INF431

Graphes eulériens et problème du facteur chinois
CORRIGÉ
Version: 4:2459M

Un postier se prépare à parcourir toutes les rues d'un village pour distribuer le courrier ; en fonction du plan des rues il risque de devoir passer plusieurs fois dans la même rue, et il se pose la question de minimiser la longueur de son trajet.

Nous modélisons ce problème comme un problème d'*optimisation sur un graphe* : il s'agit de trouver dans un graphe dont les arêtes sont pondérées, un cycle de poids minimal parmi les cycles qui visitent toutes les arêtes. Lorsque chaque sommet est incident à un nombre pair d'arêtes, nous verrons qu'une approche gloutonne peut donner le résultat en temps linéaire. D'un autre côté dès qu'il y a des sommets incidents à un nombre impair d'arêtes, l'approche gloutonne échoue et il n'est pas évident que ce problème soit plus simple que, par exemple, le problème du voyageur de commerce, notoirement difficile (NP-complet).

Pourtant nous allons voir que le problème peut être résolu en temps polynomial par une analyse combinatoire assez fine : c'est à la fois le charme ou le défaut (suivant les goûts) de ces questions d'optimisation sur les graphes que de souvent nécessiter une étude ad-hoc astucieuse pour savoir si le problème est facile (polynomial) ou difficile (NP-complet)... Pour conclure cette introduction, remarquons que la situation est vraiment compliquée : le même problème reste polynomial sur un graphe orienté, mais devient difficile (NP-complet) sur un graphe mélangeant arêtes orientées et non-orientées. D'où l'importance dans ce domaine du paradigme polytechnicien : savoir reconnaître un problème connu et s'y ramener...

1 Cycles et chemins eulériens

L'étude des graphes eulériens est souvent associée à une généralisation de la notion de graphe, où deux sommets peuvent être reliés par plus d'une arête. Nous utiliserons la définition suivante :

Définition 1 (multigraphe) Un *multigraphe non orienté sans boucle* $\mathcal{G} = (S, A)$ est composé d'un ensemble de sommets S et d'un ensemble d'arêtes A. Chaque arête $a \in A$ est de la forme $(i, \{s, s'\})$, où i est un identifiant unique pris parmi un ensemble d'identifiants I, et où s et s' sont deux sommets distincts. ◇

(Dans la suite, on dira parfois « graphe » au lieu de « multigraphe ».) La notion de chemin est adaptée à ce cadre :

Définition 2 (chemin) Un *chemin* P dans $\mathcal{G}$ est une liste alternée $(s_0, i_1, s_1, i_2, \ldots, i_k, s_k)$ de sommets et d'identifiants d'arêtes, commençant et terminant par un sommet ($k \geq 0$), telle que, pour tous (s, i, s') consécutifs, $(i, \{s, s'\})$ est une arête de $\mathcal{G}$. On note $|P| = k$ la *longueur*, c'est-à-dire le nombre d'arêtes, d'un tel chemin P. ◇

Définition 3 (concaténation) Si P et P' sont deux chemins tels que l'extrémité droite de P coïncide avec l'extrémité gauche de P', alors $P \# P'$ dénote le chemin obtenu par concaténation de P et P'. ◇

Définition 4 (cycle) Un *cycle* est un chemin dont les sommets extrémaux sont identiques. ◇

Définition 5 (chemin eulérien, cycle eulérien) Un chemin (ou un cycle) est dit *eulérien* s'il emprunte une et une seule fois chaque arête. Un multigraphe possédant un cycle eulérien est lui-même dit eulérien. ◇

On notera qu'un chemin eulérien peut passer plusieurs fois par un même sommet.

Définition 6 (chemin simple, cycle simple) Un chemin (ou, par extension, un cycle) est dit *simple* s'il n'emprunte pas deux fois la même arête (même en des sens opposés) et s'il contient au moins une arête. ◇

L'énoncé suivant caractérise de façon particulièrement simple l'existence d'un cycle eulérien. Dans la suite, nous démontrerons cet énoncé de façon constructive.

Théorème 1 (Euler, 1736) Un multigraphe connexe possède un cycle eulérien si et seulement si chaque sommet est de degré pair. ◇

Construction efficace d'un cycle eulérien

Dans cet exercice, on se donne un multigraphe $\mathcal{G} = (S, A)$. On pose $n = |S|$ et $m = |A|$.

Question 1 On suppose que tous les sommets de $\mathcal{G}$ sont de degré pair et qu'il existe un chemin $P = (s_0, i_1, \ldots, i_k, s_k)$ passant au plus une fois par chaque arête. Montrer que soit P est un cycle, soit il peut être prolongé en un cycle qui emprunte au plus une fois chaque arête. (Toutefois, ce cycle n'emprunte pas nécessairement toutes les arêtes du multigraphe, et n'est donc pas en général un cycle eulérien.) ◇

Solution. Montrons tout d'abord que si les extrémités du chemin sont distinctes alors il peut être prolongé d'une arête tout en restant simple : en effet si s_0 et s_k sont distincts alors le chemin P comporte un nombre impair d'arêtes incidentes à s_k. D'après l'hypothèse de parité du degré de s_k, il existe un sommet s_{k+1} et un identifiant i_{k+1} tel que l'arête $(i_{k+1}, \{s_k, s_{k+1}\})$ appartienne à A et ne soit pas empruntée par P. On prolonge alors P en un chemin $P' = P \mathbin{\#} (s_k, i_{k+1}, s_{k+1})$. Par construction P' emprunte au plus une fois chaque arête.

Tant qu'on est pas revenu à s_0, on peut itérer la construction précédente et produire ainsi un chemin simple de plus en plus long. Or le nombre d'arête d'un chemin simple ne peut excéder le nombre d'arêtes de $\mathcal{G}$, il faut donc que l'itération s'arrête, c'est-à-dire qu'on finisse par obtenir un cycle.

Il est intéressant de noter que le choix de l'arête utilisée pour prolonger P, s'il en existe plusieurs, n'importe pas. Nous venons de démontrer que, si l'on « pose le crayon » sur n'importe quel sommet et si l'on parcourt le graphe aléatoirement, sans jamais « lever le crayon » ni emprunter deux fois la même arête, alors on reviendra nécessairement au point de départ. □

Question 2 On suppose que tous les sommets de $\mathcal{G}$ sont de degré pair. Écrire le pseudo-code d'une fonction FINDCYCLE(s) qui construit et renvoie un cycle, issu du sommet s, passant au plus une fois par chaque arête. Cette fonction retirera du multigraphe les arêtes empruntées par ce cycle. Elle fera en sorte de produire un cycle suffisamment long pour que, une fois ces arêtes retirées, le sommet s soit isolé (un sommet est isolé s'il n'est incident à aucune arête). ◇

Solution. On construit itérativement, par prolongements successifs, un chemin P. La variable s contient, initialement, le sommet de départ, puis, à chaque instant, le sommet courant, c'est-à-dire l'extrémité droite du chemin P. Lorsqu'on atteint un sommet s à partir duquel plus aucun prolongement n'est possible, alors P est nécessairement un cycle : nous sommes revenus au point de départ.

fonction FINDCYCLE(s)
  $P \leftarrow (s)$
  tant que s n'est pas isolé
    faire $\begin{cases} \textbf{soit } a = (i, \{s, s'\}) \in A \text{ une arête incidente à } s \\ A \leftarrow A \setminus \{a\} \\ P \leftarrow P \mathbin{\#} (s, i, s') \\ s \leftarrow s' \end{cases}$
  — s est isolé ; P est nécessairement un cycle
  renvoyer P

Notons qu'un simple algorithme itératif suffit : contrairement au cas du parcours en profondeur d'un multigraphe quelconque, le retour en arrière est inutile. □

Il reste à déterminer comment, à l'aide de FINDCYCLE, on construit un cycle eulérien.

Question 3 On suppose que $\mathcal{G}$ est connexe et que tout sommet de $\mathcal{G}$ est de degré pair. Écrire le pseudo-code d'une fonction EULER() qui construit et renvoie un cycle eulérien de $\mathcal{G}$. Quelle en est la complexité ? ◇

Solution. L'idée est simple. Considérons un cycle empruntant des arêtes distinctes, mais non eulérien. Alors, il existe une arête que ce cycle n'emprunte pas, et puisque $\mathcal{G}$ est connexe, il existe nécessairement une telle arête issue de l'un des sommets du cycle. On relance alors FINDCYCLE à partir de ce sommet. Ceci produit un nouveau cycle, que l'on greffe sur le cycle existant. On itère ce processus jusqu'à obtenir un cycle eulérien.

On pourrait donc écrire l'algorithme sous la forme (particulièrement abstraite !) suivante :

fonction EULER()
  soit s un sommet quelconque de S
  $C \leftarrow (s)$
  tant que C s'écrit $P_1 \# P_2$, avec à la jointure de P_1 et P_2 un sommet s non isolé
    faire $C \leftarrow P_1 \# \text{FINDCYCLE}(s) \# P_2$
  — tous les sommets sont isolés ; C est un cycle eulérien
  renvoyer C

De par ses appels à FINDCYCLE, l'algorithme retire des arêtes du graphe ; lorsqu'il termine, le graphe ne contient plus aucune arête.

Une implémentation naïve de cet algorithme doit, à chaque itération, rechercher un sommet non isolé s le long du cycle C, ce qui est inefficace. Nous allons donc proposer une seconde version de cet algorithme, plus raffinée, qui élimine cette inefficacité.

L'idée est de maintenir à jour, en permanence, la décomposition du cycle C en deux chemins P_1 et P_2, ainsi que le sommet s qui constitue l'extrémité droite de P_1 et l'extrémité gauche de P_2. À chaque itération, on greffe le cycle FINDCYCLE(s) en tête du chemin P_2. (Notons que ce cycle est potentiellement vide.) Puis, si P_2 est non vide, on transfère la première arête de P_2 vers P_1 ; sinon, l'algorithme termine. Le pseudo-code s'écrit comme suit :

fonction EULER()
  soit s un sommet quelconque de S
  $P_1 \leftarrow (s)$
  $P_2 \leftarrow (s)$
  tant que vrai

$$\textbf{faire} \begin{cases} P_2 \leftarrow \text{FINDCYCLE}(s) \# P_2 \\ \textbf{si } |P_2| = 0 \\ \textbf{alors renvoyer } P_1 \\ \textbf{sinon} \begin{cases} \textbf{soit } (s,i,s') \# P_2' = P_2 \\ P_1 \leftarrow P_1 \# (s,i,s') \\ P_2 \leftarrow P_2' \\ s \leftarrow s' \end{cases} \end{cases}$$

L'algorithme utilise *l'invariant de boucle* suivant : les sommets rencontrés le long du chemin P_1, excepté son extrémité droite, sont isolés. (Plus précisément, si P_1 est un chemin de la forme $(s_0, \ldots, s_k)$, alors les sommets $s_0, \ldots, s_{k-1}$ sont isolés.) Vérifions que cette propriété est vraie à l'entrée de chaque itération de la boucle. C'est immédiat. D'une part, initialement, P_1 est le chemin de longueur nulle (s), donc l'invariant est trivialement satisfait. D'autre part, lors de chaque itération, le chemin P_1 se voit ajouter une arête reliant s à s'. Pour que l'invariant soit préservé, il faut que s soit isolé, ce qui est garanti par l'appel à FINDCYCLE(s).

Lorsque l'instruction **renvoyer** P_1 est exécutée, P_2 est vide. Cette propriété, combinée à l'invariant ci-dessus, implique que tous les sommets de P_1 sont isolés. Il en découle, comme dans la première version de l'algorithme, que P_1 forme alors un cycle eulérien.

Il reste à déterminer la complexité de cette version de l'algorithme, et à choisir pour cela des structures de donnés appropriées.

On remarque que la structure de données qui représente le chemin P_2 doit permettre l'insertion (ou concaténation) et le retrait d'une arête en tête, tandis que celle qui représente P_1 doit permettre l'insertion en queue. Une façon simple de répondre à ces contraintes est de représenter P_2 par une liste d'arêtes simplement chaînée, et P_1 par une liste d'arêtes simplement chaînée inversée.

Si D est le cycle construit par FINDCYCLE, le coût d'une itération de la boucle principale (appel à FINDCYCLE, concaténation, mise à jour des variables) est $\mathcal{O}(1 + |D|)$. Puisque chaque arête est empruntée par FINDCYCLE au plus une fois, la somme des $|D|$ au cours des itérations de la boucle est bornée par $\mathcal{O}(m)$. De plus, puisque chaque itération accroît $|P_1|$, le nombre d'itérations de la boucle est lui aussi borné par $\mathcal{O}(m)$. Le coût total de cette boucle est donc $\mathcal{O}(m)$. Cette complexité asymptotique est optimale, puisque le temps dépensé est proportionnel à la taille du résultat recherché – tout cycle eulérien est constitué de m arêtes. □

Nous avons ainsi démontré constructivement le théorème d'Euler.

Question 4 Proposer une structure de données pour représenter les graphes qui permette d'implémenter l'algorithme FINDCYCLE (et donc EULER) en temps linéaire. ◇

Solution. Les opérations que l'on veut pouvoir effectuer en temps constant sont :
- étant donné l'identifiant d'un sommet, trouver l'identifiant d'une arête qui lui est incidente,
- étant donné l'identifiant d'une arête, trouver les identifiants des deux sommets qui lui sont incidents,
- supprimer une arête à partir de son identifiant.

Une solution pour ce faire est d'opter pour une représentation par liste d'adjacence de demi-arêtes complétée par un tableau de pointeurs explicites des arêtes vers les demi-arêtes correspondantes.
- un tableau indexé par les sommets dont l'entrée d'indice i est une liste doublement chainée de cellules contenant chacun l'identifiant d'une arête incidente au sommet d'indice i.
- un tableau indexé par les arêtes dont l'entrée d'indice j comporte les indices des deux sommets extrémités ainsi qu'un pointeur vers chacune des deux cellules contenant l'identifiant j dans les listes d'adjacence.

La suppression d'une arête se fait en identifiant les deux cellules à supprimer à partir du tableau indexé par les arêtes et en les court-circuitant dans les listes doublement chainées. □

2 Le problème du facteur chinois

On étudie maintenant un problème d'optimisation appelé *problème du facteur chinois* ou *Chinese postman problem*. Il a été posé par le mathématicien chinois Mei Ko Kwan en 1962. Ce problème dérive de celui dit des ponts de Königsberg, résolu par Euler en 1736.

Présentation du problème d'optimisation

Soit $\mathcal{G} = (S, A)$ un multigraphe non orienté connexe, et *coût*$(\cdot) : A \to \mathbb{N}$ une valuation des arêtes de $\mathcal{G}$. On notera que deux arêtes reliant la même paire de sommets peuvent avoir des valuations différentes. On suppose par la suite que $\mathcal{G}$ ne contient pas de cycle simple de valuation nulle.

Le *problème du facteur chinois* consiste à construire un *cycle de valuation minimale* passant au moins une fois par chaque arête. La figure 1 propose un exemple de problème. La métaphore du facteur considère les arêtes comme les rues d'un quartier et la valuation comme la distance entre deux carrefours : le facteur, partant du bureau de poste, doit parcourir toutes les rues et revenir au bureau de poste tout en minimisant la distance totale.

Bien entendu, si $\mathcal{G}$ est eulérien, alors tout cycle eulérien est solution optimale à ce problème, et ce quelle que soit la valuation *coût*$(\cdot)$. Dans la suite on appelle *tournée* un cycle passant au moins une fois par chaque arête.

Si l'on parcourt un cycle eulérien en dessinant les arêtes au fur et à mesure du parcours, on reconstitue le graphe eulérien sous-jacent. Plus généralement, si l'on parcourt une *tournée* C d'un graphe $\mathcal{G}$, qui emprunte éventuellement plusieurs fois certaines arêtes, en dessinant *de nouvelles arêtes* au fur et

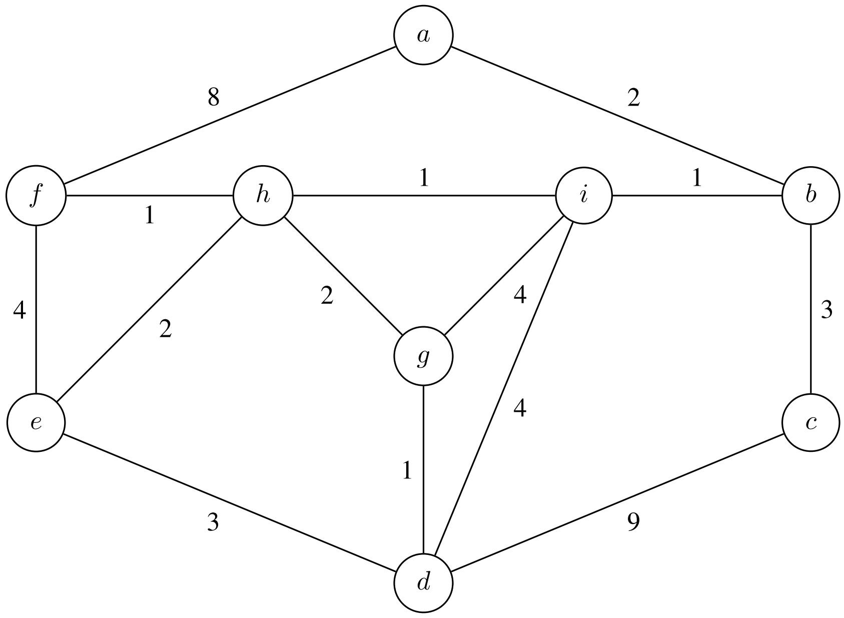

FIG. 1 – Une instance du problème du facteur chinois

à mesure du parcours, ces nouvelles arêtes forment un *multigraphe eulérien* $\mathcal{G}'$, dans lequel les arêtes empruntées plusieurs fois par la tournée ont été *dupliquées*. (C'est ici que nous exploitons réellement la notion de multigraphe.) Nous allons utiliser cette construction pour ramener l'étude des tournées à celle de cycles eulériens. Le lemme suivant formalise la construction esquissée ci-dessous et pourra être utile pour la rédaction de preuves formelles des résultats suivants.

Lemme 1 Le graphe $\mathcal{G} = (S, A)$ admet une tournée C de valuation c si et seulement si il existe un cycle eulérien C' de valuation c dans un graphe $\mathcal{G}' = (S, A')$ muni d'une surjection $\phi : A' \to A$ qui préserve les extrémités des arêtes et leurs valuations. ◇

Preuve : Soit $C = (s_0, i_1, \ldots, i_k, s_k)$ une tournée de $\mathcal{G}$. Notons que, en général, les identifiants i_j ne sont pas deux à deux distincts, puisque C n'est pas un cycle eulérien. Nous allons choisir des identifiants uniques, par exemple les entiers de 1 à k, pour obtenir un cycle eulérien. Soit donc C' le chemin $(s_0, 1, \ldots, k, s_k)$. Le chemin C' est un cycle eulérien vis-à-vis du multigraphe $\mathcal{G}' = (S, A')$, où A' est l'ensemble des arêtes parcourues par C', à savoir $\{(j, \{s_{j-1}, s_j\}) \mid 1 \leq j \leq k\}$. La fonction $\phi : A' \to A$ qui à l'arête d'identifiant j associe l'arête d'identifiant i_j est surjective : en effet, puisque C est une tournée, l'ensemble $\{i_1, \ldots, i_k\}$ contient (les identifiants de) *toutes* les arêtes de $\mathcal{G}$.

Réciproquement, soit C' un cycle eulérien d'un graphe $\mathcal{G}'$ muni d'une telle surjection ϕ. Soit $C = \phi(C')$. La conservation des extrémités des arêtes par ϕ assure que C est un cycle de $\mathcal{G}$. La conservation des valuations assure que C' et C ont même valuation. Enfin, la surjectivité de ϕ garantit que toutes les arêtes de $\mathcal{G}$ sont empruntées au moins une fois par C, donc que C est une tournée. □

Sur la structure de l'ensemble des arêtes réutilisées dans une tournée

Question 5 Montrer que, parmi les tournées de valuation minimale sur un graphe $\mathcal{G}$, il en existe une qui emprunte au plus deux fois chaque arête. ◇

Solution. Première preuve : Soit C une tournée qui emprunte au moins trois fois une arête $a = (i, \{s, s'\})$. Cette arête est donc empruntée au moins deux fois dans le même sens, disons de s vers s'. Modulo une rotation, le cycle C peut donc s'écrire $C = (s, i, s') \# C_1 \# (s, i, s') \# C_2$, où C_1 et C_2 sont deux chemins de s' vers s. Soit C_2' le chemin inverse de C_2, donc un chemin de s vers s'. Alors,

$C_1 \# C'_2$ est un cycle de $\mathcal{G}$; emprunte au moins une fois chaque arête (car C_1 ou C_2 emprunte a) ; est de valuation inférieure ou égale à celle de C ; et emprunte l'arête a deux fois de moins que C.

En itérant cette construction à partir d'une tournée de valuation minimale, tant qu'il existe une arête empruntée au moins trois fois, on obtient finalement une tournée, également de valuation minimale, qui emprunte au plus deux fois chaque arête. (Le processus termine bien, puisque chaque itération produit une tournée strictement plus courte, en termes de nombre d'arêtes, que la précédente.)

Deuxième preuve : (plus dans l'esprit de la suite) Soit C une tournée de valuation minimale et soit $\mathcal{G}'$ le multigraphe associé. Supposons que la tournée C emprunte au moins 3 fois l'arête e. Alors $\mathcal{G}'$ contient au moins 3 copies de cette arête et le graphe $\mathcal{G}''$ obtenu en éliminant deux de ces copies est toujours eulérien. Ce graphe est de valuation inférieure ou égale à $\mathcal{G}'$, donc la tournée associée est toujours minimale. On termine la preuve en itérant comme au dessus. □

Il est donc licite de rechercher la solution du problème du facteur chinois parmi les tournées empruntant chaque arête de $\mathcal{G}$ au plus deux fois. Pour les trois questions suivantes, on fixe une telle tournée C, de valuation minimale. On note D l'ensemble des arêtes de $\mathcal{G}$ empruntées deux fois dans C, et on étudie le graphe $\mathcal{D} = (S, D)$ formé par ces arêtes. Notons que ces définitions impliquent $|C| = |D| + |A|$.

Question 6 Montrer, en utilisant le graphe $\mathcal{G}'$, que le graphe $\mathcal{D}$ ne contient pas de cycle simple. ◇

Solution. idée de la preuve : s'il y a un cycle simple dans D la suppression d'une copie de ce cycle dans $\mathcal{G}'$ donne un graphe eulerien qui contient toujours $\mathcal{G}$ et de valuation inférieure à $\mathcal{G}'$...
Détails : Supposons que $\mathcal{D}$ contienne un cycle simple T.

On construit le graphe eulérien $\mathcal{G}' = (S, A')$ associé à la tournée C, ainsi que la surjection $\phi : A' \to A$, suivant le lemme 1.

Puisque ϕ est surjective, il existe dans $\mathcal{G}'$ un cycle simple T' dont l'image par ϕ est T. Retirons ce cycle de $\mathcal{G}'$: formons le graphe $\mathcal{G}'' = (S, A'')$, où $A'' = A' \setminus T'$.

Par hypothèse, tout cycle simple est de valuation non nulle, donc $\mathcal{G}''$ est de valuation totale strictement inférieure à celle de $\mathcal{G}'$.

Parce que le cycle T est inclus dans D, $\mathcal{G}''$ est toujours connexe et la restriction de ϕ à $A'' \to A$ est toujours surjective. (En bref, $\mathcal{G}''$ contient toujours $\mathcal{G}$.)

Enfin parce que T' est un cycle *simple*, tout sommet est incident à un nombre pair d'arêtes de T'. Par conséquent, $\mathcal{G}''$ est toujours eulérien.

De ces trois points, il résulte (lemme 1) qu'il existe dans $\mathcal{G}$ une tournée de valuation strictement inférieure à celle de C, ce qui contredit la minimalité de C. □

Question 7 Montrer que les chemins simples du graphe $\mathcal{D}$ sont des plus courts chemins du graphe $\mathcal{G}$. (On pourra se contenter de décrire comment adapter la preuve de la question précédente.) ◇

Solution. Idée de la preuve : on suppose qu'il existe un chemin Q plus cours de s à s' dans $\mathcal{G}$ et on observe que $\mathcal{G}'' = \mathcal{G}' - P \cup Q$ est un graphe eulerien qui contient toujours $\mathcal{G}$ et de valuation inférieure à $\mathcal{G}'$.
Détails : Comme précédemment, on construit le graphe eulérien $\mathcal{G}' = (S, A')$ associé à la tournée C, ainsi que la surjection $\phi : A' \to A$.

D'après la question 6, $\mathcal{D}$ ne contient pas de cycle simple, donc un chemin simple ne peut passer deux fois par un même sommet. En particulier, s et s' sont distincts.

Soit $Q = (s_0 = s, i_1, \ldots, i_k, s_k = s')$ un plus court chemin entre s et s' dans $\mathcal{G}$, supposé de valuation strictement inférieure à celle de P. On peut toujours supposer que Q est simple. La construction qui suit consiste à supprimer de $\mathcal{G}'$ un chemin isomorphe à P, puis à ajouter un chemin isomorphe à Q.

Puisque ϕ est surjective, il existe dans $\mathcal{G}'$ un chemin simple P' dont l'image par ϕ est P. Retirons ce chemin de $\mathcal{G}'$: formons le graphe $\mathcal{G}'' = (S, A'')$, où $A'' = A' \setminus P'$. Parce que P' est simple, tout sommet est incident à un nombre pair (en fait, zéro ou deux) d'arêtes de P', sauf s et s', qui sont incidents à un nombre impair (en fait, une) d'arêtes de P'. Par conséquent, dans $\mathcal{G}''$, tout sommet est de degré pair, à l'exception de s et s', dont le degré est impair. Parce que le chemin P est inclus dans D, la restriction de ϕ à $A'' \to A$ est toujours surjective. (En bref, $\mathcal{G}''$ contient toujours $\mathcal{G}$.)

Soit maintenant $\mathcal{G}''' = (S, A''')$ le graphe obtenu à partir de $\mathcal{G}''$ en y insérant le chemin simple Q. On pose $A''' = A'' \uplus \{(i'_j, \{s_{j-1}, s_j\}) \mid 1 \leq j \leq k\}$, où les i'_j sont des identifiants arbitraires distincts n'apparaissant pas dans A''. La surjection ϕ s'étend naturellement à $A''' \to A$ en posant $\phi(i'_j) = i_j$. Par le même raisonnement que ci-dessus, on constate que le graphe $\mathcal{G}'''$ est à nouveau eulérien. Enfin, par construction, la valuation totale de $\mathcal{G}'''$ est strictement inférieure à celle de $\mathcal{G}'$, parce que nous avons supposé la valuation de Q strictement inférieure à celle de P.

De ceci, il résulte (lemme 1) qu'il existe dans $\mathcal{G}$ une tournée de valuation strictement inférieure à celle de C, ce qui contredit la minimalité de C. □

Question 8 Montrer que, pour tout sommet s, la parité du degré de s dans $\mathcal{G}$ est égale à la parité du degré de s dans $\mathcal{D}$. En déduire qu'on peut partitionner l'ensemble d'arêtes D en un ensemble de chemins simples dont les extrémités sont toutes distinctes et sont formées de l'ensemble des sommets de degré impair de $\mathcal{G}$. ◇

Solution. On a, pour tout sommet s, la relation $d_{\mathcal{G}'}(s) = d_{\mathcal{G}}(s) + d_{\mathcal{D}}(s)$. De plus, $\mathcal{G}'$ est eulérien, d'où le premier résultat.

Il reste donc à démontrer que les arêtes du graphe acyclique $\mathcal{D}$ peuvent être partitionnées en un ensemble de chemins simples dont les extrémités sont (tous) les sommets de degré impair dans $\mathcal{D}$. Vérifions ceci par induction généralisée sur le nombre d'arêtes de $\mathcal{D}$. S'il est nul, le résultat est immédiat. S'il est non nul, alors $\mathcal{D}$ contient au moins une arête, donc au moins un chemin simple. Soit alors P un chemin simple maximal (c'est-à-dire non prolongeable) de $\mathcal{D}$. L'acyclicité de $\mathcal{D}$ et la maximalité de P impliquent que les extrémités s et s' de P ont degré 1 dans $\mathcal{D}$. Soit alors $\mathcal{D}' = \mathcal{D} \setminus P$. Par hypothèse d'induction, $\mathcal{D}'$ peut être partitionné en un ensemble $\mathcal{P}$ de chemins simples dont les extrémités sont les sommets de degré impair dans $\mathcal{D}'$. Parce que P est simple, à l'exception de s et s', tout sommet a un degré de même parité dans $\mathcal{D}$ et dans $\mathcal{D}'$. Les sommets s et s' sont de degré nul dans $\mathcal{D}'$. Les sommets de degré impair dans $\mathcal{D}$ sont donc s, s', et les sommets de degré impair dans $\mathcal{D}'$. L'ensemble $\mathcal{P} \cup \{P\}$ forme donc la partition recherchée. □

Réduction à un problème de couplage parfait de poids minimum

Dans le but de résoudre le problème du facteur chinois, on cherche donc à apparier (c'est-à-dire à relier deux à deux) tous les sommets de $\mathcal{G}$ de degré impair, et ce au moyen de plus courts chemins.

On construit un graphe auxiliaire non orienté $\mathcal{G}_{\text{aux}}$, dont les sommets sont les sommets de degré impair de $\mathcal{G}$. Le graphe $\mathcal{G}_{\text{aux}}$ est *complet* : toute paire de sommets distincts forme une arête. La valuation associée à une arête $\{s, s'\}$ de $\mathcal{G}_{\text{aux}}$ est la valuation d'un plus court chemin entre s et s' dans $\mathcal{G}$.

L'algorithme de Floyd-Warshall [1] permet de construire l'ensemble des plus courts chemins de $\mathcal{G}$ en temps $\mathcal{O}(n^3)$. Si le nombre k de sommets de degré impair est faible devant n, ou bien si le nombre m d'arêtes du graphe est faible devant n^2, on peut préférer appliquer l'algorithme de Dijkstra pour calculer les distances à partir de chacun des sommets de degré impair, pour une complexité totale de $\mathcal{O}(k \cdot (n + m) \log n)$.

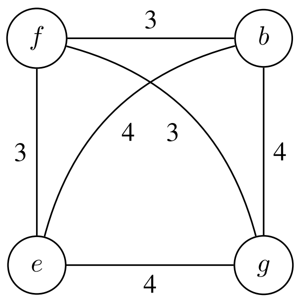

FIG. 2 – Graphe complet auxiliaire

Question 9 Construire le graphe auxiliaire, dans le cas où $\mathcal{G}$ est le graphe de la figure 1. ◇

Solution. Voir la figure 2. □

Définition 7 (couplage parfait) Un *couplage* d'un graphe $\mathcal{G}$ est un sous-graphe de $\mathcal{G}$ (comportant donc les mêmes sommets, et en général moins d'arêtes, que $\mathcal{G}$) dont chaque sommet appartient à au plus une arête. Un couplage est dit *parfait* lorsque chaque sommet appartient à une arête. ⋄

Question 10 Montrer que $\mathcal{G}_{\text{aux}}$ admet un couplage parfait. ⋄

Solution. Ce résultat découle directement de la question 8. (On peut aussi observer directement que le nombre de sommets impair d'un graphe est pair (handshaking lemma) et apparier les sommets d'indices impairs avec les sommets d'indices pairs. □

Question 11 Montrer que le problème du facteur chinois se réduit à la recherche d'un couplage parfait de valuation minimale pour le graphe $\mathcal{G}_{\text{aux}}$. ⋄

Solution. Soit c la valuation d'une tournée de valuation minimale sur $\mathcal{G}$. On peut écrire $c = c_0 + d$, où c_0 est la somme des valuations des arêtes de $\mathcal{G}$, et d est la somme des valuations des arêtes empruntées deux fois par cette tournée. Par ailleurs, soit d' la valuation d'un couplage parfait de valuation minimale sur $\mathcal{G}_{\text{aux}}$.

À un couplage parfait de valuation minimale sur $\mathcal{G}_{\text{aux}}$ correspond un ensemble de chemins de $\mathcal{G}$ dont les extrémités sont distinctes et sont (tous) les sommets de degré impair de $\mathcal{G}$. (A priori, on n'a pas prouvé que ces chemins n'ont pas d'arête commune.) La somme des valuations de ces chemins est égale à d'. Construisons un graphe $\mathcal{G}'$ à partir de $\mathcal{G}$ en dupliquant chacun de ces chemins. Un sommet de degré impair dans $\mathcal{G}$ est l'extrémité d'exactement un chemin, donc devient de degré pair dans $\mathcal{G}'$. Un sommet de degré pair de $\mathcal{G}$ ne peut apparaître qu'à l'intérieur des chemins dupliqués, donc reste de degré pair. Le graphe $\mathcal{G}'$ est donc eulérien. Il admet un cycle eulérien C', qui détermine une tournée C sur $\mathcal{G}$. La valuation de cette tournée C est égale à la valuation c_0 de $\mathcal{G}$ plus celle d' du couplage. On en déduit que $c \leq c_0 + d'$. On note, au passage, que la construction de la tournée C à partir du couplage parfait donné est effective (nous avons donné un algorithme) et peut être effectuée en temps $\mathcal{O}(m)$.

Réciproquement, étant donné une tournée C de valuation minimale sur $\mathcal{G}$, on partitionne l'ensemble D des arêtes empruntées deux fois par C en un ensemble de chemins simples dont les extrémités sont (tous) les sommets de degré impair de $\mathcal{G}$ (question 8). Ces chemins déterminent un couplage parfait de $\mathcal{G}_{\text{aux}}$. De plus, ces chemins sont des plus courts chemins dans $\mathcal{G}$. La valuation du couplage obtenu est donc égale à la valuation d de D, c'est-à-dire à $c - c_0$. On en déduit que $d' \leq c - c_0$, d'où $c \geq c_0 + d'$, et $c = c_0 + d'$.

La tournée construite à partir d'un couplage parfait de valuation minimale est donc bien de valuation minimale. □

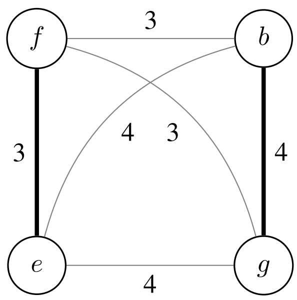

FIG. 3 – Couplage parfait de valuation minimale

Question 12 Trouvez un couplage parfait de valuation minimale pour le graphe auxiliaire construit à la question 9, et déduisez-en une solution optimale au problème du facteur chinois pour le graphe valué de la figure 1. ⋄

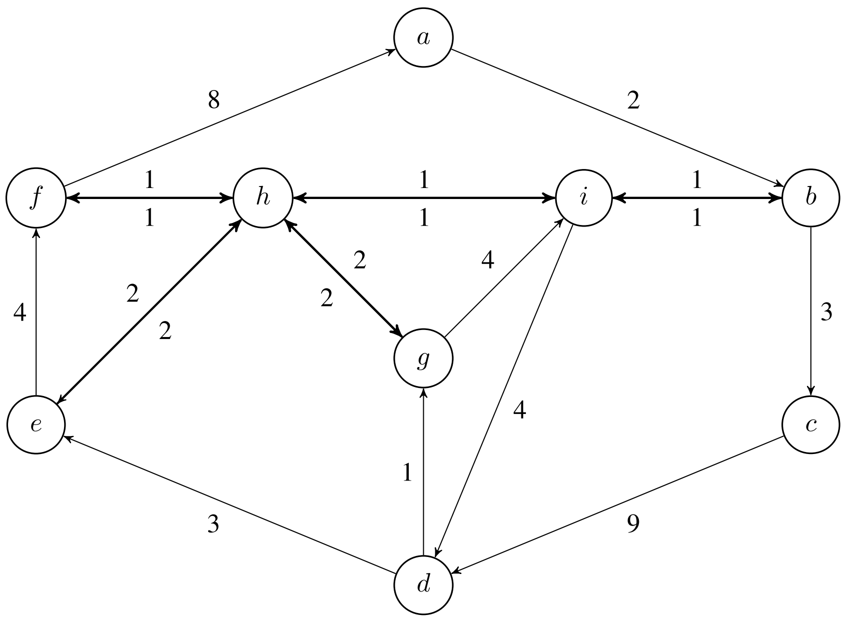

FIG. 4 – Solution : $(g, i, d, g, h, i, h, e, f, h, f, a, b, i, b, c, d, e, h, g)$

Solution. Voir les figures 3 et 4. Un couplage parfait de valuation minimale relie e à f et b à g. (Cette solution n'est pas unique.) Dans le graphe original $\mathcal{G}$, les plus courts chemins entre e et f et entre b et g sont donc dupliqués. Chacune des arêtes qui constituent ces chemins sera empruntée deux fois (en des sens opposés). La construction d'une tournée se fait à l'aide de l'algorithme de la question 3. □

Dans le cas eulerien la complexité de la construction d'une tournée optimale est linéaire en m. Pour k fixé (petit) la complexité résultante est à peine moins bonne : nous avons perdu un facteur $\log n$ lors du calcul des plus courts chemins. Pour des valeurs de k arbitraires il faut par contre tenir compte du temps de calcul du couplage parfait de valuation minimale : Le meilleur algorithme connu pour déterminer un couplage parfait de valuation minimale est dû à Gabow [2]. Cet algorithme dépasse largement le cadre de ce cours ; sa complexité asymptotique pour un graphe à k sommets et ℓ arêtes est $\mathcal{O}(k(\ell + k \log k))$. Edmonds a donné le premier algorithme polynomial pour résoudre ce problème classique : sa complexité était de $\mathcal{O}(\ell k^2)$. Une bonne introduction aux problèmes de couplage est disponible dans l'ouvrage de référence de Gondran et Minoux [3]. Dans notre cas $\ell = \Theta(k^2)$ – le graphe auxiliaire est complet – de sorte que l'algorithme de Gabow donne une complexité $O(k^3)$.

Références

[1] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest et Clifford Stein. *Introduction à l'algorithmique : Cours et exercices*. Sciences Sup. Dunod, 2002.

[2] Harold N. Gabow. Data structures for weighted matching and nearest common ancestors with linking. *In ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 434–443, 1990.

[3] M. Gondran et M. Minoux. *Graphes et algorithmes*. Eyrolles, 1995.