équationnelle en ajoutant les variables d'écart x_{n+1}, \dots, x_{n+m} . Notons que A devient une matrice $(n+m)m$ et que c devient une matrice $(n+m)$ alors que b reste comme avant.

Convention:

On suppose par la suite le programme linéaire en forme équationnelle avec les lignes de la matrice A linéairement indépendantes.

Définition:

Une solution de base admissible du programme linéaire $\max c^t x$, $Ax=b$, $x \geq 0$ est une solution admissible x pour laquelle il existe $B \subseteq \{1, 2, \dots, n+m\}$ tel que:

1. La matrice A_B est non singulière.
2. $x_j=0$ pour tout $j \notin B$.

Observations:

1. A_B est une matrice carrée $m*m$.

2. Une matrice carrée est non-singulière si et seulement si:

- elle est inversible
- son déterminant est $\neq 0$

Par la suite, si $B \subseteq \{1, 2, \dots, n+m\}$ est associé à une solution de base admissible, alors on appelle B une base admissible.

Proposition:

Toute solution de base admissible est déterminée de façon unique par B . En d'autres termes, si $B \subseteq \{1, 2, \dots, n+m\}$ est A_B est non-singulière, alors il existe au plus une solution admissible x avec $x_j=0, \forall j \notin B$.

3.3 Dictionnaires, cas général

Définition:

Un dictionnaire $D(B)$, associé à une base admissible B est un système de $m+1$ équations linéaires en variables $N = \{1, \dots, n+m\} \setminus B$ qui possède le même ensemble des solutions que le système $Ax=b$, $z=c^T x$ et à la forme $\begin{cases} x_B = p + Qx_N \\ z = z_0 + dx_N \end{cases}$, avec:

- x_B : vecteur de variables de base
- x_N : vecteur de variables hors-base
- p : vecteur avec m coordonnées
- d : vecteur avec n coordonnées
- Q : matrice $n*m$
- z_0 : réel

Solution de base associée à $D(B)$: $x_N=0, x_B=p, z=z_0$.

Proposition:

Si B est une base admissible, alors $D(B)$ est défini de façon unique pour les formules:

- $Q = -A_B^{-1} A_N$
- $p = A_B^{-1} b$
- $z_0 = c_B^T A_B^{-1} b$
- $d = c_N - (c_B^T A_B^{-1} A_N)^T$

L'algorithme du simplexe

0. Calculer une base admissible et une solution base admissible (en utilisant le prog. lin. auxilliare). Si une telle base et solution de base existe pas, alors stop. Sinon,
1. Soit B la base admissible courante et soit $D(B)$ le dictionnaire associé.
2. Si $d \leq 0$, alors la solution base admissible $x_N = 0, x_B = b, z = z_0$ est optimale. Stop.
3. Sinon, choisir $x_u, u \in N$ avec $d_u > 0$ comme variable entrante. Si il y a plusieurs possibilités, utiliser la règle du pivot.
4. Choisir $x_v, v \in B$ la plus contrainte sur l'augmentation de x_u comme variable sortante.
5. Faire $B \leftarrow B \setminus \{u\} \cup \{v\}$, calculer $D(B)$ et aller à l'étape 1.

Optimalité:

Comme $z = z_0 + dx$ de $D(B)$ est égale à $c^t x$, si on choisit une autre solution admissible $x' = (x'_1, \dots, x'_{n+m})$, alors, comme $x'_j \geq 0, \forall j \in N$ et $d_j \leq 0, \forall j \in N$, on obtiendras que $c^t x' = z_0 + dx' \leq z_0$.

Règle du pivot:

1. (Dantzig) plus grand coefficient: choisir comme variable entrante la variable avec le plus grand coefficient positif.
2. plus grande croissance: la variable avec le coefficient positif qui augmente le plus la fonction objectif

3. (Bland) la variable avec coefficient positif et plus petit indice
4. steepest edge: maximiser le produit scalaire avec le vecteur
 $c: \max \frac{c^T(x_{NEW} - x_{OLD})}{\|x_{new} - x_{OLD}\|}$

4 Cours 3

4.1 Dualité dans la programmation linéaire

maximiser $2x_1 + 3x_2$ avec:

- $4x_1 + 8x_2 \leq 12$
- $2x_1 + x_2 \leq 3$
- $3x_1 + 2x_2 \leq 4$
- $x_1, x_2 \geq 0$

L'objectif est de calculer les bornes supérieures pour l'optimum sans résoudre le programme linéaire.

1. Notons que $2x_1 + 3x_2 \leq 4x_1 + 8x_2 \leq 12 \Rightarrow \text{OPT} \leq 12$
2. Notons aussi $2x_1 + 3x_2 \leq \frac{1}{2}(4x_1 + 8x_2) \leq 6 \Rightarrow \text{OPT} \leq 6$
3. Aussi $2x_1 + 3x_2 = \frac{1}{3}(4x_1 + 8x_2) + \frac{1}{3}(2x_1 + x_2) \leq \frac{12}{3} + \frac{3}{3} = 5 \Rightarrow \text{OPT} \leq 5$

On note (D), le programme linéaire dual du programme linéaire.

Cas général:

$$\left\{ \begin{array}{l} \text{maximiser } c_1x_1 + \dots + c_nx_n \\ a_{11}x_1 + \dots + a_{1n}x_n \leq b_1 \quad (y_1) \cancel{(y_1)} \\ \vdots \\ a_{m1}x_1 + \dots + a_{mn}x_n \leq b_m \quad (y_m) \end{array} \right. \quad (PL)$$

$$\left\{ \begin{array}{l} \text{minimiser } b_1y_1 + \dots + b_my_m \\ a_{11}y_1 + \dots + a_{m1}y_m \geq c_1 \\ \vdots \\ a_{1n}y_1 + \dots + a_{mn}y_m \geq c_n \end{array} \right. \quad (D)$$

$$\left\{ \begin{array}{l} \text{maximiser } c^T x \\ Ax \leq b \quad (PL) \\ x \geq 0 \end{array} \right. \quad \left\{ \begin{array}{l} \text{minimiser } b^T y \\ A^T y \geq c \quad (D) \\ y \geq 0 \end{array} \right.$$
(3)

Proposition:

Si x est une solution admissible du programme linéaire, et y est une solution admissible du dual, alors $c^T x \leq b^T y$.

Théorème:

Si $x^* = (x_1^*, \dots, x_n^*)$ est une solution optimale du programme linéaire, alors le dual admet une solution optimale $y^* = (y_1^*, \dots, y_m^*)$, et $\sum_{j=1}^n c_j x_j^* = \sum_{i=1}^m b_i y_i^*$.

Donc, le programme linéaire et le dual ont la même valeur de l'optimum.

Proposition:

Soit $z = z_0 + \sum_{i=1}^{n+m} d_i x_i$, l'expression de la fonction objectif dans le dernier dictionnaire des programmes linéaires, avec

- $d_i = 0$ si $i \in B$
- $d_i \leq 0$ si $i \in N$

Alors, $y_1^* = -d_{n+1}, \dots, y_m^* = -d_{n+m}$ est une solution optimale du dual.

Conclusion:

Pour résoudre le programme linéaire et son dual à la fois, il faut résoudre soit l'un, soit l'autre. Le plus simple, de préférence.

4.2 Théorème des écarts complémentaire

Théorème:

une solution admissible x_1^*, \dots, x_n^* du primal est optimale si et seulement si il existe des nombres y_1^*, \dots, y_m^* tels que:

- si $\sum_{j=1}^n a_{ij}x_j^* < b_i$, alors $y_i^* = 0$
- si $x_j^* > 0$, alors $\sum a_{ij}y_i^* = c_j$

et tels que;

- $\sum a_{ij}y_i^* \geq c_j, j = 1, \dots, n$
- $y_i^* \geq 0, i = 1, \dots, m$

Dans ce cas, $y^* = (y_1^*, \dots, y_m^*)$ est une solution optimale du dual.

4.3 La signification économique du dual

- b_i - la quantité de la ressource i ;
- a_{ij} - la quantité de la ressource i consommé pour la fabrication d'une unité de produit j ;
- c_j - la valeur unitaire du produit j .

La relation à l'optimum $Z_0 = \sum_{j=1}^n c_j x_j^* = \sum_{i=1}^m b_i y_i^*$

implique que y_i doit représenter la "valeur unitaire de la ressource i ", appelée aussi "prix implicite" de la ressource i ", i.e. le montant maximum que l'on serait prêt à payer pour obtenir une unité supplémentaire de la ressource i .

4.4 Problème dual-réalisable

Un programme linéaire est dual-réalisable si les coefficients c_j de la fonction objectif sont tous négatifs. Dans ce cas, à la place de la méthode à deux phases, on peut utiliser le dual.

Exemple: $\left\{ \begin{array}{l} \text{maximiser } -x_1 - x_2 \\ -3x_1 - x_2 \leq -4 \quad (\text{P}) \\ 7x_1 - x_2 \leq 7 \\ x_1, x_2 \geq 0 \end{array} \right.$

$$\left\{ \begin{array}{l} \text{minimiser } -4y_1 + 7y_2 \\ -3y_1 + 7y_2 \geq -1 \quad (\text{D}) \\ -y_1 - y_2 \geq -1 \\ y_1, y_2 \end{array} \right. \Leftrightarrow \left\{ \begin{array}{l} \text{maximiser } 4y_1 - 7y_2 \\ 3y_1 - 7y_2 \leq 1 \\ y_1 + y_2 \leq 1 \\ y_1, y_2 \geq 0 \end{array} \right.$$

5 Cours 4

5.1 Modélisation par flots

L'objectif de la modélisation par flots est de:

- modéliser des problèmes par des modèles de flots
- modéliser les problèmes de flots comme programmes linéaires

Excès d'un sommet:

L'excès du sommet v est le flot entrant dans v moins le flot sortant de v .

Lemme:

La valeur d'un flot est l'excès du sommet T , ou l'opposé de l'excès du sommet s .

Théorème:

\forall flot f , et tout (s,t) -coupe (S, \bar{S}) , $\text{val}(f) \leq c(S, \bar{S})$.

\exists un flot f^* et une (s,t) -coupe (S^*, \bar{S}^*) tels que $\text{val}(f^*) = c(S^*, \bar{S}^*)$.

Alors, f^* est un flot max, et (S^*, \bar{S}^*) une coupe minimum.

Lemme:

Si il existe une circulation avec demandes, alors $\sum_{v \in V} d_v = 0$, c'est à dire $\sum_{v: d_v > 0} d_v = -\sum_{v: d_v < 0} d_v =: D$.

Le problème de flot de valeur d se réduit à la circulation eb posant $d_s = -d$, $d_t = d$, et $d_v = 0$, $\forall v \neq s, t$. Le problème de circulation se réduit au flot max.

5.2 Circulation et flots avec demandes et bornes inférieures et bornes supérieures et bornes nippées de capacités

Circulation: $N = (V, A, l, c, d)$

$l: A \rightarrow \mathbb{R}^+$ - les bornes inf des capacités

$c: A \rightarrow \mathbb{R}^+$ - les bornes sup (classiques) des capacités

$d: V \rightarrow \mathbb{R}$ - les demandes des sommets

Si $l(\vec{uv}) = c(\vec{uv})$, cela oblige de passer par l'arc \vec{uv} en quantité $c = (\vec{uv})$.

Flot de coût minimum: $N = (V, A, s, t, c, q)$

$c: A \rightarrow \mathbb{R}^+$ - les capacités des arcs

$q: A \rightarrow \mathbb{R}^+$ - les coûts des arcs

$q(\vec{uv})$ - le coût de transport d'une unité de flot sur l'arc \vec{uv}

Cout d'un flot f : $\text{cout}(f) = \sum_{\vec{uv} \in A} q(\vec{uv}) \cdot c(\vec{uv})$.

Le flot de cout minimal généralise deux problèmes classiques:

1. le problème de transport
2. le problème d'affectation

Problème de transport:

Déterminer, de façon optimale, l'acheminement des biens à partir de n entrepôts vers m destinataires.

But: Minimiser le coût de transport.

Cas général: n entrepôts E_1, \dots, E_n ,
m destinataires D_1, \dots, D_m

a_i - la quantité du bien dans
l'entrepôt E_i , $i = \overline{1, \dots, n}$

b_j - la demande du destinataire
 D_j , $j = \overline{1, \dots, m}$

On suppose $\sum_{i=1}^n a_i = \sum_{j=1}^m b_j$ (tout le bien doit
être transporté)

d_{ij} - la distance entre E_i et D_j

c'_{ij} - le coût de transport d'une
unité de bien/km de l'entre-
pôt E_i vers D_j .

Le problème d'affectation:

n agents $A_1 \dots A_n$ et n tâches $T_1 \dots T_n$.

Soit c_{ij} , le prix demandé par l'agent A_i pour effectuer la tâche T_j .

Chaque agent peut réaliser une seule tâche, et chaque tâche ne peut être réalisée que par un seul agent.

But: Trouver une affectation de coût total minimal.

D'autres problèmes de flot exprimables en PL :

- **multiflots** : dans le réseau circule k types de flots entre les sources s_i et les puits $t_i, i=1, \dots, k$.
- **flot de coût minimum généralisé** : chaque sommet v possède aussi une demande de flot $b(v)$, qui représente la quantité totale de flot passant par v .
- **flot avec multiplicateurs** : une unité de flot passant par un arc \vec{uv} peut résulter dans un quantité $m(\vec{uv}) > 1$ (on gagne du flot) ou une quantité $M(\vec{uv}) < 1$ (on perd du flot) quand il arrive à l'extrémité v de l'arc,

6 Cours 5

6.1 Beyond flows

6.1.1 MAX-SAT comme PL(en nombre entiers)

Entrée: Une formule booléenne $f = c_1 \wedge \dots \wedge c_m$ en forme normale conjonctive avec n variables x_1, \dots, x_n .

Sortie: Une affectation 0-1 des variables dont le poids total des clauses satisfait soit max.

Algorithme d'approximation:

1. Dériver l'affectation (x'_1, \dots, x'_n) où

$$\forall j, x'_j = \begin{cases} 1 & \text{avec la probabilité } \frac{1}{2} \\ 0 & \text{avec la probabilité } \frac{1}{2} \end{cases}$$

2. Résoudre le PL relaxe et soit (x^*_1, \dots, x^*_n) une solution optimale.

3. Dériver l'affectation (x''_1, \dots, x''_n) où

$$\forall j, x''_j = \begin{cases} 1 & \text{avec la probabilité } x_i^* \\ 0 & \text{avec la probabilité } 1 - x_i^* \end{cases}$$

4. Retourner la meilleure parmi (x'_1, \dots, x'_n) et (x''_1, \dots, x''_n) .

Théorème:

Le poids espéré des clauses satisfaites par l'affectation de l'algorithme est $\geq \frac{3}{4}$ poids optimal. Donc, l'algo est un algorithme d'approximation facteur $\frac{3}{4}$.

6.1.2 Couverture de sommets

Entrée: Un graphe $G=(V,E)$ où chaque sommet $v \in V$ a un poids positif $w_v > 0$.

Une couverture de sommet est un sous ensemble $S \subset V$ tel que, \forall arrête uv de G , $\{u,v\} \cap S \neq \emptyset$. Le poids de S est $w(S) = \sum_{v \in S} w_v$.

But: Trouver une couverture de poids minimal de G .

Difficulté: NP-complet déjà dans le cas non pondéré ($w_v = 1$, $\forall v \in V$).

Lemme:

Les solutions admissibles de PLNE sont en bijection avec les couvertures de sommets de G .

Théorème:

$S = \{v \in V : x_v^+ = 1\}$ est une couverture de sommet de G . De plus, $w(S) \leq 2OPT$.

7 Informations importantes

Il n'y aura que deux séances de tds et tps, il faut donc bosser chez soi.