RCP104 - Optimisation en informatique

Mis à jour le 10-02-2025

Code : RCP104

Unité d'enseignement de type mixte

6 crédits

Volume horaire de référence (+/- 10%) : **50 heures**

Responsabilité nationale : EPN05 - Informatique / Cédric BENTZ

Contact national :

EPN05 - Informatique

2 rue Conté
33.1.9A
75003 Paris
01 58 80 87 99
Jean-mathieu Codassé
jean-mathieu.codasse@lecnam.net

Présentation

Prérequis

Élèves ingénieurs, étudiants de master M1.
Prérequis : avoir des connaissances de base en algorithmique, réseaux informatiques, programmation, graphes et recherche opérationnelle.

Objectifs pédagogiques

A partir de problèmes concrets en informatique (majoritairement issus des réseaux de télécommunication), l'UE vise à apprendre à traiter des problèmes d'optimisation difficiles. En particulier, elle vise à apprendre à écrire un problème d'optimisation sous la forme d'un modèle mathématique, puis à proposer des méthodes, exactes ou non (mais efficaces malgré tout), utilisant des outils pratiques pour résoudre un tel problème (méthodes heuristiques, logiciels de programmation linéaire, programmation dynamique, etc.).

Lorsque l'UE est dispensée en modalité hybride, cela signifie que, non seulement tous les supports de cours/ED/TP sont évidemment disponibles dans l'espace Moodle de l'UE, mais en outre toutes les séances de cours/ED ont lieu en présentiel et sont enregistrées puis mises à disposition dans ce même espace Moodle pour pouvoir être (re)visionnées par la suite.

Compétences

L'élève ayant suivi cet enseignement sait comment traiter des problèmes d'optimisation dans les réseaux de différents types.

Il ou elle sait notamment les modéliser sous la forme de programmes linéaires, en nombres entiers ou non.

Il ou elle sait également identifier certains problèmes simples, et les résoudre ensuite à l'aide d'algorithmes efficaces connus.

Enfin, il ou elle sait résoudre les problèmes difficiles (souvent des problèmes d'optimisation combinatoire) à l'aide d'outils issus de la recherche opérationnelle, comme la programmation dynamique, les méthodes de résolution de PLNE (à travers des solveurs de PLNE, qui implémentent ces méthodes), ou des méthodes de résolution approchée (heuristiques et méta-heuristiques).

Programme

Contenu

1) Présentation de problèmes simples d'optimisation dans les réseaux : routage (acheminement de flux de données sous contraintes de bandes passantes), tables de routage (via l'algorithme de Dijkstra), arbres de connexion simples (via l'algorithme de Kruskal).

2) Modélisation de problèmes d'optimisation par la programmation linéaire (PL) : choix des variables, détermination de leurs domaines, écriture de la fonction objectif et des contraintes. Formulation par la PL de problèmes d'acheminement de flux de données (problèmes de flots maximums ou à coût minimum). Méthode de résolution graphique pour la PL.

3) Mise en oeuvre informatique utilisant un solveur (logiciel de résolution) de PL libre d'accès (a priori, GLPK), par le biais d'un modeleur (GMPL) ou du format de fichier LP.

4) Problèmes difficiles en optimisation dans les réseaux : retour sur les arbres de connexion dans le cas général, et leurs liens avec les arbres de Steiner. Notions de complexité des algorithmes et des problèmes, et catégorisation des problèmes simples et difficiles (P versus NP).

5) Modélisation et résolution de problèmes difficiles (essentiellement combinatoires) d'optimisation dans les réseaux à l'aide de la programmation linéaire en nombres entiers (PLNE) : particularités des modèles en variables binaires ou entières, et formulations pour des problèmes classiques (comme les arbres de connexion dans le cas général, l'allocation de fréquences, le routage multicast, etc.). Linéarisation de problèmes d'optimisation non linéaires (contenant, par exemple, des produits de variables) de façon à pouvoir les résoudre via des solveurs de PLNE. Introduction aux techniques de résolution de PLNE, et aux inégalités valides. Mise en oeuvre informatique pendant une ou deux séances, puis étude d'un cas réel, sous la forme d'un TP noté.

6) Résolution de certains problèmes difficiles par des méthodes basées sur la programmation dynamique, lorsque cela est possible : principe d'optimalité de Bellman, et exemples d'algorithmes de programmation dynamique pour des problèmes classiques en optimisation dans les réseaux.

7) Résolution approchée de problèmes d'optimisation difficiles par des méthodes générales (recuit simulé, méthode tabou, algorithmes génétiques, etc.) ou par des méthodes spécifiques (heuristiques ad-hoc, gloutonnes, par recherche locale, etc.). Validation des résultats obtenus par ces heuristiques à l'aide de bornes basées, par exemple, sur la résolution du problème (ou d'une relaxation) par un solveur. Etude de la résolution approchée d'un cas réel, sous la forme d'un TP noté.

Il est à noter que le plan de cette UE a notamment vocation à refléter la démarche de résolution de problèmes d'optimisation dans les réseaux suivie dans l'UE. On commencera par identifier le problème à traiter comme simple ou difficile (du point de vue de la complexité). Si ce problème est difficile, on cherchera alors dans un premier temps à le formuler comme un programme mathématique, ou à utiliser la programmation dynamique pour le résoudre. Si, à l'issue de cette première étape, la formulation du problème utilisée ne permet pas d'obtenir de façon suffisamment efficace une solution optimale à l'aide d'un solveur, ou bien si le problème ne peut pas être résolu efficacement par la programmation dynamique, on s'intéressera dans un second temps à l'obtention d'une solution approchée. En particulier, on cherchera à répondre à la question suivante : comment obtenir rapidement une bonne solution approchée, et ensuite évaluer la qualité de cette solution ?

Modalités de validation

- Contrôle continu
- Examen final

Description des modalités de validation

Examen noté (sur 16 points).
Un ou deux TP noté(s) (sur 4 points en tout).

Bibliographie

Titre	Auteur(s)
Optimisation discrète (Dunod)	Alain Billionnet
Algorithmes de graphes (Eyrolles)	Philippe Lacomme, Christian Prins, Marc Sevaux
Métaheuristiques pour l'optimisation difficile (Eyrolles)	Johann Dréo, Alain Pétrowski, Patrick Siarry, Eric Taillard
Réseaux informatiques : conception et optimisation	Malek Rahoual et Patrick Siarry