



1816
2016



ÉCOLE NATIONALE SUPÉRIEURE DES MINES

FOUNDEUR
1816 - 1817

Challenge Optim 2025

23/06/2025



Un problème de chargement de camions



Inspiré du challenge Renault ROADEF/EURO 2022 :
<https://roadeff.org/challenge/2022/en/>

Le sujet

Données d'entrée (ensembles) :

- I : ensemble de marchandises (objets) à transporter
- S : ensemble de fournisseurs
- T : ensemble de types de camions à disposition

Objectif général du problème : Positionner (sous forme de piles, en 3D) les marchandises à transporter dans les camions pour les acheminer des fournisseurs vers l'usine (unique) de fabrication

Les données

Données associées aux marchandises $i \in \mathcal{I}$

- L_i : longueur
- W_i : largeur
- H_i : hauteur
- M_i : poids
- S_i : code d'empilement
- O_i : fournisseur où doit être chargée la marchandises (origine)
- HE_i : jour d'arrivée au plus tôt à l'usine de fabrication (interdit avant)
- HL_i : jour d'arrivée au plus tard à l'usine de fabrication (interdit après)
- C_i : coût de stockage journalier à l'usine de fabrication

Les données

Données associées aux types de camions $t \in \mathcal{T}$

- L_t : longueur
- W_t : largeur
- H_t : hauteur
- M_t : poids maximal de chargement autorisé
- \mathcal{O}_t : **liste** de fournisseurs où le camion peut charger des marchandises
- HC_t : jour d'arrivée du camion à destination
- C_t : coût fixe d'utilisation du camion

Remarque :

- Il s'agit bien de types de camions : il est possible d'utiliser autant de camions que l'on souhaite pour un type donné ; il y aura un coût fixe C_t à payer pour chaque camion
- \mathcal{O}_t est une liste de fournisseurs : il est possible de charger des marchandises dans un camion à partir de n'importe quel fournisseur de cette liste, sans impact sur le coût

Contraintes

C1 : Toutes les marchandises doivent être chargées dans un camion

C2 : Le fournisseur d'origine d'une marchandise doit appartenir à la liste de fournisseurs du camion sur lequel il est chargé

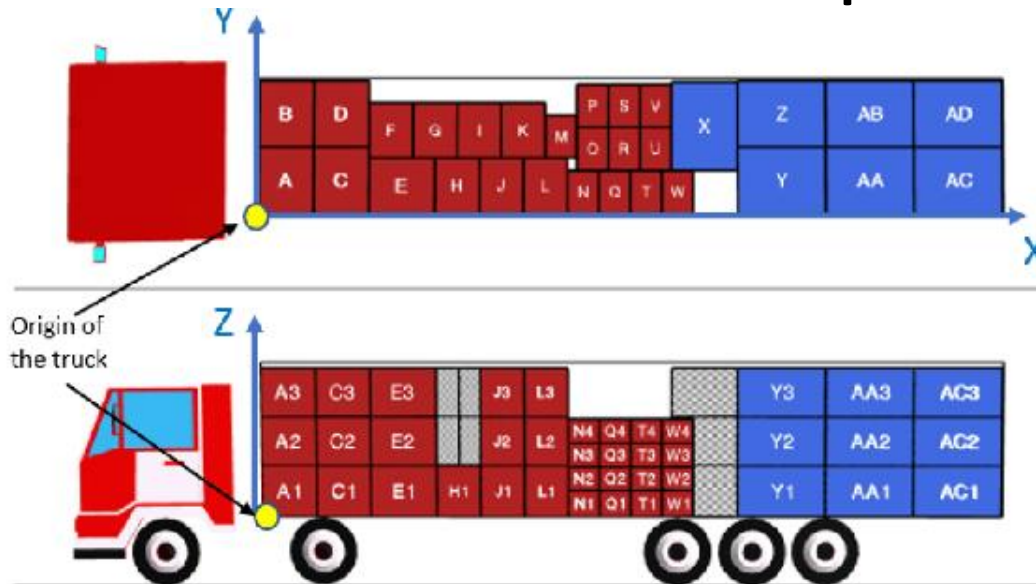
C3 : Le jour d'arrivée d'un objet $i \in \mathcal{I}$ à l'usine de fabrication doit appartenir à la plage $[HE_i, HL_i]$

C4 : Des marchandises empilées doivent avoir le même code d'empilement S_i

C5 : Toute marchandise ou pile de marchandises doit être calée vers le fond du camion (contre un autre objet ou contre le fond du camion)

C6 : Le poids maximal de chargement du camion doit être respecté

Référentiel pour le positionnement



La coordonnée (0,0,0) correspond au point le plus :

- proche de la cabine (X=0)
- à gauche vue de l'arrière (Y=0)
- bas (Z=0)

C5 : Toute marchandise ou pile de marchandises doit être calée vers le fond du camion → la coordonnée X ne peut physiquement pas être diminuée

Remarques :

- des marchandises empilables ont toujours la même longueur et largeur (et auront les mêmes coordonnées X et Y si elles sont empilées)
- Il est autorisé de changer l'orientation d'une pile (longueur ↔ largeur)

Objectif

Minimiser la somme de deux types de coûts :

- **Les coûts associés au stockage des objets en usine**

Coût de stockage de l'objet i : $C_i \times (HL_i - H)$ où H est le jour d'arrivée effectif du camion

- **Les coûts associés à l'usage d'un camion**

Coût d'usage d'un camion de type t : C_t

Les entrées / sorties

15 instances sont proposées

3 fichiers par instances :

- **In**_input_items.txt,
- **In**_input_trucks.txt,
- **In**_supplier_trucks.txt,

où **n** est le numéro de l'instance

Format du fichier In_input_items.txt

nbObjets	1672								
Id_item	Supplier_code	Length	Width	Height	Weight	Stackability_code	Earliest_arrival_time	Latest_arrival_time	Inventory_cost
0	12	7	5	4	81	5	46	49	20
1	12	7	5	4	96	5	16	17	30
2	7	12	10	9	577	3	9	10	70
3	4	14	12	11	278	7	52	53	150
4	12	7	5	4	100	5	4	7	10
...

i **O_i** **L_i** **W_i** **H_i** **M_i** **S_i** **HE_i** **HL_i** **C_i**

Les entrées / sorties

Format du fichier In_input_trucks.txt

nbCamions	237					
Id_truck	Arrival_time	Length	Width	Height	Max_weight	Cost
0	45	145	24	28	30000	1500
1	3	150	25	30	24000	1500
2	3	150	25	30	24000	1500
3	15	150	25	30	24000	1500
4	15	150	25	30	24000	1500
...

t H_t L_t W_t HC_t M_t C_t

Format du fichier In_supplier_trucks.txt

nbCamions	747				
Id_truck	Number_of_suppliers	Supplier_code			
0	1	7			
1	1	15			
2	2	66	42		
3	4	72	47	48	23
4	1	56			
...

t $| \mathcal{Q}_t |$ liste des fournisseurs dans \mathcal{Q}_t

Les entrées / sorties

Sorties : un fichier par instance résolue

Nom proposé pour les fichiers solutions : **res_n.txt** où n est le numéro de l'instance

Attention : extension .txt obligatoire

Format du fichier :

Nom de l'équipe

Numéro de l'instance

EQUIPE	ZeProf							
INSTANCE	1							
TYPECAMION	76							
PILE	7	5	0	0				
OBJETS	0	215	244	361	391	443	450	
PILE	12	10	0	5				
OBJETS	11	313	545	562				
...
TYPECAMION	25							
PILE	12	4	0	0				
...

Attention : Nom de l'équipe au choix mais **sans caractères spéciaux ni accents**

Les entrées / sorties

Bloc à répéter
pour chaque pile

Bloc à répéter pour
chaque camion

EQUIPE	ZeProf							
INSTANCE	1							
TYPECAMION	76							
PILE	7	5	0	0				
OBJETS	0	215	244	361	391	443	450	
PILE	12	10	0	5				
OBJETS	11	313	545	562				
...
TYPECAMION	25							
PILE	12	4	0	0				
...

Longueur (axe des X), largeur (axe des Y) et coordonnées (x,y) du point d'ancrage de la pile*

Liste des marchandises
dans la pile

* : le point d'ancrage d'une pile est son point d'abscisse et d'ordonnée minimale ; les informations sur la longueur et la largeur permettent de savoir comment sont orientées les marchandises dans la pile

Algorithme ALGO1

Le principe de l'algorithme est d'affecter successivement chaque marchandise i à un camion, en partant d'une solution vide dans laquelle aucun camion n'est utilisé.

Pour chaque marchandise i , cette affectation se fait selon trois mécanismes successifs, un mécanisme n'étant déclenché qu'en cas d'échec du précédent :

1. Insertion au sommet d'une pile existante (et compatible) au sein des camions déjà planifiés
2. Insertion par la création d'une nouvelle pile dans un camion déjà planifié (et compatible)
3. Planification d'un nouveau camion selon l'un des types existants (compatible) en donnant la priorité à un type de camion arrivant à destination au plus proche de la date souhaitée

Algorithme (suite)

Structures de données

Il est conseillé d'introduire des types structurés pour les marchandises, les camions et les piles.

Ces structures seront en particulier très utiles si vous souhaitez faire des tris et continuer à accéder facilement aux données associées.

Structure de données (suite)

Pour les données dont on ne connaît pas le nombre d'éléments a priori (nombre de camions d'un certain type dans la solution, nombre de piles dans un camion, nombre de marchandises dans une pile), il ne faudra pas hésiter à utiliser des tableaux alloués de manière dynamique mais avec une taille suffisamment grande

Par exemple : le nombre de piles dans un camion ne peut pas dépasser $L_t \times W_t$, on peut donc allouer un tableau de cette taille pour éviter d'utiliser une liste chaînée plus complexe à manipuler...

Algorithme (suite)

Création d'une nouvelle pile dans un camion

Il sera utile de définir une fonction qui tente de créer une nouvelle pile dans un camion.

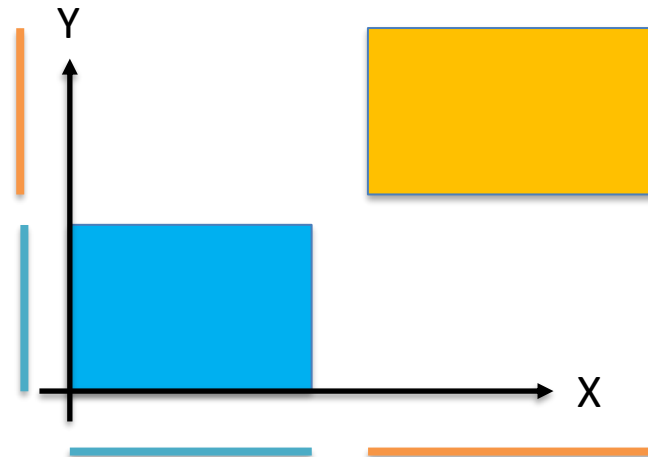
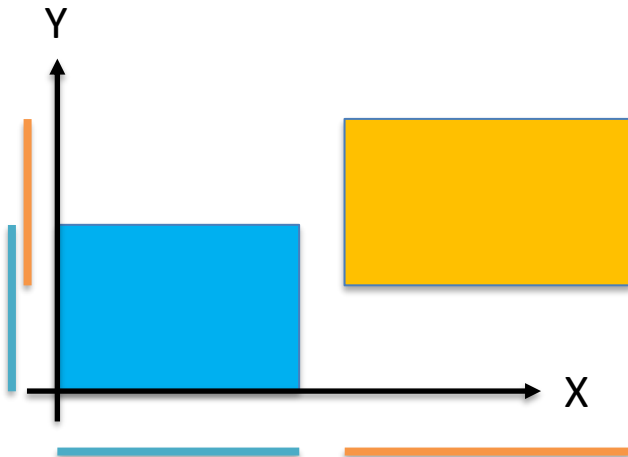
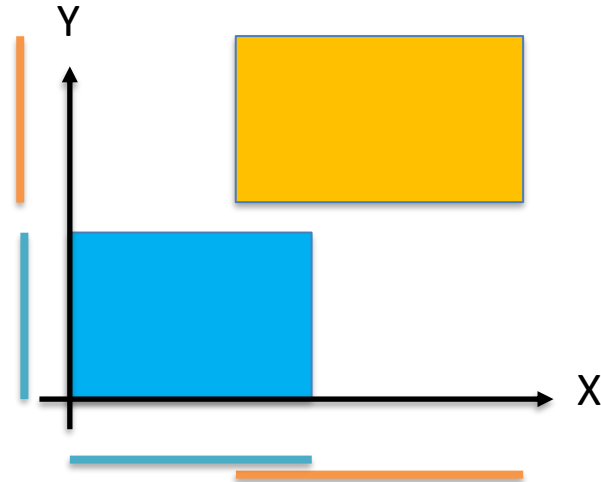
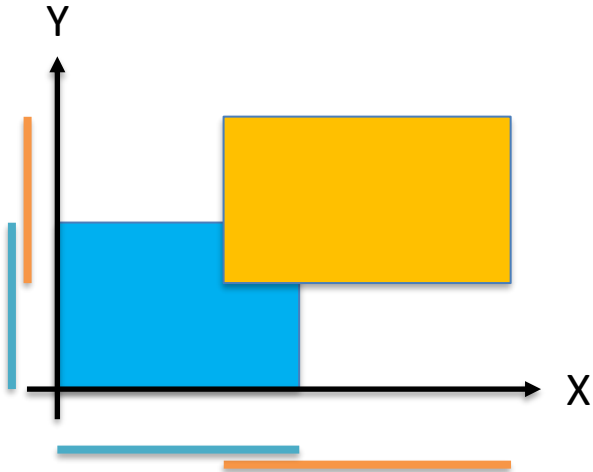
Un algorithme possible pour cette fonction est le suivant :

```
x* = -1, y* = -1
pour y variant de 0 à  $W_t - 1$ 
  pour x variant de 0 à  $L_t - 1$ 
    ok = true
    pour chaque pile p déjà présente dans le camion
      si placer i en (x,y) n'est pas compatible avec p
        ok = false
    si ok == true
      x* = x, y* = y, return true
return false
```

En cas de retour « true », x^* et y^* contiennent les coordonnées ou positionner la nouvelle pile. Un retour « false » indique que la pile ne peut pas être créée.

La condition pour que deux piles soient compatibles est que leurs projections sur l'axe des abscisses ou sur l'axe des ordonnées ne se superposent pas

Algorithme (suite)



Deux intervalles $[a,b]$ et $[c,d]$ ne se superposent pas si ...

Evaluation

Note : moyenne des notes sur l'ensemble des instances

Note pour une instance :

- Aucune solution aussi bonne que ALGO1 n'a été transmise : 10
- Une solution de score supérieur à ALGO1 a été transmise :

Note proportionnelle à la qualité de la solution, sur l'intervalle [12,20]
(note 12 pour un score ALGO1, note 20 pour la meilleure solution reçue)

Organisation

1. A effectuer en binôme. Choisir son nom d'équipe.
2. Se rendre sur campus et récupérer les fichiers : le sujet (ces transparents), les 45 fichiers d'instance, le checker
3. Choisir son langage (et son environnement)
4. Mettre à jour les fichiers solutions obtenus sur campus jusqu'à **18h00**
5. 18h00 : remise des prix
6. Avant 20h : déposer votre code sur campus

Les meilleurs résultats sont projetés tout au long de la journée

Présence obligatoire jusqu'à la remise des prix

Attention : Nom de l'équipe au choix mais **sans caractères spéciaux ni accents**

Organisation

Fonctionnement du checker

`check.exe res_1.txt res_2.txt ...`

ou

`check.exe`, puis indiquer le nom du fichier résultat à vérifier

Attention : les fichiers décrivant l'instance doivent se trouver dans le même répertoire

Disponible sous Mac, sous Linux et sous Windows

Pour une exécution sous Linux, bien vérifier les droits
(commande `chmod u+x nom`)

Quelques conseils

Le choix du langage est libre (Python...) mais c'est l'occasion de progresser en C !

Le challenge est noté mais les enseignants sont à votre disposition pour vous aider tout au long de la journée et répondre à vos questions.

Coder l'algorithme 1 n'est pas obligatoire, c'est pour vous guider.

Toute l'équipe enseignante sera ravie de vous aider à imaginer des algorithmes plus efficaces.

les 4.001 FAÇONS *(dingo)* de CHARGER *une* VOITURE

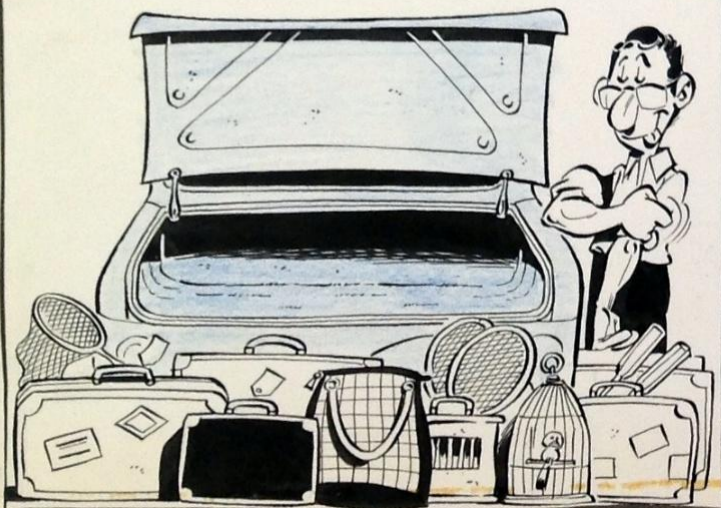
Avant d'affronter en voiture, la route des vacances, il y a lieu de résoudre un tas de petits problèmes ayant trait au véhicule ...



...et le chargement de celui-ci n'est pas le moindre de ces problèmes.



Le chargement du coffre d'une voiture demande un tour de main pratiquement inné. Prenez le même coffre, le même nombre de valises ...



...que précédemment. Vous trouverez des êtres d'élite, ayant reçu de la providence l'instinct quasi-scientifique...



...d'utiliser rationnellement l'espace dont ils disposent. (Ces types-là existent et ils sont très énervants.)

