

**Etude de cas : Résolution du problème
d'ordonnancement de véhicules en
Choco**

Sommaire

1- Ordonnancement de véhicules.....	3
Description d'instance théorique de problème de "car-sequencing"	4
Exemple de solution de l'instance tab 3.1	5
2- Formalisation.....	5
Définition.....	5
Exemple CSPLib	6
Solution CSPLib	6

1- Ordonnancement de véhicules



Le Problème d'Ordonnancement de Voitures (POV), mieux connu sous le nom de car-sequencing problem, est apparu dans le secteur de la production automobile lorsque les fabricants ont abandonné la production de masse pour adopter la personnalisation des véhicules. Le POV consiste à donner l'ordre dans lequel des automobiles doivent être produites, en considérant les options de chacune et les contraintes de capacité de la chaîne de montage.

Le problème d'ordonnancement de voitures consiste à séquencer des voitures sur une chaîne de montage pour leur installer des options (e.g., toit ouvrant ou climatisation). Chaque option est montée par une station différente conçue pour traiter au plus un certain pourcentage de voitures passant le long de la chaîne. Par conséquent, les voitures demandant une même option doivent être espacées de telle sorte que la capacité des stations ne soit jamais dépassée. On peut facilement le formuler en un problème de satisfaction de contraintes (CSP) et il s'agit d'un benchmark classique pour les solveurs de contraintes. La plupart de ces solveurs sont basés sur une approche complète qui explore l'espace de recherche de façon systématique jusqu'à ce qu'une solution soit trouvée, ou le problème soit montré inconsistant.

Problème : NP-complet, problème N.1 classique de CSPLib.

Variables booléennes : utilisation des variables soit vraies soit fausses Pour produire une permutation de V respectant les contraintes de fréquences.

Description d'instance théorique de problème de "car-sequencing"

Une instance de problème de "car-sequencing" est décrite par un ensemble de voitures à produire qui peuvent être, pour faciliter le traitement, groupées en catégories de voitures semblables. À cet ensemble, s'ajoute les contraintes de capacité du problème ainsi que la demande pour chacune des catégories de véhicule. Un exemple d'instance de problème tiré de Smith [1996] est présenté au **Tableau 3.1**. Une matrice d'éléments binaires de la taille du nombre de catégories multiplié par le nombre d'options précise quelles sont les options présentes dans chacune des catégories de véhicules. Par exemple les voitures de la catégorie 1 possèdent l'option 2 tandis que celles de la catégorie 2 possèdent les options 1, 3 et 5.

		Catégories											
Option	contrainte	1	2	3	4	5	6	7	8	9	10	11	12
1	1/2		•	•				•	•	•			•
2	2/3	•		•	•	•		•				•	•
3	1/3		•					•		•	•	•	
4	2/5				•		•		•				•
5	1/5		•			•							
	nb de voitures	3	1	2	4	3	3	2	1	1	2	2	1

Tableau 3.1 - Instance théorique de problème de "car-sequencing"

Par exemple, certaines voitures nécessitent l'installation de toit ouvrant, d'air climatisé ou de lecteur de disques compacts ou autres. Ces véhicules doivent par conséquent être dispersés dans la séquence de production de façon à lisser la charge de travail. Des contraintes de capacité précisent alors le nombre maximum de voitures q possédant l'option i qui peuvent être produites sur une sous-séquence de taille p . Ces contraintes de capacité sont généralement exprimées par le ratio $C_i = q_i / p_i$ pour chacune des options i du problème. Notons également que la dernière ligne de ce tableau indique la demande pour chacune des catégories de véhicules. Par exemple, la catégorie 1 a une demande de 3 véhicules et la catégorie 2 une demande de 1 véhicule.

Exemple de solution de l'instance tab 3.1

La séquence de production doit respecter toutes les contraintes de capacité du problème et remplir la demande de chacune des catégories de véhicule. Par exemple, pour l'instance de problème présentée au Tableau 3.1, une solution valide est décrite au Tableau 3.4.

option \ Catégories	7	4	8	5	3	10	12	4	2	1	3	10	4	4	9	5	1	6	7	1	6	11	5	6	11
1, 1/2	•		•		•		•		•		•				•			•							
2, 2/3	•	•		•	•		•	•		•	•		•	•		•	•		•	•		•	•		•
3, 1/3	•					•			•			•			•			•			•				•
4, 2/5		•	•				•	•					•	•				•			•			•	
5, 1/5				•					•							•							•		

Tableau 3.4 - Exemple de solution

2- Formalisation

Définition

Soit le quintuplet (V, O, p, q, r) avec :

V : Ensemble des voitures à produire.

O : Ensemble des options disponibles.

p, q : Listes d'entier, pour chaque option i , p^i/q^i représente la fréquence d'apparition.

r : Matrice de booléen $r_{ij} = 1$, représente le fait que l'option O^j est présente sur le véhicule V_i , 0 sinon.

Exemple CSPLib

v		o				
10	5	6				
1	2	1	2	1	p	
2	3	3	5	5	q	
0	1	1	0	1	1	0
1	1	0	0	0	1	0
2	2	0	1	0	0	1
3	2	0	1	0	1	0
4	2	1	0	1	0	0
5	2	1	1	0	0	0

Solution CSPLib

Classe	Options requises					
0	1	0	1	1	0	
1	0	0	0	1	0	
5	1	1	0	0	0	
2	0	1	0	0	1	
4	1	0	1	0	0	
3	0	1	0	1	0	
3	0	1	0	1	0	
4	1	0	1	0	0	
2	0	1	0	0	1	
5	1	1	0	0	0	

Travail demandé

Modéliser et résoudre ce problème par la plateforme Choco.