Recherche Opérationnelle:
Apprentissage par la pratique et par projets

Sophie Demassey

GIPAD GS2
UV Optimisation Combinatoire
2011-2012

Ce document présente l'organisation, le déroulement et les objectifs de formation du module *Recherche Opérationnelle : apprentissage par la pratique et par projets*.

1 Recherche Opérationnelle par la pratique

Le module Recherche Opérationnelle vise trois objectifs de formation :
- **savoir optimiser** (rechercher, identifier, étudier/appliquer et mettre en œuvre/implémenter une approche de résolution sur une problématique d'optimisation donnée) ;
- **savoir valider et argumenter** le choix d'une approche ;
- **découvrir**, à travers des études de cas, un large éventail de contextes applicatifs de l'optimisation combinatoire.

La pratique à travers la réalisation de projets est la forme d'apprentissage qui nous semble la mieux convenir à ces objectifs. En espérant que vous apprécierez !

1.1 Principal objectif de formation : savoir optimiser

Résoudre une problématique d'optimisation, se fait en trois grandes phases : modéliser le problème, sélectionner la méthode, expérimenter.

Une infinité d'applications, un nombre limité de modèles. Des problèmes d'optimisation combinatoire se rencontrent dans pratiquement tous les secteurs de l'industrie, des services et des loisirs : conception des réseaux de télécommunication ou d'acheminement du gaz ; déplacements de troupes militaires ou de déneigeuses ; localisation d'usines ou de forages pétroliers ; agencement d'entrepôts et jeux géométriques de placement ; ordonnancement de processeurs ou de la production de biens manufacturiers ; planification de personnels ou de tournois sportifs ; reconnaissance de texte ou séquencement d'ADN ; vérification de code informatique, configuration de programmation musicale, valorisation de portefeuilles boursiers, etc.

Les problématiques d'optimisation sont indénombrables, mais elles se regroupent en un nombre fini de catégories de problèmes qui ont fait l'objet d'études plus ou moins intensives en Recherche Opérationnelle, depuis un demi-siècle.

Analyser une problématique d'optimisation, c'est avant tout faire abstraction du contexte applicatif pour se ramener à une catégorie – ou des catégories si la problématique est composée – classique de problèmes : autrement dit, c'est formaliser puis **modéliser**.

Exemple 1 *On modélisera un problème d'alignement de séquences ADN en un problème de plus court chemin dans un graphe dédié [3].*

Reconnaître les techniques appropriées. La seconde étape consiste à rechercher les méthodologies de résolution de la littérature qui ont été appliquées dans cette catégorie, et sélectionner la méthodologie la plus appropriée à la problématique, au vu de ses spécificités, et l'adapter au besoin.

Exemple 2
1. *Pour le calcul d'un plus court chemin dans un graphe, on considérera l'algorithme de Dijkstra* $O(n^2)$ *plutôt que celui de Bellman-Ford* $O(n^3)$ *si les longueurs des arcs sont toutes positives ;*
2. *pour le calcul de parcours avec conditions de trafic en temps-réel (par un GPS typiquement), où le graphe est dynamique et excessivement large et le temps de réponse court, on considérera un algorithme de plus court chemin incrémental en temps linéaire au pire : obligatoirement une méthode d'approximation (réduction du graphe par clustering, par exemple [2]).*

Expérimenter. Enfin, le comportement théorique (pire cas) d'une méthode ne permet pas exactement de prévoir le comportement expérimental (sur une instance donnée). Pour l'évaluer, il est nécessaire de prototyper, de tester (validité et performance du code et de la méthode) sur les instances types de la problématique, d'ajuster l'implémentation en fonction de ces résultats.

1.2 Acquis scientifiques et techniques

Ce module vise à vous faire acquérir un socle de connaissances théoriques et pratiques de la recherche opérationnelle.

Cadres de modélisation. Les **graphes** et la **programmation mathématique** sont des environnements de modélisation puissants qui s'appliquent à une large majorité des problèmes combinatoires. Vous les appliquerez à pratiquement toutes les problématiques étudiées : parcours, chemins, flots, couplages et recouvrements dans les graphes ; programmation linéaire, programmation linéaire en nombre entiers et décompositions. On abordera également les modèles de **programmation dynamique** à travers une application.

Techniques de résolution. À défaut d'être exhaustif, on manipulera les principales techniques de résolution de la Recherche Opérationnelle et leur combinaison : algorithmes polynomiaux de graphe ; relaxations linéaires, branch-and-bound, méthodes de coupes, génération de colonnes ; heuristiques, metaheuristiques et recherche locale ; algorithmes ad-hoc d'ordonnancement et de transport.

Implémentation et tests. Chaque projet donnera lieu à l'implémentation d'une méthode (graphe, programmation linéaire et metaheuristique) et à l'expérimentation sur différents jeux de tests. Le choix des langages et des outils est libre.

1.3 Autres compétences cibles

Au-delà des connaissances en recherche opérationnelle, les différentes activités que comportent ce module visent à vous faire acquérir ou maîtriser de nombreuses compétences :
- analyser et formaliser des cas d'étude en contextes variés
- appliquer les techniques de modélisation
- effectuer une recherche bibliographique
- analyser et synthétiser un document scientifique
- transposer, adapter, combiner des méthodes génériques à des cas d'étude
- supposer, vérifier, prouver
- implémenter à partir d'une spécification d'algorithme
- valider un programme
- analyser les performances d'un programme
- rédiger un rapport scientifique
- réaliser une présentation orale scientifique
- travailler en équipe, coordonner une équipe

2 Organisation du module

Le module de recherche opérationnelle GIPAD-GS2 occupe 54 heures à l'emploi du temps, auxquelles s'ajoute un temps de travail personnel estimé à environ 30 heures. Il est organisé en une séquence de trois projets thématiques dirigés.

2.1 Projets dirigés

Déroulement. Trois projets dirigés de 15h chacun sont prévus sur les thématiques suivantes :

projet 1 : réseaux et transport (sep-oct ; Sophie Demassey)

projet 2 : packing et découpe (nov ; Sophie Demassey)

projet 3 : ordonnancement et sur-contraint (Thierry Petit)

Chaque projet est composé de 6 séances de 2h30. Il s'agit de séances de travail de groupe encadrés, à l'exception éventuellement de quelques séances de cours académiques pour la présentation des concepts les plus complexes. À la suite de chaque projet, une séance de 3h est consacrée à l'évaluation individuelle écrite. Enfin, une partie du temps de travail personnel et en séance du groupe doit être consacrée à l'étude approfondie d'une problématique au choix dans le thème du projet ; cette étude donnant lieu au développement d'un moteur de résolution et à la rédaction d'un rapport technique par groupe.

Travail en séance. Un *livret étudiant* est délivré à chacun contenant les exercices à réaliser lors d'une séance, ou en préparation d'une séance. Les exercices portent sur : la définition de concepts, la modélisation de problématiques, la recherche, l'étude et l'analyse de méthodes de résolution. **La recherche d'informations faisant partie intégrante de cette formation**, aucune référence bibliographique précise ne vous est donnée (sauf cas exceptionnels). En séance, la consultation de tous documents est autorisée ; elle est même obligatoire : le web [4] ; vos anciens cours ; des livres seront mis à votre disposition mais vous pouvez également en apporter ; des articles ; etc.

Les livrets sont volontairement longs : certains exercices pourront ne pas être traités. Tout exercice abordé devra cependant être traité avec rigueur. L'énoncé du livret au format LaTeX est fourni et peut être utilisé comme base de rédaction du document de travail du groupe. Ce document pourra être fait validé au fur et à mesure de son avancement par les encadrants. Le document de travail n'est pas évalué directement. En revanche, l'évaluation individuelle écrite est basée entièrement sur les exercices du livret.

Implémentation et rapport technique. Au cours de chacun des 3 projets, chaque groupe devra mener l'étude approfondie théorique et expérimentale d'une problématique dans la thématique du projet. Problématiques et méthodes sont imposées dans le cadre du dernier projet, et libres dans le cadre des deux premiers projets.

À l'issue du projet, chaque groupe remettra le code informatique du moteur d'optimisation qu'il a développé ainsi qu'un un rapport technique comprenant :

- la description de la problématique et des techniques de résolution choisies
- la documentation du code et des tests de validation et de performance,
- l'analyse des résultats expérimentaux
- l'analyse de la méthode choisie

Composition des groupes. Les projets dirigés sont réalisés en groupes de 3 ou 4 personnes. Au début de chaque projet, les fonctions principales seront attribuées selon les règles suivantes :

- par équipe : 1 *co-ordonnateur*, 1 *rédacteur*, 2 *codeurs/testeurs*
- le *co-ordonnateur* est responsable de l'avancement de la réalisation de l'équipe : il tient l'horloge ;
- le *rédacteur* est responsable du rapport écrit : il assemble le document, en est le principal rédacteur ;
- les deux *programmeurs* sont responsables de l'implémentation et travaillent sur le même code. Le *codeur* est plutôt en charge du moteur de résolution, le *testeur* est en charge de la vérification, des benchmarks et de la sortie des résultats ;
- les rôles de chacun alterneront à chaque projet.

Encadrant. Les encadrants sont présents durant les séances de projet dans le but : de guider et de vérifier l'avancement des projets, d'aider à la compréhension des sujets et des concepts abordés, de valider vos choix. En dehors des heures planifiées à l'emploi du temps, ils peuvent également être sollicités (rendez-vous par mail) en tant que *consultants*. Enfin, ils sont responsables de l'évaluation des projets.

3 Évaluations et critères retenus

Les évaluations portent directement (code+rapport) ou indirectement (DS) sur un le travail de groupe : **le groupe est responsable de son homogénéité**. [1]

Chaque projet donne lieu à 3 évaluations :
- code informatique (groupe)
- rapport technique (groupe)
- évaluation individuelle écrite

Les **critères d'évaluation du code informatique** portent sur, dans l'ordre : la validité, la clarté, la documentation, les interfaces (input/output), les performances.

Le principal **critère d'évaluation des exercices écrits** (rapport technique, DS, livret étudiant) est la **rigueur** : les rapports doivent être synthétiques (directs, concis et propres) et honnêtes à défaut d'être complets. En particulier :
- le vocabulaire précis doit être employé ;
- tout document consulté et utilisé (livre, web, article, cours,...) doit être **cité** ;
- toute définition doit être attestée par un document existant (à référencer) ;
- toute proposition doit être clairement énoncée et **argumentée** soit, également, par une référence à un document qui en atteste, soit par une preuve formelle écrite [2] ;
- à défaut (manque de temps ou difficulté technique), il faudra explicitement signaler que la supposition n'est pas vérifiée.

Le document [1] vous présente un exemple-type de rédaction attendue. Tous les sources LaTeX vous sont également fournis comme base de rédaction des documents de travail (le format LaTeX n'est pas obligatoire).

Enfin, les exercices sont pour la plupart dirigés mais une grande part d'autonomie vous est laissée, dans le choix de la documentation et des outils, par exemple, et surtout dans les exercices d'approfondissement. Les prises d'initiative sont donc considérées très favorablement – quelqu'en soient les résultats – si elles sont correctement motivées. N'hésitez pas : les encadrants sont présents pour vous conseiller.

Références

[1] S. Demassey. *GIPAD GS2 – APP Recherche Opérationnelle : exemple*. École des Mines de Nantes, 2008.

[2] G. Nannicini, P. Baptiste, D. Krob, and L. Liberti. Fast point-to-point shortest path queries on dynamic road network with interval data. In *CTW*, 2007.

[3] M.S. Waterman. *Mathematical methods for DNA sequences*. CRC Press, 1988.

[4] http://www.google.com.

1. Cela fait également partie de la formation.
2. ou bien les deux – une référence et une preuve rédigée – si le résultat est important et la preuve courte

GIPAD GS2
APP Recherche Opérationnelle: un exemple

Sophie Demassey

5 septembre 2011

Ce document donne un bref exemple de présentation de rapport pour les projets d'APP Recherche Opérationnelle,

- pour la forme : le document LaTeX montre comment utiliser une figure, un théorème ou une définition, une référence, une citation et une bibliographie, etc.
- pour le fond : il insiste sur les points importants attendus, comme :
  - la citation des sources : dans cet exemple, la définition de graphe eulérien est extraite de [1]. Elle n'est pas strictement équivalente à celle donnée en cours de graphe, pourtant ni l'une, ni l'autre n'est fausse ! Cela se produit fréquemment. C'est pour cette raison qu'il est nécessaire de bien définir les notions employées et de citer les références bibliographiques correspondantes ;
  - la précision et la concision : en gros, mieux vaut une citation qu'un discours approximatif. Parfois, il peut être utile (mais pas nécessaire) de relever des points importants qui aident (qui vous ont aidés) à la compréhension (voir par exemple, la preuve du théorème 3) ;
  - l'argumentation systématique (voir, la preuve de la proposition 4), ou à défaut, le signalement des observations non prouvées (voir la preuve de la proposition 5) ;
  - l'emploi du vocabulaire consacré (voir par exemple, les notions de *bornes* ou bien d'*incidence*) ;

1 Graphes eulériens

Question 1 *Le facteur d'un quartier de Pékin distribue chaque jour le courrier et doit pour cela parcourir l'ensemble des rues de son quartier, dont le plan est représenté par le multigraphe* G *non-orienté de la figure 1 : les arêtes sont les rues et les sommets les carrefours. La poste (dont part et revient sa tournée) est située sur le carrefour* 2*. Il lui faut exactement 10 minutes pour parcourir chaque rue. Estimez la durée minimale d'une tournée.*

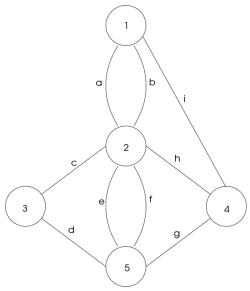

FIGURE 1 – Modélisation du quartier par un graphe G

Par définition, une tournée correspond à une chaîne (séquence d'arêtes ou séquence de sommets reliés par des arêtes [2]) fermée dans G passant au moins une fois par chaque arrête de G.

Proposition 1 *La durée minimale d'une tournée est exactement de* ***100 minutes****.*

Nous établissons la preuve de cette conjecture ci-après :
- la proposition 2 montre, avec une solution particulière, qu'il s'agit bien d'une borne supérieure de la durée minimale d'une tournée ;
- la proposition 6 montre qu'il s'agit également d'une borne inférieure.

Proposition 2 *La durée minimale d'une tournée est inférieure ou égale à 100 minutes.*

Preuve. La chaîne définie par la séquence des arêtes ci-dessous, commence et termine au sommet 2, elle contient 10 arêtes et au moins une fois chaque arête de G.

$$(c, d, e, h, g, f, a, i, i, b)$$

C'est donc une tournée possible de durée égale à 100 minutes. ■

La seconde partie de la preuve de la proposition 1 est basée sur la notion de chemin eulérien :

Définition 1 *Un* chemin eulérien *dans un graphe* G *est une chaîne fermée contenant chaque arête de* G *exactement une fois. Un graphe* G *possédant un chemin eulérien est dit* graphe eulérien. *[1, p.36]*

Théorème 3 *Un graphe* G *est eulérien si et seulement s'il est connexe et si chaque sommet de* G *est de degré pair.*

Preuve. Se référer à [1, p.37]. [L'intuition de la condition nécessaire est la suivante : Si G possède un chemin eulérien alors ce chemin connecte tous les sommets deux à deux (G est donc connexe) et le chemin entre dans un sommet autant de fois qu'il en ressort et par des arêtes distinctes (le degré des sommets est donc pair). La preuve de la condition suffisante se fait par induction sur le nombre d'arêtes du graphe.] ■

Proposition 4 *Le multigraphe* G *de la figure 1 n'est pas eulérien.*

Preuve. Les sommets 1 et 4 ont 3 arêtes incidentes, ils sont donc de degrés impairs, et, d'après le théorème 3, G n'est pas eulérien. ■

Proposition 5 *Une tournée contient au moins 10 arêtes.*

Preuve. NON TERMINÉE. Intuition : Le graphe G n'est pas eulérien, d'après la proposition 4, et il contient 9 arêtes. ■

Proposition 6 *La durée d'une tournée est au moins égale à 100 minutes.*

Preuve. C'est une conséquence directe de la proposition 5. ■

Références

[1] E. Lawler. *Combinatorial Optimization : Networks and Matroids*. Holt, Rinehart & Winston, 1976.

[2] http://www.apprendre-en-ligne.net/graphes/lexique/.