

# Descriptif du projet de Programmation Linéaire en Nombres Entiers (PLNE)

Nicolas DUPIN

dupin@lri.fr

4 décembre 2019

Cours de PLNE  
Polytech' Paris-Sud, Université Paris-Saclay,



# Evaluation du cours de PLNE

- ▶ Note du cours de PLNE : 50% sur l'examen (le 14 janvier), 50% sur la partie projet.
- ▶ Des questions générales à l'examen évaluent la compréhension globale du cours et des TD/TP. La meilleure révision pour cela : le projet.
- ▶ Le projet : application de techniques exactes et heuristiques sur un problème d'application.
- ▶ Problématique 1, clustering : p-median/ k-medoids/ p-centre/ Min-Sum-k-radii tels que définis sur les planches du cours 1.
- ▶ Problématique 2, graphes : maxStable/maxClique/coloration (sommets) tels que définis sur les planches du cours 1.

# Vision générale du projet de PLNE

- ▶ Le projet : application de techniques exactes et heuristiques sur un problème d'application.
- ▶ Méthode exacte : implémentation modèle PLNE avec un solveur, analyse de caractéristiques de convergence PLNE.
- ▶ Heuristique : coder au moins une approche constructive et une approche perturbative.
- ▶ Travail d'analyse : différence de résultats empiriques entre méthodes exactes et heuristiques, voire possibilités de collaboration.
- ▶ Approfondissement(s) au choix : vous permettent d'appliquer des points spécifiques du cours.
- ▶ Travail de projet en binôme
- ▶ Programmation dans le langage de programmation de votre choix parmi : OPL, C/C++, Python, Java, OCaml, Julia

# Jeux de données

- ▶ Sur les problèmes d'une même problématique, les données d'entrée sont identiques.
- ▶ Des données fournies, définissant un format OPL .dat.
- ▶ Graphe : des jeux de données OPL sont fournis dans le dossier minCover du TP.
- ▶ Clustering : un fichier sera fourni dans le dossier Projet. Coder un générateur d'instances au même format que les données fournies

# Génération de jeux de données pour projet clustering

- ▶ Créer un générateur de données aléatoire.
- ▶ Paramètres : nb de points, dimension, mode de génération.
- ▶ Plusieurs modes de génération suivant l'aléatoire : comment générer des données réalistes pour la problématique ?
- ▶ N.B : génération uniforme sur  $\mathbb{R}^d$  donne des données très faciles pour le clustering ...
- ▶ Forme de données à générer : avoir un nombre de clusters naturellement formés (paramètre  $N$ ), un nombre de points aléatoirement (pas d'équirépartition) répartis dans les  $M$  clusters naturels.
- ▶ Analyse sur projet : Quel impact sur la difficulté de résolution ?
- ▶ Les programmes d'optimisation doivent s'appeler sur les jeux de données au format OPL, demander en entrée un nom/chemin de fichier, une vérification sera effectuée sur des fichiers autres que ceux que vous aurez généré...

# Méthodes exactes

- ▶ Implémentation modèle PLNE avec Cplex ou GLPK : peut utiliser les modeleurs (OPL ou MathProg GLPK) ou générer les fichiers .lp depuis un code.
- ▶ Clustering : implémentation directe pour p-centre/ Min-Sum-k-radii. Pour p-median (et donc k-medoids), plusieurs variantes.
- ▶ Graphe : implémentation directe pour maxStable/maxClique. Plus de variantes avec problématique des symétries sur le pb de coloration (sommets).
- ▶ Pour les problèmes à plusieurs formulations, analyser l'impact de la formulation sur la convergence exacte.
- ▶ Comparer l'efficacité de la résolution PLNE pour les différentes variantes, sur mêmes données. Pouvez vous expliquer les différences observées ?
- ▶ Graphe : stables et cliques sont complémentaires. Les résolution PLNE avec passage au graphe complémentaire sont elles identiques ?
- ▶ optionnel Graphe : Coder pour maxStable avec l'algo B&B de l'amphi 2. Etude : utilisation heuristiques maxClique pour accélérer convergence.

# Analyse d'une formulation PLNE

	varBin	LP	time	inf0	sup0	time	inf	sup	time
A1	3892	0,17%	0,09	0,00%	0,00%	1,1	0,00%	0,00%	1,1
A2	78896	0,22%	0,4	0,02%	0,00%	3,8	0,00%	0,00%	4,9
A3	8162	0,36%	0,11	0,05%	0,00%	4,4	0,00%	0,00%	3
A4	17465	0,95%	3,1	0,42%	0,09%	21,4	0,00%	0,00%	334
A5	15357	0,90%	7,5	0,53%	0,44%	55	0,15%	0,00%	3600
B6	24563	2,23%	11,2	0,61%	0,51%	64	0,13%	0,00%	3600
B7	35768	2,90%	66,7	0,78%	1,99%	346	0,52%	0,98%	3600
B8	69653	9,47%	244	7,95%	NS	3320	7,95%	NS	3600
B9	69306	8,51%	350	6,69%	NS	3600	6,69%	NS	3600
B10	29948	3,45%	14,7	0,66%	0,27%	77	0,07%	0,00%	3600

Indicateurs pour analyser une convergence PLNE : qualité des bornes primales et duales à la relaxation continue, au noeud 1 B&B et au bout d'un temps défini. (gaps pris sur meilleure solution connue)

Autres indicateurs : nombre de variables et contraintes,

Indicateur pour convergence à optimalité : nb de noeuds nécessaires à convergence PLNE, temps de calcul

# Matheuristiques

- ▶ Heuristiques basées des résolutions PL/PLNE.
- ▶ Codage avec OPL script ou itérations à des appels OPL ou Cplex depuis le langage de votre choix.
- ▶ Les heuristiques peuvent être codées avec OPL.
- ▶ Heuristique constructive simple : lancer résolution PLNE en ayant figé des variables heuristiquement.
- ▶ ATTENTION : relaxation LP ne donne aucune information pour maxStable et max Clique!!!
- ▶ Heuristique constructive générale : Algo glouton avec optimisations successives par PLNE.
- ▶ Une heuristique hill-climbing : itérer des résolution PLNE, peut être codé en OPL script ou depuis un langage de programmation. Diversification par ILS ou multi-start.
- ▶ Bien définir les opérateurs/voisinages/opérateurs, travail d'analyse de l'impact d'opérateurs.



# (Méta-)heuristiques

- ▶ Codage dans le langage de programmation choisi.
- ▶ Parseur depuis un fichier externe, pour reproductibilité des résultats, au format de données d'OPL.
- ▶ Heuristique(s) constructive(s) : design, comparaison empirique
- ▶ Heuristique(s) perturbative : définir structure voisinages, coder hill climbing et au moins une technique apportant de la diversification
- ▶ Travail d'analyse : comparaison de résultats empiriques avec des variantes d'opérateurs/voisinages (qualité de solutions, peuvent être comparées avec résolution B&B).
- ▶ Approfondissement : coder un algo génétique et/ou tester plusieurs opérateurs de diversification

# (Méta-)heuristiques : problématique clustering

- ▶ Coder les heuristiques de manière générique pour p-median et k-medoids. Adapter ou code générique avec MinSum-k-radii également.
- ▶ Pour p-median et k-medoids : les algos classiques phase constructive et perturbative type Hill Climbing sont donnés dans le cours.
- ▶ Important : techniques de diversification ...
- ▶ Analyse : y a t'il bcp de minimums locaux ?

# (Méta-)heuristiques : problématique graphe

- ▶ Max Stable et max clique sont le même problème par passage au graphe complémentaire : en implémenter un seul et l'opération passage complémentaire pour résoudre les deux problèmes.
- ▶ Adaptation à la coloration : optionnel. L'heuristique max Clique peut être utilisé dans phase constructive.
- ▶ Analyse spécifique : Les heuristiques permettent d'elles d'accélérer significativement les résolutions exactes comme indiquées dans les cours 1 et 2 ? (problématique de symétries pour coloration maximale)

# Optionnel : génération de colonnes

- ▶ Une heuristique de génération de colonne peut être codée pour ceux qui souhaiterait cet approfondissement.
- ▶ Graphes : pb de coloration
- ▶ Clustering : problème du p-median.
- ▶ Un exemple OPL donne un canevas exemple à suivre : cutting-stock.
- ▶ Peut être codé en OPL script ou depuis un langage de programmation.
- ▶ Optionnel : utilisation de résolution heuristique de sous problèmes

# Elements d'organisation

- ▶ Travail en binôme : permet de se répartir le travail.
- ▶ 3 séances de TP consacrées au travail en projet : profitez en pour poser des questions.
- ▶ Rendus pour le 15 janvier : rapport, code, instructions pour lancer le code sous Linux, instances générées.
- ▶ Rapport : travail d'analyse empirique, description précise des algorithmes/opérateurs implémentés, tableaux de résultats.
- ▶ Annexe du rapport : structure du code.
- ▶ Vous pouvez aussi joindre un fichier tableur avec des résultats exhaustifs (et présentation dans le rapport de résultats agrégés sur des classes d'instances)

# DES QUESTIONS ?