

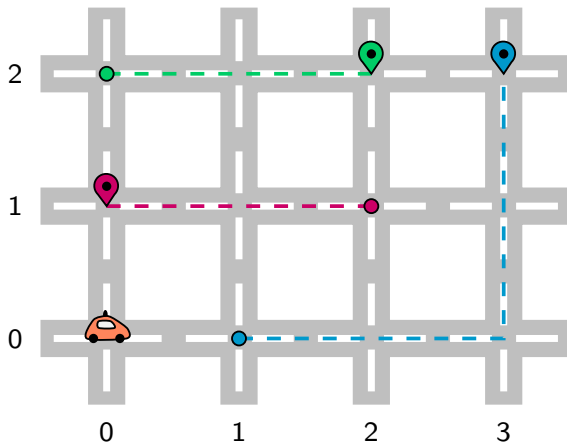
Google Hash Code

Self-driving rides

Hash Code 2018, Online Qualification Round

Énoncé

Représentation du problème



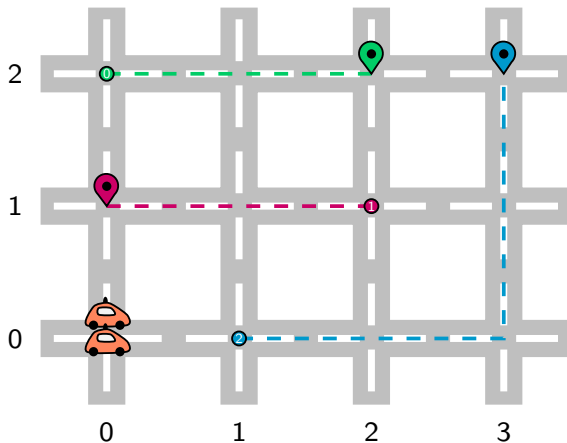
Représentation du problème

- R, C nombre de lignes et de colonnes de la grille
- F véhicules
- N courses
 - $\forall r \in [1, N], s_r, f_r$: le point de début et le point d'arrivée de la course
 - $\forall r \in [1, N], e_r, l_r$: le temps au plus tôt de début et le temps au plus tard de fin de la course
- B bonus par course commençant à l'heure
- T horizon de temps
- Score d'une course : distance de la course plus un éventuel bonus si elle est commencée à l'heure au plus tôt

Objectif : Maximiser le score de toutes les courses effectuées

Exemple

Example



$$T = 15$$

$$B = 2$$

$$e_0 = 2$$

$$l_0 = 14$$

$$e_1 = 4$$

$$l_1 = 14$$

$$e_2 = 0$$

$$l_2 = 14$$

Exemple

Example

- Grille de 3 lignes et 4 colonnes
- 2 véhicules
- 3 courses
 - $s_0 = (0, 2), f_0 = (2, 2), e_0 = 2, l_0 = 14$
 - $s_1 = (2, 1), f_1 = (0, 1), e_1 = 4, l_1 = 14$
 - $s_2 = (1, 0), f_2 = (3, 2), e_2 = 0, l_2 = 14$
- Bonus de 2
- Horizon de 15 pas de temps

Les variables ?

- Les courses affectées aux véhicules
 - $\forall v \in [0, F - 1], L_v$: la liste des courses affectées au véhicule v

Algorithme glouton

Principe

- À chaque étape, on fait un choix, celui qui semble le meilleur à cet instant
- Construit une solution pas à pas
 - sans revenir sur ses décisions
 - en effectuant à chaque étape le choix qui semble le meilleur
 - en espérant obtenir un résultat optimum global
- Approche glouton
 - suivant les problèmes pas de garantie d'optimalité (heuristique gloutonne)
 - peu coûteuse (comparée à une énumération exhaustive)
 - choix intuitif

Algorithme glouton

Example

- 2 véhicules, 3 courses
 - $s_2 = (1, 0)$, $f_2 = (3, 2)$, $e_2 = 0$, $l_2 = 14$, $d_2 = 4$
 - $s_0 = (0, 2)$, $f_0 = (2, 2)$, $e_0 = 2$, $l_0 = 14$, $d_0 = 2$
 - $s_1 = (2, 1)$, $f_1 = (0, 1)$, $e_1 = 4$, $l_1 = 14$, $d_1 = 2$

Objectif : Maximiser le score de toutes les courses effectuées

- On trie les courses par ordre décroissant de distance
- On parcourt les courses et on essaye de l'affecter à un véhicule de façon à maximiser le score (distance + bonus)

Algorithme glouton

Example

- 2 véhicules, 3 courses

- $s_2 = (1, 0), f_2 = (3, 2), e_2 = 0, l_2 = 14, d_2 = 4$
- $s_0 = (0, 2), f_0 = (2, 2), e_0 = 2, l_0 = 14, d_0 = 2$
- $s_1 = (2, 1), f_1 = (0, 1), e_1 = 4, l_1 = 14, d_1 = 2$

- $L_0 = [2, 1]$
- $L_1 = [0]$
- score = 10

$$t_0 = 9, p_0 = (0, 1)$$
$$t_1 =, p_1 = ()$$

Algorithme glouton

Example

- 2 véhicules, 3 courses
 - $s_2 = (1, 0)$, $f_2 = (3, 2)$, $e_2 = 0$, $l_2 = 14$, $d_2 = 4$
 - $s_0 = (0, 2)$, $f_0 = (2, 2)$, $e_0 = 2$, $l_0 = 14$, $d_0 = 2$
 - $s_1 = (2, 1)$, $f_1 = (0, 1)$, $e_1 = 4$, $l_1 = 14$, $d_1 = 2$

Amélioration

On peut changer de stratégie

- 1 On trie les courses par ordre décroissant de distance
- 2 On trie les courses par possibilité de bonus
- 3 Une combinaison des 2 précédentes