Chapitre 9
Conclusion

Claude-Guy Quimper

Révision

- Nous avons commencé ce cours par un chapitre d'introduction.
- Il est donc naturel de le terminer par un chapitre de conclusion.
- Ce chapitre se veut donc être une révision de tous les chapitres du cours et permet ainsi d'avoir une vue d'ensemble sur ce que nous avons étudié.

Chapitre 1

- Dans le premier chapitre, nous avons vu différentes applications de l'optimisation combinatoire.
- Nous en avons vu d'autres dans les travaux pratiques et les exercices.
- Vous avez aussi pu explorer des applications dans le cadre de votre projet.

Chapitre 2

- Nous avons vu comment procéder à une fouille dans un arbre de recherche.
- Nous avons vu comment le filtrage des contraintes et l'apprentissage de clauses permettent de réduire la taille de cet arbre de recherche.
- Nous avons vu différents niveaux de cohérence
  - Cohérence de domaine
  - Cohérence d'intervalle
  - Cohérence de bornes
- Nous avons aussi vu différents niveaux de cohérence locale
  - Cohérence de singleton
  - Cohérence de chemin

Chapitre 3

- Ce chapitre nous a appris les différentes composantes d'un solveur.
- Nous avons vu les structures de données pour encoder le domaine des variables.
- Vous avez eu l'occasion de vous familiariser avec un solveur de contraintes.

Chapitre 4

- Nous avons étudié l'arithmétique des intervalles utilisée pour le filtrage des contraintes arithmétiques.
- Nous avons vu que la répétition des variables dans une contrainte peut mener à un filtrage plus faible.
- Nous avons aussi vu comment exploiter la monotonie des contraintes afin d'augmenter le filtrage.

Chapitre 5

- Nous avons vu comment les structures traitables permettent de modéliser un problème de sorte à augmenter le niveau de filtrage.
- Les structures d'arbre et d'hyperarbre nous permettent, entre autres, d'encoder simplement et efficacement les contraintes Regular et Lexicographique.
- Lorsque le graphe de contrainte a une largeur d'arbre bornée par une constante, il est possible de résoudre ce problème en temps polynomial.

Chapitre 6

- Ce chapitre nous a appris différentes heuristiques utilisées dans les solveurs de contraintes.
- Nous avons vu qu'il existe des fouilles telles que LDS qui explorent simultanément différentes sections de l'arbre de recherche.
- Nous avons vu comment les redémarrages permettent de diversifier l'exploration de l'arbre de recherche.

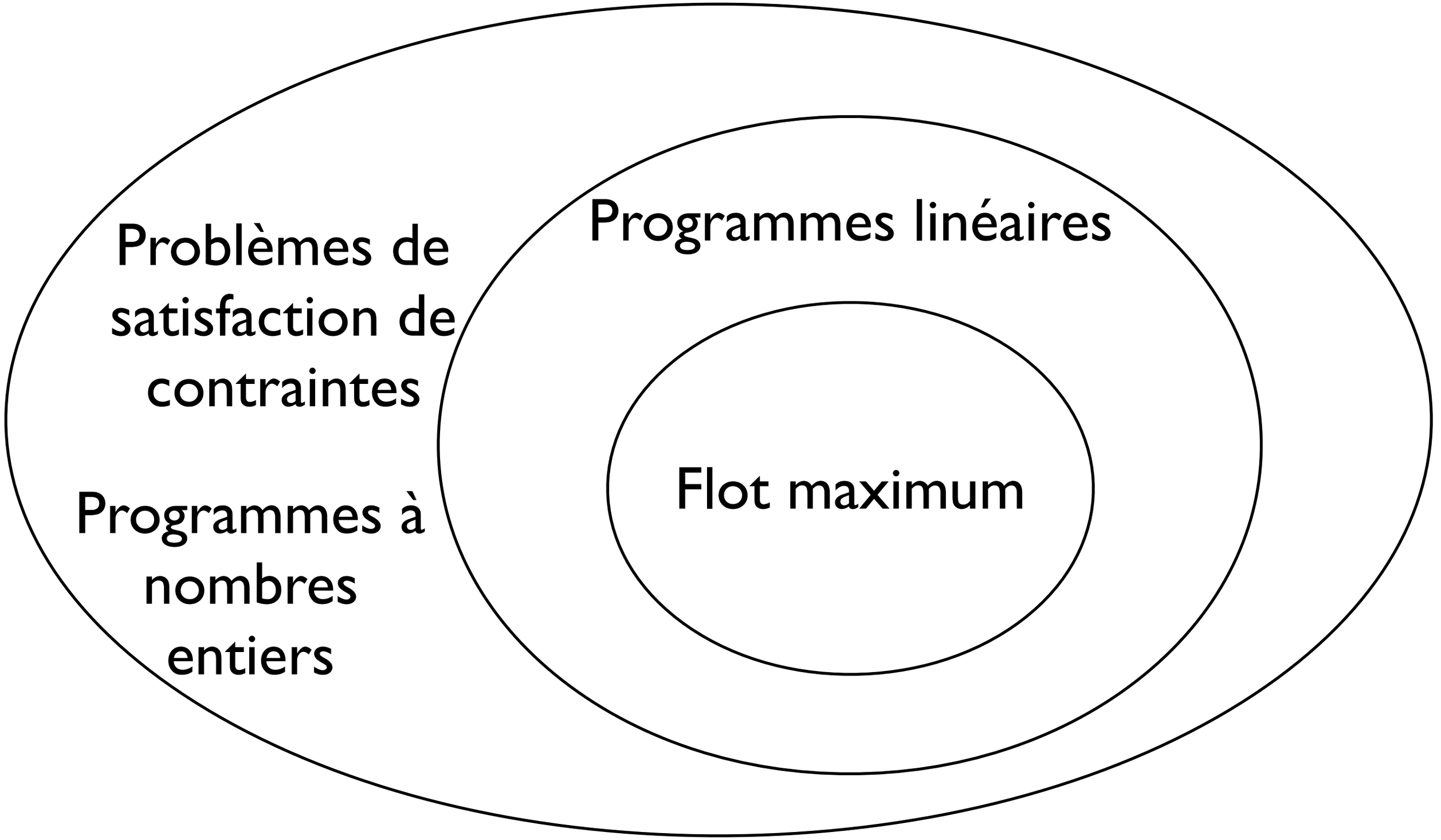
Chapitre 7

- Certains problèmes peuvent être modélisés sous la forme d'un flot maximum dans un graphe.
- Si tel est le cas, l'algorithme de Ford-Fulkerson permet de résoudre efficacement ces problèmes.
- Nous avons vu que trouver le flot maximum d'un graphe est équivalent à trouver la coupe minimum de ce graphe.
- L'algorithme de Régin qui filtre la contrainte All-Different est basé sur les algorithmes de flot et les couplages.

Chapitre 8

- Un programme linéaire est un problème d'optimisation sous contraintes d'inégalités linéaires.
- Nous avons vu comment la méthode du simplexe peut résoudre efficacement ces problèmes.
- Le théorème de dualité nous en apprend sur la nature des problèmes combinatoires en démontrant que pour chaque programme linéaire de maximisation, il existe un programme linéaire dual de minimisation. Chaque contrainte devient alors une variable et chaque variable devient une contrainte.
- Finalement, nous avons vu les bases de la programmation en nombres entiers.

Relations entre les problèmes



Outils de résolution