Action adjointe sur les graphes et la preuve de la conjecture P=NP

Par :
Mohamed Sghiar
msghiar21@gmail.com
Présenté à :
UNIVERSITÉ DE BOURGOGNE DIJON
Faculté des sciences Mirande
Département de mathématiques et informatiques
9 Av Alain Savary
21078 DIJON CEDEX

Abstract : I study the link between the adjoint action and the Hamiltonian cycles in a symmetric graph. Then by a simple algebraic resolution of a system of equations with several variables I find all the Hamiltonian cycles of the graph. Finally I will apply the results found to give an algorithm of order $\mathcal{O}(n^3)$ allowing to quickly give all the Hamiltonian cycles with their distance. This gives a proof of the conjecture $P = NP$. (see [1]).

Résumé : J'étudie le lien entre l'action adjointe et les cycles Hamiltoniens dans un graphe symétrique. Puis par une simple résolution algébrique d'un système d'équations à plusieurs variables on trouve tous les cycles Hamiltoniens du graphe. Enfin j'appliquerai les résultats trouvés pour donner un algorithme de l'ordre de $\mathcal{O}(n^3)$ permettant de donner rapidement tout les cycles Hamiltoniens avec leur distance. Ce qui donne une preuve de la conjecture $P = NP$. (voir [1]).

Keywords : Graph, Hamilton cycles, P=NP, the travelling salesman problem, TSP, Analysis of algorithms, adjoint action.

Code : 68R10, 05CXX, 68R05, 05XX, 15AXX, 15B10, 68W99, 68XX, 14-XX, 14LXX

1 Introduction, notations et définitions

Le problème du voyageur de commerce (TSP), qui est un problème NP-difficile en optimisation combinatoire (voir [1] à [8]), très important dans la recherche opérationnelle et informatique théorique, pose la question suivante : Étant donné une liste de villes et les distances entre chaque paire de villes, quel est l'itinéraire le plus court possible qui visite chaque ville exactement une fois et retourne à la ville d'origine ?. Un tel itinéraire, sans tenir compte de la distance est dit un **cycle Hamiltonien**.

Les n villes sont représentées par un graphe symétrique : $G = G_n(\delta, d)$ où $\delta(v_i, v_j) = \delta_{i,j} = 1$ si les deux villes v_i et v_j sont reliées par un chemin (non orienté) sinon $\delta(v_i, v_j) = \delta_{i,j} = 0$ si les deux villes ne sont pas reliées par un chemin. Par convention on pose $\delta(v_i, v_j) =< G(v_i), v_j >= 0$ si $j = i$.

La fonction $d(v_i, v_j) = d_{i,j}$ représente la distance entre les deux villes v_i et v_j.

Si $G = G_n(\delta, d)$ est un graphe sur un ensemble E à n éléments. (E représente les n villes). Alors à toute permutation σ sur E de matrice M_σ^t , faisons agir σ sur G comme suit : $\sigma.G = M_\sigma G M_\sigma^t$ avec $< M_\sigma G M_\sigma^t(e_i), e_j >=< GM_\sigma^t(e_i), M_\sigma^t(e_j) >= \delta(M_\sigma^t(e_i), M_\sigma^t(e_j))$. Cette action est dite une **action adjointe** sur les graphes.

Dans cette article j'étudie le lien entre l'action adjointe et les cycles Hamiltoniens dans un graphe symétrique. Puis par une simple résolution algébrique d'un système d'équations à plusieurs variables on trouve tout les cycles Hamiltoniens du graphe. Enfin j'appliquerai les résultats trouvés dans le corollaire 2.2 pour donner un algorithme de l'ordre de $\mathcal{O}(n^3)$ permettant de donner rapidement tout les cycles Hamiltoniens avec leur distance.

2 Le lien entre les cycles Hamiltoniens et les actions adjointes

Proposition 2.1 *Soit $G = G_n(\delta, d)$ un graphe sur un ensemble E à n éléments.*
$G_n(\delta, d)$ possède un cycle Hamiltonien si et seulement si il existe P^t une matrice d'une permutation σ sur E telle que :
$\sigma.G = PG_n(\delta, d)P^t = M$ avec $M = (m_{i,j})$ et $m_{i,i-1} = 1$, $\forall i \in \{1, \cdots, n\}$ (i-1 est considéré mod n)

Preuve :

À toute permutation σ sur E de matrice M_σ^t , faisons agir σ sur G comme suit :
$\sigma.G = M_\sigma G M_\sigma^t$ avec $< M_\sigma G M_\sigma^t(e_i), e_j > = < GM_\sigma^t(e_i), M_\sigma^t(e_j) >$
Si G possède un cycle Hamiltonien, alors il existe une suite $x_1, \cdots, x_n$ telle que $< G(x_i), x_{i+1} > = 1$ $\forall i \in \{1, \cdots, n\}$. (i+1 est considéré mod n).
Soit σ la permutation telle que $\sigma(x_i) = x_{i+1}$ $\forall i \in \{1, \cdots, n\}$. (i+1 est considéré mod n).
On a donc $\sigma^i(x_1) = x_{i+1}$ $\forall i \in \{1, \cdots, n\}$. (i+1 est considéré mod n).
En posant $M_\sigma G M_\sigma^t = M$ et $e_i = \sigma^{i-1}(x_1)$, on en déduit le résultat car : $< M_\sigma G M_\sigma^t(e_i), e_j > = < GM_\sigma^t(e_i), M_\sigma^t(e_j) > = < G\sigma^i(x_1), \sigma^j(x_1) > = 1$ si $j =$

$i+1$

Le sens inverse est facile à voir.

Corollaire 2.1 *Soit $G = G_n(\delta, d)$ un graphe sur un ensemble E à n éléments.*

Posons $\delta = (\delta_{i,j})$ et $d = (d_{i,j})$

$G_n(\delta, d)$ possède un Hamiltonien si et seulement si ces n équations ont des solutions :

$$\forall i \in \{1, \cdots, n\}, \sum_{k,l} p_{i,k}\delta_{k,l}p_{i-1,l} = 1, avec\ p_{i,j} \in \{0,1\}, p_{i,i} = 0, \sum_{k} p_{i,k} = 1, \sum_{k} p_{k,i} = 1$$

(i-1 est considéré mod n)

Et en posant $P = (p_{i,j})$, alors $\{P^t(e_1), \cdots, P^t(e_n)\}$ est un cycle Hamiltonien ayant pour distance : $\sum_i d(P^t(e_i), P^t(e_{i+1}))$, et tout les cycles Hamiltoniens sont trouvés de cette façon.

Corollaire 2.2 *Soit $G = G_n(\delta, d)$ un graphe sur un ensemble E à n éléments.*

Posons $\delta = (\delta_{i,j})$ et $d = (d_{i,j})$

$G_n(\delta, d)$ possède un cycle Hamiltonien si et seulement si ces n équations ont des solutions :

$$\forall i \in \{1, \cdots, n\}, p_{i,k}\delta_{k,l}p_{i-1,l} = 1, avec\ p_{i,j} \in \{0,1\}, \sum_{k} p_{i,k} = 1, \sum_{k} p_{k,i} = 1, p_{i,i} = 0$$

(i-1 est considéré mod n)

Et en posant $P = (p_{i,j})$, *alors* $\{P^t(e_1), \cdots, P^t(e_n)\}$ *est un cycle Hamiltonien ayant pour distance :* $\sum_i d(P^t(e_i), P^t(e_{i+1}))$, *et tout les cycles Hamiltoniens sont trouvés de cette façon.*

Preuve :

Toute les solutions du système du corollaire 2.2 sont solutions du système du corollaire 2.1.

Le sens inverse se déduit du fait que si une $p_{i,j}(\neq 0)$ *de* $\sum_{k,l} p_{i,k}\delta_{k,l}p_{i-1,l} = 1$ *n'est pas dans une solution du système du corollaire 2.2 , alors* $p_{i,j}\delta_{j,l}p_{i-1,l} = 0\ \forall l$, *et par suite* $\sum_{k,l} p_{i,k}\delta_{k,l}p_{i-1,l} = 0$, *ce qui est absurde .*

3 Algorithme de Gauss-Jordan-sghiar pour trouver les cycles Hamiltoniens

Algorithme 3.1 (GJS-Algorithme pour trouver les cycles Hamiltoniens)

"Construction de la matrice P à partir du corollaire 2.2" :

Posons $\delta_{\bullet,l} = \{k \in \{1, \ldots, n\} \diagup \delta_{k,l} = 1\}$

Posons $\delta_{k,\bullet} = \{l \in \{1, \ldots, n\} \diagup \delta_{k,l} = 1\}$

Pour i=2, :

On pose : $P_1 = \{l \neq 1 \diagup l \in \delta_{k,\bullet}, k \neq 2\}$ *et* $P_2 = \{k \neq 2 \diagup \delta_{k,\bullet} \setminus \{1\} \neq \emptyset\}$

Si $P_1 = \emptyset$ *ou* $P_2 = \emptyset$ *: Affiche " Il n' y a pas de cycle Hamiltonien "*

——————————— *Et on arrête le programme* ———————————

Sinon :

Supposons construit $P_1, \cdots, P_{i-1}$, *et construisons* P_i *pour* $1 \leq i \leq n$

$P_i = \{k \neq i \diagup \delta_{k,\bullet} \diagdown \{i-1\} \neq \emptyset \ \}$

Si $P_i = \emptyset$ *:Affiche " Il n' y a pas de cycle Hamiltonien "*

—————————— *Et on arrête le programme* ——————————

Sinon :

Si $P_1 \diagdown \{1, n\} = \emptyset$ *: Affiche " Il n' y a pas de cycle Hamiltonien "*

—————————— *Et on arrête le programme* ——————————

Si $P_1 \diagdown \{1, n\} \neq \emptyset$ *alors :*

Si $\cup_i P_i \neq \{1, \cdots, n\}$ *: Affiche " Il n' y a pas de cycle Hamiltonien "*

—————————— *Et on arrête le programme* ——————————

Sinon, comme :

$$\forall i \in \{1, \cdots, n\}, p_{i,k}\delta_{k,l}p_{i-1,l} = 1, avec\ p_{i,j} \in \{0,1\}, \sum_k p_{i,k} = 1, \sum_k p_{k,i} = 1, p_{i,i} = 0$$

(i-1 est considéré mod n)

Pour procéder à l'élimination de certains $p_{i,k}$ *afin que* P *soit une matrice orthogonale* ($PP^t = I_n$), *on pose* $P = (p_{i,j})$ *avec* $p_{i,k} \in \{0,1\}$ *si* $k \in P_i$ *sinon* $p_{i,j} = 0$

Trouvons les $p_{i,j}$ *par la résolution du système :*

$\forall i \in \{1, \cdots, n\} : \sum_k p_{i,k} = 1$

$\forall i \in \{1, \cdots, n\} : \sum_k p_{k,i} = 1$

Qu'on peut écrire :

$\forall i \in \{1, \cdots, n\} : \sum_j \epsilon_{i,j} p_{i,j} = 1$ *avec* $\epsilon_{i,j} = 1$ *si* $j \in P_i$ *sinon* $\epsilon_{i,j} = 2$. *avec* $p_{i,j} \in \{0, 1\}$

$\forall i \in \{1, \cdots, n\} : \sum_j \epsilon_{j,i} p_{j,i} = 1$ *avec* $\epsilon_{j,i} = 1$ *si* $i \in P_j$ *sinon* $\epsilon_{i,j} = 2$. *avec* $p_{i,j} \in \{0, 1\}$

Par l'élimination de Gauss-Jordan, si ce système n'a pas de solution : Affiche " Il n' y a pas de cycle Hamiltonien"

————————————— *Et on arrête le programme* —————————————

Sinon pour toute solution on pose $P = (p_{i,j})$ *avec* $p_{i,j} \in \{0, 1\}$ *et* P *est une matrice orthogonale* ($PP^t = I_n$)

$\{P^t(e_1), \cdots, P^t(e_n)\}$ *est un cycle Hamiltonien ayant pour distance :* $\sum_i d(P^t(e_i), P^t(e_{i+1}))$, *et tout les cycles Hamiltoniens sont trouvés de cette façon.*

Affiche : " Il y' a des cycles Hamiltonien les voici :"

$\{P^t(e_1), \cdots, P^t(e_n)\}$ *est un cycle Hamiltonien ayant pour distance :* $\sum_i d(P^t(e_i), P^t(e_{i+1}))$

——————

——————

————————————— *Fin du programme* —————————————

4 Conclusion et preuve de la conjecture P=NP

En algèbre linéaire, l'élimination de Gauss-Jordan, aussi appelée méthode du pivot de Gauss - nommée en hommage à Carl Friedrich Gauss et Wilhelm

Jordan - est un algorithme permettant de déterminer les solutions d'un système d'équations linéaires, le rang d'une matrice ou trouver l'inverse d'une matrice (carrée) inversible. Lorsqu'on applique l'élimination de Gauss à une matrice, on obtient sa forme échelonnée réduite. La complexité algorithmique asymptotique de l'élimination de Gauss est $\mathcal{O}(n^3)$, il en résulte que mon algorithme ci-dessus, sa complexité algorithmique asymptotique reste lui aussi de l'ordre de $\mathcal{O}(n^3)$ et résout donc le problème du voyageur de commerce (TSP) en temps polynomiale (car de l'ordre $\mathcal{O}(n^3)$). Ce qui confirme bien l'ordre $\mathcal{O}(n^3)$ trouvé dans [5] , [6], [7] and [8].
À la différence des algorithmes trouvés dans [5] and [6], [7] qui nécessitent beaucoup de mémoire quoique ils sont de l'ordre $\mathcal{O}(n^3)$, l'algorithme de cet article ne demande pas assez de mémoire puisque **il se ramène à une simple résolution des équations d'un système linéaire.** Et il semble que mon algorithme peut être généralisé pour les graphes **orientés** mais il faut cette fois se ramener à la résolution d'un système d'équations polynomiales à plusieurs variables.
Par ailleurs, Il est connu que le problème du voyageur de commerce (TSP) est un problème NP-Complet et que la résolution d'un problème complet entraîne la preuve de la conjecture $P = NP$. (voir [1]).
On conclut donc que $P = NP$.

Références

[1] Stephen Cook. The p versus np problem. *http://www.claymath.org/sites/default/files/pvsnp.pdf*, pages 1–12.

[2] L.Lovasz. Combinatorial problems and exercises. *Noth-Holland, Amsterdam*, 1979.

[3] D.S.Johnson M.R.Garey. Computers and intractability :a guid to the theory of np-completeness. *Freeman,San Francisco*, 1979.

[4] R.Diestel. Graph theory. *Springer, New York*, 2000.

[5] M. Sghiar. Algorithmes quantiques, cycles hamiltoniens et la k-coloration des graphes. *Pioneer Journal of Mathematics and Mathematical Sciences*, 17-Issue 1 :51–69, May 2016.

[6] M. Sghiar. Atomic algorithm and the servers' s use to find the hamiltonian cycles. *International Journal of Engineering Research and Applications (IJERA), ISSN : 2248-9622*, 6-Issue 6 :23–30, jun 2016.

[7] M. Sghiar. An electronic algorithm to find the optimal solution for the travelling salesman problem. *IOSR Journal of Mathematics (IOSR-JM), e-ISSN : 2278-5728, p-ISSN : 2319-765X*, 12 :82–86, August 2016.

[8] M. Sghiar. Les nombres graphiques et le problème p=np. *IOSR Journal of Mathematics*, 14.3 :26–29, 2018.

M.Sghiar

msghiar21@gmail.com

9 Allée capitaine J.B. Bossu, 21240, Talant, France

Tel : (France) 0033669753590