

Introduction

MTH8415

S. Le Digabel, Polytechnique Montréal

H2020

(v2)

Plan

1. Introduction
2. Exemples de problèmes
3. Algorithmes
4. Documentation

1. Introduction

2. Exemples de problèmes

3. Algorithmes

4. Documentation

Termes importants du cours

- ▶ **Recherche opérationnelle (RO)** : *Ensemble de techniques mathématiques appliquées à la modélisation, l'optimisation et l'analyse d'un processus*
- ▶ Modélisation
- ▶ **Optimisation** :
 - ▶ Continue
 - ▶ **Linéaire** (OL)
 - ▶ **Non linéaire** (ONL)
 - ▶ Combinatoire (OC)
 - ▶ En **nombre entiers** (ONE)
- ▶ **Théorie des Graphes** :
 - ▶ Cheminements optimaux
 - ▶ Flots.
 - ▶ Problèmes de transport

Problème d'optimisation

L'**optimisation** est un domaine qui étudie les problèmes de la forme

$$\min_{x \in \mathcal{X}} \{f(x) : x \in \Omega\}$$

où

- ▶ \mathcal{X} est un ensemble de dimension n :
Les **variables d'optimisation**
- ▶ $\Omega \subseteq \mathcal{X}$ est l'ensemble des **solutions réalisables** :
Les **contraintes**
- ▶ La **fonction objectif** f prend ses valeurs sur \mathcal{X}

Modèle d'optimisation

- ▶ Pour un problème donné, l'expression de f , \mathcal{X} et Ω permet d'obtenir un **modèle d'optimisation**
- ▶ Optimisation continue (OL et ONL) : $\mathcal{X} = \mathbb{R}^n$
- ▶ OC : \mathcal{X} est un ensemble **discret**
- ▶ ONE : $\mathcal{X} = \mathbb{Z}^n$ ou \mathbb{N}^n ou $\{0, 1\}^n$
- ▶ En théorie des graphes, il n'est pas forcément nécessaire d'exprimer un modèle d'optimisation. On se sert directement d'un **graphe** pour représenter le problème

Modèle d'optimisation (continue) non linéaire

$$\min_{x \in \mathcal{X} = \mathbb{R}^n} f(x)$$
$$\text{s.c.} \quad \begin{cases} g_1(x) \leq 0 \\ g_2(x) \leq 0 \\ \dots \\ g_m(x) \leq 0 \\ \ell \leq x \leq u \end{cases}$$

- ▶ $f : \mathbb{R}^n \rightarrow \mathbb{R}$ différentiable : **Fonction objectif**
- ▶ $g_i(x) \leq 0, i \in \{1, 2, \dots, m\}$: **Contraintes**
- ▶ $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ différentiable, $i \in \{1, 2, \dots, m\}$: Membres de gauche des contraintes
- ▶ $\ell, u \in \mathbb{R}^n$: **Bornes** sur les variables x . Peuvent être $\pm\infty$
- ▶ $\Omega = \{x \in \mathbb{R}^n : g_i(x) \leq 0 \forall i \in \{1, 2, \dots, m\}, \ell \leq x \leq u\}$

Modèle d'optimisation linéaire (forme standard)

$$\begin{aligned} \max_{x \in \mathbb{R}^n} \quad & f(x) = \sum_{j=1}^n c_j x_j \\ \text{s.c.} \quad & \begin{cases} \sum_{j=1}^n a_{ij} x_j \leq b_i & i \in \{1, 2, \dots, m\} \\ x_j \geq 0 & j \in \{1, 2, \dots, n\} \end{cases} \end{aligned}$$

Peut être exprimé de façon matricielle :

$$\begin{aligned} \max_{x \in \mathbb{R}^n} \quad & f(x) = c^\top x \\ \text{s.c.} \quad & \begin{cases} Ax \leq b \\ x \geq 0 \end{cases} \end{aligned}$$

avec $c \in \mathbb{R}^n$ (**coûts**), $b \in \mathbb{R}^m$ (**membres de droite**), et $A \in \mathbb{R}^{m \times n}$

Notes

- ▶ Fonction objectif (pas objective)
- ▶ Optimisation et pas programmation
- ▶ min et max sont équivalents
- ▶ Contraintes égalité ($=$) et contraintes inégalité (\leq et \geq). On peut transformer des égalités en inégalités et vice-versa

Optimisation combinatoire (OC)

L'ONE et la théorie des graphes sont de l'OC.

- ▶ En théorie, une solution optimale peut être obtenue en énumérant toutes les solutions réalisables et en conservant la meilleure. En pratique, ce procédé est trop long.
- ▶ Pour les problèmes faciles, une résolution exacte en un temps court est envisageable.
- ▶ Un grand nombre de problèmes sont difficiles. Des solutions exactes sont envisageables, mais dans un délai acceptable, on se contentera de **solutions approchées** obtenues par des méthodes **heuristiques**.

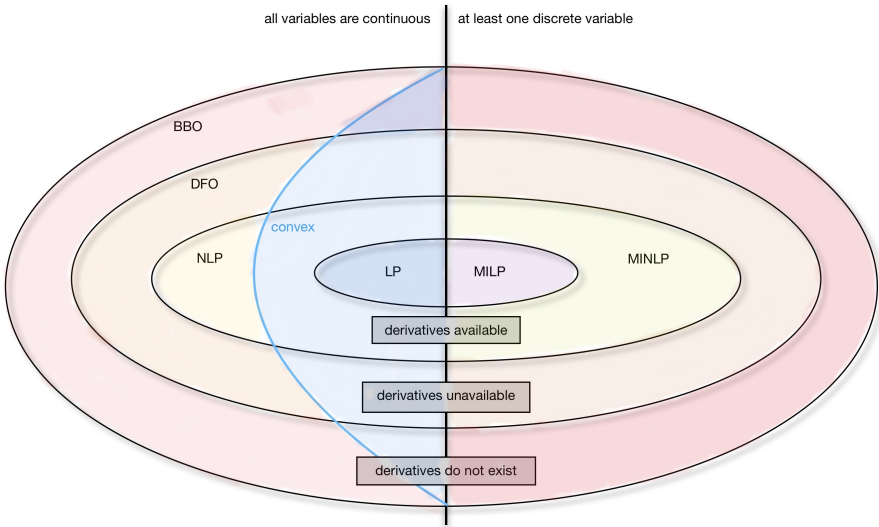
Termes importants

- ▶ Optimum **local** vs **global**.
- ▶ Algorithme **exact** vs **heuristique**.
- ▶ En OL, on aura un optimum global.
- ▶ En ONL, la plupart du temps, un optimum local, et si le problème est **convexe**, on aura un optimum global.
- ▶ En OC, on aura soit une solution exacte (=un optimum global), soit un optimum local qui dépend d'un **voisinage**, ou alors une solution heuristique.

Extensions

- ▶ Optimisation sans dérivées
- ▶ Optimisation multiobjectifs
- ▶ Optimisation multi-niveaux
- ▶ Optimisation stochastique
- ▶ Optimisation robuste
- ▶ Optimisation conique
- ...

Schéma global



1. Introduction

2. Exemples de problèmes

3. Algorithmes

4. Documentation

Problème v.s. Instance

- ▶ Un **problème** correspond au modèle

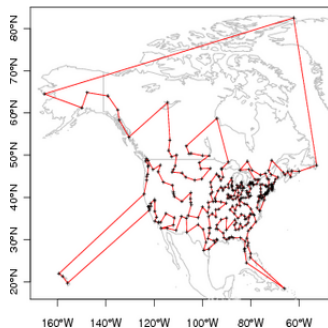
$$\min_{x \in \mathcal{X}} \{f_a(x) : x \in \Omega_a\}$$

dans lequel a est un ensemble de **paramètres** non déterminé (c'est le cas général)

- ▶ Une **instance** du problème est une formulation du modèle dans laquelle a est déterminé (c'est un cas particulier)
- ▶ En pratique, on conçoit souvent des algorithmes pour des problèmes, qu'on teste sur plusieurs instances
- ▶ On peut aussi s'intéresser à une seule instance ou à une famille d'instances particulières. Dans ce cas on concevra une méthode plus spécialisée

Voyageur de commerce (TSP)

Un voyageur de commerce doit visiter un certain nombre de villes, et chaque ville une et une seule fois. Étant donné des distances entre chaque paire de villes, il doit minimiser la distance totale parcourue



Problème d'affectation

- ▶ n tâches à affecter à n machines de telle sorte que chaque machine soit affectée à une seule tâche et chaque tâche soit affectée à une seule machine
- ▶ Le coût d'affecter la tâche j à la machine i est c_{ij}
- ▶ L'objectif est de minimiser la somme des coûts
- ▶ Ω est l'ensemble de toutes les affectations possibles des n tâches aux n machines. Il y en a $n!$
- ▶ Modèle en variables binaires :

$$\min_{x \in \mathcal{X} = \{0,1\}^{n^2}} f(x) = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$$

sujet aux $2n$ contraintes $\left\{ \begin{array}{l} \sum_{j=1}^n x_{ij} = 1, i = 1, 2, \dots, n \\ \sum_{i=1}^n x_{ij} = 1, j = 1, 2, \dots, n \end{array} \right.$
(c'est l'ensemble Ω)

Liste non exhaustive de problèmes et applications

- ▶ OC (théorie des graphes et ONE) :
 - ▶ Coloration de graphes
 - ▶ Routage
 - ▶ Satisfaisabilité
 - ▶ Classification
 - ▶ Horaires
 - ▶ etc.
- ▶ ONL : Génie chimique, génie industriel, génie mécanique, estimation de paramètres, apprentissage machine, etc.
- ▶ OL : Applications dans quasiment tous les domaines (transport, énergie, planification, production, distribution, télécommunications, finance, etc.)

1. Introduction

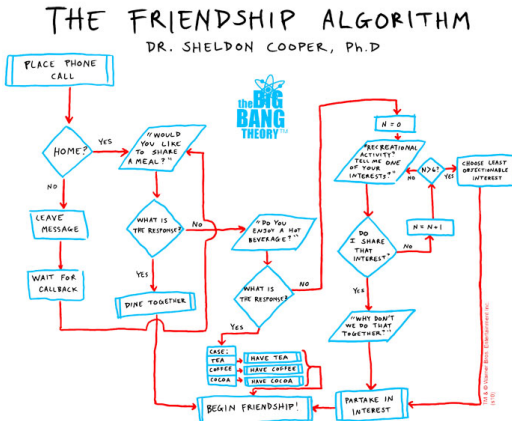
2. Exemples de problèmes

3. Algorithmes

4. Documentation

Algorithmme

Un algorithme est la description d'une séquences d'instructions à entreprendre afin de résoudre un problème. Par exemple :



Il faut choisir un algorithme selon les exigences sur la **qualité de la solution** et le **temps de calcul**. Ces objectifs sont souvent contradictoires.

- ▶ **Algorithmes exacts** : garantissent une solution optimale globale mais peuvent être très longs si le problème est difficile.
 - ▶ Séparation et évaluation (*branch and bound*) pour l'ONE.
 - ▶ Le simplexe pour l'OL.
 - ▶ Programmation dynamique.
- ▶ **Algorithmes locaux** : garantissent une solution locale.
 - ▶ Méthodes de descente (OC et ONL).
 - ▶ Gradient / Newton pour l'ONL.
- ▶ **Heuristiques et métaheuristiques** : très peu de garanties sur la qualité de la solution, mais convergent rapidement.
 - ▶ Algorithme glouton.
 - ▶ Recherche tabou, à voisinages variables (VNS).
 - ▶ Algorithmes génétiques.

La plupart des métaheuristiques sont utilisées en OC.

1. Introduction

2. Exemples de problèmes

3. Algorithmes

4. Documentation

Liste non exhaustive de références

- ▶ OL : V. Chvátal, *Linear programming*, 1983
- ▶ OL et ONL : D.G. Luenberger et Y. Ye, *Linear and Nonlinear Programming*, 2016
- ▶ Graphes et réseaux : R.K. Ahuja, T.L. Magnanti, et J.B. Orlin, *Network Flows : Theory, Algorithms, and Applications*, 1993
- ▶ Général : D. de Werra, T.-M. Liebling, et J.-F. Hêche, *Recherche opérationnelle pour ingénieurs, volume 1*, 2003