Introduction à l'optimisation

Quentin Fortier

September 15, 2021

Cours

Le cours est disponible sur Internet :
https://github.com/fortierq/oc-m1-2021

Optimisation

Soit $f : \Omega \longrightarrow \mathbb{R}$ une fonction et $\mathcal{S} \subseteq \Omega$ l'ensemble des solutions admissibles.

Objectif

Trouver $\min_{s \in \mathcal{S}} f(s)$ et s^* tel que $f(s^*) = \min_{s \in \mathcal{S}} f(s)$

Optimisation

Soit $f : \Omega \longrightarrow \mathbb{R}$ une fonction et $\mathcal{S} \subseteq \Omega$ l'ensemble des solutions admissibles.

Objectif

Trouver $\min\limits_{s \in \mathcal{S}} f(s)$ et s^* tel que $f(s^*) = \min\limits_{s \in \mathcal{S}} f(s)$

- Si Ω est fini : **optimisation combinatoire**
- Sinon : **optimisation continue/différentiable**

Optimisation

Objectif

Trouver $\min_{s \in \mathcal{S}} f(s)$ et s^* tel que $f(s^*) = \min_{s \in \mathcal{S}} f(s)$

Pour prouver qu'un algorithme renvoie une solution optimale :

1. Montrer que l'algorithme renvoie une solution s admissible ($s \in \mathcal{S}$)

Optimisation

Objectif

Trouver $\min_{s \in \mathcal{S}} f(s)$ et s^* tel que $f(s^*) = \min_{s \in \mathcal{S}} f(s)$

Pour prouver qu'un algorithme renvoie une solution optimale :

1. Montrer que l'algorithme renvoie une solution s admissible ($s \in \mathcal{S}$)
2. Montrer que s est optimale. Pour cela on raisonne souvent par l'absurde : on considère s^* une solution meilleure que s et on trouve une contradiction.

Optimisation

Objectif

Trouver $\min_{s \in \mathcal{S}} f(s)$ et s^* tel que $f(s^*) = \min_{s \in \mathcal{S}} f(s)$

Problème	Ω	f
Arbre couvrant min d'un graphe G	Ensemble des arbres couvrants de G	Poids d'un arbre
Plus court chemin	Ensemble des chemins	Poids d'un chemin
Problème du sac à dos	Ensemble des façons de remplir un sac à dos	Valeur des objets dans le sac

Complexité

Considérons un problème d'optimisation combinatoire de minimisation de f sur un ensemble Ω.
On suppose de plus que chaque élément de Ω a une certaine taille n.

Complexité

Considérons un problème d'optimisation combinatoire de minimisation de f sur un ensemble Ω.
On suppose de plus que chaque élément de Ω a une certaine taille n.

Comme Ω est fini, on pourrait énumérer toutes les possibilités et conserver le mininmum.

Complexité

Considérons un problème d'optimisation combinatoire de minimisation de f sur un ensemble Ω.
On suppose de plus que chaque élément de Ω a une certaine taille n.

Comme Ω est fini, on pourrait énumérer toutes les possibilités et conserver le mininmum. Mais la complexité serait exponentielle en n...
on cherche donc des algorithmes plus efficaces.

NP-difficile

De nombreux problèmes intéressants sont NP-difficiles et il n'existe probablement pas d'algorithme polynomial pour les résoudre... On pourra alors essayer de chercher une approximation de la solution optimale.

Plan du cours

1. **Algorithmes glouton**
   $\rightarrow$ Kruskal, Prim, Dijkstra
2. **Programmation dynamique**
   $\rightarrow$ Bellman-Ford, plus longue sous-suite croissante
3. **Problèmes de flots**
   $\rightarrow$ Ford-Fulkerson, max flow - min cut, max flow - min cost
4. **Recherche locale**
   $\rightarrow$ Heuristique, hill climbing, recuit simulé
5.