MINES
Saint-Étienne

1816
2016

Challenge Optim 2022

14/06/2022

INSPIRING
INNOVATION
SINCE 1816

Le contexte

Optimisation de la production sur une chaîne de montage automobile

Inspiré du challenge ROADEF 2005 :
Ordonnancement de véhicules pour une chaîne de montage automobile
proposé par RENAULT

https://www.roadef.org/roadef-le-challenge

Le sujet

Données d'entrée :

- **V** : ensemble des véhicules à faire passer sur la ligne
- **O** : ensemble des options possibles pour les véhicules

avec leurs caractéristiques

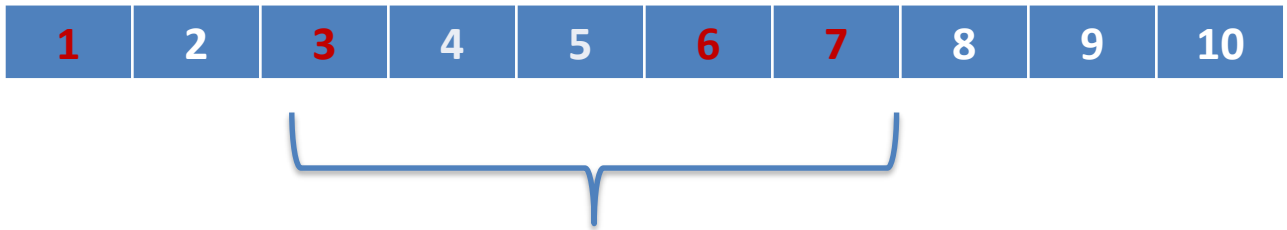
Ratio N/P : on ne souhaite pas programmer plus de N véhicules ayant cette option parmi P véhicules consécutifs

Planning : Décider de l'ordre de passage des |V| véhicules sur la chaîne

La seule contrainte est de faire passer tous les véhicules. Le respect des ratio permet de discriminer les différents plannings.

Le sujet

Illustration pour une option de ratio 2 sur 5 :
Deux plannings avec 10 véhicules. En rouge les véhicules ayant l'option.



L'option est présente sur 3 véhicules dans cette séquence de taille 5.
Le ratio n'est pas respecté

Le ratio est respecté sur toutes les séquences

Données

Données sur les options :

- **Ratio N/P**
- **Poids** : pénalité si le ratio n'est pas respecté
- **Identifiant**

Données sur les véhicules :

- **Vecteur caractéristique δ** des options présentes sur le véhicule

Calcul du coût

somme sur toutes les options

$$\sum_{o \in O} \sum_{v=1}^{|V|} p_o \times \max(0, \sum_{k=v}^{v+P-1} \delta^*_{ko} - N)$$

nombres de véhicules en excès pour l'option o dans cette fenêtre

somme sur toutes les fenêtres glissantes

poids

avec $\delta^*_{kO} = \delta_{k,o}$ pour k ≤ |V], 0 sinon

Le sujet

Illustration pour une option de ratio 2 sur 5 et de poids 10 :
En rouge les véhicules ayant l'option.

1	2	3	4	5	6	7	8	9	10	Pénalité
0	0	0	0	0						
	0	0	0	0	0					
		0	0	0	0	1				
			0	0	0	1	1			
				0	0	1	1	1		+10
					0	1	1	1	1	+20
						1	1	1	1	+20
							1	1	1	+10
								1	1	
									1	

Les entrées / sorties

15 instances sont proposées

1 fichiers cs**n**.txt par instance, où **n** est le numéro de l'instance

OPTIONS	12											
VEHICULES	704											
RATIO		POIDS	NOM									
1	3	10	HPRC1									
5	6	10	HPRC2									
1	3	10	HPRC3									
1	6	1	LPRC1									
1	10	1	LPRC2									
1	3	1	LPRC3									
2	3	1	LPRC4									
1	3	1	LPRC5									
1	4	1	LPRC6									
10	15	1	LPRC7									
1	4	1	LPRC8									
1	5	1	LPRC9									
VEHICULE	HPRC1	HPRC2	HPRC3	LPRC1	LPRC2	LPRC3	LPRC4	LPRC5	LPRC6	LPRC7	LPRC8	LPRC9
1	0	0	0	0	0	0	1	0	0	1	0	0
2	0	0	0	0	0	0	1	0	0	1	0	0
3	0	0	0	0	0	0	0	0	0	1	0	0
4	0	0	0	0	0	0	1	0	0	1	0	0
5	0	0	0	0	0	0	1	0	0	1	0	0
6	0	0	0	0	0	1	0	0	0	1	0	0
7	0	1	1	0	0	0	1	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	1	0	0

Fichier de données (extrait)

***Remarque** : le matrice δ respecte toujours l'ordre des véhicules et des options*

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 (par contre le nom est au choix)

Format du fichier :

Nom de l'équipe

EQUIPE nom de l'équipe
INSTANCE n
v1 v2 v3 v4 …

Numéro de l'instance

Séquence proposée pour les véhicules

Heuristique proposée

ALGO1 : meilleure insertion (référence basse)

```
pour i = 0 à |V|-1
   bestVeh ← -1, bestVal ← infini
    pour k = 0 à |V|-1
      si le véhicule k n'est pas déjà ajouté à la séquence
         val = evaluerAjout(k)
         si val < bestVal
            bestVeh=k
            bestVal=val
         fin si
      fin si
   fin pour
   ajouter véhicule k en position i
fin pour
```

Remarque *: dans les fichiers les véhicules sont numérotés de 1 à |V|. Dans cet algorithme ils ont été renumérotés de 0 à [V|-1*

Evaluation

Note : moyenne des notes sur l'ensemble des instances

Note pour une instance :

- Aucune solution aussi bonne que ALGO1 n'a été transmise : 8
- Une solution de score supérieur à ALGO1 a été transmise

  Note proportionnelle à la qualité de la solution, sur l'intervalle [10,20]
  (note 10 pour un score ALGO1, note 20 pour la meilleure solution reçue)

Organisation

1. A effectuer en binôme

2. Se rendre sur campus et récupérer les fichiers : le sujet (ces transparents, les 15 fichiers d'instance, le checker)

3. Choisir son langage (et son environnement)

4. Déposer les fichiers solutions obtenus au fur et à mesure sur campus jusqu'à **19h00**

5. A 19h00, déposer votre code

Les meilleurs résultats sont mis à jour tout au long de la journée

Fonctionnement du checker :

check.exe nom1.txt nom2.txt …

ou

check.exe et indiquer le fichier résultat à vérifier

Quelques conseils

N’hésitez pas à solliciter de l’aide

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, comme pour un TP

Quelques conseils

Ne mettez pas votre GP en péril

Si une note de 8 suffit pour votre GP, faites-vous plaisir.

Sinon, si vous êtes peu à l'aise en programmation, nous vous conseillons fortement de programmer en C et de suivre la méthodologie suivante :

1. Commencer par définir sur papier les structures de données qui seront utilisées
2. Faire un planning (horaire) de développement, étape par étape
3. Ne pas hésiter à faire appel aux enseignants pour valider ces structures et ce planning
4. Nous consulter si vous constatez un retard inquiétant par rapport à ce planning

Je vous souhaite un EXCELLENT CHALLENGE