M2 MPRO

Bases de l'Optimisation dans les Graphes : couplages, flots et liens avec la programmation linéaire

Quelques notions, techniques de preuves et théorèmes de base en optimisation dans les graphes (concernant les couplages et les flots)

Théorie des graphes, optimisation dans les graphes, et paramètres de graphes

- Graphes = objets discrets :
  - Utiles pour représenter des structures discrètes rencontrées en pratique (réseaux télécoms, réseaux sociaux, réseaux de transport).
  - Nombreuses notions pertinentes associées : chemins, couplages, stables, flots, colorations, etc.
  - Théorie des graphes = étude des propriétés des graphes et des notions associées.
  - A chaque notion, un problème d'optimisation associé : valeur min ou max = valeur d'un *paramètre* du graphe considéré.
  - Ainsi, résoudre de nombreux problèmes réels = calculer des paramètres de graphes : itinéraires ou ordonnancement de projet sans ressources (chemins), emplois du temps (colorations), résistance aux pannes (k-connexité), affectation de tâches (couplages), débit (flots), plans de transport (flots avec coûts), etc.

Programme abordé (2 prochaines séances)

- Notions et théorèmes emblématiques :
  - Couplages : lemme de Berge, théorème de König-Egerváry dans les graphes bipartis (preuve par l'absurde), théorème de Hall et conséquences.
  - Flots : théorème de Ford-Fulkerson (flot max/coupe min), via un algorithme de marquage (2 versions) et avec König(-Egerváry) comme corollaire, et extensions algorithmiques aux flots à coût minimum (Busacker-Gowen).
  - Chemins disjoints par les arêtes/sommets et k-connexité : théorème de Menger, par réduction à Ford-Fulkerson.

ACTE I :

Couplages, lemme de Berge, et théorèmes de König(-Egerváry) et de Hall

Couplages dans les graphes

- Couplage = ensemble d'arêtes sans extrémités communes
  - Paramètre : taille d'un couplage maximum (au plus la moitié du nombre de sommets).
  - Si M couplage : arête uv dans M ==> arête vw pas dans M, pour tout w dans N(v)\{u} (N(v) = *voisinage* de v).
  - Couplage *parfait* : tout sommet est *saturé*, c'est-à-dire incident à une arête du couplage (taille = moitié du nombre de sommets).
- Notion essentielle : *chaîne alternée* par rapport à un couplage M
  - Chaîne qui débute par une arête uv hors de M, et dont toutes les arêtes sont alternativement dans et hors de M.
  - Si extrémité initiale u non saturée par M, et si nombre pair d'arêtes OU si extrémité terminale non saturée par M, alors inverser les arêtes (arête hors de M ==> dans M, et arête dans M ==> hors de M) sur une telle chaîne produit un autre couplage.

Chaînes augmentantes et lemme de Berge

- Chaîne *augmentante* par rapport à un couplage M
  - Chaîne alternée dont aucune des 2 extrémités n'est saturée par M.
  - Observation : inverser les arêtes sur une telle chaîne produit un autre couplage, plus grand (de 1).
- *Lemme de Berge* : couplage M maximum <==> il n'existe pas de chaîne augmentante par rapport à M.
  - Sens ==> : cf observation ci-dessus.
  - Sens <== : supposons M non maximum, et soit M* un couplage maximum (on a |M*|>|M|). Dans la différence symétrique de M et M*, il y a une chaîne augmentante par rapport à M (*pourquoi ?*).

Graphe biparti

- Graphe dont les sommets peuvent être partitionnés en 2 parties :
  - Ensembles L (pour « L*eft* ») et R (pour « R*ight* ») qui partitionnent l'ensemble des sommets.
  - Pour toute arête uv, on a u dans L et v dans R (ou l'inverse).
- *Remarque :* un arbre est un cas particulier de graphe biparti !
- *Caractérisation :* graphe non orienté biparti <==> il ne contient aucun cycle de longueur impaire.
  - Cycle de longueur impaire ==> un sommet sur 2 dans L, et un sommet sur 2 dans R, donc une arête dans L (ou dans R).
  - Aucun tel cycle ==> à l'issue d'un parcours en largeur d'abord à partir d'un sommet arbitraire r, les sommets à distance paire et impaire de r définissent L et R, respectivement (*pourquoi ?*).

Transversaux et théorème de König(-Egerváry) dans les graphes bipartis

- Transversal (« *vertex cover* » en anglais)
  - Ensemble de sommets couvrant toutes les arêtes.
  - On a : taille couplage maximum ≤ taille transversal minimum.
  - Pas d'égalité en général (cycle de taille 3).
- *Théorème de König* : dans tout graphe biparti G = ((L,R), E), taille couplage maximum = taille transversal minimum.
  - *Observation* : théorème de König « facile » si degré maximum 2.
  - Tout contre-exemple au théorème de König possède donc au moins un sommet de degré au moins 3.
  - Et ensuite ?

Théorème de König(-Egerváry) dans les graphes bipartis : preuve par l'absurde

- Preuve par l'absurde du théorème de König :
  - Soit un contre-exemple *minimal* G, et soit u un sommet de degré > 2.
  - Soit v un voisin de u, et soit M un couplage maximum tel que v n'est pas saturé par M. (*Pourquoi M existe-t-il ?*)
  - Soit f une arête incidente à u mais pas à v, et qui n'est pas dans M. (*Pourquoi f existe-t-elle ?)*
  - Alors, tout transversal minimum pour G\{f} est un transversal pour G (*pourquoi ?*), de taille |M| (*pourquoi ?*) ==> contradiction.

Théorème de Hall et conséquences

- *Théorème de Hall* : dans un graphe biparti G=((L,R),E), il existe un couplage saturant tous les sommets de L <==> pour tout sous-ensemble L' de L, $|L'| \leq |N(L')|$.
  - Sens ==> : facile (*pourquoi ?*).
  - Sens <== : pas de couplage saturant L ==> il existe un transversal T de taille $|T \cap L| + |T \cap R| < |L|$ (*pourquoi ?*). On peut alors en déduire que $|L \backslash (T \cap L)| > |N(L \backslash (T \cap L))|$ (*pourquoi ?*).
- Conséquences :
  - CNS pour l'existence d'un couplage parfait quand $|L| = |R|$.
  - *Tout graphe biparti k-régulier admet k couplages parfaits disjoints*.
    - Car $k|L|=|E|=k|R|$, et $k|L'| \leq k|N(L')|$ pour tout sous-ensemble L' de L.

ACTE II :

Flots, coupes, théorème de Ford-Fulkerson et algorithme associé (2 versions)

Flots dans les réseaux de transport

- *Réseau de transport :* graphe (orienté) G=(V,A) avec une source s, un puits p, et une capacité (entière) capa(a) sur chaque arc a de A.
- *Flot :* fonction f : A $\rightarrow$ IR^+, qui vérifie :
  - f(a) $\leq$ capa(a) pour chaque arc a (*contraintes de capacité*),
  - Conservation des valeurs de f en chaque nœud, sauf s et p (*contrainte de conservation du flot*).
- Intuitivement : f(a) = nombre unités de flot circulant sur l'arc a, toute unité de flot étant émise par la source et reçue par le puits.
- Valeur d'un flot f = nombre d'unités de flot émises par la source.

Coupes et flots

- *Coupe dans un réseau de transport :*
  - Coupe associée à un sous-ensemble V' de V (contenant s mais pas p) = ensemble des arcs de V' vers V\V'.
  - Capacité d'une telle coupe = somme des capacités des arcs concernés.
- Toute unité de flot doit traverser les arcs d'une coupe arbitraire :

  ==> Valeur de tout flot f ≤ capacité de toute coupe
- Vrai en particulier pour un flot maximum et une coupe minimum.
- Comment calculer (efficacement) un flot max. ? Une coupe min. ?
- Peut-il y avoir un écart (important) entre ces deux quantités ?

Algorithme de Ford-Fulkerson (marquage)

- Calcul d'un flot maximum et d'une coupe minimum
- Basé sur des règles de marquage des sommets
  - Choisir un flot initial f (nul ou non), puis répéter jusqu'à STOP : marquer la source s, et ensuite
  - Tant que c'est possible, marquer un sommet via une de ces 2 règles
    - (i) Arc (u,v) avec u marqué et f(u,v) < capa(u,v) ==> marquer v avec « + »
    - (ii) Arc (u,v) avec v marqué et f(u,v) > 0 ==> marquer u avec « - »
  - Si puits p non marqué, STOP ; sinon, on a une *chaîne améliorante*
- Chaîne améliorante = suite de sommets adjacents marqués de s à p
  - Quantité minimum qmin sur la chaîne = min sur arcs (u,v) de la chaîne (capa(u,v)-f(u,v) si (i), f(u,v) si (ii))
  - Mise à jour du flot f(u,v) := ? : f(u,v)+qmin si (i) ; f(u,v)-qmin si (ii)

Algorithme de Ford-Fulkerson (graphes d'écart)

- Chaque itération améliore le flot de qmin > 0 unités
- Coupe obtenue à la fin : entre V' = {sommets marqués} et V\V'
- Description alternative de cet algorithme, via les *graphes d'écart*
  - Graphe d'écart G(f) associé à un réseau de transport G et un flot f :
    - Mêmes sommets que G,
    - Arc (u,v) dans G ==> arc (u,v) étiqueté *capa(u,v)-f(u,v)* dans G(f) si f(u,v) < capa(u,v) ET/OU arc (v,u) étiqueté *f(u,v)* dans G(f) si f(u,v) > 0.
  - Marquages remplacés par la recherche d'un chemin de s à p dans G(f)
  - Mise à jour identique des flots, et coupe obtenue à la fin = celle entre les sommets accessibles à partir de s et les autres

Preuve d'optimalité de l'algorithme, et théorème de Ford-Fulkerson

- En fait, flot et coupe calculés optimaux *car de même valeur*

==> *Théorème de Ford-Fulkerson* : valeur flot maximum (à valeurs entières) = capacité coupe minimum.

- *Remarque* : en fait, ce théorème implique celui de König (à suivre)
- Preuve du théorème de Ford-Fulkerson ?
  - Valeur de tout flot f = somme f(u,v) pour les arcs (u,v) avec u dans V' et v dans V\V' – somme f(u,v) pour les arcs (u,v) avec u dans V\V' et v dans V', pour tout V' avec une coupe associée (*pourquoi ?*)
  - Appliquer cette égalité au flot f obtenu à la fin de l'algorithme et à la coupe associée aux sommets marqués ==> CQFD (*pourquoi ?*)

Problème du flot à coût minimum (avec coûts unitaires), et algorithme de Busacker-Gowen

- On peut adapter la version de l'algorithme de Ford-Fulkerson utilisant les graphes d'écart pour résoudre le problème du flot maximum à coût minimum (si les coûts des arcs sont unitaires) :
  - Ajout d'un coût unitaire $c(u,v) \geq 0$ sur chaque arc du graphe G :

    (i) Si arc (u,v) dans G non saturé ➔ arc (u,v) dans G(f) de coût c(u,v),

    (ii) Si arc (u,v) dans G avec $f(u,v) > 0$ ➔ arc (v,u) dans G(f) de coût –c(u,v).
  - **Un flot f de valeur F donnée est de coût minimum parmi tous les flots de valeur F <==> pas de circuit de coût < 0 dans G(f).**
  - Flot initial nul (car flot à coût minimum parmi tous les flots nuls !)
  - A chaque étape, au lieu de chercher un chemin de s à p dans G(f), on calcule **un plus court chemin de s à p** (via Bellman-Ford, par ex.)
- Si flot à coût minimum de valeur F fixée, on adapte le graphe :
  - On ajoute un unique arc de capacité [F] en sortie de s.

ACTE III :

Chemins disjoints par les arêtes/sommets, k-connexité, et théorème de Menger

Chemins disjoints et ensembles déconnectant

- *Chemins disjoints* entre 2 sommets x et y d'un graphe non orienté :
  - Disjoints par les sommets ==> seuls sommets communs = x et y,
  - Disjoints par les arêtes ==> aucune arête en commun.
- *Ensemble déconnectant* entre x et y :
  - Ensemble de sommets (différents de x et y) ou d'arêtes dont la suppression ne laisse aucun chemin entre x et y.
- Clairement, s'il existe k chemins disjoints entre 2 sommets, alors la taille d'un ensemble déconnectant entre x et y est au moins k.

k-connexité et théorème de Menger

- *k-connexité* (résistance aux pannes) :
  - Graphe k-arête-connexe ==> le rendre non connexe nécessite de supprimer au moins k arêtes,
  - Graphe k-sommet-connexe ==> le rendre non connexe nécessite de supprimer au moins k sommets,
  - Graphe 1-connexe mais pas 2-connexe ==> présence d'un *isthme* (version arêtes) ou d'un *sommet d'articulation* (version sommets).
- *Théorème de Menger* (forme générale) : un graphe est k-connexe <==> il existe k chemins disjoints entre toute paire de sommets.
  - Sens <== : facile (*pourquoi ?*).
  - Sens ==> : ? (s'il n'existe pas k chemins disjoints entre x et y, alors supprimer moins de k arêtes/sommets suffit à les déconnecter).

Preuve du théorème de Menger (sens ==>)

- Preuve de Menger utilisant le théorème de Ford-Fulkerson :
  - Orienter le graphe, en remplaçant chaque arête par un ensemble approprié de 5 arcs.
  - Si chemins disjoints par les sommets, remplacer chaque sommet v par 2 nouveaux sommets (l'un incident aux arcs entrant en v, l'autre incident aux arcs sortant de v), reliés par un arc.
  - Fixer la capacité de tous les arcs à 1.
  - Alors : nombre maximum de chemins disjoints (par les sommets ou les arêtes) de x à y = valeur d'un flot maximum de x à y, et taille minimum d'un ensemble déconnectant = capacité d'une coupe min.

La suite ?

- Preuves de *théorèmes* classiques sur des *paramètres* essentiels :
  - Basées sur des *notions purement combinatoires* (chaînes augmentantes/améliorantes, dénombrement, réduction entre problèmes, contre-exemple minimal...).

  ==> Lien entre l'optimisation dans les graphes et la programmation math. ? *Preuves alternatives basées sur la programmation linéaire*.