Outils de Recherche Opérationnelle en Génie - MTH 8414

Astuce de Modélisation

- Problème des N-Reines
- Problème d'affectation
- Problème du voyageur de commerce

Louis-Martin Rousseau

Office: A520.21 Tel.: #4569
Louis-Martin.Rousseau@polymtl.ca

SEND + MORE = MONEY

- Trouver la valeur de chaque chiffre
- Toutes les lettres doivent avoir des valeurs différentes

$$\begin{array}{cccccc}
 & & S & E & N & D \\
+ & & M & O & R & E \\
\hline
= & M & O & N & E & Y
\end{array}$$

SEND + MORE = MONEY

- Trouver la valeur de chaque chiffre
- Toutes les lettres doivent avoir des valeurs différentes

$$
\begin{array}{cccccc}
 & & S & E & N & D \\
+ & & M & O & R & E \\
\hline
= & M & O & N & E & Y
\end{array}
$$

$$
\begin{aligned}
& 1000 * S + 100 * E + 10 * N + D \\
+ \quad & 1000 * M + 100 * O + 10 * R + E \\
= \; & 10000 * M + 1000 * M + 100 * N + 10 * E + Y
\end{aligned}
$$

$$AllDifferent(S, E, N, D, M, O, R, Y)$$

$$S, E, N, D, M, O, R, Y \in \{0,1,2,3,4,5,6,7,8,9\}$$

Problème d'affectation

Modéliser le problème d'affectation suivant en PC

- Étant donné 5 tâches (t_1 to t_5) et 5 employés (e_1 to e_5)
- Affecter une et seulement une tâche à chaque employé tel que l'affectation minimise les coûts suivants

T\E	1	2	3	4	6
1	2	3	5	1	8
2	3	4	3	4	5
3	1	3	4	7	9
4	3	3	2	6	4
5	5	7	2	8	5

- Comparer avec la version PLNE du problème

Problème d'affectation

MIP

Min $\sum_{et} c_{et} x_{et}$

st:

$$\sum_{t} x_{et} = 1 \quad \forall\, e$$

$$\sum_{e} x_{et} = 1 \quad \forall\, t$$

$$x_{et} \in \{0,1\}$$

x_{et} =1 si l'employé e fait la tâche t.

CP option 1

Min $\sum_{e} c_{e x_e}$

st:

$$\sum_{e} (x_e = t) = 1 \quad \forall\, t$$

Or

$$\text{Alldifferent}(x_e)$$

$$x_e \in \{t\}$$

x_e détermine quelle tache doit faire l'employé e

CP option 2

Min $\sum_{tt} c_{x_t t}$

st:

$$\sum_{t} (x_t = e) = 1 \quad \forall\, e$$

Or

$$\text{Alldifferent}(x_t)$$

$$x_t \in \{e\}$$

x_t détermine quel employé doit faire la tâche t

Contraintes redondantes

Série magique

- Une série $S = (s_0, \dots, s_n)$ est magique si s_i représente le nombre d'occurrence de i dans S

0	1	2	3	4
?	?	?	?	?

$n \; = \; 5$
$D \; = \; \{0, \dots, n-1\}$
var $s[D]$ in D
forall $(k$ in $D)$ $s[k] \; == \; sum(i \; in \; D) \; (s[i] == k))$

Pouvez-vous trouver des contraintes redondantes qui améliore la résolution ?

Contraintes redondantes

Série magique

$n = 5$
$D = \{0, \dots, n-1\}$
$\text{var } s[D] \text{ in } D$
$\text{forall } (k \text{ in } D)\ s[k] == sum(i\ in\ D)\ (s[i] == k))$

Redundant #1
$\text{sum } (k \text{ in } D) s[k] = n$

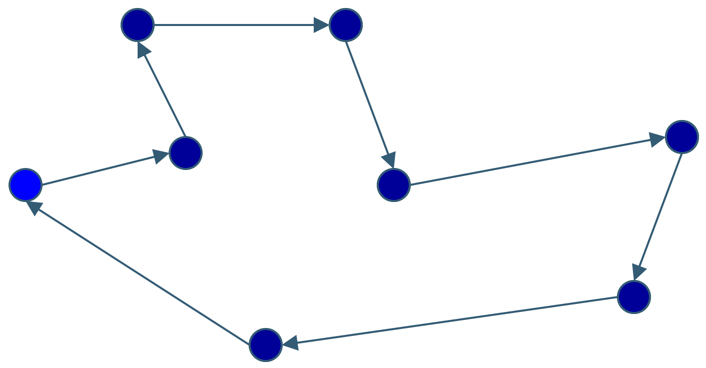
Redundant #2
$\text{sum } (k \text{ in } D)\ k * s[k] = n$

0	1	2	3	4
2	1	2	0	0

Le voyageur de commerce

Le problème du voyageur de commerce (VdC) cherche à trouver un circuit fermé passant par un ensemble donné de n villes, avec une longueur totale minimale.

Données : ensemble de villes et distances d_{ij} entre deux villes i et j



TSP: MIP model

Classical model based on 'assignment problem'

Binary variable x_{ij} represents whether the tour goes from i to j

Objective

$$\min \ \sum_{ij} d_{ij} x_{ij}$$

Need to make sure that we leave and enter each location exactly once

$$\sum_{j} x_{ij} = 1 \text{ for all } i$$

$$\sum_{i} x_{ij} = 1 \text{ for all } j$$

Remove all possible subtours: there are exponentially many; impossible to model concisely in MIP

MIP Solvers therefore resort to specialized solving methods for the TSP

Voyageur de commerce : modèle PC

La variable x_i représente la i-ème ville que le circuit visite (le domaine est {1,2,…,n})

Fonction objectif

$$\min \quad d_{x_n,x_1} + \sum_{i=1}^{n-1} d_{x_i,x_{i+1}}$$

Element constraints (variables en indice)

Constraint

alldifferent(x_1, x_2, …, x_n)

Contrainte "globale"

Comparaison entre PLNE et PC pour le VdC

- En PC seules *n* variables sont nécessaires, alors qu'il en faut n^2 en PLNE (*n* nombre de villes)
- Le modèle PLNE est de taille exponentielle, le modèle PC ne nécessite qu'une contrainte
- Le modèle PC est intuitif car il s'intéresse directement à la structure du problème : l'ordonnancement des villes dans le circuit

Remarque : les solveurs PLNE spécifiques au VdC sont plus performants que PC pour le VdC classique. En présence de contraintes additionnelles (fenêtres de temps,…), la programmation par contraintes devient compétitive.