REVUE FRANÇAISE D'AUTOMATIQUE, D'INFORMATIQUE ET DE RECHERCHE OPÉRATIONNELLE. RECHERCHE OPÉRATIONNELLE

CHOUKHMANE EL-ARBI

Une heuristique pour le problème de l'arbre de Steiner

Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle, tome 12, n° 2 (1978), p. 207-212.

<http://www.numdam.org/item?id=RO_1978__12_2_207_0>

R.A.I.R.O. Recherche opérationnelle/Operations Research
(vol. 12, n° 2, mai 1978, p. 207 à 212)

UNE HEURISTIQUE POUR LE PROBLÈME DE L'ARBRE DE STEINER (1)

CHOUKHMANE EL-ARBI (2)

Résumé. — *On dit qu'un algorithme résout un problème de minimisation dans un rapport $r \geq 1$ si la valeur de la solution obtenue par cet algorithme est inférieure au produit de la valeur de la solution optimale par r. On construit une heuristique et on montre qu'elle résout le problème de l'arbre de Steiner dans un rapport $r = 2\,[(k-1)/k]$ (ou k est le nombre de sommets à connecter).*

Soit $G = (N, E)$ un graphe connexe, non orienté, avec une valuation des arêtes :

$$e \in E \rightarrow W(e) > 0.$$

Étant donné $S \subseteq N$ trouver un sous-graphe $G_1 = (Y, F)$ de G tel que : G_1 est connexe, $S \subseteq Y$ et minimisant $\sum_{e \in F} W(e)$.

Ce problème connu sous le nom de « arbre de Steiner » fait partie de la classe des problèmes NP-complet [4].

L'une des approches pratiques de cette classe de problèmes « difficiles » est de construire des heuristiques et de mesurer leur efficacité.

On dira qu'une heuristique H résout le problème P dans le rapport $r \geqq 1$ si [3] : $\varphi_H/\varphi_{\text{opt}} \leqq r$, *où* φ_H; *valeur de la solution obtenue par H;* φ_{opt}, *valeur de la solution optimale.*

Dans ce qui suit on construit une heuristique simple qui approche le problème de « Steiner » tree et on démontre qu'elle le résout dans un rapport $r = 2\,[(k-1)/k]$ où $k = |S|$.

LEMME : *Soient $T = (N, E)$ un arbre valué et $S \subseteq N$ tel que tous les sommets pendants de T appartiennent à S, alors :*

1° *quelle que soit la façon avec laquelle on indice les sommets S*

$$S = [S_1, S_2, \ldots, S_k]$$

on a

$$\varphi_T = \sum_{e \in E} W(e) \leqq \frac{1}{2}\left[\sum_{i=1}^{k-1} W_T(S_i S_{i+1}) + W_T(S_k S_1)\right];$$

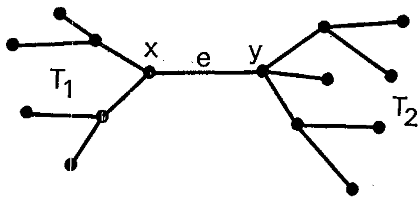
(1) Manuscrit reçu mars 1977, révisé mai 1977.
(2) Équipe de recherche « Graphes et Programmation combinatoire » Université Paris-VI, Tour 45-46.

2° *il existe une façon d'indicer les sommets* S $S = [S_1, S_2, \ldots, S_k]$ *telle que*

$$\varphi_T = \sum_{e \in E} W(e) = \frac{1}{2}\left[\sum_{i=1}^{k-1} W_T(S_i\, S_{i+1}) + W_T(S_k\, S_1)\right],$$

où $W_T(S_i\, S_{i+1})$ = *valeur de la chaîne élémentaire joignant* S_i *à* S_{i+1} *dans l'arbre* T.

Démonstration : 1° Dans l'évaluation $\sum_{i=1}^{k-1} W_T(S_i\, S_{i+1}) + W_T(S_k\, S_1)$ la valeur $W(e)$ de toute arête e est comptée au moins deux fois; en effet, soit une arête $e = xy$, cette arête est un isthme de T appelons T_1 et T_2 les sous-arbres de T qui résultent de la suppression de e.



Deux cas se présentent :

(a) S_1 et S_k appartiennent à T_1, T_2 contient alors un S_i avec $1 < i < k$, $S_{i-1} \in T_1$ et $S_{i+1} \in T_1$ la valeur de l'arête e est alors comptée dans $W_T(S_{i-1}\, S_i)$ et dans $W_T(S_i\, S_{i+1})$;

(b) $S_1 \in T_1$ et $S_k \in T_2$; il existe i, $1 \leq i \leq k$ avec $S_i \in T_1$ et $S_{i+1} \in T_2$, la valeur de l'arête e est alors comptée une fois dans $W_T(S_i\, S_{i+1})$ et une fois dans $W_T(S_k\, S_1)$.

2° Si en plus la façon d'indicer les sommets de S, vérifie la propriété suivante :

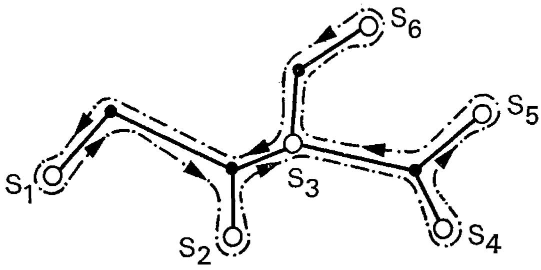
Pour toute arête e, les sommets de S qui appartiennent à $T_1 \cap S$ sont indicés $[S_1, S_{i+1}, \ldots, S_p]$ et ceux appartenant à $T_2 \cap S$ sont indicés

$$[S_1, \ldots, S_{i-1}, S_{p+1}, \ldots, S_k]$$

alors on a

$$\varphi_T = \sum_{e \in T} W(e) = \frac{1}{2}\left[\sum_{i=1}^{k-1} W_T(S_i\, S_{i+1}) + W_T(S_1\, S_k)\right].$$

Avec une telle façon d'indicer les sommets de S, chaque arête est comptée exactement deux fois, une fois dans $W_T(S_{i-1}\, S_i)$ et une fois dans $W_T(S_p\, S_{p+1})$. On indice les sommets de S selon l'ordre de parcours en suivant un circuit eulerien (voir *fig.*) contournant l'arbre T, sans jamais couper une arête.



HEURISTIQUE H : (1) *Construire le graphe complet* $G(S) = (S, E_S)$ *avec la valuation :* W_G; $W_G(S_i S_j)$ = *valeur d'une plus courte chaîne joignant* S_i *à* S_j *dans le graphe initial* G *valué par* W;

(2) *construire un arbre* T_S, *maximal de poids minimal du graphe* $G_{(S)}$.

(3) *poser* $R = \{ \Phi \}$; *pour toute arête* $e_S = S_i S_j \in T_S$ *faire*

$R := R \cup \{$ *sommet appartenant à une plus courte chaîne joignant* S_i *à* S_j *dans le graphe* G $\}$,

(4) *construire un arbre maximal de poids minimal du sous-graphe engendré par* $Y = S \cup R$;

(5) *s'ils existent des sommets pendants n'appartenant pas à* S *les supprimer.*

Cette heuristique nécessite le calcul de la matrice symétrique des distances qui est en $O(n^3)$ *et de deux recherches d'arbre maximal de poids minimal qui est en* $O(m \text{ Log Log } n)$.

THÉORÈME :

$$\frac{\varphi_H}{\varphi_{\text{opt}}} \leqq 2\frac{k-1}{k}.$$

Démonstration : Soit T_0 une solution optimale, tous les sommets pendants de T_0 appartiennent à S.

D'après le lemme, il existe une façon d'indicer les sommets S telle que

$$\varphi_{\text{opt}} = \varphi_{T_0} = \frac{1}{2}\left[\sum_{i=1}^{k-1} W_{T_0}(S_i S_{i+1}) + W_{T_0}(S_k S_1)\right]$$

comme $W_G(S_i S_{i+1}) \leqq W_{T_0}(S_i S_{i+1})$ on a

$$\varphi_{\text{opt}} \geqq \frac{1}{2}\left[\sum_{i=1}^{k-1} W_G(S_i S_{i+1}) + W_G(S_k S_1)\right].$$

Considérons le cycle $C = S_1, S_2, \ldots, S_k, S_1$ dans le graphe $G(S)$:

$$\varphi_c = \sum_{i=1}^{k-1} W(S_i S_{i+1}) + W(S_k S_1),$$

$$\varphi_{\text{opt}} \geqq \frac{1}{2} \varphi_c .$$

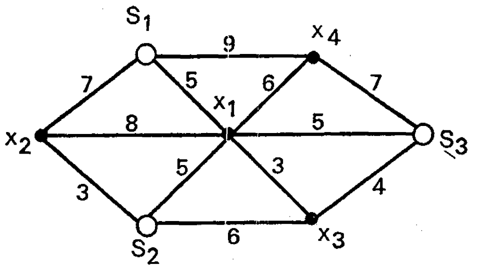
Le cycle C contient un arbre maximal du graphe $G(S)$:

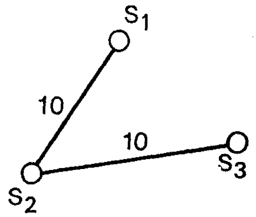
$$\varphi_H \leqq \varphi_{T_S} \leqq \varphi_c \frac{k-1}{k},$$

d'où

$$\frac{\varphi_H}{\varphi_{\text{opt}}} \leqq 2 \frac{k-1}{k}.$$

Exemple 1 :

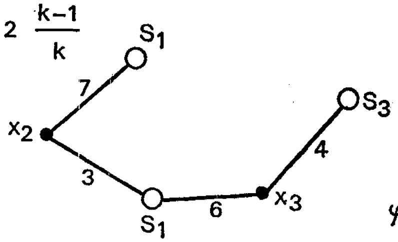




T_S, arbre maximal de poids minimal de $G_{(S)}$.

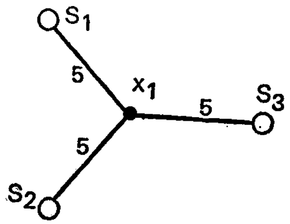
Si on prend les plus courtes chaînes $[S_1, x_2, S_2]$ et $[S_2, x_3, S_3]$ on obtient une solution extrême.

$$\frac{\varphi_H}{\varphi_{\text{opt}}} = 2 \frac{k-1}{k}$$



$$\frac{\varphi_H}{\varphi_{\text{opt}}} = \frac{20}{15} = \frac{4}{3}$$

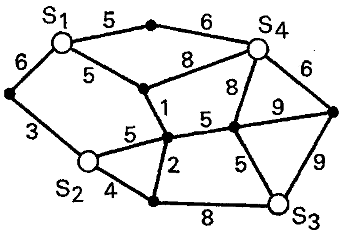
Par contre, si on prend les plus courtes chaînes $[S_1, x_1, S_2]$ et $[S_2, x_1, S_3]$ on obtient une solution optimale.



$$\varphi_H = \varphi_{opt} = 15.$$

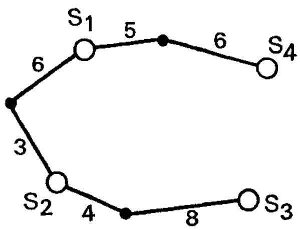
Cet exemple montre que dans le cas où la plus courte chaîne entre S_i et S_j, n'est pas unique, on génère à partir de l'étape (3) de l'heuristique, les solutions relatives aux différentes plus courtes chaînes et on prend la meilleure des solutions obtenues.

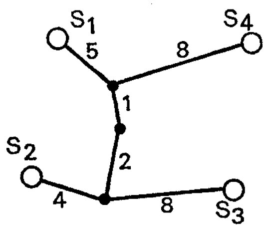
Exemple 2 :



$|S| = 4$

$S = \{S_1, S_2, S_3, S_4, \}$





Solution obtenue par l'heuristique H. $\varphi_H = 32$. Solution optimale $\varphi_{opt} = 28$.

$$\frac{\varphi_H}{\varphi_{opt}} = \frac{32}{28} = \frac{8}{7} < 2\frac{4-1}{4} = \frac{3}{2}.$$

Ce deuxième exemple montre que l'optimum n'est pas toujours atteint par cet algorithme.

BIBLIOGRAPHIE

1. M. CHEIN, *Un algorithme pour relier N Points*, Calcolo, Fasc. 4, vol. 5, 1968, p. 537-547.
2. S. E. DREYFUS, *The Steiner Problem in Graphs*, Networks, vol. 1, n° 3, 1972, p. 195-207.
3. R. M. KARP, *The Fast Approximation Solution of Hard Problems*, Proc. 6th S.E. Conf. Combinatorics Graphs Theory and Computing, 1975, p. 15-51.
4. R. M. KARP, *Reductibility Among Combinatorial Problems*, Complexity of Computer Computations, R. E. MILLER and J. W. THATCHER, eds., Plenum Press, New York, 1972, p. 85-104.