

Remplissage d'un camion (container packing).

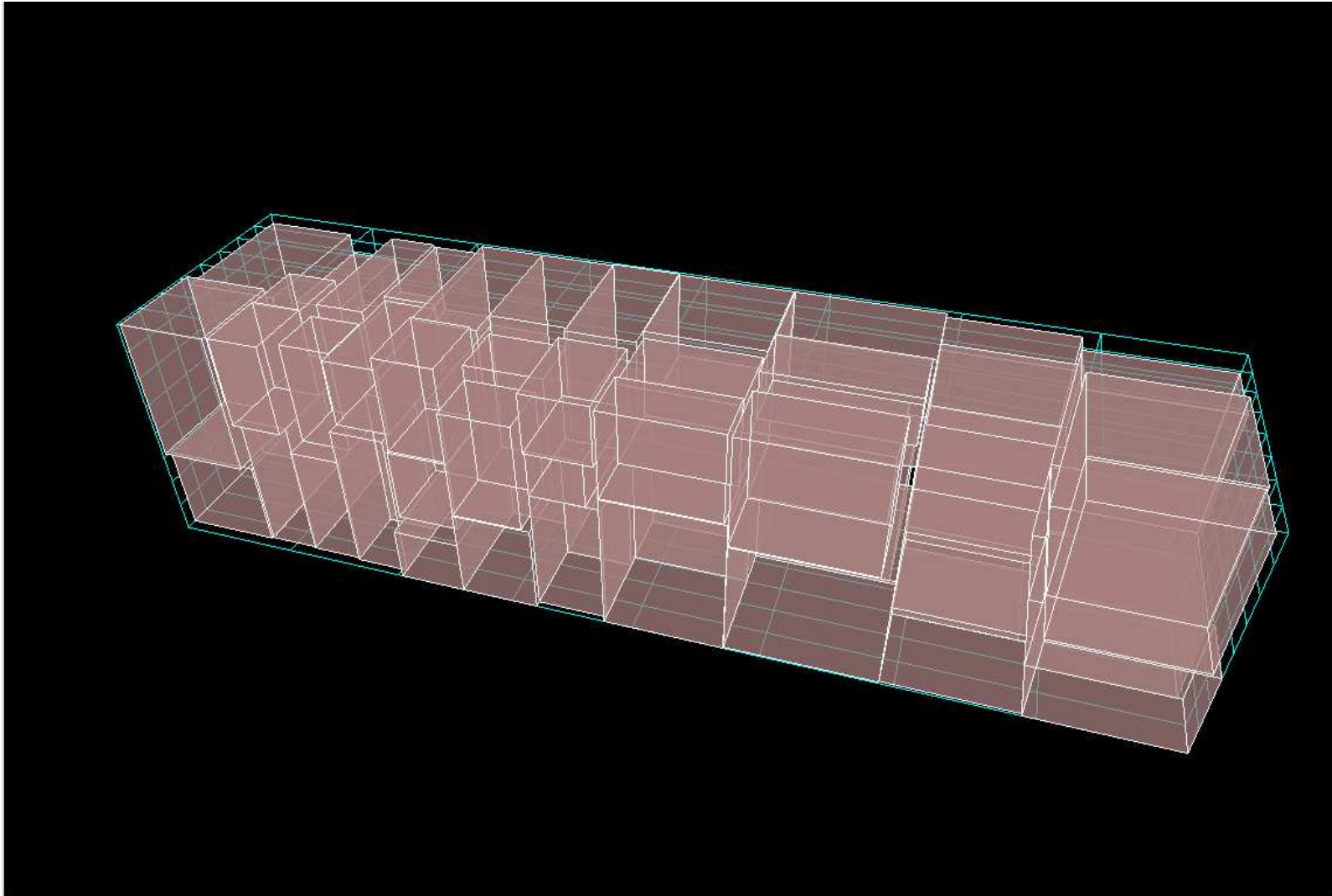
Données.

- Un camion C de dimensions $\langle l, p, h \rangle$,
- des colis C_1, \dots, C_n de dimensions $\langle l, p, h \rangle_i$ sans orientation définie.

Problème. Remplir au mieux le camion avec les colis.



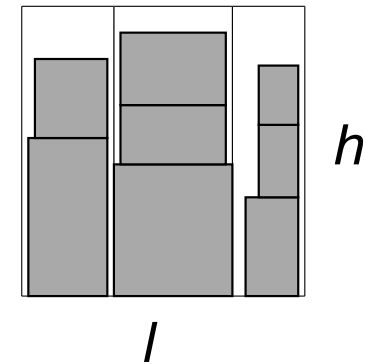
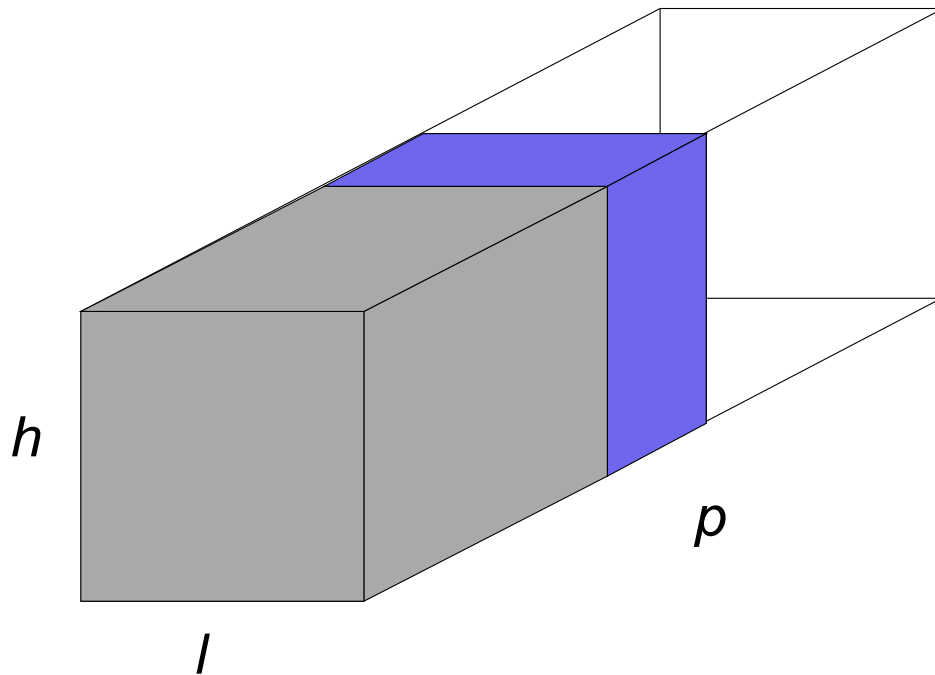
Remplissage d'un camion (container packing).



Remplissage d'un camion.

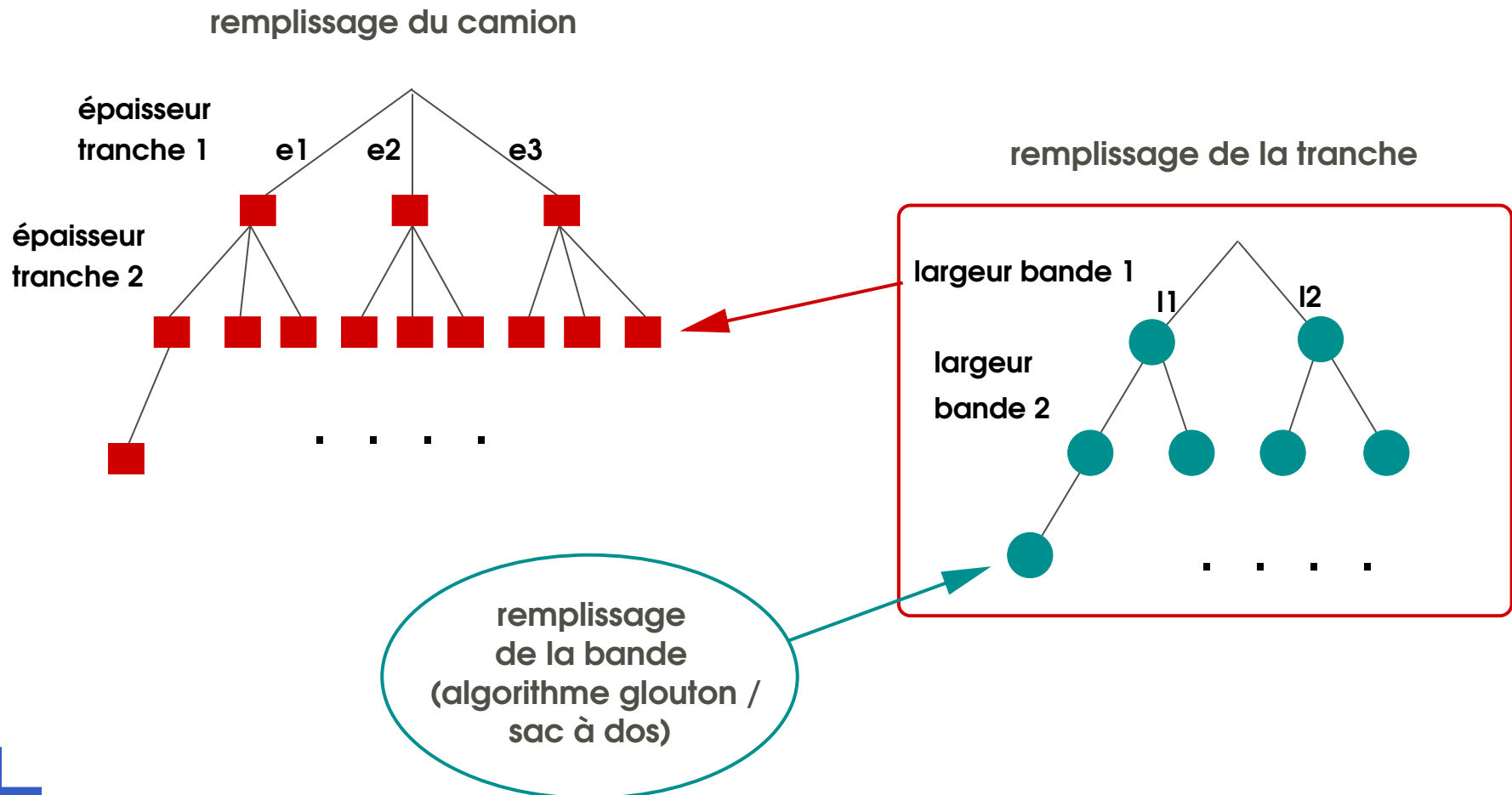
Méthode.

- Découper le volume du camion en *tranches*,
- découper chaque tranche en *bandes*.



Remplissage d'un camion.

Méthode. Pour les tranches (resp. les bandes) on énumère plusieurs épaisseurs (resp. largeurs) possibles



Remplissage d'une bande.

- calculer l'ensemble des colis *libres* et *éligibles* (réorientation),
 - **Colis éligibles** : ceux qui entrent dans la bande quite à les réorienter.
- sélectionner des colis éligibles pour mettre dans la bande.
 - **Sélection** : à partir des hauteurs des colis (relativement à leurs orientations) et de la hauteur de la bande (hauteur du camion), constituer une sélection de colis avec une approche glouton dans un premier temps, dans un deuxième temps calculer la meilleure sélection à l'aide d'une méthode de type sac à dos (approche programmation dynamique)..

Remplissage d'un camion.

- **données** : la taille du camion, les tailles des colis,
- **paramètres** : les largeurs d'énumération sur les épaisseurs des tranches et les largeurs des bandes.
- **résultat** : une sélection de colis avec leurs positions et leurs orientations, qui remplit plutôt bien le camion.

Projet : comment s'y prendre.

- Structures de données.
- Développement des méthodes de calcul.
- Validation.

Structures des données.

- représentation des données du problème,
- représentation des solutions du problème,
- éventuellement représentation de divers objets utilisés dans le calcul.

Développement.



- méthodes propres aux structures de données (construction, affichage, recherche, . . .),
- méthodes de calcul (basées sur des algorithmes) pour la résolution du problème.

Conseil 1. Vérifier chaque fois que la méthode est juste (en faisant afficher, en testant sur des exemples).

Conseil 2. Passer du temps pour écrire des méthodes simples, courtes et lisibles.



Validation.



- validation au fur et à mesure du développement.
- validation du projet dans son intégralité, par exemple en le lançant sur des gros problèmes générés aléatoirement.
- réglage, recherche des meilleurs paramètres, tests expérimentaux pour évaluer l'efficacité de différentes options.



Projet : notation.



Phase 1 : structures des données

- **structures des données (8 points)** à rendre avant les vacances de février. Avec une présentation écrite des structures.

Phase 2 : développement et validation

- **développement (4 points)** Lisibilité, clarté, commentaires, propreté du code.
- **analyse (4 points)** Compréhension des algorithmes de calcul, utilisation de méthodes appropriées.
- **validation (4 points)** Validation du programme sur des jeux de tests pour être sûr de sa justesse.



Projet : dates importantes.



- **structures : au plus tard le 14 février par mail**
- **projet : lundi 10 avril de 14h00 à 16h00 et mardi 11 avril de 9h00 à 12h00**

