US336C - Programmation mathématique Avancée

Mis à jour le 12-02-2024

Code : US336C

Unité spécifique de type cours

2 crédits

Responsabilité nationale :
EPN05 - Informatique / Safia KEDAD SIDHOUM

Contact national :
Recherche opérationnelle
2D4P20, 33-1-10, 2 rue Conté
75003 Paris
01 40 27 22 67

secretariat.ro@cnam.fr

Présentation

Prérequis

Cours de programmation mathématique

Objectifs pédagogiques

Les objectifs de ce cours sont d'apprendre les techniques de résolution d'applications réelles ayant des non-linéarités:

- Modéliser un problème réel par un problème d'optimisation non-linéaire
- Étudier les méthodes exactes pour résoudre des :
  - problèmes de programmation quadratique, non-convexe avec variables entières
  - problèmes de programmation non-linéaire, convexe
  - problèmes de programmation non-linéaire, non-convexe en nombres entiers
- Apprendre un langage de modélisation
- Coder les méthodes pour résoudre les problèmes étudiés

Les séances théoriques seront alternées avec des séances pratiques. Pendant les séances pratiques, des applications réelles dans le domaine de l'énergie ou de la science des données seront présentées.

Compétences

- Modéliser un problème issu d'une application réelle comme un programme non-linéaire en nombres entiers (PNLNE)
- Connaître les méthodes exactes pour les différentes classes de problèmes
- Identifier la méthode de résolution la plus appropriée pour résoudre une application réelle
- Implémenter des formulations d'applications réelles grâce aux langages de modélisation
- Combiner les résultats des solveurs disponibles pour résoudre des applications réelles

Programme

Contenu

- **Introduction aux MIQPs** (A. Lambert)
  Nous introduisons les problèmes quadratiques en variables mixtes (MIQP) et étudions leurs propriétés. Nous proposons ensuite des applications réelles qui peuvent se modéliser par un MIQP. Enfin, après avoir introduit les bases des langages de modélisation, nous implémentons les modèles introduits et résolvons des petites instances.

- **Algorithmes exacts pour la résolution des MIQPs -- basés sur les relaxations SDP** (A. Lambert)
  Nous introduisons les concepts de base de l'optimisation semi-définie (SDP), et montrons comment utiliser les relaxations SDP dans des algorithmes de résolution exacts de MIQPs non-convexes. Nous montrons également comment utiliser, en pratique, les solveurs d'optimisation semi-définie.

- **Algorithmes exacts pour la résolution des MIQPs -- basés sur les relaxations quadratiques convexes** (A. Lambert)
  Nous introduisons les concepts de base de la construction de relaxations quadratiques convexes basées sur les relaxations SDP. Nous montrons ensuite comment ces relaxations peuvent être utilisées dans un algorithme de résolution global. Enfin, nous implémentons cette approche combinant l'utilisation des langages de modélisation et des solveurs d'optimisation SDP.

- **Introduction aux problèmes de programmation non-linéaire, convexes en nombres entiers et motivations** (C. D'Ambrosio).
  Exemple de problèmes PNLNE dans différents domaines comme la finance et la production d'énergie. Introduction de l'extension des méthodes issues de la PLNE aux problèmes PNLNE convexes, en particulier : branch-and-bound, outer-approximation, et méthodes hybrides. Nous allons analyser les spécificités des problèmes PNLNE convexes et les défis qu'ils représentent. La session pratique consistera en la modélisation d'applications par un PNLNE convexe et en sa résolution avec un langage de modélisation.

- **Introduction aux problèmes de programmation non-linéaire, non-convexes en nombres entiers et motivations** (C. D'Ambrosio)
  Pourquoi les méthodes exactes pour les problèmes PNLNE convexes s'avèrent inexactes pour les problèmes non-convexes ? En particulier : Optimalité globale vs. locale ; les coupes de outer-approximation ne sont pas valides pour les PNLNE non-convexes. Nous allons introduire et analyser des algorithmes exacts pour les problèmes PNLNE (relaxation de Mc Cormick, branch-and-bound spatial). Dans la session pratique, nous allons modéliser des applications par un PNLNE non-convexe et le résoudre avec un langage de modélisation.

Modalités de validation

- Examen final