Parcours eulériens et hamiltoniens

Un graphe est **connexe** si pour toute paire de sommets x,y il existe une chaîne entre x et y.
Un graphe est **fortement connexe** si pour toute paire de sommets x,y il existe un chemin de x vers y.
Un graphe est **quasi fortement connexe** s'il contient une **racine**, c'est-à-dire un sommet r tel qu'il existe un chemin de r vers tout autre sommet du graphe.
Une **arborescence** est un arbre orienté comportant une (et une seule) racine.
Lorsqu'on inverse l'orientation de chaque arc d'une arborescence, on obtient une **anti-arborescence** et la racine de l'arborescence est transformée en une **anti-racine**.
Un graphe est **eulérien** s'il contient un cycle eulérien.

Pour un graphe orienté G=(V,A), pour un sous-ensemble W de sommets, et pour un sommet x, notons

- $\omega^+(W)$ l'ensemble des arcs ayant leur extrémité initiale dans W et l'autre hors de W.
- $\omega^-(W)$ l'ensemble des arcs ayant leur extrémité finale dans W et l'autre hors de W.
- $d_G^+(x)$ le nombre d'arcs dans $\omega^+(\{x\})$ (et aussi le nombre de sommets dans $N^+(x)$).
- $d_G^-(x)$ le nombre d'arcs dans $\omega^-(\{x\})$ (et aussi le nombre de sommets dans $N^-(x)$).

Le degré d'un sommet x est donc égal à $d_G^+(x)+d_G^-(x)$, et les arcs de $\omega^+(W)\cup\omega^-(W)$ constituent le cocycle [W,V-W] lorsqu'on oublie leur orientation.

Propriétés

- G=(V,E) est connexe $\Leftrightarrow$ $[W,V-W]\neq\emptyset \quad \forall W$ tel que $\emptyset\subset W\subset V$.
- G=(V,A) est fortement connexe $\Leftrightarrow$ $\omega^+(W)\neq\emptyset \quad \forall W$ tel que $\emptyset\subset W\subset V$.
- G=(V,A) est fortement connexe $\Rightarrow$ chaque arc de A fait partie d'un circuit
- G est fortement connexe $\Rightarrow$ G est quasi-fortement connexe
- G est quasi-fortement connexe $\Rightarrow$ G est connexe
- G est quasi-fortement connexe $\Leftrightarrow$ G contient une arborescence comme sous-graphe partiel
- G est fortement connexe $\Leftarrow$ G connexe et $d_G^+(x)=d_G^-(x)$ pour tout x dans G
- G contient un circuit eulérien $\Leftrightarrow$ G connexe et $d_G^+(x)=d_G^-(x)$ pour tout x dans G
- G contient un cycle eulérien $\Leftrightarrow$ G connexe et chaque sommet de G est de degré pair.
- Si G contient p sommets de degré impair alors il existe une partition de ses arêtes en p/2 chaînes

CONSTRUCTION D'UNE ARBORESCENCE DANS UN GRAPHE QUASI FORTEMENT CONNEXE

Soit G=(V,A) un graphe quasi fortement connexe et r une racine dans G. L'algorithme ci-dessous produit une arborescence (V,F) de racine r.

(1) Poser W :={r} et F :=$\emptyset$;
(2) Tant que W$\neq$V faire
    Choisir un arc (x,y) dans $\omega^+(W)$ et poser W :=W$\cup${y} et F :=F$\cup${(x,y)}

CONSTRUCTION D'UN CIRCUIT EULÉRIEN DANS UN GRAPHE CONNEXE OÙ $d_G^+(x)=d_G^-(x)$ POUR TOUT SOMMET x.

(1) Choisir un sommet quelconque r dans le graphe et construire une anti-arborescence à l'aide d'un algorithme similaire à celui ci-dessus.
(2) Pour chaque sommet x faire
    Numéroter de 1 à $d_G^+(x)$ les arcs sortant de x, en mettant le plus grand numéro sur l'arc sortant appartenant à l'anti-arborescence (pour l'anti-racine, la numérotation est quelconque)
(3) Partir de r, et tant qu'il existe un arc non encore utilisé permettant de quitter le sommet où on se trouve, choisir celui de plus petit numéro et le parcourir.

CONSTRUCTION D'UN CYCLE EULÉRIEN DANS UN GRAPHE OÙ CHAQUE SOMMET EST DE DEGRÉ PAIR

(1) Orienter les arêtes de telle sorte que $d_G^+(x)=d_G^-(x)$ pour tout x dans G
(2) Construire un circuit eulérien et 'oublier' l'orientation

Problème du postier chinois

Soit G=(V,E). Déterminer un cycle de coût minimum passant au moins une fois par chaque arête de E

Résolution

- Si G est eulérien il suffit de construire un cycle eulérien.
- Si G n'est pas eulérien, il faut déterminer un couplage M de coût minimum sur les sommets de degré impair (le coût d'un couple (x,y) étant égal à la longueur d'une plus courte chaîne entre x et y dans G).
  Le graphe G'=(V,E∪M) est alors eulérien et il suffit de construire un cycle eulérien.

Remarque : La solution fournie par la méthode décrite ci-dessus est toujours optimale.
La détermination d'un couplage de coût minimum sera vue dans un chapitre ultérieur.

Problème du postier rural

Soit G=(V,E), et soit R un sous-ensemble d'arêtes de G.
Déterminer un cycle de coût minimum passant au moins une fois par chaque arête de R.

Notation

Soit V_R l'ensemble des sommets incidents à une arête de R
On notera G_R le graphe contenant les sommets de V_R et les arêtes de R (c'est-à-dire $G_R=(V_R,R)$)

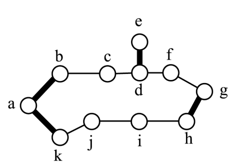
Définition

On dit que les distances satisfont l'inégalité triangulaire si $d_{xy} \le d_{xz} + d_{zy}$ pour tout x,y,z dans G.

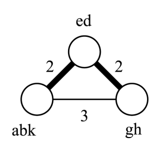
Résolution

- Si R=E il s'agit du problème du postier chinois
- Si R≠E alors on peut utiliser l'algorithme suivant proposé par Frederickson

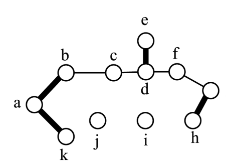
(1) Construire un graphe H ayant un sommet par composante connexe de G_R et tel que chaque paire de sommets x,y de H est reliée par une arête de coût égal à la longueur de la plus courte chaîne entre x et y dans G.
(2) Déterminer un arbre de coût minimum dans H. Soit A le sous-ensemble d'arêtes de G correspondant à cet arbre de coût minimum dans H.
(3) Déterminer un couplage de coût minimum sur les sommets de degré impair de G'=(V,R∪A), le coût d'un couple (x,y) étant égal à la longueur d'une plus courte chaîne entre x et y dans le graphe original G). Soit M le sous-ensemble d'arêtes de G correspondant à ce couplage.
(4) Déterminer un cycle eulérien dans G''=(V,R∪A∪M)



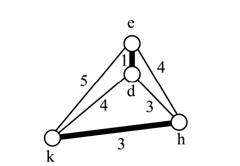
Graphe original avec R en gras.
Toutes les arêtes sont de longueur 1



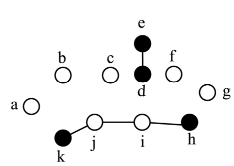
Graphe H avec un arbre de coût min en gras



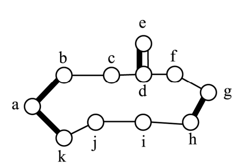
graphe G'=(V,R∪A)



graphe dans lequel on recherche un couplage de coût minimum, et couplage de coût minimum en gras



arêtes de M correspondant au couplage de coût minimum



graphe eulérien G''=(V,R∪A∪M)

Remarques

- Si G_R est connexe alors la solution fournie par l'algorithme de Frederickson est toujours optimale (et les étapes (1) et (2) de l'algorithme ci-dessus sont inutiles puisque H ne contient qu'un sommet)
- Si G_R n'est pas connexe et si les distances dans G satisfont l'inégalité triangulaire, alors la valeur de la solution fournie par l'algorithme de Frederickson est au pire égale à 3/2 fois la valeur de la solution optimale.
- Si G_R n'est pas connexe et si les distances ne satisfont pas l'inégalité triangulaire, alors on ne peut pas borner l'écart entre la valeur de la solution fournie par l'algorithme de Frederickson et celle de la solution optimale.

Problème du voyageur de commerce

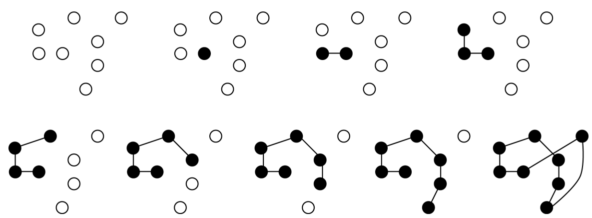
Soit G=(V,E) un graphe complet (c'est-à-dire tel qu'il existe une arête entre toute paire de sommets). Déterminer un cycle hamiltonien de coût minimum.

Remarque : Considérons un graphe H dans lequel certains sommets correspondent aux clients qu'un voyageur de commerce doit visiter. En général, ce graphe n'est pas complet, et le voyageur a le droit de ne pas passer par les sommets ne correspondant pas à des clients, et il a le droit de passer plusieurs fois par un même sommet si cela l'arrange (le cycle recherché n'est donc pas hamiltonien). On transforme cependant ce problème en construisant un nouveau graphe G contenant comme sommets uniquement ceux de H qui correspondent à des clients, et en reliant chaque paire de clients par une arête de coût égal à la longueur de la plus courte chaîne entre ces deux clients dans H. Déterminer l'ordre dans lequel les clients doivent être visités dans H pour minimiser la distance totale parcourue est alors équivalent à déterminer un cycle hamiltonien de coût minimum dans le graphe complet G.

Algorithme du plus proche voisin

(1) Choisir un sommet de départ x
(2) Tant que tous les sommets ne sont pas encore visités faire
    Se rendre au sommet le plus proche pas encore visité

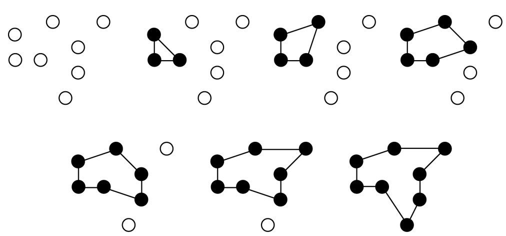
Exemple



Algorithme du plus petit détour

(1) Choisir trois sommets x, y et z et créer une tournée (un triangle) avec ces trois clients
(2) Tant que tous les sommets ne sont pas encore visités faire
    Déterminer le client non encore visité dont l'insertion dans la tournée courante provoque le plus petit détour et effectuer cette insertion

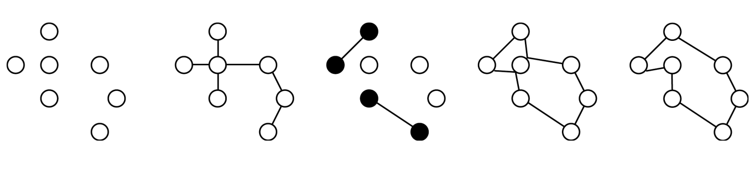
Exemple



Algorithme de Christofides

(1) Déterminer un arbre de coût minimum dans G. Soit A l'ensemble des arêtes de cet arbre
(2) Déterminer un couplage M de coût minimum sur les sommets de degré impair dans G'=(V,A). Soit M les arêtes de ce couplage.
(3) Déterminer un cycle eulérien dans G''=(V,A$\cup$M)
(4) Raccourcir la tournée si elle passe plusieurs fois par un même sommet.

Exemple



Remarques

- Les algorithmes du plus proche voisin et du plus petit détour ne donnent aucune garantie de performance, en ce sens que l'écart entre la valeur déterminée par ces méthodes et la valeur optimale peut être arbitrairement grand.
- Si les distances satisfont l'inégalité triangulaire, la valeur de la solution fournie par l'algorithme de Christofides est au pire égale à 3/2 fois la valeur de la solution optimale.

  Preuve

  Notons c(F) le coût total d'un ensemble F d'arêtes.

  Soit T un cycle hamiltonien de coût minimum. La valeur d'une solution optimale est donc c(T), et on a $c(A) \leq c(T)$ car T est un arbre+une arête et A est un arbre de coût minimum

  Soit W l'ensemble des sommets de degré impair dans G'=(V,A). Soit T_W la tournée obtenue à partir de T en ôtant les clients qui ne sont pas dans W (ce qui permet de raccourcir T puisqu'on a les inégalités triangulaires). On a donc $c(T_W)$)$\leq c(T)$.

  T_W est une tournée comportant un nombre pair d'arêtes qu'on peut colorer alternativement en bleu et vert. Les arêtes bleues forment un couplage M_1 des sommets de W, et les vertes forment un couplage M_2 de ces sommets.

  Comme $c(T_W)=c(M_1)+c(M_2)$, on a $\min\{c(M_1),c(M_2)\} \leq c(T_W)/2 \leq c(T)/2$.

  Comme M est un couplage de coût minimum, on a $c(M) \leq \min\{c(M_1),c(M_2)\}$ et donc $c(M) \leq c(T)/2$

  La solution produite par l'algorithme de Christofides à la fin de l'étape (3) est de valeur c(A)+c(M) et on a donc montré que $c(A)+c(M) \leq c(T)+c(T)/2=3/2\ c(T)$

  L'étape (4) ne peut que raccourcir la tournée puisqu'on a les inégalités triangulaires

Procédure 2-opt de postoptimisation

Soit $T=(v_1,v_2,\ldots,v_n,v_1)$ un cycle hamiltonien.
Pour toute paire d'indices i,j notons T_{ij} la tournée obtenue en remplaçant les arêtes (v_i,v_{i+1}) et (v_j,v_{j+1}) par les arêtes (v_i,v_j) et (v_{i+1},v_{j+1})

| Tant qu'il existe deux indices i et j tel que T_{ij} est plus court que T faire : poser $T := T_{ij}$

Sur des exemples de la vie courante, les trois premiers algorithmes donnent des solutions qui sont à environ 15% de l'optimum. La procédure 2-opt ci-dessus permet de réduire cet écart à environ 2 à 3 %. Il existe des méthodes qui permettent de réduire cet écart à moins d'un 1%, mais ces méthodes ne sont pas l'objet de ce cours.

Classification des problèmes de par leur difficulté

- Problème du postier chinois $\Rightarrow$ facile
- Problème du postier rural $\Rightarrow$ facile si G_R est connexe
  $\Rightarrow$ NP-dur sinon (mais bonne approximation si inégalités triangulaires)
- Problème du voyageur de commerce $\Rightarrow$ NP-dur (mais bonne approximation si inégalités triangulaires)

Remarques

- La recherche d'un cycle eulérien de coût minimum dans un graphe G est équivalente à la recherche d'un cycle hamiltonien dans son graphe de ligne L(G).
- On peut donc transformer un problème de postier chinois sur G en un problème de voyageur de commerce sur L(G). Mais ce n'est pas vraiment une bonne idée car le problème du postier chinois est facile à résoudre optimalement alors que celui du voyageur de commerce est NP-dur en général.