Un exemple de résolution de problème

Complexité et Calculabilité

Pascal Berthomé

Plan

- Problème concret
- Modélisation
- Résolution
  - NP-Complétude
  - Approximation

Problème

- Réseau d'interconnexion
  - Routeurs
  - Utilisateurs finaux
  - Liens entre les routeurs
- Communauté voulant un réseau dédié
  - Couvrant tous les utilisateurs
  - Pour un coût minimum

Modélisation

- Le réseau → un graphe G
  - Les routeurs, utilisateurs → nœuds : V
  - Les liens → les arêtes du graphe : E
    - (arcs si on suppose que les liens sont orientés)
  - Le coût de location d'un lien → fonction sur les arêtes : c : E → IR
  - La communauté → un sous-ensemble des nœuds : V'

Reformulation du problème

- On cherche un sous-ensemble d'arêtes qui permet de relier tous les éléments de la communauté pour un coût minimum
- Propriété : c'est un arbre!
  - Preuve:

Formulation mathématique

- L'ensemble des arêtes $E = (e_1, e_2, \ldots, e_m)$
- On définit l'ensemble des variables booléennes (0 ou 1) x_i telles que
  - $x_i = 1$ ssi l'arête e_i est choisie dans la solution
- La fonction de coût par arête $c(e_i) = c_i$
- Problème
  - Minimiser la somme des produits x_i ci
  - De sorte que l'ensemble des arêtes soit connexe et couvre V'

Cas particuliers

- V' contient deux éléments
  - Problème du plus court chemin dans les graphes
    - Algorithme de Dijkstra
    - → polynomial
- V' = V
  - Problème de l'arbre couvrant de poids minimum
    - Algorithme de Prim ou de Kruskal
    - → polynomial
- V' contient trois éléments
- Tous les autres cas sont « difficiles »

Problème de l'arbre de Steiner

- Problème précédent dans le cas général
- Le problème de décision associé est NP-Complet [ND12, GJ79]!
  - Même dans le cas de graphes simples (grilles, graphes bipartis, …)
  - Problème [ND13]: Arbre de Steiner Géométrique
    - Points sur Z^2, distance euclidienne.

Problème d'optimisation

- Données
  - Des données « simples »
  - Une fonction de coût f qui mesure la qualité d'une solution
- Problème
  - Trouver la valeur minimale (ou maximale) de la fonction de coût correspondant au problème initial

Problème de décision associé

- Données
  - Les mêmes données « simples »
  - Un entier k
- Question
  - Existe-t-il une solution du problème telle que la fonction de coût est inférieure à k ?

Que faire ?

- Rechercher des solutions exactes ?
  - Algorithmes exponentiels
  - Dreyfus et Wagner, 1972
    - $O(n\ 3^k + n^2\ 2^k + n^3) = O^*(3^k)$
    - Programmation dynamique
  - Wang, 2008
    - $O^*(2.684^k)$
    - Utilisation astucieuse de l'algorithme précédent

Que faire ?

- Algorithmes approchés ?
  - Trouver un algorithme qui donne une solution « acceptable »
  - Si on sait évaluer la qualité → algorithme d'approximation
  - Sinon → heuristique
- Moralité : une heuristique donne souvent de très bons résultats, mais ne possède pas de garantie de résultats.

Approximations pour Steiner

- Il existe une approximation à un facteur 2 (1968)
  1. Calculer un nouveau graphe G' avec V' comme ensemble de sommets
  2. Le poids de l'arête (i,j) = distance (coût) entre les sommets i et j dans G
  3. Calculer un arbre de poids minimum dans G'
  4. Si l'arête (i,j) est sélectionnée dans G', prendre le chemin allant de i à j dans G
  5. Éliminer les arêtes en trop dans G

Autres algorithmes

- 1968 : Gilbert Pollack 2
- 1993 : Zelikovski 11/6
- 1994 : Berman Ramaiyer 1.75
- 1997 : Zelikovski Karpinski 1.65
- 1999 : Hougardy Prömel 1.60
- 2000 : Zelikovski Robins 1 + ln3/2 = 1.55

- 1999 : Clementi Trevisan approx > 1.0006

Rapport d'approximation

- Garantie sur la qualité de la solution par rapport à la solution optimale
  - ρ = Max (SolApproch/ValOpt)
    - (pour une minimisation)
- Plus ρ est proche de 1, meilleure est l'approximation

Cas particuliers

- Il existe des problèmes pour lesquels il existe une famille d'algorithmes polynomiaux (fonction de ρ) qui produisent une approximation de la solution
  - → schéma d'approximation
  - La complexité de l'algorithme dépend de ρ
  - Problème du sac à dos

Cas particuliers

- Problèmes non approximables à une constante près
- Toutes les classes ainsi définies sont strictement incluses les une dans les autres à moins que P=NP

Références

- Garey, Johnson: Computers and Intractability, A Guide to the Theory of NP-Completeness, 1979
- Crescenzi, Kann, A compendium of NP optimization problems. http://www.csc.kth.se/~viggo/problemlist/compendium.html
- Cours d'algorithmique 3A/ Graphes 3A