Chapitre 2 : Problèmes d'optimisation et algorithmes à garantie de performances

MPRO - Complexité : algorithmes approchés

Dimitri Watel (dimitri.watel@ensiie.fr)

2024

Problème de décision

Définition (informelle)

Un problème de décision est un problème possédant des entrées (*les instances*) et une question concernant l'entrée dont la réponse est **Oui** ou **Non** en fonction de l'entrée.

La classe de complexité P

Définition de la classe P

Un problème de décision Π est dit polynomial ou appartenant à la classe P si sa complexité est polynomiale. Autrement dit, il existe une constante c telle que la complexité de Π est $O(n^c)$.

La classe de complexité EXPTIME

Définition de la classe EXPTIME

Un problème de décision Π est dit exponentiel ou appartenant à la classe EXPTIME si sa complexité est exponentielle. Autrement dit, il existe une constante c telle que la complexité de Π est $O(2^{(n^c)})$.

$$\mathrm{P} \subsetneq \mathrm{EXPTIME}$$

La classe de complexité NP

Définition de la classe NP

Un problème de décision Π est appartient à la classe NP s'il existe une constante c et s'il existe un vérifieur $\mathcal{V}$ pour les réponses **OUI** de complexité $O(n^c)$ et $O(n^c)$.

!!!

$$\mathrm{P} \subset \mathrm{NP} \subsetneq \mathrm{EXPTIME}$$

La classe de complexité P

Définition de la classe P

Un problème de décision Π est dit polynomial ou appartenant à la classe P si sa complexité est polynomiale. Autrement dit, il existe une machine de Turing **déterministe** qui résoud Π en temps polynomial.

$$\mathrm{P} = \bigcup_{c \in \mathbb{N}} \mathrm{DTIME}(n^c)$$

La classe de complexité EXPTIME

Définition de la classe EXPTIME

Un problème de décision Π est dit exponentiel ou appartenant à la classe EXPTIME si sa complexité est exponentielle. Autrement dit, il existe une machine de Turing **déterministe** qui résoud Π en temps exponentiel :

$$\text{EXPTIME} = \bigcup_{c \in \mathbb{N}} \text{DTIME}(2^{n^c})$$

$$\text{P} \subsetneq \text{EXPTIME}$$

La classe de complexité NP

Définition de la classe NP (non-deterministic Polynomial)

Un problème de décision Π appartient à la classe NP s'il existe une machine de Turing **non déterministe** qui résoud Π en temps polynomial.

$$\mathrm{NP} = \bigcup_{c \in \mathbb{N}} \mathrm{NTIME}(n^c)$$

$$\mathrm{P} \subset \mathrm{NP} \subset \mathrm{PSPACE}$$

Réduction polynomiale de Karp

Définition (informelle)

Soit deux problèmes de décision Π_1 et Π_2, une réduction polynomiale de Π_1 vers Π_2 transforme en temps polynomial toute instance positive (resp. negative) de Π_1 en une instance positive (resp. negative) de Π_2.

Si Π_1 se réduit à Π_2, alors on dit que Π_2 est *plus difficile* que Π_1 et on note :

$$\Pi_1 \preccurlyeq \Pi_2$$

Difficulté

Problème NP-Difficile

Soit Π_1 un problème de décision. On dit que Π_1 est NP-Difficile si, pour tout problème Π_2 de NP, $\Pi_2 \preccurlyeq \Pi_1$.

Complétude

Problème NP-complet

Soit Π un problème de décision. On dit que Π est NP-Complet si $\Pi \in$ NP et s'il est NP-difficile.

Réduction polynomiale de Turing

Définition : Réduction polynomiale de Turing

Soient deux problèmes Π_1 et Π_2, une réduction polynomiale de Turing de Π_1 vers Π_2 est un algorithme $\mathcal{R}$ tel que

- $\mathcal{R}$ peut appeler un algorithme $\mathcal{R}'$ (imaginaire), appelé *oracle*, résolvant Π_2 en temps constant ;
- $\mathcal{R}$ résoud Π_1 en temps polynomial.

On note $\Pi_1 \preccurlyeq_T \Pi_2$.

Π_1 et Π_2 ne sont pas nécessairement des problèmes de décision.

Problème d'optimisation

Définition (informelle)

Un problème d'optimisation est un problème possédant des entrées (les *instances*), des solutions au problème (les *solutions réalisables*), une *mesure* ou *poids* associant à chaque solution un entier positif. L'objectif est de trouver une solution maximisant ou minimisant la mesure.

Problèmes associés à un problème d'optimisation

Définition (informelle)

À un problème d'optimisation Π est associé 3 problèmes :

- un *problème de décision* Π_D : existe-t-il une solution réalisable de poids
  - inférieur à un entier K ?, si l'objectif est de minimiser,
  - supérieur à un entier K ?, si l'objectif est de maximiser ;
- un *problème d'évaluation* Π_E : trouver le poids d'une solution optimale ;
- un *problème de construction* Π_C : trouver une solution optimale et sa valeur.

PO et NPO

Définition

PO et NPO sont les équivalents de P et NP pour les problèmes d'optimisation : les problèmes d'optimisation que l'on peut résoudre en temps polynomial déterministe ou non déterministe.

En particulier,

- $\Pi \in PO \Rightarrow \Pi_D \in P$
- $\Pi \in NPO \Rightarrow \Pi_D \in NP$

Problème d'optimisation

Définition : problème d'optimisation

Un problème d'optimisation Π est un quadruplet $(\mathcal{I}, \mathcal{S}, \mathcal{M}, \mathcal{O})$ tel que :

- $\mathcal{I}$ est un ensemble d'*instances* de Π ;
- $\mathcal{S}$ est une fonction qui à $x \in \mathcal{I}$ associe un ensemble de *solutions réalisables* de x ;
- $\mathcal{M}$ est une fonction de *mesure* qui à $x \in \mathcal{I}$ et $y \in \mathcal{S}(x)$ associe un entier positif non nul ;
- $\mathcal{O}$ est l'*objectif* et vaut min ou max et détermine si on cherche trouver une solution de mesure maximum ou minimum.

$\mathcal{I}$ et $\mathcal{S}(x)$ peuvent contenir n'importe quel objet mathématique.

Solution optimale

Définition

Soit $\Pi = (\mathcal{I}, \mathcal{S}, \mathcal{M}, \mathcal{O})$ un problème d'optimisation et $x \in \mathcal{I}$. On appelle *solution optimale* une solution $y^* \in \mathcal{S}(x)$ telle que :

$$\mathcal{M}(x, y^*) = \min_{y \in \mathcal{S}(x)} \mathcal{M}(x, y) \text{ si } \mathcal{O} = \min$$

$$\mathcal{M}(x, y^*) = \max_{y \in \mathcal{S}(x)} \mathcal{M}(x, y) \text{ si } \mathcal{O} = \max$$

On note $\mathcal{S}^*(x)$ *l'ensemble des solutions optimale* de x et $\mathcal{M}^*(x)$ la valeur $\mathcal{M}(x, y^*)$ des solutions optimales.

Taille d'une instance ou d'une solution

Définition

La taille d'une instance ou d'une solution réalisable est définie comme le nombre de bits nécessaires pour l'encoder.

On note généralement $|x|$ ou n la taille de l'entrée x et $|y|$ la taille d'une solution réalisable y de x.

Problèmes associés à un problème d'optimisation

Définition

À un problème d'optimisation $\Pi = (\mathcal{I}, \mathcal{S}, \mathcal{M}, \mathcal{O})$ est associé 3 problèmes :

- un *problème de décision* Π_D : soit $x \in \mathcal{I}$ et un entier K, déterminer si $\mathcal{M}^*(x) \leq K$ si $\mathcal{O} = \min$, ou si $\mathcal{M}^*(x) \geq K$ si $\mathcal{O} = \max$.
- un *problème d'évaluation* Π_E : soit $x \in \mathcal{I}$, calculer $\mathcal{M}^*(x)$;
- un *problème de construction* Π_C : soit $x \in \mathcal{I}$, calculer une solution optimale $y \in \mathcal{S}^*(x)$ et $\mathcal{M}^*(x)$.

Quelques résultats

Théorème

Soit Π un problème d'optimisation :

$$\Pi_D \preccurlyeq_T \Pi_E \preccurlyeq_T \Pi_C$$

$$\Pi_E \preccurlyeq_T \Pi_D \text{ si } \mathcal{M}^*(x) = O(2^{|x|^c})$$

NPO

Définition

Soit un problème d'optimisation $\Pi = (\mathcal{I}, \mathcal{S}, \mathcal{M}, \mathcal{O})$, le problème Π appartient à la classe NPO si

- soit x un nombre binaire, on peut vérifier en temps polynomial si x encode une instance de $\mathcal{I}$;
- il existe un polynome q tel que, pour tout $x \in \mathcal{I}$
  - pour tout $y \in \mathcal{S}(x)$, $|y| \leq q(|x|)$,
  - pour tout nombre binaire y tel que $|y| \leq q(|x|)$, on peut vérifier en temps polynomial si y encode une solution réalisable de x ;
- pour tout $x \in \mathcal{I}$ et tout $y \in \mathcal{S}(x)$, on peut calculer $\mathcal{M}(x, y)$ en temps polynomial.

PO

Définition

Un problème $\Pi \in$ NPO appartient à la classe PO si on peut résoudre Π_C en temps polynomial.

Par définition PO $\subseteq$ NPO.

PO, NPO, P et NP

Théorème

$$\Pi \in PO \Rightarrow \Pi_D \in P$$

$$\Pi \in NPO \Rightarrow \Pi_D \in NP$$

Problème d'optimisation NP-Difficile

Définition

Soit un problème d'optimisation Π_1 , Π_1 est NP-Difficile si, pour tout problème de décision $\Pi_2 \in \text{NP}$, $\Pi_2 \preccurlyeq_T \Pi_1$.

Théorème

Soit $\Pi \in \text{NPO}$, si Π_D est NP-Difficile alors Π est NP-difficile.

Théorème

$\text{PO} = \text{NPO} \Rightarrow \text{P} = \text{NP}$

Approximation polynomiale

Est-il nécessaire de renvoyer une solution optimale ?

Approximation polynomiale

Définition

Un *algorithme d'approximation polynomial* pour un problème Π est un algorithme polynomial qui renvoie une solution réalisable de Π "proche" d'une solution optimale.

Approximation polynomiale d'un problème de minimisation

Définition

Un *algorithme d'approximation polynomial* de rapport r pour un problème de minimisation Π ou r approximation polynomiale est un algorithme polynomial qui, pour toute instance x de Π, renvoie une solution réalisable y de Π telle que $M(x, y) \leq r(x) \cdot M^*(x)$.

r est une fonction de $\mathcal{I}$ dans $\mathbb{R}$.

Approximation polynomiale d'un problème de maximisation

Définition

Un *algorithme d'approximation polynomial* de rapport r pour un problème de maximisation Π ou r approximation polynomiale est un algorithme polynomial qui, pour toute instance x de Π, renvoie une solution réalisable y de Π telle que $M(x, y) \geq r(x) \cdot M^*(x)$.

r est une fonction de $\mathcal{I}$ dans $\mathbb{R}$.

Classes d'approximabilité : APX

Définition

Un problème $\Pi \in$ NPO appartient à la classe APX s'il existe une constante c et une c-approximation polynomiale pour Π.

$$\text{APX} \subset \text{NPO (Si P} \neq \text{NP, APX} \subsetneq \text{NPO.)}$$

Classes d'approximabilité : r-APX

Soit r une fonction ($r : \mathbb{N} \to \mathbb{R}^+$)

Définition

Un problème $\Pi \in$ NPO appartient à la classe r-APX s'il existe une $r(|\mathcal{I}|)$-approximation polynomiale pour Π.

De maninère similaire, on définit la classe $O(r)$-APX.
r-APX $\subset$ NPO (Si P $\neq$ NP, r-APX $\subsetneq$ NPO.)

Classes d'approximabilité : PTAS

Définition : PTAS

Un problème $\Pi \in$ NPO de minimisation appartient à la classe PTAS s'il existe un schémas d'approximation pour Π : pour tout $\varepsilon > 0$, il existe une $(1+\varepsilon)$-approximation polynomiale pour Π.

PTAS $\subset$ APX (Si P $\neq$ NP, PTAS $\subsetneq$ APX.)

Classes d'approximabilité : FPTAS

Définition : FPTAS

Un problème $\Pi \in$ NPO de minimisation appartient à la classe FPTAS s'il existe un schémas d'approximation fortement polynomial pour Π : pour tout $\varepsilon > 0$, il existe une $(1+\varepsilon)$-approximation polynomiale pour Π dont la complexité est polynomiale en la taille de l'instance et en $\frac{1}{\varepsilon}$.

FPTAS $\subset$ PTAS (Si P $\neq$ NP, FPTAS $\subsetneq$ PTAS.)

Classes d'approximabilité : EPTAS

Définition : EPTAS

Un problème $\Pi \in$ NPO de minimisation appartient à la classe EPTAS s'il existe un schémas d'approximation efficace pour Π : pour tout $\varepsilon > 0$, il existe une $(1+\varepsilon)$-approximation polynomiale pour Π dont la complexité est $O(f(\frac{1}{\varepsilon}) \cdot n^c)$.

FPTAS $\subset$ EPTAS $\subset$ PTAS (Si P $\neq$ NP, FPTAS $\subsetneq$ EPTAS, et si ??? $\neq$??? (cf cours de complexité paramétrée), EPTAS $\subsetneq$ PTAS.)

Peut-on caractériser les problèmes qui ont ou n'ont pas de PTAS/FPTAS ?

Codage binaire VS Codage unaire

Codage binaire

Coder un entier k en *binaire* sur une machine de Turing consiste à l'écrire en base 2 avec $\log_2(k)$ bits entourés par deux symboles blancs. La place mémoire de k est $log_2(k) + 2$.

Codage unaire

Coder un entier k en *unaire* sur une machine de Turing consiste à l'écrire avec k cases remplies de 1 entourées par deux cases blanches. La place mémoire de k est $k + 2$.

Complexité et encodage

Constatation

Soit un problème de décision Π, une instance x de Π contenant un entier W et un algorithme $\mathcal{A}$ de complexité $O(W)$ résolvant Π.

- Si W est codé en unaire, $\mathcal{A}$ est polynomial.
- Si W est codé en binaire, $\mathcal{A}$ est exponentiel.

NP-Complétude forte

Définition

Soit un problème de décision Π, on dit que Π est *fortement NP-Complet* (ou *NP-Complet au sens fort*) s'il est NP-Complet quand on encode les entiers contenus dans les entrées en unaire.

NP-Complétude faible

Définition

Soit un problème de décision Π, on dit que Π est *faiblement NP-Complet* (ou *NP-Complet au sens faible*) s'il est NP-Complet mais qu'il existe un algorithme polynomial pour le résoudre si on encode les entiers contenus dans les entrées en unaire.

Taille de l'entrée usuelle : 2^{e} essai

Définition

Soit une entrée x contenant un ensemble x_0 d'objets non numériques et un ensemble de k entiers $(x_1, x_2, \ldots, x_k)$, on définit la taille de x de plusieurs manières :

- sa taille totale $|x|$ usuelle : le nombre de bits pour tout encoder ;
- la taille $l(x)$ indépendante de la valeur des entiers : $|x_0| + k$;
- la taille $\max(x)$ relative aux entiers : $\max_{[\![1;k]\!]} x_i$.

NP-Complétude forte : définition alternative

Définition

Soit un problème de décision Π, on dit que Π est *fortement NP-Complet* (ou *NP-Complet au sens fort*) s'il est NP-Complet quand on se restreint aux instances où $\max(x)$ est polynomialement borné par rapport à $I(x)$.

Théorème

Les deux définitions de NP-Complétude forte sont équivalentes.

NP-Complétude faible

Définition : complexité pseudo-polynomiale

Soit un problème de décision Π, un algorithme résolvant Π est *pseudo-polynomial* si sa complexité en temps est polynomiale en $I(x)$ et $\max(x)$.

Définition

Soit un problème de décision Π, on dit que Π est *faiblement NP-Complet* (ou *NP-Complet au sens faible*) s'il est NP-Complet mais qu'il existe un algorithme pseudo-polynomial pour le résoudre.

Théorème

Les deux définitions de NP-Complétude faible sont équivalentes.

Fort $\neq$ Faible

Théorème

Si P $\neq$ NP alors les problèmes NP-Complets au sens fort ne sont pas NP-Complets au sens faible et inversement.

Montrer qu'un problème est fortement NP-Complet.

Théorème

Soit Π un problème NP-Complet ne contenant pas d'entier, alors Π est NP-Complet au sens fort.

Théorème

Soit Π_1 un problème fortement NP-Complet et Π_2 un problème de décision. S'il existe une réduction polynomiale de Karp telle que, pour toute instance x de Π_1 transformée en instance y de Π_2 , $\max(y)$ est borné polynomialement par rapport à $l(x)$ et/ou $\max(x)$ alors Π_2 est NP-Complet au sens fort.

NPO fort

Définition

Soit un problème d'optimisation $\Pi \in$ NPO, on dit que Π est *fortement NP-Difficile* (ou *NP-Difficile au sens fort*) s'il est NP-Difficile quand on se restreint aux instances où $\max(x)$ est polynomialement borné par rapport à $l(x)$.

NPO faible

Définition

Soit un problème d'optimisation $\Pi \in$ NPO, on dit que Π est *faiblement NP-Difficile* (ou *NP-Difficile au sens faible*) s'il est NP-Difficile mais qu'il existe un algorithme pseudo-polynomial pour le résoudre.

FPTAS et algorithme pseudo-polynomial

Théorème

Soit $\Pi \in$ FPTAS tel que $\mathcal{M}^*(x)$ est polynomialement borné par $I(x)$ et $\max(x)$, alors il existe un algorithme pseudopolynomial pour résoudre Π.

Corollaire

Soit $\Pi \in$ NPO un problème NP-Difficile au sens fort tel que $\mathcal{M}^*(x)$ est polynomialement borné par $I(x)$, alors $\Pi \notin$ FPTAS.

Rapport d'approximation absolu

Définition

Un *algorithme d'approximation polynomial* de rapport **absolu** r pour un problème de minimisation Π est un algorithme polynomial qui, pour toute instance x de Π, renvoie une solution réalisable y de Π telle que $M(x, y) \leq M^*(x) + r(x)$.

r est une fonction de $\mathcal{I}$ dans $\mathbb{R}$.

Rapport d'approximation absolu

Définition

Un *algorithme d'approximation polynomial* de rapport **absolu** r pour un problème de maximisation Π est un algorithme polynomial qui, pour toute instance x de Π, renvoie une solution réalisable y de Π telle que $M(x, y) \geq M^*(x) - r(x)$.

r est une fonction de $\mathcal{I}$ dans $\mathbb{R}$.

Classes d'approximabilité : AAPX

Définition

Un problème $\Pi \in$ NPO appartient à la classe AAPX s'il existe une constante c et une c-approximation polynomiale absolue pour Π.

AAPX $\subset$ NPO (Si P $\neq$ NP, AAPX $\not\subset$ PTAS et PTAS $\not\subset$ AAPX.)
On définit de même APTAS, AFPTAS, A-$O(r)$-APX, ...

Rapport d'approximation asymptotique

Définition

Un *algorithme d'approximation polynomial* de rapport **asymptotique** r pour un problème de minimisation Π est un algorithme polynomial tel qu'il existe une constante $k \in \mathbb{R}$ et telle que, pour toute instance x de Π, renvoie une solution réalisable y de Π telle que $M(x, y) \leq M^*(x, y) \cdot r(x) + k$.

r est une fonction de $\mathcal{I}$ dans $\mathbb{R}$.

Rapport d'approximation asymptotique

Définition

Un *algorithme d'approximation polynomial* de rapport **asymptotique** r pour un problème de maximisation Π est un algorithme polynomial tel qu'il existe une constante $k \in \mathbb{R}$ et telle que, pour toute instance x de Π, renvoie une solution réalisable y de Π telle que $M(x, y) \geq M^*(x) \cdot r(x) - k$.

r est une fonction de $\mathcal{I}$ dans $\mathbb{R}$.

Classes d'approximabilité : APX_∞

Définition

Un problème $\Pi \in NPO$ appartient à la classe APX_∞ s'il existe une constante c et une c-approximation polynomiale asymptotique pour Π.

$$APX_\infty = APX$$

On définit de même $PTAS_\infty$, $FPTAS_\infty$, $O(r)$-APX_∞, ...

Si $P \neq NP$, $PTAS_\infty \neq PTAS$.

Autres rapports

- Rapport moyen : se comparer à la moyenne des valeurs des solutions réalisables
- Différentiel : biaiser le rapport solution/optimal avec le poids ω d'une pire solution : $\frac{M(x,y)-\omega}{M^*(x)-\omega}$.