Résumé du cours d'Algorithmique et Recherche Opérationelle
INFO-F310

Théo Verhelst

10 février 2017

Table des matières

1 Introduction

Parmi les nombreux domaines compris dans l'algorithmique et la recherche opérationelle, dans ce cours seront abordés la *programmation mathématique* et les *méthodes combinatoires dans les graphes.*

2 Programmation mathématique

2.1 Définitions

La programmation mathématique est une modélisation de problèmes (qui peuvent provenir d'une large gamme de domaines) ainsi que leur résolution. On représente un problème comme suit :

Définition : Résoudre un problème de programmation mathématique consiste à trouver les valeurs $x_i \ \forall i \in [n]$ qui maximisent une fonction

$$f(x_1, \dots, x_n)$$

tout en vérifiant

$$g_j(x_1, \dots, x_n) \begin{Bmatrix} \leq \\ \geq \\ = \end{Bmatrix} b_j \quad \forall j \in [m]$$

La fonction f (on notera parfois $f(x_1, \dots, x_n) = z$) est appellée fonction de coût, et les fonctions g_j sont les contraintes.
Note : Le problème peut également consister en la minimisation d'une fonction, mais il suffit alors de poser f comme étant l'opposé de cette fonction. Cet énoncé ne perds donc pas en généralité.

Définition : Quand les fonctions f et g_j sont linéaires en x_i, alors le problème est appelé problème de programmation linéaire.

2.2 Exemples

Le problème du sac à dos Supposons que, dans l'optique d'une randonnée en montagne, l'on cherche à remplir au mieux un sac à dos avec des aliments ayant chacun un poids et une valeur énergétique, en sachant que l'on ne peut pas porter plus de 16 unités de poids :

	A	B	C
Énergie	5	4	1
Poids	4	2	1

On va représenter le problème comme suit :

$$f(x_A, x_B, x_C) = 5x_A + 4x_B + x_C$$

$$4x_A + 2x_B + x_C \leq 16$$

où x_A, x_B, x_C sont le nombre de fois que l'on mettra un objet A, B ou C respectivement dans le sac.
On cherche donc à maximiser f, car il faut maximiser la valeur nutritionelle totale des aliments dans le sac.
C'est un problème de programmation linéaire.

Le problème du voyageur de commerce Problème bien connu, passons directement à la modélisation en programmation mathématique :
Soit n villes, posons $x_{ij} = 1$ si le voyageur va de la ville i à la ville j, $x_{ij} = 0$ sinon.
Le cout du trajet entre chaque ville est décrit par la matrice $[c_{ij}]_{i,j \in [n]^2}$ Nous cherchons à minimiser la fonction de coût

$$z = \sum_i \sum_j x_{ij} c_{ij}$$

Les contraintes sont les suivantes :

— Il faut que chaque ville soit visitée une seule fois :

$$\sum_i x_{ij} = 1$$

— Il faut aussi que le voyageur parte aussi de chaque ville :

$$\sum_j x_{ij} = 1$$

— Mais il faut également que le graphe ainsi formé soit connexe ! Pour toute partition de l'ensemble V des villes en deux sous-ensembles S et $\bar{S}$ tels que $S \cup \bar{S} = V$ et $S \cap \bar{S} = \emptyset$, il faut qu'il existe un chemin reliant S et $\bar{S}$:

$$\sum_{s \in S} \sum_{\bar{s} \in \bar{S}} x_{s\bar{s}} \geq 1$$

pour $x_{s\bar{s}} = x_{ij}$ si s et $\bar{s}$ correspondent aux villes i et j respectivement.

Difficulté : L'énumération de toutes les possibilités de cette dernière contrainte prends beaucoup, beaucoup de temps. C'est ce qui explique la nature difficile de ce problème, et qui le rends impossible à résoudre en temps polynomial.

On pourrait être tenté de résoudre le problème en énumérant toutes les solutions possibles, et en prenant la meilleure. On peut trouver facilement que le nombre de solutions possibles est $n!$. Pour $n = 52$ (le problème tel que posé pour visiter tous les états des États-Unis), on a à une vache près 10^{69} solutions. Leur énumération est simplement impossible, même avec le plus puissant des ordinateur connus.