Miniprojet de recherche opérationnelle

A rendre pour le 14 janvier 2021

Les projets sont à effectuer par groupe de trois élèves, et à rendre sur `educnet`, avant le 14 janvier, 23h59. Merci de rendre

- Un rapport au format `pdf` nommé `rapport_GROUPE.pdf`. Ce rapport devra :
  - contenir le numéro du groupe et le nom et prénom de chacun des élèves
  - être d'une longeur d'*au plus 4 pages* dans une police au moins 11 (-*2 points par page excédentaire*)
- Le fichier `europe_GROUPE.txt` contenant la solution de l'instance `europe.csv` de la partie 3.2

Dans les noms des deux fichiers, `GROUPE` est à remplacer par votre numéro de groupe. Merci de respecter la nomenclature et les formats txt et pdf (-*1 point pour ceux qui ne les respectent pas*).

1 Position du problème

1.1 Contexte de l'étude et concepts principaux

Cadre de l'étude : dispatch des emballages propres et vides. Renault met à disposition de ses fournisseurs des emballages pour assurer le transport des pièces depuis ses fournisseurs vers ses usines. La logistique des emballages peut être décrite de la manière suivante.

- Les fournisseurs utilisent des emballages pleins pour envoyer des pièces vers les usines.
- Une fois arrivés à l'usine, les emballages sont vidés. Les emballages vides sont ensuite renvoyés vers les fournisseurs.

Ce document introduit une modélisation du problème de dispatch des emballages propres et vides chez Renault. Dans la suite du document, le mot emballage fait référence à des emballages propres et vides.

Usines et fournisseurs. On considère le problème suivant. Renault souhaite organiser le dispatch de ses emballages vides depuis ses usines et sites de lavages vers ses *fournisseurs*. Un *site libérateur* est un site qui libère des emballages vides, c'est à dire une usine ou un site de lavage. On notera respectivement U l'*ensemble des sites libérateurs*, F l'*ensemble des fournisseurs*, et l'on indexera par u les sites libérateurs dans U, et par f les fournisseurs dans F.

Les *emballages* sont utilisés pour le transport de pièces depuis n'importe quel fournisseur vers n'importe quelle usine. On note E l'ensemble des types emballages.

Horizon. Le dispatch des emballages est optimisé sur un horizon temporel fixé. On note J le nombre de jours dans cet horizon. On note $[J]$ l'ensemble $\{1, \ldots, J\}$. Les jours de l'horizon sont notés j. De manière générale, étant donné un entier naturel n, on note $[n]$ l'ensemble $\{1,\ldots,n\}$.

Diversité et régularité des livraisons chez un fournisseurs. Un fournisseur n'utilise pas forcément tous les types d'emballages. Pour les emballages qu'il utilise, un fournisseur n'est pas forcément livré toutes les semaines en emballages vides : une livraison peut lui permettre de tenir plusieurs semaines.

Nom	Description
j	Jour
J	Horizon
u	Site libérateur
U	Ensemble des sites libérateurs
f	Fournisseur
F	Ensemble des fournisseurs
e	Type d'emballage
E	Ensemble des types d'emballage
b^+_{euj}	Nombre de piles d'emballages de type e libéréss le jour j en u
b^-_{efj}	Nombre de piles de e consommées le jour j en f
r_{euj}	Stock maximum de piles de e en u le soir du jour j
r_{efj}	Stock maximum de piles de e en f le soir du jour j
V	Ensemble $U \cup F$ des sommets du graphe des lieux
v	Sommets de V
A	V^2 : ensemble des arêtes
a	Arête
d_a	Distance kilométrique correspondant à l'arête a
P	Chemin dans le graphe
R	Route suivie par un camion
ℓ_e	Métrage linéaire (longueur) d'une pile de e
L	Métrage linéaire (longueur) d'un camion
$Q = (Q_{ev})$	Chargement détaillé d'une route (par type d'emballage et fournisseur)
$\boldsymbol{q} = (q_e)$	Chargement agrégé d'une route au départ d'une usine (par type d'emballage)
c^{s}_{eu}	Coût unitaire d'un stock excédentaire de e en u par nuit
c^{s}_{ef}	Coût unitaire d'un stock excédentaire de e en f par nuit
c^{exc}_{ef}	Coût d'expédition d'une pile de pièces par f dans un carton au lieu d'un emballage de type e
γ	Coût kilométrique des trajets de camion
c^{cam}	Coût unitaire d'utilisation d'un camion
c^{stop}	Coût unitaire d'un arrêt d'un camion

TABLE 1: Données

Nom	Description
z^-_{euj}	Nombre de e expédiés de u le jour j
z^+_{efj}	Nombre de e livrés à f le jour j
s_{euj}	Stock de e en u le soir du jour j
s_{efj}	Stock de e en f le soir du jour j
x_R	Nombre de camions parcourant la route R

TABLE 2: Variables

1.2 Demandes, stocks et coûts des usines et fournisseurs

Libération par les sites libérateurs Chaque site libérateur libère chaque jour un certain nombre d'emballages. Notons b^+_{euj} le nombre d'emballages de type e libérés le jour j par le site libérateur u. Ces emballages ne sont pas nécessairement envoyés le même jour. On considère que tous ces emballages sont libérés le matin, et peuvent être envoyés pendant la journée. On note z^-_{euj} la variable qui indique le nombre d'emballages de type e expédiés le jour j, et s_{euj} la variable qui indique le stock de e en u le soir du jour j. Le stock initial est une donnée d'entrée. On a

$$s_{euj} = s_{eu(j-1)} + b^+_{euj} - z^-_{euj}, \quad \forall e, u, j. \tag{1}$$

Un site libérateur ne peut évidemment pas libérer des emballages qu'il ne possède pas. Le stock doit donc satisfaire la contrainte

$$0 \leq s_{euj}, \quad \forall e, u, j. \tag{2}$$

Chaque site libérateur u aimerait stocker au plus r_{euj} piles d'emballages de type e le soir du jour j. Ne pas satisfaire cette limite a un coût unitaire de c^{s}_{eu}. Le coût total de stockage dans les usines pour le jour j est donc modélisé par l'équation

$$\sum_{e,u} c^{\mathrm{s}}_{eu} \max(s_{euj} - r_{euj}, 0). \tag{3}$$

Demande par les fournisseurs. Les fournisseurs ont un planning de consommation d'emballages, qui suit leur planning de production de pièces. On note b^-_{efj} le nombre d'emballages consommés par le fournisseur pour emballer ses pièces le jour j. On note s_{efj} le stock d'emballages chez le fournisseur f le soir du jour j, et z^+_{euj} le nombre d'emballages e livrés à f le jour j. Le stock d'emballages chez le fournisseur devrait donc suivre la dynamique

$$s_{efj} = s_{ef(j-1)} - b^-_{efj} + z^+_{efj}, \quad \forall e, f, j \tag{4}$$

Et comme le stock doit toujours être positif, on impose

$$0 \leq s_{efj}, \quad \forall e, f, j. \tag{5}$$

En pratique, il n'est pas toujours possible de satisfaire la contrainte (5). Par conséquent, lorsqu'un fournisseur n'est pas livré en temps et en heure, il utilise des cartons ordinaires pour emballer ses pièces. On remplace ainsi les contraintes (4) et (5) par la nouvelle dynamique

$$s_{efj} = \max(s_{ef(j-1)} - b^-_{efj}, 0) + z^+_{efj}, \quad \forall e, f, j. \tag{6}$$

Cette équation modélise le fait que les pièces dont l'emballage est prévu le jour j sont effectivement emballées le jour j, et que s'il n'y a pas d'emballage disponible, celles-ci sont emballées dans des cartons.

Deux coûts sont liés à la dynamique (6) :

— Un coût c^{exc}_{ef} par pièce expédiée dans des cartons, ce qui donne pour le jour j un coût total

$$\sum_{e,f} c^{\text{exc}}_{ef}(\max(b^-_{efj} - s_{ef(j-1)}, 0)) \tag{7a}$$

— Et un coût lié à un stock trop important chez le fournisseur, qui dépasse sa capacité idéale r_{efj}. Pour le jour j, on obtient un coût total de

$$\sum_{e,f} c^{\text{s}}_{ef} \max(s_{efj} - r_{efj}, 0). \tag{7b}$$

Unité de mesure L'unité de mesure pour le transport est la pile d'emballages : toutes les quantités d'emballages sont exprimées en piles.

1.3 Transport depuis les usines et coûts afférents

Le transport des emballages entre les sites libérateurs u et les fournisseurs f est réalisé par des camions.

Chemin Un camion de livraison charge dans une seule usine, puis il peut aller livrer au plus 4 fournisseurs. Ceci est modélisé avec la notion de chemin admissible, introduite ci-dessous. On note P un *chemin* dans le graphe (V, A). Un chemin est *admissible* s'il est de la forme $v_0, v_1, \ldots, v_n$ où $u := v_0$ est une usine, $n \leq 4$ et $v_1, \ldots, v_n$ sont des sommets de F. Nous soulignons que les chemins ne retournent pas à l'usine à la fin : ils se terminent au dernier fournisseur livré.

Route Une *route* R est un triplet $(j_R, P_R, \boldsymbol{q}_R)$ tel que :

— j_R est le jour de départ d'un camion suivant la route R

— P_R est un chemin admissible $v_0 = u, v_1, \ldots, v_n$ indiquant l'usine de départ et les fournisseurs livrés par un camion suivant R

— $\boldsymbol{q}_R$ désigne le chargement d'un camion suivant R (détaillé juste après)

Il n'y a pas de limite sur le nombre de routes utilisées. On note x_R la variable dans $\mathbb{Z}_+$ qui indique le nombre de camions empruntant la route R, et c_R le coût (par camion) de la route R. Décrivons maintenant ce que contiennent les camions suivant une route.

Chargement On note ℓ_e le métrage linéaire (la longueur) d'une pile d'emballages e, et L le métrage linéaire d'un camion (identique pour tous).

Un chargement détaillé est un vecteur $Q_R = (Q_{evR})_{e\in E, v\in\{v_1,\dots,v_n\}}$ contenant le nombre d'emballages de type e livrés par un camion individuel à chaque arrêt v_i de la route R. On note $\boldsymbol{q}_R = (q_{eR})_{e\in E}$ le chargement agrégé associé à Q_R, qui contient les quantités de chaque emballage chargées dans ce même camion en partant de l'usine :

$$q_{eR} = \sum_{i=1}^{n} Q_{ev_iR}$$

Bien entendu, le chargement agrégé doit pouvoir rentrer dans le camion au départ, donc sa longeur totale doit être inférieure à L :

$$\sum_{e\in E} \ell_e q_e = \sum_{e\in E}\sum_{i=1}^{n} \ell_e Q_{ev_iR} \leq L \tag{8}$$

On définit aussi le chargement livré par un camion au fournisseur v_i comme le vecteur $\boldsymbol{q}_{v_iR} = (Q_{ev_iR})_{e\in E}$.

Simplification sur la durée des routes. En théorie, la route R peut durer plusieurs jours. Elle visite alors le sommet v_i au jour j_{v_iR} défini comme

$$j_{v_0R} = j_R \quad \text{et} \quad j_{v_iR} = j_{v_{i-1}R} + t_{(v_{i-1},v_i)}.$$

Pour simplifier, on supposera qu'une route est effectuée sur une seule journée : dans la même journée, le camion charge les emballages auprès du site libérateur et les livre auprès des fournisseurs, quelle que soit la distance parcourue. Et le fournisseur pourra utiliser les emballages livrés le jour j dès le lendemain $j+1$. Ce n'est évidemment pas le cas dans la réalité, mais nous ferons cette hypothèse pour éviter des difficultés techniques impactant peu l'optimisation.

Coût d'une route Le coût d'une route est de la forme :

$$c_R = c^{\text{cam}} + c^{\text{stop}}|P_R - 1| + \gamma \sum_{a\in P} d_a \tag{9}$$

où γ est le coût kilométrique, c^{stop} est un coût unitaire par arrêt, et c^{cam} un coût unitaire par route (coût fixe de contractualisation, etc.).

Remarque 1. Il peut être pertinent d'envoyer deux camions sur une même route R qui contient plusieurs fournisseurs. Par exemple, supposons que l'on ait deux piles de e_1 de longueur $3L/5$ à livrer chez un premier fournisseur f_1, et deux piles de e_2 de longueur $2L/5$ à envoyer chez un second fournisseur f_2 très proche de f_1, alors toute solution autre que (u, f_1, f_2) nécessite au minimum 3 camions. △

1.4 Contraintes liantes entre les stocks et les routes

Routes incidentes Un camion suivant la route R va charger le chargement $\boldsymbol{q}_R$ et quitter l'usine $u = v_0$ le jour j_R, pour livrer à chaque fournisseur v_i le chargement $\boldsymbol{q}_{v_iR}$ qui lui correspond. Étant donnés un sommet v et un jour j, on note $\delta^+(v,j)$ l'*ensemble des routes quittant* v *le jour* j, et $\delta^-(v,j)$ l'*ensemble des routes qui livrent* v *le jour* j.

Routes et stock Les équations suivantes modélisent l'évolution des stocks en fonction des routes utilisées :

$$z^-_{euj} = \sum_{R\in\delta^+(u,j)} x_R\, q_{eR}, \qquad \forall e, \forall u, \forall j, \tag{10a}$$

$$z^+_{efj} = \sum_{R\in\delta^-(f,j)} x_R\, Q_{efR}, \quad \forall e, \forall f, \forall j. \tag{10b}$$

où l'on rappelle que si le fournisseur f est livré par la route R, alors f est un des sommets v_i de R et Q_{efR} correspond à Q_{ev_iR}.

1.5 Problème de dispatch

En résumé, le problème d'optimisation à résoudre est le suivant :

$$\min_{s,x,z} \left| \begin{array}{l} \displaystyle\sum_{r\in R} c_R x_R \\ + \sum_{euj} c^{\mathrm{s}}_{eu} \max(s_{euj} - r_{euj}, 0) \\ + \sum_{efj} c^{\mathrm{s}}_{ef} \max(s_{efj} - r_{efj}, 0) \\ \displaystyle + \sum_{e,f,j} c^{\mathrm{exc}}_{ef} (\max(b^-_{efj} - s_{efj-1}, 0)) \end{array} \right. \tag{11a}$$

$$\text{s.t.} \quad (1), (2), (6) \text{ et } (10) \tag{11b}$$

2 Instance et solution

2.1 Instance

Les instances sont données au format `.csv`. Voici ci-dessous l'exemple de l'instance `petite.csv`. Le symbole $\hookrightarrow$ indique que c'est la même ligne du fichier qui est affichée sur plusieurs lignes. Toutes les quantités donnée sont des entiers. Les métrages linéaires sont données en mm.

```
J 3 U 2 F 3 E 2 L 13000 Gamma 1 CCam 100 CStop 10
e 0 l 300
e 1 l 200
u 0 v 0 coor 48.8666672 2.3514623 ce e 0 cr 2 b 4 e 1 cr 4 b 5 lib
    ↪ j 0 e 0 b 5 r 5 e 1 b 0 r 5 j 1 e 0 b 10 r 5 e 1 b 5 r 5 j
    ↪ 2 e 0 b 0 r 5 e 1 b 5 r 5
u 1 v 1 coor 43.6044622 1.4442469 ce e 0 cr 2 b 3 e 1 cr 4 b 6 lib
    ↪ j 0 e 0 b 2 r 5 e 1 b 3 r 5 j 1 e 0 b 2 r 5 e 1 b 3 r 5 j 2
    ↪ e 0 b 2 r 5 e 1 b 3 r 5
f 0 v 2 coor 48.8666672 2.3514623 ce e 0 cr 2 cexc 55 b 2 e 1 cr 4
    ↪ cexc 30 b 6 dem j 0 e 0 b 2 r 5 e 1 b 1 r 5 j 1 e 0 b 2 r 5
    ↪ e 1 b 1 r 5 j 2 e 0 b 2 r 5 e 1 b 1 r 5
f 1 v 3 coor 45.7578137 4.8320114 ce e 0 cr 2 cexc 45 b 4 e 1 cr 4
    ↪ cexc 40 b 5 dem j 0 e 0 b 0 r 5 e 1 b 2 r 4 j 1 e 0 b 0 r 6
    ↪ e 1 b 2 r 4 j 2 e 0 b 0 r 3 e 1 b 2 r 4
f 2 v 4 coor 45.7578137 4.8320114 ce e 0 cr 2 cexc 35 b 7 e 1 cr 4
    ↪ cexc 50 b 4 dem j 0 e 0 b 3 r 5 e 1 b 0 r 5 j 1 e 0 b 3 r 5
    ↪ e 1 b 0 r 5 j 2 e 0 b 3 r 5 e 1 b 0 r 5
a 0 0 d 0
a 0 1 d 10
a 0 2 d 0
a 0 3 d 8
a 0 4 d 8
a 1 0 d 10
a 1 1 d 0
a 1 2 d 8
a 1 3 d 5
a 1 4 d 5
a 2 0 d 0
a 2 1 d 8
a 2 2 d 0
a 2 3 d 8
a 2 4 d 8
a 3 0 d 8
a 3 1 d 5
a 3 2 d 8
a 3 3 d 0
a 3 4 d 0
a 4 0 d 8
a 4 1 d 5
```

```
a 4 2 d 8
a 4 3 d 0
a 4 4 d 0
```

La première ligne contient des constantes générales :

- `J`, suivi du nombre de jours dans l'horizon,
- `U`, suivi du nombre d'usines (sites libérateurs)
- `F`, suivi du nombre de fournisseurs
- `E`, suivi du nombre de types d'emballages
- `L`, suivi du métrage linéaire L d'un camion
- `Gamma`, suivi du coût kilométrique γ
- `CCam`, suivi du coût fixe par route c^{cam}
- `CStop`, suivi du coût fixe par arrêt camion c^{stop}

Les `E` lignes suivantes commencent par `e` et contiennent les propriétés des emballages :

- `e` suivi de l'id de l'emballage (les emballage sont numérotés de 0 à `E-1`.)
- `l` suivi du métrage linéaire ℓ_e de l'emballage

Les `U` lignes suivantes commencent par `u` et contiennent les propriétés des sites libérateurs (usines) :

- `u` suivi de l'id de l'usine (les usines sont numérotées de 0 à `U-1`)
- `v` suivi de l'id du sommet où est situé l'usine (identique à l'id de l'usine)
- `coor` suivi des coordonnées GPS de l'usine (non utilisées dans la définition du problème, mais utiles pour des visualisations ou heuristiques)
- `ce` suivi des constantes de l'usine pour les emballages : pour chaque emballage
  - `e` suivi de l'id e de l'emballage
  - `cr` suivi du coût de stock excédentaire c^{s}_{eu} de e en u
  - `b` suivi du stock initial d'emballage e en u
- `lib` suivi des données de libération de l'usine : pour chaque jour j
  - `j` suivi de l'index du jour j, puis pour chaque emballage
    - `e` suvi de l'id e du type d'emballage
    - `b` suivi de b^+_{euj}, la quantité d'emballage e libérée par u le jour j
    - `r` suivi de r_{euj}, le stock maximal d'emballage de type e en u le soir du jour j.

Les `F` lignes suivantes commencent par f et contiennent les propriétés des fournisseurs :

- `f` suivi de l'id de du fournisseur (les fournisseurs sont numérotés de 0 à `F-1`)
- `v` suivi de l'id du sommet où est situé le fournisseur (égal à `f` + `U`)
- `coor` suivi des coordonnées GPS du fournisseur (non utilisées dans la définition du problème, mais utiles pour des visualisations ou heuristiques)
- `ce` suivi des constantes du fournisseur pour les emballages : pour chaque emballage
  - `e` suivi de l'id e de l'emballage
  - `cr` suivi du coût de stock excédentaire c^{s}_{ef} de e en f
  - `cexc` suivi du coût d'expédition en carton c^{exc}_{ef} de e en f
  - `b` suivi du stock initial d'emballages e en f
- `dem` suivi des données de demande du fournisseur : pour chaque jour j
  - `j` suivi de l'index du jour j, puis pour chaque emballage

— `e` suvi de l'id e du type d'emballage

— `b` suivi de b^-_{efj}, la quantité d'emballages e consommés par f le jour j

— `r` suivi de r_{efj}, le stock maximal d'emballages de type e en f le soir du jour j.

Les $(\mathtt{U}+\mathtt{F})^2$ lignes suivantes commencent par a et décrivent les arcs du graphe des routes. Plus précisément, elles contiennent :

— `a` suivi de du sommet origine o de l'arc puis du sommet destination d de l'arc $a = (o, d)$

— `d` suivi de la distance kilometrique de o à d

Le graphes est complet : ces arcs peuvent être vus comme la matrice des distances entre chaque paire de sommets du graphe.

2.2 Solutions

Les solutions doivent être retournées au format `.txt`. Voici ci-dessous l'exemple d'une solution `petite.txt` de l'instance `petite.csv` définie ci-dessus.

```
R 3
r 0 j 0 x 1 u 0 F 1 f 0 e 0 q 0 e 1 q 2
r 1 j 1 x 1 u 0 F 3 f 0 e 0 q 1 e 1 q 1 f 1 e 0 q 1 e 1 q 1 f 2 e
    0 q 1 e 1 q 1
r 2 j 1 x 2 u 1 F 2 f 1 e 0 q 3 e 1 q 1 f 2 e 0 q 0 e 1 q 1
```

La première ligne commence par `R` et contient le nombre de routes dans la solution. Le reste du fichier contient une ligne par route. Chaque ligne est composée de la manière suivante.

— `r` suivi de l'id de la route (utilisé dans les messages d'erreur du code de vérification des solutions)

— `j` suivi du jour où la route est effectuée

— `x` suivi d'un entier donnant le nombre de fois que la route est opérée (nombre de camions qui l'empruntent)

— `u` suivi de l'id de l'usine u d'où part la route

— `F` suivi d'un entier indiquant le nombre de fournisseurs visités par la route. Ensuite, pour chaque fournisseur, dans l'ordre où ils sont visités par la route :

  — `f` suivi de l'id du fournisseur f, puis pour chaque type d'emballage e (par index croissant de 0 à $\mathtt{E}-1$)

    — `e` suivi de l'index de l'emballage e

    — `q` suivi de Q_{efr} la quantité de e livrée à f par r

Lorsqu'une même route est prise plusieurs fois, il est possible de l'écrire sur une seule ligne (grâce à la variable `x`) ou sur plusieurs.

2.3 Codes de vérification des solutions

Il vous est fourni un code `C++` permettant de vérifier les solutions. Le document `Readme.md` qu'il contient indique comment le compiler. Vous pouvez en réutiliser tout ou partie à votre convenance si vous réalisez votre projet en `C++`. Lancer l'exécutable sans arguments permet d'obtenir la liste des arguments attendus.

Un code `Julia` est également inclus, sous forme de notebook Jupyter `.ipynb` et de fichier au format `.jl`.

3 Ce qui est demandé

3.1 Questions (5 points)

1. Montrer que si il y a un seul type d'emballage ($E = 1$) et que celui-ci vérifie $\ell_1 > L/2$, le problème peut être modélisé comme un problème de flot (*2.5 points*).
2. Pourquoi le nombre de variables de la formulation PLNE (11) est il polynomial en la taille de l'instance ? Est-il vraisemblable de résoudre l'instance `europe.csv` fournie avec ce PLNE (*0.5 points*)
3. Sous l'hypothèse qu'il y a au plus 5 routes qui livrent un fournisseur donné chaque jour, donner une formulation PLNE du problème (avec tous les emballages) comprenant $O\big(|E|J(|F|+|U|)^2\big)$ variables. (*2 points*)

3.2 Algorithme (15 points)

Il vous est fourni un instance `europe.csv`. Par une/des stratégie(s) de votre choix, proposer une solution pour cette instance.

- Vous rendrez un fichier au format voulu donnant votre solution pour l'instance fournie.
- La qualité des stratégies de résolution utilisées et de leur présentation dans le rapport sera notée sur *9 points*.
  - Qualité de l'approche fournie (*5 points*)
    - Prouver tout résultat intéressant sur ces approches (exactitude, complexité, etc.).
  - La rigueur et la qualité de la rédaction seront particulièrement prises en compte. (*4 points*)
    - Les algorithmes doivent être fournis de manière lisible (https://en.wikipedia.org/wiki/Pseudocode)
    - Pour les résultats numériques, fournir les coûts des solutions, leur optimalité / gap.
    - Les résultats numériques doivent être présentés de manière lisible (tableau, figure légendée).
- La qualité des solutions fournies sera notée sur *6 points*.
  - Le score d'une équipe est le coût minimal obtenu sur l'instance `europe.csv`. Les équipes seront classées par score (l'équipe avec le score le plus faible aura la meilleure note).
    Sur les 6 points, 4 seront liés au classement :
    - 4 points + 2 points bonus pour le premier groupe
    - 4 points + 1 points bonus pour le second
    - 4 points le troisième groupe

    - 3 points pour ceux qui sont dans le premier quart
    - 2 pour ceux qui sont dans le deuxième quart
    - 1 point pour ceux qui sont dans le troisième quart
    - 0 pour les autres
  - Vos solutions seront vérifiées avec le code de vérification qui vous est fourni. Merci de m'indiquer par mail à axel.parmentier@enpc.fr si vous identifiez une erreur dans le sujet ou dans le code de vérification.

Le problème est difficile à résoudre.

- Ne commencez pas le projet à la dernière minute. Posez des questions aux enseignants de vos groupes.
- N'hésitez pas à le simplifier / décomposer pour obtenir des solutions admissibles.
- Si vous le souhaitez, vous pouvez utiliser des solveurs pour résoudre le problème. Néanmoins, l'instance étant de grande taille, il est peu probable qu'un solveur puisse trouver une bonne solution s'il est appliqué à la totalité de l'instance. Cela ne vous empêche pas d'utiliser un solveur pour un sous-problème.
- Pour contribuer à la motivation des groupes, un Google Sheets a été créé [1] pour vous permettre d'y inscrire vos résultats et de les comparer avec les autres, sans pour autant partager vos méthodes.

1. https://docs.google.com/spreadsheets/d/17pbo_dCn_YzBPpsNPZ3pMP4B2vuVJ5pzxF4FxBQ3tB8/edit?usp=sharing